Evaluating Deep Learning Models for Vocabulary Alignment at Scale in the UMLS Metathesaurus

Goonmeet Bajaj[1], Vinh Nguyen[2], Thilini Wijesiriwardene[3], Hong Yung Yip[3], Vishesh Javangula[4], Srinivasan Parthasarathy[1], Amit Sheth[3], Olivier Bodenreider [2]

[1] The Ohio State University
[2] National Library of Medicine
[3] AI Institute, University of South Carolina
[4] George Washington University
{bajaj.32, parthasarathy.2}@osu.edu, vinh.nguyen@nih.gov, obodenreider@mail.nih.gov, {thilini, amit}@sc.edu, hyip@email.sc.edu, visheshj123@gwu.edu

Abstract

The current UMLS (Unified Medical Language System) Metathesaurus construction process for integrating over 200 biomedical source vocabularies is expensive and error-prone as it relies on the lexical algorithms and human editors for deciding if the two biomedical terms are synonymous. Recent work has aimed to improve the Metathesaurus construction process using a deep learning approach with a Siamese Network initialized with BioWordVec embeddings for predicting synonymy among biomedical terms. Recent advances in Natural Language Processing, such as Transformer models, and Graph Neural Networks (GNN), such as Graph Attention Networks (GAT), have achieved state-of-the-art (SOTA) on different downstream tasks. Therefore, these techniques are therefore logical candidates for a synonymy prediction task as well.

In this paper, we evaluate different approaches of employing biomedical BERT-based Transformer models and Graph Attention Networks for synonymy prediction. We employ BERT models in two model architectures: (1) Siamese Network, and (2) Transformer for predicting synonymy in the UMLS Metathesaurus. We aim to validate if using the BERT models or GNNs can actually outperform the existing approaches for synonymy prediction. In the existing Siamese Networks with LSTM and BioWordVec embeddings, we replace the BioWordVec embeddings with the biomedical BERT embeddings extracted from each BERT model using different ways of extraction. For the Transformer architecture, we evaluate the use of the different biomedical BERT models that have been pre-trained using different datasets and tasks. For the GNN architecture, we formulate synonymy prediction as a link prediction task use a graph neural network (GNN) with a graph attention layer to predict if two terms are synonymous in the UMLS Metatharsus.

Given the SOTA performance of these BERT models for other downstream tasks, our experiments yield surprisingly interesting results: (1) employing these biomedical BERT-based models do not outperform the existing approaches using Siamese Network with BioWordVec embeddings for the UMLS synonymy prediction task, (2) the original BioBERT large model that has not been pre-trained with the UMLS outperforms the SapBERT models that have been pre-trained with the UMLS, and (3) using the Siamese Networks yields better performance for synonymy prediction when compared to using the biomedical BERT models and the GNN architecture.

1 Introduction

The Unified Medical Language System (UMLS) (Bodenreider 2004) is a biomedical terminology integration system that includes over 200 source vocabularies[1], including CPT, ICD-10, MeSH, and SNOMED CT. The UMLS Metathesaurus construction process organizes synonymous terms from these source vocabularies into *concepts*. The Ontology Alignment Evaluation Initiative[2] (OAEI) uses three well-defined vocabularies (NCI, FMA, and SNOMED CT) from the Metathesaurus for their ontology alignment task (Euzenat et al. 2011). Unlike the ontologies used by OAEI, not all vocabularies in the UMLS are well-defined or represented as ontologies. Therefore, when we refer to Metathesaurus construction process we use the phrase *vocabulary alignment* instead of ontology alignment.

The Metathesaurus construction process uses a lexical similarity model and semantic pre-processing to determine synonymy. Human editors determine the final set of synonymous terms. The large scale and diversity of the Metathesaurus make the construction process very challenging, tedious, and error-prone for human editors. To assist the UMLS Metathesaurus construction process, Nguyen et al. introduce the UMLS Vocabulary Alignment (UVA) task, or synonymy prediction (Nguyen, Yip, and Bodenreider 2021). The synonymy prediction task takes in terms (or "atoms") as input to determine synonymy among them. The authors design and train a Siamese network to predict if two atoms from UMLS are synonymous (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). The Siamese network is initialized using BioWordVec embeddings that are learned using fastText (Bojanowski et al. 2017) to encode atom strings. The authors use the Manhattan distance to compute the (dis)similarity in the final output representations from the Siamese network. Synonymous pairs can be

[1] https://www.nlm.nih.gov/pubs/techbull/mj21/mj21_umls_2021aa_release.htmll

[2] http://oaei.ontologymatching.org/

predicted using different similarity thresholds. We describe this approach in more detail in Section 5.

