Can Embeddings Adequately Represent Medical Terminology? New Large-Scale Medical Term Similarity Datasets Have the Answer![*]

Claudia Schulz and **Damir Juric**
Babylon Health
London, SW3 3DD, UK
{firstname.lastname}@babylonhealth.com

Abstract

A large number of embeddings trained on medical data have emerged, but it remains unclear how well they represent medical terminology, in particular whether the close relationship of semantically similar medical terms is encoded in these embeddings. To date, only small datasets for testing medical term similarity are available, not allowing to draw conclusions about the generalisability of embeddings to the enormous amount of medical terms used by doctors. We present multiple automatically created large-scale medical term similarity datasets and confirm their high quality in an annotation study with doctors. We evaluate state-of-the-art word and contextual embeddings on our new datasets, comparing multiple vector similarity metrics and word vector aggregation techniques. Our results show that current embeddings are limited in their ability to adequately encode medical terms. The novel datasets thus form a challenging new benchmark for the development of medical embeddings able to accurately represent the whole medical terminology.

1 Introduction

AI has recently enabled major breakthroughs in health-care (**?**; **?**), but it often requires to develop and adapt AI algorithms specifically to the domain (**?**). Especially *medical terminology* differs largely from commonly used language, so a crucial step towards the successful use of AI in health-care is to ensure that medical terminology is adequately encoded. Doctors know a vast amount of medical terms, including which of them are similar (e.g. synonyms of a disease), but it is so far unclear whether embeddings share this deep understanding of medical terminology.

To investigate this, small datasets of a few hundred medical concept pairs with a similarity score have been created (**?**; **?**; **?**). However, testing medical language representation models on such restricted datasets does not allow to draw any reliable conclusions about the generalisability of these models to the whole medical terminology.

In this paper, we aim to overcome the generalisation problem by creating *large-scale* medical term similarity datasets, the largest consisting of more than 600,000 term pairs. Semantically similar medical terms are extracted from the SNOMED ontology (**?**) and we propose a novel strategy for creating pairs of dissimilar terms, resulting in datasets that are highly challenging for embeddings. To ensure the correctness and reliability of the completely *automatically created* datasets, we perform a manual evaluation with doctors, confirming the datasets' *high quality* and correctness in representing medical term similarity. We make our code for dataset construction freely available[1], allowing the easy recreation for future research.[2]

We evaluate publicly available medical word and contextual embeddings on both our new and existing datasets to compare what conclusions can be drawn from either. We also compare and analyse the effects of using different similarity metrics, including the commonly used cosine similarity as well as recently suggested rank-based measures (**?**). We find that existing datasets are too small to realistically reflect the complexity of medical terminology and that they do not reveal significant performance differences between embeddings. In contrast, our new benchmark datasets highlight significant differences between embeddings as well as their inability to adequately represent medical terminology.

As a second evaluation of embeddings' ability to represent medical terminology, we propose a category separation task and a new error metric. A good medical terminology representation should identify terms in similar categories as being closer than terms in dissimilar categories.

Importantly, our large-scale datasets are not only of interest for testing embeddings on the term similarity task, but also less obvious tasks, such as reducing the time to manually create and verify medical ontologies.

Our contributions are: 1) we introduce highly challenging large-scale medical term similarity benchmarks, 2) we reveal that existing datasets are too small to discover significant performance differences between embeddings, whereas our datasets do, and 3) we find that current embeddings cannot adequately represent medical terminology.

[*]Please refer to this version for up-to-date experimental results.

[1]https://github.com/babylonhealth/medisim

[2]IHTSDO prohibits to publish data derived from SNOMED CT.

2 Related Work

Many benchmark datasets are available to evaluate semantic textual similarity (STS) methods, both on word and sentence level (**?**), but most of them are concerned with everyday words and sentences. However, a method with good performance on these datasets is likely to utterly fail when applied to medical terminology. For the medical domain, only a handful of similarity datasets exist, as summarised in Table 1, all of them manually curated and comprising only commonly used medical concepts. Furthermore, half contain only single-word terms, although medical terms are frequently made of multiple words.

Note that we here focus on medical *terms* rather than *concepts*. Concepts are abstract entities, represented as codes in ontologies such as SNOMED, which are described by some (potentially more than one) term.

Dataset	Size	Scores	Source	MW
Hliaoutakis (?)	36	0-1	MeSH	47%
MiniMayoSRS (?)	29	1-4	UMLS	47%
MayoSRS (?)	101	1-4	UMLS	44%
UMNSRS-Sim (?)	566	0-1600	UMLS	2%
UMNSRS-Sim-mod (?)	449	0-1600	UMLS	0%
UMNSRS-Rel (?)	587	0-1600	UMLS	2%
UMNSRS-Rel-mod (?)	458	0-1600	UMLS	0%
Bio-SimLex (?)	988	0-10	PubMed	0%
Bio-SimVerb (?)	1000	0-10	PubMed	0%

Table 1: Existing datasets and % of multi-word (MW) terms.

Regarding the automatic creation of medical terminology datasets, **?** (**?**) extract pairs of related medical concepts using a bootstrapping approach, resulting in various datasets of related medical UMLS codes extracted from different sources. In contrast, our dataset focuses on *similar* medical terms. **?** (**?**) use the same approach as **?** to create a dataset from SNOMED's 'is-a' and other relationships between disorders and drugs. **?** (**?**) extract 8000 synonym concepts from relationships in UMLS and then randomly create 1.6M negative pairs, whereas we also apply a more sophisticated negative sampling strategy. Neither of these automatically created datasets has been evaluated regarding its quality nor are these datasets publicly available or easy to recreate.

