Conceptual Grounding Constraints for Truly Robust Biomedical Name Representations

Pieter Fivez
CLiPS Research Centre
University of Antwerp
pieter.fivez@uantwerpen.be

Simon Šuster
Faculty of Engineering and Information Technology
University of Melbourne
simon.suster@unimelb.edu.au

Walter Daelemans
CLiPS Research Centre
University of Antwerp
walter.daelemans@uantwerpen.be

Abstract

Effective representation of biomedical names for downstream NLP tasks requires the encoding of both lexical as well as domain-specific semantic information. Ideally, the synonymy and semantic relatedness of names should be consistently reflected by their closeness in an embedding space. To achieve such robustness, prior research has considered multi-task objectives when training neural encoders. In this paper, we take a next step towards truly robust representations, which capture more domain-specific semantics while remaining universally applicable across different biomedical corpora and domains. To this end, we use conceptual grounding constraints which more effectively align encoded names to pretrained embeddings of their concept identifiers. These constraints are effective even when using a Deep Averaging Network, a simple feedforward encoding architecture that allows for scaling to large corpora while remaining sufficiently expressive. We empirically validate our approach using multiple tasks and benchmarks, which assess both literal synonymy as well as more general semantic relatedness. Our code is open-source and available at www.github.com/clips/conceptualgrounding.

1 Introduction

Biomedical and clinical free-text contain mentions of biomedical terms which can provide valuable information for text mining applications. Such textual mentions, as well as their corresponding reference names in biomedical ontologies, can often be expressed in various synonymous surface forms (e.g. *pleuritic pain* vs. *pain breathing*), which is challenging for downstream applications. Effective dense representation of these biomedical names has been mainly investigated through the normalization task of disorder linking, which consists of matching disease mentions to reference terms of concept identifiers in ontologies (e.g. matching the mention *myocardial depression* to the reference term *Myocardial Dysfunction)* (Leaman et al., 2015). While past research has gradually shifted its focus from lexical representations (Leaman et al., 2013; D'Souza and Ng, 2015) to dense distributed representations (Limsopatham and Collier, 2016; Li et al., 2017; Phan et al., 2019; Sung et al., 2020), encoders are still typically optimized towards normalization tasks, which are focused on resolving word-level analogies between synonymous biomedical names.

ICD-10	SNOMED-CT
F60.1	C0564504 *schizoid fantasy* *schizoid fantasy - mental defense mechanism*
	C0338969 *introverted personality disorder* *introverted personality*
	C0036339 *schizoid personality disorder* *unspecified schizoid personality disorder*

Table 1: Example of SNOMED-to-ICD-10 mappings. The synonym sets for the SNOMED-CT concepts *C0564504*, *C0338969*, and *C0036339*, are fused into one large set of semantically related names for the ICD-10 code *F60.1*.

Recent research has focused more explicitly on encoding domain-specific biomedical semantics by training biomedical name representations that are *robust*, i.e., reflecting the synonymy and semantic relatedness of names by their closeness in the embedding space, preferably in a consistent way

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 2440–2450
April 19 - 23, 2021.

that generalizes across different biomedical subdomains and corpora. To date, the most effective approaches have applied some form of *conceptual grounding*: minimizing the distance between on the one hand representations of names, and on the other hand pretrained embeddings of their concept identifiers. These concept embeddings are supposed to reflect domain-specific semantics, and are constructed using a variety of different techniques, including distributional similarity of graph relations and distributional similarity of textual occurrences in large-scale free-text, as well as combinations thereof (Kartsaklis et al., 2018; Phan et al., 2019).

While knowledge graph embeddings of biomedical concepts can encode a variety of semantic relations, Kartsaklis et al. (2018) show that such graph embeddings need to incorporate textual features to make them effective targets for conceptual grounding. Such features help to translate textual representations of names to the topology of the concept embedding space, which otherwise reflects only ontological information. In other words, concept embeddings are mostly useful targets for grounding to the extent that name representations can be efficiently mapped to them by the encoder architecture. This raises the question whether we can increase the effectiveness of conceptual grounding by better aligning the topology of the created name embedding space and the pretrained concept embedding space. In this paper, we investigate how to maximally exploit low-cost concept embeddings, which can be constructed using only pretrained word embeddings and sets of biomedical synonyms or semantically related names.

