Evaluating Biomedical Word Embeddings for Vocabulary Alignment at Scale in the UMLS Metathesaurus Using Siamese Networks

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] 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*, *visheshj123@gwu.edu*, *hyip@email.sc.edu*,

Abstract

Recent work uses a Siamese Network, initialized with BioWordVec embeddings (distributed word embeddings), for predicting synonymy among biomedical terms to automate a part of the UMLS (Unified Medical Language System) Metathesaurus construction process. We evaluate the use of contextualized word embeddings extracted from nine different biomedical BERT-based models for synonymy prediction in the UMLS by replacing BioWordVec embeddings with embeddings extracted from each biomedical BERT model using different feature extraction methods. Surprisingly, we find that Siamese Networks initialized with BioWordVec embeddings still outperform the Siamese Networks initialized with embedding extracted from biomedical BERT model.

1 Introduction

The UMLS (Bodenreider, 2004) is a biomedical terminology integration system that includes over 200 source vocabularies[1]. The UMLS Metathesaurus construction process organizes synonymous terms from these source vocabularies into *concepts*. The current Metathesaurus construction process uses a lexical similarity model and semantic preprocessing to determine synonymy, followed by a human review. The large scale and diversity of the Metathesaurus make the construction process very challenging, tedious, and error-prone. Therefore, to assist the UMLS Metathesaurus construction process, Nguyen et al. introduced the UMLS Vocabulary Alignment (UVA) task, or synonymy prediction task (Nguyen et al., 2021). They designed and train a Siamese Network to predict if two UMLS atoms are synonymous. The Siamese Network is initialized using BioWordVec embeddings, learned using fastText (Bojanowski et al., 2017). Given the recent successful use of contextualized word embeddings, extracted from Transformer models, for different downstream NLP tasks (Devlin et al., 2019; Vaswani et al., 2017; Peters et al., 2019), we explore the use of contextualized embeddings extracted from several distinct biomedical BERT-based language models.

Objectives. 1) Find which type of word embeddings, including contextualized embeddings, achieves the best performance when used with the Siamese Network for the synonymy prediction (or UVA) task. 2) Find which feature extraction method works best to extract word embeddings from the biomedical BERT models for optimal performance. 3) Find the best hyperparameters and optimization of the prediction task to train the Siamese Networks for the UVA task.

Approach. 1) We analyze the performance of the Siamese Networks initialized with embeddings from nine different biomedical BERT models for synonymy prediction. 2) We explore different feature extraction techniques to extract BERT embeddings. 3) We conduct a grid search and optimization of the prediction task to train the Siamese Networks.

Contributions. 1) We conduct an extensive analysis to extract embeddings from nine different biomedical BERT models using four feature extraction techniques. 2) Somewhat surprisingly, we find that Siamese Networks still achieve the highest performance for synonymy prediction when initialized with BioWordVec embeddings. 3) We find that no single feature extraction method works well across the different biomedical BERT models. 4) With a thorough grid search, we find substantial increases in F1-Score (e.g., 2.43%), when compared to the default hyperparameters. 5) Overall, our work contributes to defining best practices for the use of embeddings in Siamese Networks. See https://arxiv.org/abs/2109.13348 for an extension of this paper as it presents an extended analysis of the experiments and additional results.

2 UMLS - Knowledge Representation

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. All atoms in the UMLS Metathesaurus are assigned a unique identifier (AUI). Atoms that are synonymous are grouped into a single concept identified with a concept unique identifier (CUI). Table 1 contains examples of synonymous atoms and the identifiers assigned to each respective atom for a

[1] https://uts.nlm.nih.gov/uts/

Proceedings of the Third Workshop on Insights from Negative Results in NLP, pages 82 - 87

Tuple	Atom String	Source	AUI	CUI
t_1	Headache	MSH	A0066000	C0018681
t_2	Headaches	MSH	A0066008	C0018681
t_3	Cephalodynia	MSH	A26628141	C0018681
t_4	Cephalodynia	SNOMEDCT_US	A2957278	C0018681

Table 1: Examples tuples from UMLS consisting of an atom string, its source vocabulary name, its unique atom identifier (AUI), and its concept unique identifier (CUI). All tuples in the example table are synonymous and, hence, have the same CUI.

particular concept. For example, the term "Cephalodynia" appearing in both MSH and SNOMEDCT_US has different AUIs as shown in Table 1. Additionally, the strings "Headache" and "Headaches" have different AUIs because of the lexical variation (see Table 1). We use the 2020AA version of the UMLS, which contains 15.5 million atoms from 214 source vocabularies grouped into 4.28 million concepts.