Like us, **?** (**?**) compare different methods for aggregating embeddings of words to measure similarity between multi-word medical concepts. They train their own medical word embeddings and compare summing and averaging these vectors to the performance of concept embeddings. Instead of training yet another embedding model, we use existing embeddings and experiment with a larger variety of word vector aggregation techniques. Furthermore, we use not only cosine similarity to measure vector similarity but also apply rank-based metrics.

3 New Large-Scale Datasets

We choose SNOMED Clinical Terms (CT) as the basis for our medical term similarity datasets as it is the "most comprehensive, multilingual clinical healthcare terminology in the world"[3]. As of the January 2019 release, SNOMED CT comprises 349,548 medical concepts. SNOMED CT is thus ideal for our purpose of creating datasets that adequately represent the whole medical terminology used by doctors. We create *binary classification* datasets, consisting of pairs of medical terms classified as semantically similar (1) or dissimilar (0). Note that the dataset creation is fully automatic, not requiring any costly manual annotation.

3.1 Extracting Positive Instances

In the first step of the dataset creation, pairs of semantically similar terms are extracted from SNOMED CT.

SNOMED CT Synonyms. Each SNOMED CT concept is associated with a unique *fully specified name* (FSN) and may have one or more *synonyms*, e.g. the FSN 'Sprain of ankle' has a synonym 'Ankle sprain'. Clearly, synonyms are semantically very similar to the FSN, so we construct a dataset consisting of all FSN-SYNONYM term pairs as positive instances.

We first filter out concepts from the model component module, which provides metadata and organisational concepts such as 'Fully specified name' and 'Entire term case sensitive'. For each remaining active concept, we obtain its current FSN and delete parentheses indicating the concept's category, e.g. 'Malaria (disorder)'. We pair the modified FSN with all its active synonyms that are not equivalent to the modified FSN, resulting in the positive instances of our FSN-SYNONYM medical term similarity dataset. Since each synonym of an FSN is similar to the FSN, we expect that synonyms are also similar to each other. Based on this assumption, we obtain a second dataset SYNONYM-SYNONYM, by adding synonym-synonym term pairs to the FSN-SYNONYM dataset.

SNOMED CT Deactivated Concepts. Synonyms are the most obvious similar terms, but we can leverage another type of information about similar terms in SNOMED CT: in every release, some concepts are deactivated and replaced by a different active concept. An *association* between the concepts gives the reason for replacement: 1) POSSIBLY-EQUIVALENT-TO indicates that the deactivated concept is ambiguous and that the active concept represents one of its possible meanings, 2) REPLACED-BY applies to erroneous or obsolete deactivated concepts and their suitable replacement, and 3) SAME-AS refers to (semantically) duplicate concepts. Clearly these associations describe pairs of similar concepts, which we transform into similar term pairs.

Again, we first disregard pairs containing concepts from the model component module. For each concept we then use the most recent FSN as the term and again drop parentheses specifying medical categories. In addition, we drop any

[3] https://www.snomed.org/snomed-ct/five-step-briefing

Dataset	Size	Pos	Neg-R	Neg-L
FSN-SYN.	451,256	16.53	37.40	7.97
– easy	78,466	2.08	36.90	8.37
– hard	372,790	19.57	37.51	7.89
SYN.-SYN.	726,158	16.57	35.10	8.00
– easy	122,864	2.21	35.33	8.66
– hard	603,294	19.50	35.05	7.86
POSS.-EQUIV.-TO	57,528	33.92	49.33	17.95
– easy	1,474	3.96	30.10	12.33
– hard	56,054	34.71	49.84	18.10
REPLACED-BY	7,082	20.00	33.93	11.49
– easy	654	2.94	30.59	11.29
– hard	6,428	21.74	34.27	11.51
SAME-AS	20,324	22.71	33.30	10.71
– easy	2,570	2.62	27.17	10.81
– hard	17,754	25.62	34.19	10.70

Table 2: Our new datasets, respective number of (positive & negative) term pairs (Size), average Levenshtein distance of the Pos(itive) and Neg(ative) instances with the R(andom) and L(evenshtein) strategies.

"[D]" at the start or end of a FSN, which SNOMED CT uses to indicate deprecated names. We collect the three types of term pairs in three separate datasets to investigate if any of them are easier or more difficult to identify as similar.

Easy vs. Hard Datasets. The extracted positive instances are expected to all be semantically similar terms. However, *lexically* the terms can be very similar, e.g. 'Sacrum sprain' and 'Sacral sprain', or completely different, e.g. 'Malaria' and 'Paludism'. The latter requires a much deeper understanding of medical terminology, whereas the former can be guessed from the surface similarity. To investigate how deep the understanding of term representation models is, we split the positive instances of each dataset into *easy* and *hard* ones. The difficulty is measured in terms of *Levenshtein distance* between the two terms. We experimentally choose a threshold of 5, so that the hard splits mainly contain term pairs with fundamentally different words. Table 2 illustrates the average Levenshtein distance of term pairs in the easy and hard datasets.

3.2 Creating Negative Instances

SNOMED CT explicitly specifies similar terms (e.g. synonyms), but not dissimilar ones. Naïvely, we can thus consider all term pairs not explicitly specified as similar to be dissimilar. For each dataset, our *random* negative sampling strategy therefore matches the first term of each positive instance to a randomly selected term from another instance. As can be seen in Table 2, this leads to negative term pairs with very high average Levenshtein distance, i.e. they are mostly made of completely different words with no lexical overlap. This may make it easy for models to correctly identify these term pairs as dissimilar.