To this end, we enrich a siamese neural network encoder for biomedical names with 2 novel constraints which are meant to effectively map encoded names to pretrained concept embeddings. The first constraint, which we call the *linear constraint*, applies canonical correlation analysis (CCA) to pretrained embeddings of names and their concepts to project them into a space which improves their linear mapping. These transformed embeddings are then used as input representations for the neural encoder. The second constraint adds a training objective which we call *prototypical grounding*: minimizing the distance between a pretrained concept embedding and the average of all the encoded names belonging to that concept. This average is an approximation of the prototypical representation of a concept in the name embedding space.

While the linear constraint involves a simple preprocessing step, the prototypical grounding constraint can be computationally expensive for large-scale corpora. Therefore, we use a simple Deep Averaging Network (DAN) (Iyyer et al., 2015) as encoder to prove the effectiveness and scalability of our approach, even for a neural architecture that has no access to word order like LSTMs have or cannot apply attention over specific word combinations like Transformers can. We train and evaluate our encoder on different categorizations of biomedical names. For instance, Table 1 shows how concepts from the SNOMED-CT ontology capture literal synonymy, while these concepts can also be grouped into the ICD-10 coding system which reflects more general semantic relatedness. Our experimental results show that our approach is effective for both types of categorizations, as well as for various ontologies and benchmarks.

2 Related work

Biomedical name encoders A variety of neural architectures have been proposed for encoding biomedical names. Kartsaklis et al. (2018) use a multi-sense LSTM with attention over different word senses. This attention is conditioned on the context of the biomedical name. Phan et al. (2019) include a character-level Bidirectional LSTM in a word-level Bidirectional LSTM which extracts a fixed-size representation using max pooling over all dimensions, followed by a linear transformation. Sung et al. (2020) finetunes pretrained context-sensitive BioBERT (Lee et al., 2019) representations and uses them in tandem with lexical TF-IDF representations. While past research has explicitly investigated the role of various training objectives, even jointly in multi-task training regimes, the specific impact of encoder architectures has not received much attention or comparison.

Averaging networks Research on sentence embeddings and paraphrasing has consistently found that simple encoding procedures such as averaging of word embeddings can rival or even outperform complex neural architectures on tasks for which those are finetuned (Wieting et al., 2016; Shen et al., 2018; Wieting and Kiela, 2019). Moreover, research on Deep Averaging Networks (Iyyer et al., 2015) has found that feedforward neural networks that use averaged word embeddings as input can be tuned to textual classification tasks such as sentiment analysis if the network is sufficiently large

and/or deep. This way, small differences in the input can be magnified by the network where relevant.

Prototypical networks While successful approaches to few-shot learning such as Matching Networks (Vinyals et al., 2016) optimize representation models on the level of single instances, follow-up work has shown the benefits of simultaneously learning class representations using those same models. For instance, prototypical networks (Snell et al., 2017) train a neural encoder with objectives that involve class prototypes, which are created by averaging the encodings of all instances that belong to a single class. In this paper, we include a training objective for our encoder which forces synonymous or semantically related biomedical names to form class prototypes that approximate the pretrained embedding of their concept identifier.

3 Encoding model

3.1 Encoder architecture

Our encoder is a Deep Averaging Network (DAN) (Iyyer et al., 2015) which extracts a fixed-size representation for an input name *n*:

$$u_n = \frac{1}{|N_t|} \sum_{t \in N_t} u_t$$
$$f(n) = enc(u_n) \quad (1)$$

where N_t is the bag of tokens from a name, u_t is a pretrained word embedding of a token, u_n is a name embedding created by averaging all the pretrained word embeddings of all tokens, and enc is a feedforward neural network with Rectified Linear Unit (ReLU) as non-linear activation function. As pretrained word embeddings we use 300-dimensional fastText (Bojanowski et al., 2017) representations which we train on 76M sentences of preprocessed MEDLINE articles released by Hakala et al. (2016). This fastText model also allows for constructing word embeddings for out-of-vocabulary tokens by composing character n-gram embeddings.