3 Problem Formulation

An essential part of the UMLS construction process is identifying similar atoms across source vocabularies to integrate knowledge from different sources accurately. The UMLS Vocabulary Alignment (UVA) – or synonymy prediction – task is to identify synonymous atoms by measuring the similarity among pairs of atoms. A machine learning model should be able to identify the synonymous atoms are that lexically: *similar but are not synonymous* and *dissimilar but are synonymous*. Let (t_i, t_j) be a pair of input tuples, where $i \neq j$. Each tuple is initialized from a different source vocabulary in the form of (str, src, aui), where str is the atom string, src is the source vocabulary, and aui is the atom unique identifier (AUI). Let $f : T \times T \rightarrow 0, 1$ 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 atom strings (str_i, str_j) from t_i and t_j are synonymous and belong to the same concept (and hence, share same the CUI).

4 Dataset

We thank Nguyen et al. for sharing the dataset used in their work (Nguyen et al., 2021; Nguyen and Bodenreider, 2021). The dataset is created using the 2020AA release of the UMLS Metathesaurus. We use the ALL dataset for our study. The training and validation dataset contains a total of 192,400,462 examples, where 88.4% of the examples are negative examples. The testing dataset set contains a total of 173,035,862 examples, where 96.8% of the examples are negative examples. We refer the readers to Section 4.2 of (Nguyen et al., 2021) for a detailed description.

5 Related Work

We first describe the Siamese Networks for the UVA then describe the biomedical BERT variants.

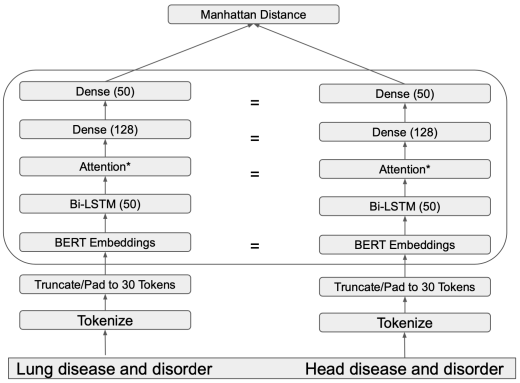

Figure 1: Siamese Network used for Synonymy Prediction. Nguyen et al. use BioWordVec embeddings, whereas we use contextualized word embeddings. "*" indicates optional attention layer.

Siamese Networks for the UVA Task

Nguyen et al. assess the similarity of atoms using lexical features of the atom strings (str). The authors design a Siamese Network that inputs a pair of atom strings, and outputs a similarity score between 0 and 1, $sim(str_i, str_j) \in [0, 1]$ (see Figure 1). The inputs are preprocessed, tokenized, and then sent through an initial embedding layer initialized with BioWordVec embeddings (Zhang et al., 2019). The word embeddings are then fed into Bidirectional Long Short Term Memory (Bi-LSTM) layers, followed by two dense layers. All atom pairs with a similarity > 0.5 are considered synonyms (using the Manhattan distance). Their deep learning model has a precision of 94.64%, recall of 94.96% and an F1-Score of 94.8% and outperforms a rule-based approach for synonymy prediction by 23% in recall, 2.4% in precision, and 14.1% in F1-Score. In their follow-up work, Nguyen et al. add an attention layer after the Bi-LSTM layers that improves the precision by +3.63% but decreases recall by 1.42%.

Biomedical BERT Models

In this section, we summarize the specific biomedical BERT variants used in this study. For brevity, we focus on biomedical BERT variants and omit the general presentation of BERT. We refer the interested reader to (Devlin et al., 2019) for details.