To test if models in fact have a deep understanding of medical terminology, we apply a second negative sampling strategy to create more difficult negative instances: the first term of each positive instance is matched to the term with closest Levenshtein distance that is not (directly or indirectly) specified to be similar. Table 2 illustrates that this leads to a much lower Levenshtein distance between negative term pairs than using the random strategy. In the hard datasets, the Levenshtein distance of negative instances is even lower than that of positive ones. Thus, for the hard datasets with Levenshtein negative sampling, lexical similarity between terms will not help at all to distinguish similar and dissimilar pairs.

For both negative sampling strategies, we construct the same number of negative instances as there are positive ones to obtain balanced datasets. The split into easy and hard combined with our two negative sampling strategies results in 20 different datasets. In contrast to existing datasets, where most medical terms are single words (see Table 1), SNOMED CT terms are mostly made of multiple words, resulting in 92% multi-word terms in our datasets. This makes the datasets both more realistic, as multi-word terms are more complex and more fine-grained, and more challenging for medical terminology representation models.

3.3 Quality Evaluation

To verify the quality of our automatically created datasets, we perform a manual evaluation with three doctors. For each dataset we randomly select 30 positive and 30 negative instances. Each doctor thus evaluates (the same) 1200 term pairs. The doctors are presented with term pairs without knowing which dataset they belong to and have to decide if the terms are similar in the sense that they could be used interchangeably in consultation notes. They are allowed to look up terms of which they do not remember the meaning and can choose "don't know" instead of "same"/"not same" for a pair of terms. To compare the automatically created similarity scores in our datasets to the doctors' assessment, we first combine the three doctors' decisions into a *ground truth score* using majority voting. If there is no majority, we assign no ground truth score (NaN).

The *difficulty* (regarding human judgement) of each dataset is measured in terms of the doctors' inter-annotator agreement (IAA) and the amount of disagreement (NaNs). The overall IAA is Krippendorff's $\alpha = 0.85$, with the lowest agreement on a dataset being $\alpha = 0.65$ and the highest $\alpha = 0.95$ (see Table 3). The doctors' decisions can thus be considered reliable. The mostly higher IAA for easy datasets compared to hard ones confirms the intended difficulty difference. Importantly, there is no notable difficulty difference between the two strategies for creating negative instances, so even negative instances with lexically very similar terms can be easily identified as negative by the doctors due to their semantic dissimilarity. REPLACED-BY datasets are the most difficult as doctors frequently disagree on the similarity of two terms.

The *quality* of datasets is given by the accuracy of the automatically created dataset scores with regards to the ground truth scores. Table 3 illustrates that FSN-SYNONYM and SAME-AS datasets are of very high quality. The accuracy of REPLACED-BY datasets is lower than for the other datasets, but even the lowest accuracy of 0.86 indicates that they are

	FSN-SYN.				SYN.-SYN.				POSS.-EQUIV.-TO				REPLACED-BY				SAME-AS			
	e-R	h-R	e-L	h-L	e-R	h-R	e-L	h-L	e-R	h-R	e-L	h-L	e-R	h-R	e-L	h-L	e-R	h-R	e-L	h-L
NaN	3%	13%	17%	10%	13%	10%	20%	20%	10%	10%	13%	10%	17%	30%	13%	37%	6%	6%	17%	3%
IAA	0.88	0.87	0.85	0.93	0.93	0.79	0.85	0.81	0.91	0.65	0.95	0.74	0.90	0.70	0.95	0.73	0.82	0.83	0.88	0.86
acc	0.98	0.96	0.98	0.98	0.96	0.96	0.93	0.94	1.00	0.86	0.98	0.91	0.96	0.86	0.95	0.86	0.98	0.95	0.98	0.97
rec	1.00	1.00	0.96	1.00	1.00	1.00	0.96	1.00	1.00	1.00	1.00	0.96	1.00	1.00	0.96	0.93	1.00	1.00	1.00	1.00
prec	0.97	0.92	1.00	0.97	0.93	0.93	0.88	0.88	1.00	0.70	0.96	0.86	0.93	0.70	0.93	0.70	0.96	0.89	0.96	0.93

Table 3: Datasets evaluation: term pairs without ground truth score (NaN), Krippendorff's α IAA, acc(uracy), rec(all), and prec(ision) between ground truth and dataset scores for e(asy)/h(ard) datasets with R(andom)/L(evenshtein) negative sampling.

good-quality datasets. We observe that all datasets using the random strategy exhibit a *recall* of 1, meaning that negative instances in these datasets are indeed dissimilar terms. In contrast, negative instances created using the Levenshtein strategy are sometimes so close that doctors indicate them as in fact being similar terms. The *precision* furthermore shows that some positive term pairs are in fact dissimilar according to the doctors, which occurs more frequently for the hard datasets. The lower precision in the POSSIBLY-EQUIVALENT-TO and REPLACED-BY datasets indicates that, as is to be expected, positive instances in these datasets do not always denote *exactly* the same. An example is the pair of terms 'Abortion in first trimester' and 'Induced termination of pregnancy', which are related but according to the doctors not the same.

In summary, our manual evaluation shows that all datasets have reliable similarity scores.

4 Methods for Measuring Term Similarity

Many embeddings specifically trained for the use in medical applications have been suggested in recent years and tested on different subsets of the existing concept similarity datasets. Instead of presenting a new embedding model to test on our dataset, we evaluate publicly available existing embeddings. Note that unfortunately many of the embeddings performing best on existing datasets are not available (**?**; **?**). We also do not test concept embeddings as our datasets are based on *terms*, so a pair of terms may belong to the same concept.