Given the success of Transformer models in Natural Language Processing (NLP) (Devlin et al. 2018; Vaswani et al. 2017), we evaluate different approaches of employing biomedical BERT-based models in two model architectures: (1) a Siamese Network (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021), and (2) a Transformer model (Devlin et al. 2018) for predicting synonymy in the UMLS Metathesaurus. We first evaluate different feature extraction techniques to replace BioWordVec embeddings with BERT embeddings in the current state-of-the-art Siamese Networks used for synonymy prediction (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). Second, we evaluate the use of the Transformer architecture using the biomedical BERT models for synonymy prediction. In particular, we use nine different biomedical BERT models: BioBERT (Lee et al. 2020), BioBERT Large (Lee et al. 2020), BlueBERT (Peng, Yan, and Lu 2019), SapBERT (Liu et al. 2020), UMLSBERT (Michalopoulos et al. 2020), BioBERT + SapBERT (Liu et al. 2020), BlueBERT + SapBERT (Liu et al. 2020), UMLSBERT + SapBERT (Liu et al. 2020), VanillaBERT + SapBERT (Liu et al. 2020).

In this work, we also explore the use of graph neural networks (GNN) for synonymy prediction. The existing Siamese Networks (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021) used for predicting synonymous terms only use lexical features. While the lexical features carry meaningful information, we hypothesis that incorporating additional graph features (i.e. structural and hierarchical features) will help with synonymy prediction boost precision. To exploit the structural features and hierarchy of the Metathesaurus we formulate the synonymy prediction as a link prediction tasks. Given two atoms from UMLS we aim to predict if an edge representing the synonymy among the atoms exists. To solve, this link prediction task, we use a GNN with a graph attention layer (Veličković et al. 2017; Zhang et al. 2019a) to incorporate information from an atom's neighbors (e.g. the semantic group, the parent source vocabulary).

Contributions. Given the SOTA performance of these BERT models for other downstream tasks, our experiments yield surprisingly interesting results: (1) employing these biomedical BERT-based models do not outperform the existing approaches using Siamese Network with BioWordVec embeddings for the UMLS synonymy prediction task, (2) the original BioBERT large model that has not been pretrained with the UMLS outperforms the SapBERT models that have been pre-trained with the UMLS, and (3) using the Siamese Networks yields better performance for synonymy prediction when compared to using the biomedical BERT models and the GNN architecture.

2 Background: Knowledge Representation in the UMLS Metathesaurus

The UMLS Metathesaurus links terms and codes between health records, pharmacy documents, and insurance documents (Bodenreider 2004). The Metathesaurus consists of several building blocks, including atoms and concepts (clusters of synonymous atoms). Each atom is a term from a specific source vocabulary and each concept is a cluster (or grouping) of atoms. All atoms in the UMLS Metathesaurus are assigned a unique identifier (AUI). Atom strings in the UMLS are also assigned a semantic group (SG) reflecting the semantics of the string in the source vocabularies. the source When the same term appears in different source vocabularies, the individual terms are assigned separate AUIs. Table 1 contains examples of synonymous atoms and the various types of identifiers assigned to each respective atom for a particular concept (i.e. C0018681). For example, the term "Cephalodynia" appearing in both MSH and SNOMEDCT_US has different AUIS as shown in Table 1: "A26628141" and "A2957278" respectively. AUIs are then linked to a unique string identifier (SUI) to represent occurrences of the same term. Any lexical variation in character set of the term, upper-lower case, or punctuation results in a separate SUI. Additionally, the strings "Headache" and "Headaches" have different AUIs because of the lexical variation (see Table 1). Finally, each concept (cluster of synonymous terms) in the Metathesaurus is labelled with a unique identifier (CUI). To recap, every atom (or term) has an unique identifier (AUI) is linked to a single string (STR), is associated with a semantic group (SG), and belongs to a single concept with a unique identifier (CUI).

Tuple	String	Source	SCUI	AUI	CUI	SG
t_1	Headache	MSH	M0009824	A0066000	C0018681	Disorders
t_2	Headaches	MSH	M0009824	A0066008	C0018681	Disorders
t_3	Cranial Pains	MSH	M0009824	A1641924	C0018681	Disorders
t_4	Cephalodynia	MSH	M0009824	A26628141	C0018681	Disorders
t_5	Cephalodynia	SNOMEDCT_US	25064002	A2957278	C0018681	Disorders
t_6	Headache (finding)	SNOMEDCT_US	25064002	A3487586	C0018681	Disorders

Table 1: Examples of synonymous atoms from a Metathesaurus concept with associated identifiers

3 Problem Formulation

To formulate the UMLS Metathesaurus construction process as a machine / deep learning task, Nguyen et al. (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021) view synonymy prediction as a similarity task. The task is to identify synonymous atoms by measuring the (dis)similarity among pairs of atoms. Finding synonymous atoms pairs is comparable to finding a cluster of synonymous atoms. Of note, the authors refrain from treating this as a classification task, because it is unfeasible to train a classifier to predict one out of 4.28 million classes (i.e. concepts) present in the 2020AA UMLS release.