3.2 Training objectives

Our training objectives optimize the mapping between an encoded name $f(n)$ and the pretrained embedding of its concept u_p. While in principle any type of pretrained concept embeddings could be used, our experiments use concept embeddings which are simply the average of all pretrained name embeddings belonging to the concept:

$$u_p = \frac{1}{|C_n|} \sum_{n \in C_n} u_n \quad (2)$$

These concept embeddings can be constructed entirely from synonym sets only, and have been proven effective in experiments by Phan et al. (2019).

Linear constraint: CCA We apply canonical correlation analysis (CCA) to find the best linear combination between pretrained name embeddings and the pretrained embeddings of their concept identifiers that maximizes their correlation. We can then project both the name embeddings and the concept embeddings to this new space for training objectives that use them as input. In order to not lose any information for further training, the projected embedding space has the same dimensionality as the original embedding space.

Siamese triplet loss To enforce embedding similarity between names that are synonyms or semantically related, we use a siamese triplet loss (Chechik et al., 2010). This loss forces the encoding of a biomedical name to be closer to the encoding of a true synonym than that of a negative sample name, within a specified (possibly tuned) margin:

$$pos = d(f(CCA(n)), f(CCA(n_{pos})))$$
$$neg = d(f(CCA(n)), f(CCA(n_{neg}))) \quad (3)$$
$$L_{syn} = max(pos - neg + margin, 0)$$

where CCA denotes that the pretrained name embedding used as input for the DAN has first been transformed by the CCA constraint. We take cosine distance as distance function *d*. To select negative names during training we apply distance-weighted negative sampling (Wu et al., 2017) over all training names.

Prototypical grounding constraint To enforce prototypical grounding, we average the name encodings of all synonyms or semantically related terms belonging to a concept identifier, in order to approximate a prototypical representation of the concept in the name embedding space. We then minimize the cosine distance between this prototypical concept representation and the pretrained

embedding of the concept:

$$f(p) = \frac{1}{|C_n|} \sum_{n \in C_n} f(CCA(n))$$
$$L_{proto} = d(f(p), CCA(u_p)) \quad (4)$$

To avoid overfitting, we enforce this objective using a random dropout of synonyms from C_n, in order to stochastically approximate prototypical similarity to the concept embedding.

This constraint implies that the dimensionality of the encoder output should be the same as the dimensionality of the pretrained concept embeddings. However, if the dimensionality of the concept embeddings is smaller than the desired output dimensionality, this could be solved using e.g. random projections, which work well for increasing the dimensionality of neural encoder inputs (Wieting and Kiela, 2019).

Multi-task setup Our multi-task setup simply sums the siamese triplet losses and prototypical grounding:

$$L = L_{syn} + L_{proto} \quad (5)$$

where both losses use either the original pretrained name and concept embeddings, or their CCA projections. While the proportion of both losses could be tuned using coefficients, our experiments prove this to be redundant, since both losses systematically converge to zero or near-zero values in all experiments.

4 Data

4.1 Disorder names

4.1.1 SNOMED-CT

Following Kartsaklis et al. (2018) and Phan et al. (2019), we use SNOMED-CT[1] disorder names as biomedical synonym sets. However, since this data is of a diverse nature and quality, we try to select the most natural and coherent data by matching it with a large target domain of processed MEDLINE articles released by Hakala et al. (2016) containing 76M sentences with 120M unique noun phrases scraped from 4K articles. We match disorder names with our target domain in 4 consecutive steps. Firstly, we only retain disorder names of which all tokens appear in the vocabulary of our target domain. Secondly, many disorder names have duplicates with a small set of redundant metatags such as *(disorder)* and *(finding)* added to the name, which very rarely appear as natural language in our target domain (we list these metatags in Appendix A). Since they do not reflect relevant synonymy, we leave out such duplicates. Thirdly, we only retain disorder names of up to 6 tokens, since this is the maximum length of the 20K disorder names which directly match noun phrases from our target domain. This is also similar to the length distribution in disorder normalization benchmarks as the NCBI Disease corpus (Doğan et al., 2014) and the ShARe/CLEF eHealth 2013 corpus (Pradhan et al., 2015). Lastly, we leave out all disorder names which belong to more than one concept identifier.