4.1 Word and Contextual Embeddings

We evaluate the following types of embeddings (see Tables **??** and **??** in the Appendix for more detail).

1) word2vec skip-gram (?):

- 4 *Bio* embeddings (**?**) trained on PubMed Central (PMC), PubMed (PM), both (PP), and both plus Wikipedia (PPW);
- 1 embedding trained on the *BioASQ* challenge dataset (**?**);
- 2 embeddings with window sizes 2 and 30 by the Language Technology Lab (*LTL*) (**?**);
- 2 embeddings by the Athens University of Economics and Business (*AUEB*) with vector dimensionalities 200 and 400 (**?**).

2) Fasttext (?):

- 2 embeddings using the *MeSH* thesaurus in addition to PM for training with window size 2 for *intrinsic* tasks and size 20 for *extrinsic* ones (**?**);
- 1 embedding (and its model (M)) based on the previous plus the *MIMIC*-III dataset (**?**).

3) Non-medical: As a comparison, we also include

- the *GloVe* word embedding (**?**);
- 2 *Fasttext* embeddings trained on Wikipedia and Common Crawl (plus its model (M)) (**?**).

The MeSH and MIMIC embeddings have least out-of-vocabulary terms (OOV) regarding our new datasets, but some of the other embeddings (esp. non-medical) can represent less than 50% of terms (see Table **??** in the Appendix).

4) Contextual embeddings: since the majority of terms in our new datasets are made of multiple words, we also experiment with *ELMo* (**?**) and its biomedical version *ELMoPubMed*, *Flair* (**?**) trained on PubMed, *BERT* (**?**) and its biomedical version *SciBERT* (**?**), and *GPT* (**?**).

4.2 Similarity Metrics

In contrast to contextual embeddings, which compute a single term vector for any multi-word input string (e.g. a term), word embeddings can only represent single words so that the different word vectors of a multi-word term need to be aggregated to form a *term vector*. To compare the similarity of embedding vectors, the most commonly applied metric is cos(ine) similarity. **?** (**?**) experimented with averaging and summing word vectors to obtain a term vector and using cos as a similarity measure, but found no significant difference.

We experiment with applying similarity measures to averaged (avg) word vectors as well as computing pairwise ($pair$) word similarities and averaging these. In addition to cos as a similarity measure, we apply the rank correlation coefficients (Pearson's) r, (Spearman's) ρ and (Kendall's) τ, as recently proposed by **?** (**?**). For word embeddings, we furthermore experiment with fuzzy Jaccard (fJ) and max Jaccard (mJ) similarity, which can handle multi-word strings (**?**).

5 Evaluation

To compare what conclusions can be drawn from existing versus our new datasets, we evaluate embeddings on both, investigating 1) which similarity metric works best for the various embeddings and whether the differences are significant and 2) which embedding performs best on each dataset