Table 2 compares the different biomedical BERT models used in this benchmarking study. To limit the scope of the biomedical BERT models, we only include models that have been pretrained with data from biomedical sources, such as biomedical terminologies (e.g., UMLS vocabularies), biomedical literature (e.g., PubMed), and clinical notes (e.g., MIMIC-III).

BioBERT: BioBERT is initialized from BERT and then pretrained on PubMed abstracts and PubMed Central (PMC) full-text articles (Lee et al., 2020). We use both BioBERT-Base and BioBERT-Large.

BlueBERT: BlueBERT is initialized with BERT weights provided by (Devlin et al., 2019) and further

Model Type	Embed. Dim.	Vocab Size	Token Size
BioWordVec	200	268,158,600	-
BioBERT (+ SapBERT)	768	28,996	13,230,336
BioBERT-Large (Cased)	1024	58,996	28,530,688
BlueBERT	1024	30,522	25,358,336
SapBERT	768	30,522	21,035,520
UMLSBERT (+ SapBERT)	768	28,996	13,230,336
BlueBERT+ SapBERT	768	30,522	19,018,752
VanillaBERT + SapBERT	768	30,522	19,018,752

Table 2: Comparison of different biomedical word embeddings in terms of the embedding dimension, vocabulary size, and the number of tokens.

pretrained with the PubMed Abstract and MIMIC-III datasets. We use BlueBERT-Large in our work.

SapBERT: SapBERT provides the current state-of-the-art (SOTA) results for six medical entity linking benchmarking datasets (Liu et al., 2021). SapBERT is trained on the UMLS with 4*M*+ concepts and 10*M*+ synonyms from over 150 vocabularies.

UMLSBERT: UMLSBERT is initialized with the pretrained Bio_ClinicalBERT model (Alsentzer et al., 2019) and pretrained with the MLM task on the MIMIC-III dataset with additional modifications.

{BioBERT, BlueBERT, UMLSBERT, VanillaBERT} + SapBERT: The SapBERT authors pretrain additional variants of SapBERT that are initialized using different BERT variants. We refer the reader to (Liu et al., 2021) for a detailed description.

6 Approach

To analyze the performance of the different embeddings extracted from the various BERT models, we train the Siamese Network end to end, similar to (Nguyen et al., 2021; Nguyen and Bodenreider, 2021). We investigate the use of the nine biomedical BERT models (mentioned in Section 5) as a source of word embeddings. Our experimental setup of consists of two primary steps for each of the Siamese Networks (with and without attention): 1) Feature extraction of word embeddings from biomedical BERT Models. 2) Grid search of optimal hyperparameters and optimization. Our code will be available at https://anonymous.4open.science/r/uva_embedding_benchmarking-8124/. For the training and testing data, we recommend reaching out to Nguyen et al. (Nguyen et al., 2021; Nguyen and Bodenreider, 2021).

Feature Extraction for the Siamese Network

BioWordVec has a fixed word embedding for each word or term (e.g., UMLS atom). For transformer models, word embedding extraction is not as straightforward because different layers of BERT capture different types of features (Jawahar et al., 2019; Liu et al., 2019; Reimers and Gurevych, 2017; Peters et al., 2018; van Aken et al., 2019; Devlin et al., 2019). We initialize Siamese Networks with token embeddings instead of word embeddings to use BERT models for the UVA task. To extract token embeddings for UMLS atoms from each BERT model, we: 1) Tokenize the atom strings using the model-specific vocabulary. 2) Create a token id tensor by mapping the token strings to their vocabulary indices. 3) Create a segment id tensor. 4) Feed the token id and segment id tensors in to the BERT model (in eval mode). 5) Create a separate token embedding matrix to initialize the Siamese Networks using each of the following methods:

- 1^{st} token embedding and last layer
- 1^{st} token embed. and avg. of last 4 layers
- Last token embedding and last layer
- Last token embed. and avg. of last 4 layers
- Avg. token embedding and last layer
- Avg. token embed. and avg. of last 4 layers

Of note, we do not use the "CLS" sentence representation as the word embedding for UMLS atoms because the Bi-LSTMs layers require a sequence as input. We only use the atom string to extract token embeddings because all vocabularies in the UMLS have this characteristic in common. In summary, we extract two sets of embeddings from each model (the 12^{th} layer and average of the 9^{th} to 12^{th} layers) and use three different types of token embeddings (the first and last occurrence of the token in the dataset and the average embedding of each occurrence of the token in the dataset).