A machine-learning model should be able to identify the (a)synonymy among atoms are that lexically:

- similar but are not synonymous, e.g., "Lung disease and disorder" versus "Head disease and disorder"
- dissimilar but are synonymous, e.g., "Addison's disease" versus "Primary adrenal deficiency"

We maintain the same problem definition proposed by (Nguyen, Yip, and Bodenreider 2021). The synonymy prediction task is defined as follows. Let $T =$

$(S_{STR}, S_{SRC}, S_{SCUI}, S_{AUI}, S_{SG})$ be the set of all input tuples in the Metathesaurus where:

S_{STR} is the set of all atom names,

S_{SRC} is the set of all source vocabulary names,

S_{SRC} is the set of all source vocabulary unique identifiers,

S_{AUI} is the set of all atom unique identifiers, and

S_{SG} is the set of all semantic groups.

Consider four sample tuples from Table 1:

t_1 = ("Headache", "MSH", "M0009824", "A0066000", "Disorders")

t_3 = ("Cranial Pains", "MSH", "M0009824", "A1641924", "Disorders")

t_4 = ("Cephalodynia", "MSH", "M0009824", "A26628141", "Disorders")

t_5 = ("Cephalodynia", "SNOMEDCT_US", "M0009824", "A3487586", "Disorders").

The tuples shown here consist of (str, src, $scui$, aui, sg), where str is the original string of the term from the source vocabulary (src), $scui$ is the source vocabulary identifier, aui is the unique atom identifier, and sg is the semantic group.

Let (t_i, t_j) be a pair of input tuples, where $i \neq j$ and each tuple is initialized from a different source vocabulary in the form of (str, src, $scui$, aui, sg). Let $f : T \times T \to 0, 1$ be a prediction function that maps a pair of input tuples to either 0 or 1. If $f(t_i, t_j) = 1$, then the two strs, (str_i, str_j), from t_i and t_j are synonymous.

4 Dataset

We thank Nguyen et al. for sharing the training data used in their work (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). The dataset is created using the 2020AA release of the UMLS Metathesaurus and only contains English terms from active source vocabularies. There are approximately 27.9M synonymous pairs (positive samples) in the UMLS and the approximately 10^{14} pairs of non-synonymous atom (negative samples). The ratio of negative samples to positive samples is large because most atoms do not share a CUI. To create a better class balance between the negative and positive samples, the authors reduce the negative samples to approximately 170M (Nguyen, Yip, and Bodenreider 2021). Additionally, Nguyen et al. (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021) created different dataset splits based on the lexical similarity and the number of negative and positive training pairs. In this work, we use the *ALL* split of the dataset used in (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). The *ALL* dataset contains the following splits:

- TOPN_SIM: negative pairs with the highest similarity
- RAN_SIM: random negative pairs having some similarity
- RAN_NOSIM: random negative pairs having no similarity

The training and testing datasets are mutually exclusive and do not contain overlapping examples. For our study, we use the ALL dataset and Table 2 contains the dataset statistics.

Type	Negative Examples	Positive Examples	Total
Training and Validation	170,075,628	22,324,834	192,400,462
Testing	167,454,653	5,581,209	173,035,862

Table 2: Dataset Statistics

We refer the readers to Section 4.2 of (Nguyen, Yip, and Bodenreider 2021) for a complete description of the dataset generation process.

5 Related Work

Siamese Networks for UVA

Nguyen et al. (Nguyen, Yip, and Bodenreider 2021) assess the similarity of atoms using lexical features of the atom strings (src). The authors design a Siamese Netowrk that inputs a pair of atom strings, (src_i, src_j), where $i \neq j$, and outputs a similarity score between 0 and 1, $sim(src_i, src_j) \in [0, 1]$. The inputs are pre-processed, then sent through an embedding layer initialized with BioWordVec embeddings (Zhang et al. 2019b). The word embeddings are then fed into Bidirectional Long Short Term Memory (Bi-LSTM) layers to learn the semantic and syntactic features of the atoms through as a sequence of tokens. The outputs from the Bi-LSTMs are then fed into two consecutive dense layers consisting of 128 hidden units and 50 respectively. The learned representation for each atom are then fed into a Manhattan distance similarity function, $exp(-||LSTM_A - LSTM_B||_1) \in [0, 1]$ to determine the similarity. In their follow-up work (Nguyen and Bodenreider 2021), Nguyen et al. add an attention layer that improves the precision of the network for synonymy prediction by +3.63% and decreases recall by -1.42%. Figure 2 displays the Siamese Network architecture. The asterisk next to the attention layer indicates that the additional layer is only used in Nguyen et al.'s follow up work (Nguyen and Bodenreider 2021). Given the success of Transformer models for different NLP tasks, the objective of this work is to investigate the performance of context-aware embeddings extracted using different methods from various domain-specific BERT models with the Siamese networks introduced in (Nguyen and Bodenreider 2021; Nguyen, Yip, and Bodenreider 2021). In this work, we replace the BioWordVec embeddings with embeddings extracted from biomedical BERT models using different feature extraction techniques. In Section 7, we outline our experimental setup.