Dataset Subset Size	**Hlia.** 36/36	**MM-av** 29/29	**Mayo** 81/101	**Sim** 352/566	**Sim-m** 340/449	**Rel** 347/587	**Rel-m** 339/458	**SimLex** 964/988	**SimVerb** 909/1000
Bio PMC	0.53^{2}	0.80^{3}	0.44^{3}	$0.48^{5/-12}$	$0.46^{5/-14}$	$0.36^{4/-16}$	$0.36^{4/-16}$	$0.71^{5/-3}$	$0.45^{4/-4}$
Bio PM	0.59^{6}	0.83^{3}	0.54^{3}	$0.58^{7/-2}$	$0.56^{7/-3}$	$0.47^{7/-4}$	$0.48^{7/-5}$	$0.69^{4/-5}$	$0.44^{4/-5}$
Bio PP	$\mathbf{0.59^{7}}$	0.78^{4}	0.50^{3}	$0.54^{7/-9}$	$0.53^{7/-9}$	$0.44^{6/-7}$	$0.45^{7/-7}$	$0.71^{5/-2}$	$0.45^{4/-4}$
Bio PPW	0.57^{2}	$\mathbf{0.85^{8}}$	0.50^{3}	$0.54^{7/-8}$	$0.53^{6/-8}$	$0.45^{7/-7}$	$0.45^{6/-7}$	$0.72^{8/-1}$	$0.47^{5/-4}$
BioASQ	0.48^{1}	0.80^{3}	0.55^{3}	$0.60^{9/-1}$	$0.59^{9/-2}$	$0.48^{7/-4}$	$0.49^{7/-4}$	$0.69^{4/-5}$	$0.42^{3/-12}$
LTL win2	0.52^{2}	0.76^{2}	$0.47^{3/-1}$	$0.60^{8/-2}$	$0.59^{8/-2}$	$0.50^{7/-2}$	$0.51^{7/-2}$	$0.72^{5/-2}$	$0.46^{5/-4}$
LTL win30	$\mathbf{0.59^{7}}$	0.81^{4}	$\mathbf{0.57^{5}}$	$\mathbf{0.66^{14}}$	$\mathbf{0.66^{13}}$	0.58^{13}	0.59^{13}	$0.69^{4/-4}$	$0.44^{4/-6}$
AUEB200	0.42^{-1}	0.78^{3}	0.51^{3}	0.62^{9}	0.62^{9}	$0.53^{9/-2}$	$0.54^{9/-2}$	$0.71^{6/-2}$	$0.46^{5/-4}$
AUEB400	0.48	0.77^{2}	0.51^{3}	0.64^{9}	0.63^{9}	$0.54^{9/-1}$	0.55^{10}	$0.72^{6/-2}$	$0.47^{5/-4}$
MeSH extr	0.46^{1}	0.82^{7}	0.50^{3}	$0.63^{9/-1}$	$0.62^{9/-1}$	$0.54^{9/-1}$	$0.55^{9/-1}$	$0.70^{5/-4}$	$0.47^{5/-4}$
MeSH intr	0.42	0.82^{4}	0.55^{3}	$\mathbf{0.66^{14}}$	$\mathbf{0.65^{14}}$	$\mathbf{0.59^{15}}$	$\mathbf{0.59^{14}}$	$0.66^{4/-14}$	$0.44^{4/-4}$
MIMIC	0.51^{1}	0.81^{4}	0.53^{3}	0.64^{9}	0.63^{9}	0.56^{11}	0.57^{11}	$0.71^{5/-2}$	$0.48^{6/-3}$
MIMIC M	0.52^{1}	0.81^{4}	0.53^{3}	0.64^{9}	0.63^{10}	0.56^{11}	0.57^{11}	$0.71^{6/-2}$	$0.48^{6/-2}$
GloVe	0.37	0.53^{-2}	0.37^{1}	$0.55^{5/-2}$	$0.54^{6/-2}$	0.49^{7}	0.49^{7}	0.75^{10}	0.56^{16}
Fastt Wiki	0.29^{-3}	0.57^{-1}	0.38	$0.53^{5/-2}$	0.55^{6}	0.49^{7}	0.52^{7}	0.75^{11}	0.56^{15}
Fastt Cr	0.32^{-3}	0.57^{-2}	0.40^{1}	0.59^{6}	0.59^{7}	0.54^{7}	0.55^{7}	0.77^{18}	$\mathbf{0.58^{18}}$
Fastt Cr M	0.30^{-3}	0.57^{-2}	0.41^{1}	0.58^{6}	0.59^{7}	0.54^{7}	0.55^{7}	$\mathbf{0.77^{19}}$	$\mathbf{0.58^{18}}$
ELMoPM	0.42^{2}	0.66^{1}	0.38^{1}	$0.44^{5/-14}$	$0.42^{5/-15}$	$0.33^{4/-15}$	$0.34^{4/-15}$	$0.72^{6/-2}$	0.51^{6}
Flair	-0.07^{-11}	0.06^{-10}	0.18^{-1}	0.19^{-18}	0.19^{-18}	0.08^{-18}	0.08^{-18}	0.38^{-19}	0.18^{-19}
SciBERT	0.40	0.59	0.26	0.19^{-18}	0.19^{-18}	0.21^{-16}	0.21^{-16}	0.35^{-19}	0.30^{-18}
BERT	-0.01^{-3}	0.21^{-9}	-0.01^{-13}	0.12^{-18}	0.11^{-18}	0.08^{-18}	0.04^{-18}	0.39^{-19}	0.18^{-19}
ELMo	0.00^{-7}	0.11^{-13}	0.08^{-17}	0.20^{-18}	0.21^{-18}	0.13^{-18}	0.14^{-18}	$0.63^{4/-9}$	$0.5^{4/-2}$
GPT	0.00^{-1}	-0.17^{-13}	0.01^{-13}	0.10^{-18}	0.09^{-18}	0.08^{-18}	0.06^{-18}	0.32^{-19}	0.26^{-19}

Table 4: Spearman's correlation of each embedding (fJ for word embeddings, avg_cos for GloVe, τ for contextual embeddings). An embedding has significantly better/worse correlation than the number of embeddings given by the positive/negative superscripts ($\alpha = 0.0002$, i.e. $\alpha = 0.05$ with Bonferroni correction). MM-av: average scores of coders and physicians.

and whether the differences are significant. For fair comparison, all analyses are performed on a subset of each dataset containing no OOV instances for any embedding. For the interested reader, detailed results are in the Appendix.

5.1 Small Existing Datasets

As in previous work, we measure the performance of embeddings in terms of Spearman's correlation. Since the similarity scores of different embeddings are not independent and we cannot assume that they are normally distributed, bias-corrected and accelerated (BCa) bootstrap confidence intervals (**?**) are applied to assess if there are significant differences between the predictions of different embeddings with different similarity metrics.

Effect of Similarity Metrics. Comparing the Spearman's correlations of a word embedding obtained with the different similarity metrics, no metric consistently performs best (see Table **??** in the Appendix). We find nearly *no significant differences* between applying different similarity metrics to an embedding on the Hliaoutakis and MiniMayoSRS datasets (see Tables **??**-**??** in the Appendix). This illustrates that these datasets are simply *too small* to draw any meaningful conclusions about performance differences of different embeddings and similarity metrics. For the larger datasets, mJ has significantly lower correlation than most other similarity metrics for various word embeddings. This is interesting as **?** (**?**) find that for sentence similarity tasks mJ outperforms avg_cos. None of the other similarity metrics performs significantly better than all others for any dataset and word embedding. We therefore use the standard avg_cos to compare the performance of word embeddings in the next section, except for GloVe where avg_r is applied as it significantly outperforms most other metrics (on the larger datasets). For contextual embeddings, ρ and τ are often significantly better than the other metrics, with the latter slightly outperforming the former. We therefore use τ for the comparison of embeddings.