Grid Search and Optimization

The performance of deep learning models highly depends on the selection of hyperparameters (Hutter et al., 2014; Bergstra and Bengio, 2012; Reimers and Gurevych, 2017). Prior work by Nguyen et al. uses a fixed set of hyperparameters. Therefore, we conduct a grid search for the best-performing models to thoroughly investigate the performance of the Siamese Networks. Hyperparameters used in our experiment include optimizer (SGD, Adam) and learning rate (0.00001, 0.0001, 0.001, 0.01, 0.1). To limit computational cost, we conduct a grid search for the following Siamese Networks: BioWordVec (**BWV**), BioWordVec + Attention (**BWV + Att.**), SapBERT avg. token embedding extracted by averaging the last 4 layers (**SB Avg_Token + Avg_Last_4**), SapBERT avg. token embedding extracted from the last layer + Attention (**SB Avg_Token + Last_Lay + Att.**). Additionally, Nguyen et al. provide no rationale for the similarity threshold of 0.5 between the learned representations of two atoms. Therefore, we search for the best threshold for prediction based on the precision-recall curve to find a threshold that maximizes the F1-Score.

7 Results and Discussion

Table 3 presents the synonymy prediction results using embeddings extracted from BERT models and BioWordVec embeddings. The *Token Type* and *Extraction Method* columns indicate the feature extraction method that was used to initialize the model.

Performance with BERT Embeddings: We find that Siamese Networks initialized with BioWordVec still outperform all models initialized with embeddings ex-

	Siamese Network without Attention (Nguyen et al., 2021)								Siamese Network with Attention (Nguyen and Bodenreider, 2021)							
Embedding Type	**Token Type**	**Extraction Method**	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**	**Token Type**	**Extraction Method**	**Threshold**	**Accuracy**	**Precision**	**Recall**	**F1-Score**	**AUC**
BioWordVec	-	-	0.5612	0.9941	0.9075	0.9127	0.9101	0.9909	-	-	0.5490	0.9939	0.9056	0.9067	0.9061	0.9907
BioWordVec w. SGD, lr = 0.001	-	-	0.5587	0.9942	0.9087	0.9146	**0.9116**	0.9913	-	-	0.5507	0.9941	0.9078	0.9102	0.9090	0.9910
SapBERT	Avg.	Avg. Last 4	0.5802	0.9892	0.8496	0.8092	0.8289	0.9848	Avg.	Last Layer	0.5607	0.9902	0.8682	0.8247	0.8459	0.9855
SapBERT w. SGD, lr = 0.0001	-	-	-	-	-	-	-	-	Avg.	Last Layer	0.5979	0.9913	0.8824	0.8459	**0.8638**	0.9830
SapBERT w. Adam, lr = 0.0001	-	-	-	-	-	-	-	-	Avg.	Avg. Last 4	0.59	0.9912	0.8840	0.8372	0.8600	0.9830
BioBERT	First	Last Layer	0.5643	0.9853	0.7955	0.7380	0.7657	0.9758	Avg.	Avg. Last 4	0.5481	0.9862	0.81	0.7504	0.779	0.9774
BioBERT_Large	Avg.	Last Layer	0.5438	0.9881	0.8400	0.7810	0.8095	0.9807	Avg.	Last Layer	0.5438	0.9881	0.84	0.781	0.8095	0.9807
BlueBERT	First	Last Layer	0.5680	0.9859	0.8066	0.7424	0.7732	0.9765	Avg.	Last Layer	0.5500	0.9872	0.8247	0.7677	0.7952	0.9792
UMLSBERT	Avg.	Avg. Last 4	0.5755	0.9852	0.7921	0.7371	0.7636	0.9754	Avg.	Avg. Last 4	0.5501	0.9862	0.8151	0.7415	0.7765	0.9764
UMLSBERT + SapBERT	Avg.	Avg. Last 4	0.5543	0.9854	0.7948	0.7432	0.7681	0.9769	Avg.	Avg. Last 4	0.5452	0.9857	0.7992	0.7485	0.773	0.9771
BlueBERT + SapBERT	Avg.	Avg. Last 4	0.5810	0.9868	0.8154	0.7651	0.7895	0.9798	Avg.	Avg. Last 4	0.5596	0.9875	0.831	0.7701	0.7994	0.9797
BioBERT + SapBERT	Avg.	Avg. Last 4	0.5756	0.9851	0.7904	0.7348	0.7616	0.9756	Avg.	Avg. Last 4	0.5511	0.9861	0.81	0.7465	0.7769	0.9769
VanillaBERT + SapBERT	Avg.	Avg. Last 4	0.5614	0.9866	0.8125	0.7633	0.7872	0.9791	Avg.	Avg. Last 4	0.5467	0.9874	0.8268	0.772	0.7984	0.9801