Contextualized Word Representations and Biomedical BERT Models

Distributed representations (e.g. word2vec, fastText) provide a single embedding for lexically similar but semantically dissimilar words (Mikolov et al. 2013; Pennington, Socher, and Manning 2014; Bojanowski et al. 2017). Recent advances in Natural Language Processing (NLP) have led to better performing contexualized embeddings that are learned using Transformer models (e.g. BERT) (Vaswani et al. 2017; Devlin et al. 2018). Transformer models produce contextualized word representations that are informed by the surrounding words in the input. Additionally, Transformers handle long range dependencies in sequences entirely through self-attention instead of sequence-dependent RNNs (Lin et al. 2017). Embeddings extracted from Transformer models have outperformed word2vec based embeddings on several NLP tasks (Devlin et al. 2018; Vaswani et al. 2017).

Bidirectional Encoder Representations from Transformers (BERT) is a key technical innovation that applies the attention mechanism found in Transformers to language models. Unlike previous efforts in language modelling, which looked at text sequences in a unidirectional manner, BERT processes text sequences bidirectionally and learns a deeper sense for the context and flow of a language (Devlin et al. 2018). The BERT architecture is designed to provide such contextualized representations. In order to achieve the contextual embeddings, BERT models are trained using two self-supervised training tasks: Masked Language Model (MLM) and Next Sentence Prediction (NSP) (Devlin et al. 2018). Current research shows these models can be pre-trained on large domain-specific corpora and fine-tuned on smaller domain specific tasks to achieve better performance on downstream tasks (Lewis et al. 2020). Biomedical NLP research follows this trend and has shown that BERT-based models such as BioBERT (Lee et al. 2020), BlueBERT (Peng, Yan, and Lu 2019), SapBERT (Liu et al. 2020) trained on domain-specific datasets outperform models that use more traditional word embeddings generated from models like word2vec and fastText. In Section 6, we provide more details about the different Transformer models used in our work.

Graph Attention Networks

Graph attention networks (GAT) are a type of GNN that use a graph attention layer to incorporate neighborhood information of nodes (Veličković et al. 2017). GATs have outperformed traditional GNNs by leveraging self-attention layers to attend over features of neighboring nodes. Zhang et al. (Zhang et al. 2019a) introduce a heterogeneous graph attention networks (HGAT) to handle large-scale graphs with different types of nodes (i.e. authors and papers in a citation graph). The HGAT consists of an multi-head graph attention encoder that is initialized with pre-trained word and structural node embeddings that are concatenated together, h. The encoder layers learn the attention weights, o_{ij}, using self-attention:

$$o_{ij} = attn(Wh_i, Wh_j)$$

The attention weight, o_{ij}, indicates the importance of a neighborhood node's, n_j, features to the input node, n_i. The vectors h_i and h_j are the feature vectors for nodes i and j respectively and W is shared projection matrix. The attention weights are normalized using the softmax function. For each node type, the authors use a different attention weight to account for the heterogeneous property of graphs. The different attention heads are passed through the sigmoid activation layer and concatenated together to get the final node representations, e.g. h'_i and h'_j. The two node representations for the input nodes are concatenated and passed through a fully connected output layer to determine the edge between the two input nodes:

$$y = fc(h'_i \oplus h'_j)$$

The authors use the negative log-likelihood loss function to optimize the model.

Model_Type	Embedding Dimension	Vocabulary Size	Token Size	# of Parameters	# of Parameters w. Attention Layer
BioWordVec	200	268,158,600	-	268,221,858	268,221,778
BioBERT (+ SapBERT)	768	28,996	13,230,336	13,407,194	13,407,114
BioBERT_Large (Cased)	1024	58,996	28,530,688	28,758,666	28,758,746
BlueBERT	1024	30,522	25,358,336	25,586,314	25,586,394
SapBERT	768	30,522	21,035,520	21,212,298	21,212,378
UMLSBERT (+ SapBERT)	768	28,996	13,230,336	13,407,114	13,407,194
BlueBERT+ SapBERT	768	30,522	19,018,752	19,195,530	19,195,610
VanillaBERT + SapBERT	768	30,522	19,018,752	19,195,530	19,195,610

Table 3: Comparison of Siamese Networks initialized with embedding from different biomedical BERT models

6 Biomedical BERT Variants

In this section, we explain the differences between the domain specific BERT variants used in this study. Table 3 compares the different biomedical BERT models used in this paper.

BioBERT: BioBERT is initialized from BERT pre-trained on Wikipedia (2.5 billion words) and Books Corpus (0.8 billion words) (Lee et al. 2020). This BERT model is then pre-trained on biomedical domain data consisting of PubMed Abstracts (4.5 billion words) and PMC Full-text articles (13.5 billion words). Then the pre-trained model was used in several Biomedical NLP tasks such as Biomeidcal Named Entity Recognition (BioNER), BioRE and question answering (Lee et al. 2020). For this study, we use BioBERT-Base v1.1, which has 768 hidden units for the embedding layer, and BioBERT-Large v1.1 (trained with a custom vocabulary), which has 1024 hidden units for the embedding layer.

BlueBERT: BlueBERT is initialized with BERT weights provided by (Devlin et al. 2018) and further pre-trained with biomedical corpus (PubMed abstract with 4,000M words) and clinical notes corpus (MIMIC-III with 500M words). Two versions of BlueBERT are released consistent with BERT-Base and BERT-Large models trained with 5M steps on the PubMed corpus and 0.2M steps on the MIMIC-III corpus. In our work, we use BlueBERT-Large trained on both PubMed and MIMIC-III datasets.