Embedding Comparison. Table 4 reports the Spearman's correlation for each embedding and indicates how many other embeddings it significantly outperforms and falls behind. Note that higher correlations have been reported for the UMNSRS-Sim/Rel datasets (e.g. (**?**; **?**)), but none of these embeddings are publicly available and thus not included here. Overall, the correlations of word embeddings are moderate to strong, *suggesting* that embeddings are able to decently encode medical terms and their similarity. For the Hliaoutakis and MiniMayo datasets, *no significant differences* between biomedical and, in most cases, even the non-medical word embeddings are observed, despite correlation differences as large as 0.2. This is due to the small size of these datasets and highlights the *need for larger datasets* to obtain more meaningful embedding comparisons. On the UMNSRS-Sim/Rel datasets, the BioNLP embeddings perform significantly worse than most other biomedical em-

Dataset	FSN-SYN. easy	FSN-SYN. hard	SYN-SYN. easy	SYN-SYN. hard	POSS.-EQU. easy	POSS.-EQU. hard	REPL.-BY easy	REPL.-BY hard	SAME-AS easy	SAME-AS hard
Subset Size	65.2%	58.6%	63.0%	56.5%	72.3%	69.0%	49.4%	62.9%	69.7%	71.0%
BioNLP PMC	$74.5^{9/-13}$	$54.9^{13/-6}$	$73.8^{9/-13}$	$52.4^{6/-10}$	$77.1^{4/-4}$	$52.7^{10/-8}$	70.6^{3}	$56.7^{9/-5}$	$79.0^{5/-3}$	$57.7^{11/-5}$
BioNLP PM	$77.0^{16/-4}$	$55.5^{17/-3}$	$76.1^{16/-4}$	$53.2^{15/-6}$	79.1^{4}	$53.3^{16/-6}$	74.3^{4}	59.7^{18}	$79.2^{5/-3}$	$58.9^{16/-4}$
BioNLP PP	$76.4^{11/-5}$	$54.5^{12/-10}$	$75.6^{11/-7}$	$52.4^{7/-10}$	$78.3^{4/-3}$	$53.0^{13/-7}$	73.1^{3}	$57.7^{10/-4}$	$79.4^{6/-3}$	$58.0^{11/-5}$
BioNLP PPW	$76.3^{11/-5}$	$53.9^{10/-11}$	$75.4^{11/-7}$	$52.4^{6/-10}$	$78.3^{4/-1}$	$52.8^{12/-7}$	72.1^{3}	$57.6^{9/-4}$	$79.1^{5/-3}$	$57.7^{11/-5}$
BioASQ	$76.6^{11/-5}$	$55.2^{14/-4}$	$75.4^{11/-7}$	$52.6^{14/-8}$	79.6^{4}	$59.4^{20/-1}$	74.3^{4}	59.9^{18}	$78.9^{5/-4}$	$60.9^{19/-1}$
LTL win2	$76.3^{11/-5}$	$52.3^{7/-15}$	$75.7^{12/-7}$	$52.4^{6/-11}$	78.7^{4}	$52.6^{10/-7}$	73.1^{3}	$56.6^{8/-6}$	$79.5^{7/-2}$	$55.2^{7/-12}$
LTL win30	$75.1^{10/-12}$	$57.4^{21/-1}$	$74.5^{10/-12}$	$54.8^{21/-1}$	$\mathbf{81.1^{9}}$	$54.3^{17/-5}$	72.8^{3}	60.6^{18}	81.1^{8}	$60.9^{19/-1}$
AUEB200	$78.8^{21/-1}$	$55.1^{13/-5}$	$77.4^{19/-1}$	$52.5^{13/-9}$	80.8^{7}	$57.8^{18/-3}$	73.4^{3}	$58.1^{11/-4}$	81.6^{9}	$57.9^{11/-4}$
AUEB400	$78.4^{19/-2}$	$55.6^{18/-3}$	$77.3^{19/-1}$	$53.3^{15/-5}$	80.1^{5}	$57.6^{18/-3}$	73.1^{3}	$58.0^{10/-5}$	82.0^{13}	$58.0^{11/-4}$
MeSH extr	$\mathbf{79.2^{22}}$	$56.7^{20/-2}$	$\mathbf{77.7^{22}}$	$53.9^{17/-2}$	81.1^{8}	$59.5^{20/-1}$	74.6^{4}	$59.3^{13/-1}$	82.3^{15}	$60.4^{19/-1}$
MeSH intr	$78.3^{19/-2}$	$\mathbf{58.5^{22}}$	$77.1^{19/-1}$	$\mathbf{55.6^{22}}$	$\mathbf{81.6^{9}}$	$\mathbf{61.4^{22}}$	74.0^{4}	$\mathbf{61.2^{19}}$	$\mathbf{82.9^{18}}$	$\mathbf{64.2^{22}}$
MIMIC	$76.7^{12/-4}$	$55.0^{13/-5}$	$76.0^{16/-4}$	$52.4^{6/-10}$	79.0^{4}	$52.5^{10/-9}$	74.3^{3}	$57.4^{9/-4}$	$80.2^{7/-1}$	$57.4^{11/-5}$
MIMIC M	$76.7^{12/-4}$	$55.0^{13/-5}$	$76.0^{16/-4}$	$52.4^{6/-10}$	79.0^{4}	$52.5^{10/-9}$	74.3^{3}	$57.4^{9/-4}$	$80.0^{7/-1}$	$57.4^{11/-5}$
GloVe	$72.2^{8/-14}$	$51.6^{6/-16}$	$71.7^{8/-14}$	$52.4^{6/-10}$	78.1^{4}	$52.0^{9/-13}$	70.3^{3}	$54.9^{6/-12}$	$77.6^{5/-5}$	$55.1^{7/-12}$
Fastt Wiki	61.0^{-22}	$53.6^{10/-11}$	61.6^{-22}	$53.6^{16/-4}$	60.0^{-20}	50.9^{-16}	54.2^{-20}	51.0^{-17}	58.9^{-21}	50.9^{-18}
Fastt Crawl	$66.2^{3/-18}$	$53.4^{9/-13}$	$66.4^{3/-18}$	$54.2^{19/-2}$	62.4^{-20}	50.9^{-16}	60.4^{-18}	51.1^{-17}	$62.1^{2/-20}$	50.9^{-18}
Fastt Crawl M	$63.3^{1/-20}$	$53.2^{8/-14}$	$63.6^{1/-20}$	$53.9^{18/-3}$	60.7^{-20}	50.9^{-16}	54.5^{-20}	51.1^{-17}	60.3^{-21}	50.8^{-19}
ELMoPubMed	$76.1^{11/-7}$	50.4^{-17}	$75.3^{11/-8}$	52.1^{-17}	78.8^{4}	$51.3^{6/-14}$	74.0^{4}	$55.9^{8/-8}$	$79.6^{7/-1}$	$54.7^{7/-12}$
Flair	$70.6^{6/-15}$	50.4^{-17}	$70.0^{6/-15}$	52.1^{-17}	$70.3^{3/-19}$	50.9^{-16}	$67.5^{2/-5}$	51.1^{-17}	$73.2^{3/-16}$	51.0^{-18}
SciBERT	$63.4^{1/-20}$	50.4^{-17}	$64.2^{1/-20}$	52.1^{-17}	$75.9^{4/-4}$	$51.1^{6/-14}$	69.0^{3}	$53.8^{6/-14}$	$72.2^{3/-17}$	$54.3^{7/-12}$
BERT	$67.4^{5/-17}$	50.4^{-17}	$67.2^{5/-17}$	52.1^{-17}	78.5^{4}	50.9^{-16}	70.3^{3}	51.3^{-17}	$75.3^{3/-10}$	$51.7^{1/-17}$
ELMo	$70.2^{6/-15}$	50.4^{-17}	$70.1^{6/-15}$	52.1^{-17}	$76.0^{4/-5}$	51.1^{-14}	70.3^{3}	$53.4^{6/-14}$	$76.0^{4/-9}$	$53.0^{5/-16}$
GPT	$65.9^{3/-18}$	50.4^{-17}	$65.8^{3/-18}$	52.1^{-17}	$77.0^{4/-1}$	50.9^{-16}	68.7^{2}	51.1^{-17}	$78.5^{5/-2}$	$52.2^{4/-16}$