Table 3: Results for Siamese Networks trained for 100 iterations initialized using different embeddings using the best prediction threshold (single run point estimates). Rows marked with "**w.**" contain the performance of the models after grid search.

tracted from biomedical BERT models. Though surprising, Schulz and Juric also find that current embeddings are limited in their ability to adequately encode medical terms when tested on large-scale datasets (Schulz and Juric, 2020).

Moreover, using a BERT model trained on more relevant domain-specific data and the right task yields more substantial gains. In particular, the SapBERT model, whose embeddings achieve the highest performance, is trained on PubMed and incorporates knowledge from the UMLS Metathesaurus by using semantic type embeddings and modifying the MLM task to indicate if which words belong to the same concept. These changes likely indicate why it outperforms the other biomedical BERT models for our task.

Feature Extraction for Biomedical BERT Models: Based on our experiments, no single feature extraction method provides the most useful embedding for all BERT models. However, results indicate that averaging all token embeddings and using the average of the last four hidden layers seems to work well for many of the models. The Siamese Network + Attention initialized with the average token embedding extracted from the last layer of SapBERT achieves the best F1-Score.

Performance after Grid Search: As mentioned in Section 6, we limit the grid search to the four best performing models: BWV, BWV + Att., SB Avg_Token + Avg_Last_4, and SB Avg_Token + Last_Lay + Att. Our grid search results indicate that the Siamese Network without attention outperforms the Siamese Network with attention when initialized with BioWordVec embeddings. Additionally, there is a 2.43% increase in F1-Score for the Siamese Network with attention and a 3.11% increase in F1-Score for the Siamese Network w.o. attention. Reducing the batch size leads to early stopping for all models but at the cost of performance (e.g, 4.67% drop in F1-Score for BWV + Att. w. SGD).

Optimizer. For the four best performing models, we see that SGD works better in three of the cases. For only one model, Adam performs similarly to SGD with a higher F1-Score by 0.16%. There is a 1% increase in F-1 Score for the Siamese Network with Attention initialized with SB + Avg_Token + Last_Lay embeddings. Using the SGD optimizer leads to earlier convergence for when using biomedical BERT embeddings.

Learning Rate. Regardless of the optimizer, increasing the learning rate (LR) to 0.01 and 0.1 leads to early stopping and results in poor F1-Scores. With a LR of 0.0001, the performance for the Siamese Networks initialized with SapBERT embeddings extracted using the average token and the last layer of the SapBERT model, F1-Score increases by about 0.6% for the model with attention and a 3.11% increase for the model without attention. Reducing the LR further decreases performance for Siamese Networks using BWV embeddings.

Threshold. The best performing thresholds range from 0.5438 to 0.581. On average using the best thresholds results in 0.0086% increase in F1-Score for the Siamese Networks without attention (results omitted due to space). Hence, 0.5 is an acceptable threshold.

8 Conclusion

We investigate if contextualized embeddings extracted from biomedical BERT-based language models can improve the performance of Siamese Networks, introduced by (Nguyen et al., 2021; Nguyen and Bodenreider, 2021), to predict synonymy in the UMLS Metathesaurus. Despite the excellent performance of BERT models on biomedical NLP tasks, BioWordVec embeddings still remain competitive for the UVA task. This confirms the importance of investigating the use of traditional distributed word embeddings. Among the biomedical BERT models, SapBERT trained on UMLS data performs best, suggesting the importance of using a model trained on datasets directly relevant to the task at hand. Finally, we demonstrate the importance of exploring different feature extraction methods and hyperparameter tuning for deep learning models.