SapBERT: SapBERT provides the current SOTA results for 6 medical entity linking (MEL) bench-marking datasets. SapBERT is trained on the UMLS with 4M+ concepts and 10M+ synonyms from over 150 vocabularies including MeSH, SNOMED CT, RxNorm. SapBERT is trained using a SOTA metric learning objective inspired by visual recognition, for learning from the positive and negative pairs of the UMLS.

BioBERT + SapBERT, BlueBERT + SapBERT, BlueBERT + SapBERT, UMLSBERT + SapBERT, VanillaBERT + SapBERT: The SapBERT authors pre-train additional variants of SapBERT that are initialized using different BERT variants.

UMLSBERT: UmlsBERT is initialized with the pre-trained Bio_ClinicalBERT model (Alsentzer et al. 2019) and further pre-trained with the MLM task on the MIMIC-III dataset. The authors modify the pre-training task in two ways: 1) by

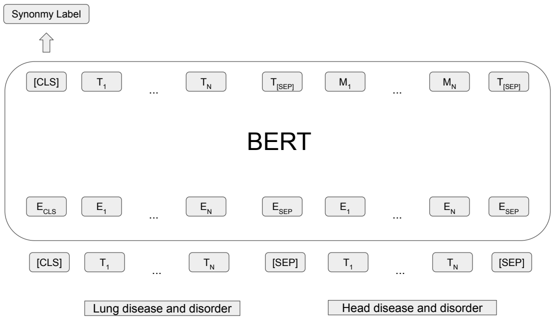

Figure 1: BERT Model for Synonymy Prediction

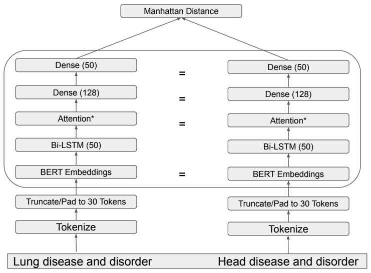

Figure 2: Siamese Model used in (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021)

introducing an additional semantic type embedding, 2) modify the MLM task by replacing the 1-hot vector that corresponds to the masked word, with a binary vector to indicate which words share them same CUI as the masked word.

7 Approach

In this section, we first explain our experimental setup to investigate the performance of the Siamese networks using embeddings extracted from BioBERT, BioBERT Large, BlueBERT, SapBERT, UMLSBERT, BioBERT + SapBERT, BlueBERT + SapBERT, UMLSBERT + SapBERT, and VanillaBERT + SapBERT. Next, we explain our setup to investigate the performance of the different domain-specific BERT models for the UVA task using the BERT architecture.

UVA with Siamese Networks

BioWordVec embeddings are generated using the fastText model (Bojanowski et al. 2017) on the PubMed text corpus and MeSH. As mentioned, embeddings extracted from Transformer models (e.g. BERT) used to train different deep learning models have achieved state-of-the-art (SOTA) performance on different down-stream tasks because these embeddings are context-dependent. For example, syntactic features are captured better using the middle layers and semantic features at the latter layers of the model (Jawahar, Sagot, and Seddah 2019). Additionally, the original inventors of

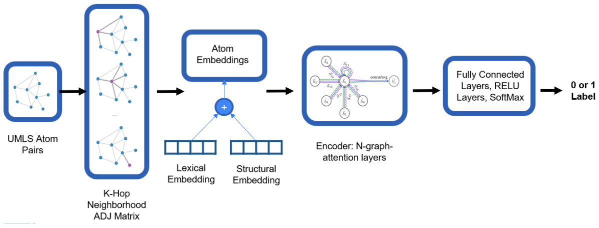
UMLS Atom Pairs
K-Hop Neighborhood ADJ Matrix
Atom Embeddings
+
Lexical Embedding
Structural Embedding
concat/avg
Encoder: N-graph-attention layers
Fully Connected Layers, RELU Layers, SoftMax
0 or 1 Label

Figure 3: HGAT adapted from (Zhang et al. 2019a)

BERT share that extracting embeddings from different layers leads to variations in performance on the down-stream task (Devlin et al. 2018).

We use this insight from (Jawahar, Sagot, and Seddah 2019; Liu et al. 2019), to experiment weather the Siamese Network model introduced in (Nguyen, Yip, and Bodenreider 2021) can benefit from contextually aware embeddings extracted from biomedical BERT models. We extract different token embeddings from different biomedical BERT models. Different layers of BERT carry different information (Jawahar, Sagot, and Seddah 2019; Liu et al. 2019; Devlin et al. 2018). Therefore, we investigate which token embeddings and which layers lead to better performance when used to initialize the Siamese Network. We extract two sets of embeddings from each model: 1) embeddings from the last layer, and 2) embeddings from the average of the last four layers. Additionally, we use the three different types of token embeddings: 1) the first occurrence of the token in the dataset, 2) the last token in the dataset, 3) the average embedding of each occurrence of the token in the dataset. It is important that we investigate which token embedding is appropriate because the BERT models generate different token embeddings for each token based on the context from the input atom string. Of note, we only use the atom string to extract token embeddings because all vocabularies have this characteristic in common. We tokenize the atom strings using the BERT tokenizer and the respective vocab from each biomedical BERT model.