Table 5: Accuracy of each embedding (fJ for word embeddings, τ for contextual embeddings) on datasets created with Levenshtein negative sampling. An embedding has significantly better/worse accuracy than the number of embeddings given by the positive/negative superscripts ($\alpha = 0.0002$, i.e. $\alpha = 0.05$ with Bonferroni correction).

beddings, even though they achieve the highest correlations on the very small datasets. This demonstrates that existing datasets *do not allow any judgements* about the generalisability of embeddings to unseen similarity instances. The other biomedical word embeddings do not exhibit significant differences and even the non-medical word embeddings do not perform significantly worse. This raises the *question if existing datasets are representative* of the highly difficult medical terminology. All word embeddings significantly outperform all contextual embeddings except ELMoPubMed, which performs significantly better than the other contextual embeddings. BERT models usually require fine tuning, so their lower performance is expected. Flair's lower performance likely stems from it not having an explicit notion of words, whereas the remaining contextual embeddings lack medical knowledge.

The results on Bio-SimLex and Bio-SimVerb are surprising: the non-medical Fasttext significantly outperforms all biomedical word embeddings, achieving much higher correlations than previously reported (**?**).

5.2 New Large-Scale Datasets

Since our new datasets frame a binary classification task, we evaluate the embeddings' separability of similar versus dissimilar term pairs using the area under the ROC curve (AUC) and accuracy based on a classification threshold optimising the accuracy (different threshold for each embedding and similarity metric). Significance between classifications of the different embeddings using the optimised thresholds is measured by McNemar's test. Since the accuracy scores follow the AUC trends (see Tables **??** and **??** in the Appendix), we present accuracy scores and their significant differences in Table 5.

Effects of Similarity Measures. In contrast to the existing datasets, fJ *significantly* outperforms most other similarity metrics for nearly all word embeddings (see Tables **??**-**??** in the Appendix). Furthermore, $pair$ metrics perform significantly worse than other metrics – a difference not observable on the existing datasets. For contextual embeddings, τ and and ρ again significantly outperform the other metrics on some datasets. For the following comparison of embeddings, we thus use fJ for all word embeddings and τ for all contextual embeddings.

Embedding Comparison. Table 5 shows *significant* performance differences between the embeddings on the new datasets created with Levenshtein negative sampling, which are not revealed by existing datasets. MeSH intr yields the best overall separation of similar and dissimilar term pairs, *significantly* outperforming the non-medical word embeddings and the contextual embeddings as well as most of the medical word embeddings – especially on the hard datasets. In contrast, the performances of all medical word embeddings on the datasets with random negative sampling are very similar and high (see details in Tables **??** and **??** in the Appendix). This shows that, as is to be expected, random negative sampling creates term pairs that are easily identi-

fiable as dissimilar. In the following, we thus focus on the datasets with Levenshtein negative sampling.