4.1.2 ICD-10

The SNOMED-to-ICD-10 mapping, which has been officially provided by the U.S. National Library of Medicine[2], groups multiple SNOMED-CT concepts together under more coarse-grained ICD-10 codes, using concept unique identifiers (CUIs) from the UMLS[3] ontology which encompass those SNOMED-CT concepts. We fuse the synonym sets of SNOMED-CT concepts belonging to the same ICD-10 concept into a single set of semantically related terms. Table 1 gives some examples of the SNOMED-to-ICD-10 mappings. These examples show how ICD-10 concepts introduce a broader range of synonymy. While many of the SNOMED-CT synonyms can be resolved using word-level analogies (e.g. *myocardial depression* vs. *myocardial dysfunction*), the ICD-10 related terms that bridge different SNOMED-CT concepts require more domain-specific semantics to be linked (e.g. for matching *myocardial dysfunction* with *muscular degeneration of heart*).

4.2 Heterogeneous names: MedMentions

The recently released MedMentions corpus (Mohan and Li, 2019) enables training and testing of biomedical name encoders on a larger scale and over a wider variety of semantic types than previous benchmarks. It maps a vast amount of biomedical names mentioned in PubMed abstracts to their corresponding concept unique identifier (CUI) in the UMLS ontology. The annotated subcorpus *MedMentions ST21pv* annotates names belonging to UMLS concepts covering 21 different semantic

[1] https://www.snomed.org

[2] https://www.nlm.nih.gov/research/umls/mapping_projects/snomedct_to_icd10cm.html

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

	Disorder		Heterogeneous
	ICD-10	SNOMED-CT	MedMentions
train concepts	5,136	20,140	18,417
train mentions	31,610	29,517	38,445
train synonym pairs	120,768	26,214	118,300
validation mentions	4,802	1,355	42,924
test mentions	7,142	2,752	43,544
zero-shot concepts	1,000	1,485	1,098
zero-shot mentions	6,490	4,199	4,705

Table 2: An overview of all the data used in our experiments.

types. We fuse these textual mentions of names into synonym sets. Since they are all verified to occur in existing biomedical free-text, we don't perform any preselection at all. This also means that there are words which are out-of-vocabulary for our fastText model: 10% of the MedMentions names contain such words, which constitute 15% of the total MedMentions vocabulary. As a result, the MedMentions data can show how reliable our approach is in cases where the vocabulary of the word embeddings does not perfectly overlap with the target domain.

5 Experiments and results

5.1 Ranking tasks and data distributions

Ranking tasks We evaluate the usefulness of biomedical name representations for synonym retrieval and concept mapping by applying 3 different performance metrics to a single ranking task. Given a mention *m* of a biomedical name which belongs to the concept identifier *c*, we have to rank a set of biomedical names *S* which includes $C_{syn} \subset S$, a set of names which belong to the same concept identifier *c* as the mention *m*. To rank the biomedical names according to their similarity to the mention, we first encode both the mention *m* as well as every name $n \in S$, and then rank every name n using the cosine similarity between the encoded mention $f(m)$ and the encoded name $f(n)$.

The aim of this task is to rank every correct synonym or semantically related name $syn \in C_{syn}$ as high as possible. We measure the synonym retrieval and concept mapping performance for this task using different metrics. For synonym retrieval, we report **Mean average precision (mAP)** over all synonyms. For concept mapping, we report **Accuracy (Acc)**, the proportion of instances where the highest ranked name *n* is a correct synonym $syn \in C_{syn}$, and **Mean reciprocal rank (MRR)** of the highest ranked correct synonym.

Data distributions Table 2 gives an overview of the data distributions after splitting. For MedMentions, we take our train, validation, test, and zero-shot data from the data splits provided by *MedMentions ST21pv*. For SNOMED-CT and ICD-10, we devise our own sampling method. Firstly, we randomly divide the synonym sets in training concepts and zero-shot test concepts. Secondly, to hold out test mentions from the training data, we randomly sample a single name from each concept which has at least two names (as to avoid empty training concepts), and repeat this procedure to get more test data. We then carry out the same procedure to sample validation data which we use to calculate the stopping criterion during training.

We calculate synonym retrieval and concept mapping performance for the test and validation mentions by ranking for a test mention *m* all names *S* present in the training data, including the synonyms C_{syn} which are present in the training data for the concept identifier *c* of the test mention. The performance of the encoders for the training data is calculated by treating a single training name at a time as test item.