UVA with Transformer Networks

To understand the usability of the domain-specific BERT Models, we use pre-trained models for the UVA task. These experiments allow us to benchmark the performance of the different domain-specific BERT models for the UVA task. Of Note, we do not fine-tune these BERT models because the size of our training data.

For the UVA, task we use the following input format to process the atom strings: A [CLS] token is inserted at the beginning of the first atom string str_i, followed by a [SEP] token, followed by the second atom string str_j, followed by a final [SEP] token to indicate the end of the sequence. The input is then processed through the BERT model and an output of 0 (synonymous) or 1 (not synonymous) is predicted.

UVA with Graph Attention Networks

To understand the usability of the domain-specific BERT Models, we adapt the HGAT introduced in (Zhang et al.

2019a). Our modified HGAT is presented in Figure 3. We treat the Metathesaurus as a graph where the set of nodes, N, consists of atoms, semantic groups, and source vocabularies, $N = (S_{AUI} \cup S_{SCUI} \cup S_{SG})$. We treat UMLS as a undirected graph and do not differentiate between the different relationships between nodes in the graph. Therefore, an edge, e exists between two nodes in UMLS if they share any relationship.

8 Experimental Setup

In this section, we provide details for our three sets of experiments: 1) feature extraction for the Siamese Network, 2) UVA with transformer networks, and 3) UVA with graph attention networks. We run all experiments using a High Performance Computing (HPC) cluster. We use the following BERT models for the first two set of experiments: BioBERT, BioBERT Large, BlueBERT, SapBERT, UMLSBERT, BioBERT + SapBERT, BlueBERT + SapBERT, UMLSBERT + SapBERT, and VanillaBERT + SapBERT.

Feature Extraction for the Siamese Network

To understand the performance of the different embeddings extracted from the various BERT models, we train the Siamese Network end to end. Our experimental setup is similar to (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). We refrain from changing the experimental setup to allow for direct comparison of models initialized with different embeddings.

For each model we extract 6 types of embeddings:

- Last layer embedding of the first occurrence of a token
- Average embedding of the last four layers of the first occurrence of a token
- Last layer embedding of the last occurrence of a token
- Average embedding of the last four layers of the last occurrence of a token
- Average of the last layer embedding of every occurrence of a token
- Average embedding of the last four layers of every occurrence of a token

We trained each Siamese network for 100 epochs using the Adam optimizer with a learning rate of 0.001 to replicate the same setup as (Nguyen, Yip, and Bodenreider 2021). We used a batch size of 8192 in our experiments. The Bi-LSTMs consist of 50 hidden units, the first and second dense layers contain 128 and 50 hidden units respectively. We did run experiments to change the number of hidden units in the dense layers, but found no improvement in performance. We use up to 30 tokens from each atom string and pad the input if needed.

UVA with BERT Models

For each BERT model (i.e., BioBERT, BioBERT Large, BlueBERT, and SapBERT), we predict the synonymy labels for each atom pair in the test set using the input format described in Section 7. Results from these experiments are presented in Section 9.

Embedding Type	Accuracy	Precision	Recall	F1-Score	AUC
BioWordVec	0.9938	0.8872	0.9274	0.9069	0.9909
SapBERT	**0.9886**	**0.8053**	**0.854**	**0.8289**	**0.9852**
BioBERT	0.9832	0.7232	0.7823	0.7516	0.9751
BioBERT_Large	0.9854	0.7579	0.8098	0.783	0.9791
BlueBERT	0.9841	0.7409	0.7833	0.7615	0.9758
UMLSBERT					
BioBERT + SapBERT	0.9831	0.7202	0.7828	0.7502	0.9756
BlueBERT + SapBERT	0.985	0.7477	0.8121	0.7786	0.9798
UMLSBERT + SapBERT					
VanillaBERT + SapBERT	0.9855	0.7637	0.7993	0.7811	0.9791

Table 4: Results for the Siamese Model trained for 100 iterations using BioWordVec embeddings and BERT embeddings extracted using the average token and average of last four layers (Nguyen, Yip, and Bodenreider 2021)

Embedding Type	Accuracy	Precision	Recall	F1-Score
BioWordVec_Attention	0.9936	0.8884	0.9198	0.9038
SapBERT_Attention	**0.9886**	**0.8109**	**0.8479**	**0.829**
BioBERT_Attention	0.9852	0.7657	0.7823	0.7739
BioBERT_Large_Attention	0.9863	0.7754	0.8128	0.7937
BlueBERT_Attention	0.9863	0.7754	0.8128	0.7937
UMLSBERT_Attention				
BioBERT + SapBERT _Attention	0.9850	0.7634	0.7801	0.7716
BlueBERT + SapBERT _Attention	0.9863	0.7791	0.807	0.7928
UMLSBERT + SapBERT _Attention				
VanillaBERT + SapBERT	0.9865	0.7861	0.8014	0.7937

Table 5: Results for the Siamese Model with Attention Layer trained for 100 iterations using BioWordVec embeddings and BERT embeddings extracted using the average token and average of last four layers (Nguyen and Bodenreider 2021)

UVA with Graph Attention Networks

Similar to (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021) we use SapBERT to extract the lexical embeddings for each atom in the UMLS. We use TransE (Wang et al. 2014) knowledge graph embeddings to extract structural node embeddings for each atom. We create the neighborhood of each node by including all nodes reachable with two hops. We restrict the max neighborhood size to 30 nodes.