Easy vs. Hard Datasets. For the hard datasets, accuracy is much lower than for the easy datasets, sometimes barely over 50% indicating *no separation* between similar and dissimilar term pairs. Recall that in these hard datasets, similar term pairs have a larger Levenshtein distance than dissimilar ones (see Table 2), making them highly challenging. In fact, contextual embeddings predict dissimilar terms to be more similar than the actual similar terms (AUC lower than 0.5, see Table **??** in the Appendix). In contrast, for the easy datasets, where similar terms have a lower Levenshtein distance than dissimilar terms, the performance of ELMoPubMed is en par with the performance of some of the medical word embeddings. This behaviour can be attributed to the fact that contextual embeddings are based on n-grams/characters, so that lexically similar medical terms are represented by similar vectors.

Conclusion of Analysis. The performance analysis of embeddings on both existing and new datasets shows: 1) Our new datasets reveal *significant* performance differences between embeddings and similarity metrics, not observable on the (too small) existing datasets. 2) Existing datasets suggest decent performance of current embeddings, whereas our datasets prove that embeddings are in fact *unable* to correctly identify difficult term pairs as (dis)similar. 3) Our datasets thus provide a challenging novel *benchmark* for future research, representing the whole medical terminology.

5.3 Category Separation

Both our new and existing datasets encode only very closely related terms as similar. An adequate representation of medical terminology, mirroring a doctor's understanding, should however go further: medical terms are also similar on a broader level, forming distinct categories. We thus propose to also use *category separation* to test medical term representations and perform a first small evaluation to motivate this type of evaluation for future research.

Again, we make use of SNOMED CT and choose the two semantically close categories *Diagnostic Procedure* (DP) and *Therapeutic Procedure* (TP) as well as the category *Organism* (Org), which is semantically distant from the other two. Intuitively, we expect that terms (of concepts) in DP and TP are more similar than terms (of concepts) in DP and Org. To quantify to what extent an embedding satisfies this intuition, we introduce a category *overlap* error metric

$$\#O = \sum_{t_i \in DP, t_j \in TP, t_k \in Org} 1 \mid sim(t_i, t_j) \leq sim(t_i, t_k)$$

counting the number of term pairs of semantically close categories that have lower sim(ilarity scores) than term pairs of distant categories. Since there may be OOV terms for some word embeddings, we report the *relative overlap*, i.e. the overlap error count compared to the maximum possible number of overlap errors, $O = \#O/(|DP| \times |TP| \times |Org|)$, where $|DP|$ (resp. $|TP|$, $|Org|$) denotes the number of terms in DP that can be encoded by the respective embedding. **?** (**?**) use a similar evaluation for non-medical terms, but apply a different metric.

Table 6 shows that, although contextual embeddings performed poorly on the term similarity task, ELMoPubMed achieves the best separation between categories (see more details in Table **??** in the Appendix). Interestingly, the best performance of ELMoPubMed is achieved using r, whereas τ – which performed best on the term similarity task – produces the worst results. Furthermore, the MeSH embeddings, performing best on the term similarity task, exhibit comparably bad performance here. These observations provide interesting first insights for future work.

	metric (best/worst)	O (best/worst)
LTL win2	$fJ/pair\_\tau$	8.6%/20.1%
AUEB200	fJ/mJ	8.6%/15.4%
MeSH intr	$avg\_cos/pair\_\rho$	13.9%/17.1%
MeSH extr	avg_cos/mJ	12.2%/19.0%
ELMoPubMed	r/τ	5.6%/13.4%
Flair	r/τ	17.5%/21.9%
SciBERT	ρ/r	21.1%/24.4%
BERT	ρ/cos	10.9%/11.9%
ELMo	r/τ	14.4%/18.8%
GPT	ρ/r	33.2%/35.4%

Table 6: Relative overlap with best/worst similarity metric of 2 best word embeddings and 2 best from similarity task.

6 Conclusion

We have shown that existing datasets for medical term similarity are too small to detect significant performance differences between embeddings and similarity metrics applied to an embedding. In contrast, using our new large-scale datasets, *significant* differences are revealed. Furthermore, the new datasets expose the enormous difficulty of current embeddings in predicting the similarity of non-obvious term pairs, i.e. semantically similar terms that are lexically dissimilar and vice versa. The datasets thus constitute a challenging *new benchmark* for medical term similarity. Our analysis also showed that the recently introduced Fuzzy Jaccard similarity measure for multi-word strings (**?**) yields better results for most medical word embeddings than the standard cosine similarity and should thus receive attention in future work. Overall, we conclude that available embeddings are *unable* to adequately represent medical terminology at scale. In contrast to doctors' explicit knowledge of term (dis)similarity, as captured in ontologies such as SNOMED, embeddings are based on terms' occurrence in context, thus making similarity much more implicit. We saw that embeddings making use of explicit knowledge (MeSH thesaurus) yield the best representations, which is thus a promising direction for future research.

Hic voluptatem nihil culpa nulla tempora, tempora quaerat magni mollitia autem vel ipsum ea sed voluptatum, molestiae alias in deleniti expedita blanditiis pariatur excepturi tempora est praesentium, et omnis perspiciatis non ex.Atque modi dolore obcaecati architecto autem qui ad similique unde mollitia, impedit illo eveniet quidem quod veritatis aliquam nostrum, nihil laborum nam vero fugit?Reprehenderit commodi labore nam quas voluptatum

architecto dolore reiciendis in, nam ad exercitationem vitae incidunt, ipsum doloremque tempora asperiores, corporis ipsum dolorem dolore aliquid debitis.