9 Acknowledgments

The authors thank Liu et al. for providing the additional pretrained SapBERT models (Liu et al., 2021) and a cooperative AI Institute grant (AI-EDGE), from the National Science Foundation under CNS-2112471.

References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

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

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 2014. An efficient approach for assessing hyperparameter importance. In *International conference on machine learning*, pages 754–762. PMLR.

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

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Fangyu Liu, Ehsan Shareghi, Zaiqiao Meng, Marco Basaldella, and Nigel Collier. 2021. Self-alignment pretraining for biomedical entity representations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4228–4238.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 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)*, pages 1073–1094, Minneapolis, Minnesota. Association for Computational Linguistics.

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

Vinh Nguyen, Hong Yung Yip, and Olivier Bodenreider. 2021. Biomedical Vocabulary Alignment at Scale in the UMLS Metathesaurus. In *Proceedings of the Web Conference 2021*, pages 2672–2683.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. *ACL 2019*, page 7.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

Claudia Schulz and Damir Juric. 2020. Can embeddings adequately represent medical terminology? new large-scale medical term similarity datasets have the answer! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8775–8782.

Betty van Aken, Benjamin Winter, Alexander Löser, and Felix A Gers. 2019. How does bert answer questions? a layer-wise analysis of transformer representations. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1823–1832.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yijia Zhang, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019. BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific data*, 6(1):1–9.

A Dataset

We thank Nguyen et al. for sharing the dataset used in their work (Nguyen et al., 2021; Nguyen and Bodenreider, 2021). To get a copy of the dataset, please sign the UMLS License Agreement and email Nguyen to receive the dataset.

B Experimental Details

We first train both Siamese Networks (with attention (Nguyen and Bodenreider, 2021) and without attention (Nguyen et al., 2021)) with the default hyperparameters for each biomedical BERT model with each of the different embedding extraction methods. The default hyperparameters rely on Adam as the optimizer with a learning rate of 0.001 and 8192 examples in batch. This results in 20 different Siamese Networks, each trained for 100 epochs. Next, we take the best performing Siamese models initialized with BERT embeddings and the two Siamese models initialized with BioWordVec embeddings and conduct a grid search to find the optimal hyperparameters. We conduct a grid search for a total of 4 Siamese Networks and evaluate each model using the following metrics: Accuracy, Precision, Recall, F1-Score, and AUC.

All experiments are run using a High Performance Computing cluster. The typical run time for a Siamese Network with BioWordVec embeddings is 48 hours for 100 iterations using a v100x NVIDIA GPU and requires about 220 GB of memory. A Siamese Network trained with BERT embeddings takes about 72 hours for 100 iterations using a v100x NVIDIA GPU and requires about 220 GB of memory. The training time is further increased to 88 hours for Siamese Networks trained with embeddings of dimensions 1024 (i.e., BioBERT-Large and BlueBERT embeddings).

C Limitations

Our work evaluates biomedical word embeddings extracted from BERT-based models for the Siamese Networks introduced by (Nguyen et al., 2021; Nguyen and Bodenreider, 2021). Our list of biomedical BERT models does not include all models; we consider the most recent biomedical BERT models that have achieved SOTA performance on NLP tasks. The narrow focus of our work allows us to conduct a thorough analysis of the embedding extraction methods and hyperparameters using nine different BERT models for two variants of the Siamese Network. However, our experimental setup is reproducible for similar NLP tasks.

As an additional exercise to test the usability of transformer based embeddings, we attempt to use the "CLS" sentence representation of the UMLS atoms. For a pair of UMLS atoms, we extract the "CLS" sentence representation of each UMLS atom and compute the similarity of the representation using both the Cosine and Manhattan distance functions. We find that this approach does not work well ($< 30\%$ accuracy). As future work, we can investigate if adding a deep neural net (different from a Siamese Network) can improve the performance.