9 Evaluation

We evaluate the performance of our models using Accuracy, Precision, Recall, F-1, and AUC.

Feature Extraction with Siamese Network Results. Table 4 presents the synonymy prediction using the Siamese Network with embeddings extracted from BERT models and BioWordVec embeddings. Due to space limitations, we only

Model_Type	Accuracy	Precision	Recall	F1
BioBERT	0.5308	0.9683	0.5326	0.6872
BioBERT_Large	**0.6255**	**0.9870**	**0.6212**	**0.7625**
BlueBERT	0.1875	0.9985	0.1607	0.2768
SapBERT	0.4929	0.6528	0.4926	0.6528
UMLSBERT				
BioBERT + SapBERT	0.5550	0.9733	0.5554	0.7072
BlueBERT+ SapBERT	0.6261	0.9722	0.6318	0.7658
UMLSBERT + SapBERT	0.3325	0.9714	0.3197	0.4811
VanillaBERT + SapBERT	0.5767	0.9732	0.5785	0.7257

Table 6: Results for Synonymy Prediction using BERT models. Input format: ([CLS] src_i [SEP] src_j [SEP])

Model_Type	Accuracy	Precision	Recall	F1
BioBERT	0.4202	0.9778	0.4102	0.5780
BioBERT_Large	**0.6364**	**0.9864**	**0.6331**	**0.7712**
BlueBERT	0.2015	0.9984	0.1752	0.2981
SapBERT	0.4500	0.9668	0.4470	0.6114
UMLSBERT				
BioBERT + SapBERT	0.5704	0.9683	0.5750	0.7215
BlueBERT+ SapBERT	0.3576	0.9636	0.3494	0.5128
UMLSBERT + SapBERT				
VanillaBERT + SapBERT	0.3890	0.9614	0.3841	0.5489

Table 7: Results for Synonymy Prediction using BERT models. Input format: ([CLS] src_j [SEP] src_i [SEP])

Embedding Type	Precision	Recall	F1
Structural Embeddings (TransE)	0.1406	0.9928	0.2640
Structural (TransE) + Lexical Embeddings (SapBERT)	0.7023	0.2938	0.4143

Table 8: Results for Synonymy Prediction using GANs

share results for the average token embedding and the average of last four layers of each BERT model. Table 5 presents the synonymy prediction using the Siamese Network with an attention layer with embeddings extracted from BERT models and BioWordVec embeddings. Our feature extraction results indicate that averaging all token embeddings and using the average of the last four hidden layers provides the most useful embedding for most models. Additionally, we find that using the Siamese Network with the attention layer achieves better performance in terms of F1-score. Surprisingly, using the embeddings extracted from the biomedical BERT model do not outperform the two baselines of the Siamese Networks with BioWordVec. We find that the embeddings extracted from SapBERT model lead to the best performance for synonymy prediction.

Synonymy Prediction with BERT Results. Tables 6 and 7 present the synonymy prediction results using the BERT architecture. The pre-trained biomedical BERT models do not outperform the current SOTA Siamese Networks for synonymy prediction. We find that the BioBERT Large model is the best performing model for synonymy prediction. We attribute the performance of this model to its size. We run additional experiments to determine if changing the order of the atoms affects the performance of the models (i.e. feeding (src_j, src_i) as input instead of (src_i, src_j)). These results are present in Table 7. We see that changing the order of the input atom strings only improves the performance for two biomedical BERT models: BioBERT and BlueBert + SapBERT. The results increase for BioBERT and BlueBert + SapBERT by about 0.026% and 0.049% in terms of F1-score.

Synonymy Prediction with GATs. Table 8 shares the results of our two experiments using graph attention networks. In our first experiment we only use structural embeddings to initialize the model. Next, we use both lexical and structural embeddings. We find that using structural embeddings only are not enough for the model to learn the relationships between nodes in the graph. We attribute these low results to the fact that the embeddings in the model remained fixed and are not updated throughout the training process.

Discussion. Our current results for both set of experiments do not outperform the current SOTA. These results indicate that the pre-trained language models on the datasets from the same domain are not enough to accurately predict synonymy among atoms in the Metathesaurus. The number of parameters for the Siamese Networks initialized with BioWordVec are one magnitude higher than all of the BERT models. The large size of the vocabulary could be an indication as to why BioWordVec performs well when compared to the biomedical BERT based variants. Additionally, from our Synonymy prediction task we find that using a BERT model trained on the right data and the right task yeilds larger gains in performance for synonymy prediction. The SapBERT model is trained on PubMed and incorporates knowledge from the UMLS Metathesaurus in two ways: 1) using semantic type embeddings and 2) modifying the MLM task to indicate if which words belong to the same concept. These changes to the model likely indicate why it outperforms the other biomedical BERT models for synonymy prediction using the Siamese Networks. We find that using the biomedical BERT embeddings with the Siamese Network yield better results than using the pre-trained BERT models for the UVA task. The AUC scores are much higher for the Siamese Networks when compared to the BERT models. These results indicate that further fine-tuning is required to use the biomedical BERT models for the UVA task. GATs for synonymy prediction do not outperform the Siamese Networks. Additionally, GATs are more computationally expensive to train. As mentioned, we attribute the low performance of the GANs to the fixed embeddings. As future work, we aim to update the embeddings during the training process.

10 Conclusion

In this paper, we evaluated different methods of using biomedical BERT-based models in two model architectures: (1) Siamese Network, and (2) Transformer for predicting synonymy in the UMLS Metathesaurus. We also examine the use of GATs for synonymy prediction. We aimed to validate if these approaches using the BERT models and GATs can actually out-perform the existing approaches. We replace the BioWordVec embeddings in the Siamese Networks with the biomedical BERT embeddings extracted from different models using different ways of extraction. We evaluate the use of the different pre-trained biomedical BERT models using the transformer architecture that. Additionally, we formulate synonymy prediction as a link prediction task to employ GANs. Our experiments yield surprisingly interesting results: (1) the approaches employing these biomedical BERT-based models do not outperform the existing approaches using Siamese Network with BioWordVec embeddings for the UMLS synonymy prediction task, (2) the original BioBERT large model that has not been pre-trained with the UMLS outperforms the SapBERT models that have been pre-trained with the UMLS, and (3) using the Siamese Networks yields better performance for synonymy prediction when compared to using the biomedical BERT models and GANs.

11 Acknowledgments

We would like to thank Nguyen et al. for sharing the training data used in (Nguyen, Yip, and Bodenreider 2021; Nguyen and Bodenreider 2021). We would also like to thank Liu et al. for providing the additional pre-trained SapBERT models (i.e., BioBERT + SapBERT, BlueBERT + SapBERT, UMLSBERT + SapBERT, VanillaBERT + SapBERT) (Liu et al. 2020).

References

Alsentzer, E.; Murphy, J. R.; Boag, W.; Weng, W.-H.; Jin, D.; Naumann, T.; and McDermott, M. 2019. Publicly available clinical BERT embeddings. *arXiv preprint arXiv:1904.03323*.

Bodenreider, O. 2004. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1): D267–D270.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5: 135–146.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Euzenat, J.; Meilicke, C.; Stuckenschmidt, H.; Shvaiko, P.; and Trojahn, C. 2011. Ontology alignment evaluation initiative: six years of experience. In *Journal on data semantics XV*, 158–192. Springer.

Jawahar, G.; Sagot, B.; and Seddah, D. 2019. What does BERT learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4): 1234–1240.

Lewis, P.; Ott, M.; Du, J.; and Stoyanov, V. 2020. Pretrained Language Models for Biomedical and Clinical Tasks: Understanding and Extending the State-of-the-Art. In *Proceedings of the 3rd Clinical Natural Language Processing Workshop*, 146–157.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, F.; Shareghi, E.; Meng, Z.; Basaldella, M.; and Collier, N. 2020. Self-alignment pretraining for biomedical entity representations. *arXiv preprint arXiv:2010.11784*.

Liu, N. F.; Gardner, M.; Belinkov, Y.; Peters, M. E.; and Smith, N. A. 2019. Linguistic Knowledge and Transferability of Contextual Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1073–1094. Minneapolis, Minnesota: Association for Computational Linguistics.

Michalopoulos, G.; Wang, Y.; Kaka, H.; Chen, H.; and Wong, A. 2020. Umlsbert: Clinical domain knowledge augmentation of contextual embeddings using the unified medical language system metathesaurus.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Nguyen, V.; and Bodenreider, O. 2021. Adding an Attention Layer Improves the Performance of a Neural Network Architecture for Synonymy Prediction in the UMLS Metathesaurus. In *MedInfo*.

Nguyen, V.; Yip, H. Y.; and Bodenreider, O. 2021. Biomedical Vocabulary Alignment at Scale in the UMLS Metathesaurus. In *Proceedings of the Web Conference 2021*, 2672–2683.

Peng, Y.; Yan, S.; and Lu, Z. 2019. Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, 58–65.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in neural information processing systems*, 5998–6008.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903*.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Zhang, F.; Liu, X.; Tang, J.; Dong, Y.; Yao, P.; Zhang, J.; Gu, X.; Wang, Y.; Shao, B.; Li, R.; et al. 2019a. Oag: Toward linking large-scale heterogeneous entity graphs. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2585–2595.

Zhang, Y.; Chen, Q.; Yang, Z.; Lin, H.; and Lu, Z. 2019b. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, 6(1): 1–9.