The zero-shot test concepts are used to observe how well our encoders can extrapolate to previously unobserved concepts, for which the encoder has not specifically learned conceptual grounding. We frame the zero-shot setup as a way of testing transfer learning within the same domain, by not including any training names at all. This setup can show that our encodings are robust enough to be used out-of-the-box in entirely novel settings. For this setup, we treat a single zero-shot name at a time as test item, and rank all correct synonyms C_{syn} present in the zero-shot data among all names *S* from the zero-shot data.

5.2 Reference model and baselines

Reference model: BNE We compare our DAN model against the Biomedical Name Encoder (BNE) by Phan et al. (2019), which we train using the exact same data. To have a direct comparison with their model, we leave out the character embeddings from their encoder architecture and only use our fastText word embeddings as input embeddings. This results in a bidirectional LSTM (BiLSTM) (Graves and Schmidhuber, 2005) with

	Train			Test			Zero-shot		
	mAP	Acc	MRR	mAP	Acc	MRR	mAP	Acc	MRR
Sent2Vec	0.27	0.42	0.51	0.30	0.47	0.56	0.43	0.67	0.74
BioBERT	0.35	0.51	0.60	0.39	0.60	0.68	0.52	0.78	0.83
fastText	0.38	0.56	0.65	0.43	0.66	0.74	0.56	0.83	0.87
CCA fastText	0.42	0.59	0.68	0.47	0.70	0.76	0.61	0.85	0.89
CCA+DAN	**0.99**	**0.99**	**0.99**	**0.79**	**0.77**	**0.80**	**0.67**	**0.87**	**0.90**
DAN	0.98	0.97	0.98	0.76	0.75	0.79	0.65	0.86	0.89
BNE	0.77	0.81	0.86	0.63	0.75	**0.80**	0.65	**0.87**	**0.90**

Table 3: Synonym retrieval and concept mapping scores for the ICD-10 encoders. The highest score is denoted in bold, the second highest is underlined.

	Train			Test			Zero-shot		
	mAP	Acc	MRR	mAP	Acc	MRR	mAP	Acc	MRR
Sent2Vec	0.41	0.35	0.45	0.38	0.44	0.54	0.55	0.57	0.67
BioBERT	0.49	0.41	0.53	0.49	0.58	0.68	0.62	0.65	0.74
fastText	0.59	0.55	0.64	0.56	0.68	0.76	0.71	0.75	0.82
CCA fastText	0.62	0.57	0.67	0.59	0.70	0.78	0.73	0.76	0.83
CCA+DAN	**0.99**	**0.99**	**0.99**	**0.84**	**0.81**	**0.85**	**0.81**	**0.85**	**0.89**
DAN	0.94	0.91	0.94	0.78	0.78	0.83	0.79	0.84	0.88
BNE	0.68	0.63	0.72	0.63	0.73	0.80	0.75	0.80	0.85

Table 4: Synonym retrieval and concept mapping scores for the SNOMED-CT encoders. The highest score is denoted in bold, the second highest is underlined.

max pooling and a linear transformation:

$$h_n = max(BiLSTM(u_{t1}, .., u_{tn}))$$
$$f(n) = W(h_n) + b \quad (6)$$

We also include the publicly released BNE model with skipgram word embeddings, BNE + SG_w, [4] which was trained on approximately 16K synonym sets of disease concepts in the UMLS, containing 156K disease names. We don't include this model for the disorder data, since it was trained on at least part of that data, and we want to avoid that data leakage affects the fairness of the model comparisons.

Baselines As baseline encoder we use the 300-dimensional **fastText** name embeddings which are used as input for the DAN (defined in Equation 1 in Section 3.1). This encoder is an example of a Simple Word-Embedding Model (SWEM) with average pooling, which has been proven to be a strong baseline for various NLP tasks (Shen et al., 2018). We also include two other pretrained baselines among our comparison of encoders: 600-dimensional **Sent2Vec** (Pagliardini et al., 2018) embeddings with word unigram and bigram representations, trained on the same MEDLINE data as our fastText embeddings; and averaged 728-dimensional context-specific token activations extracted from the publicly released **BioBERT** model (Lee et al., 2019).

[4] https://github.com/minhcp/BNE

5.3 Training details

We fit the CCA for the linear constraint using all training names and their corresponding concept prototypes constructed from the same training names. The encoder architectures of our own DAN model and the BNE reference model are implemented in PyTorch (Paszke et al., 2019). Both the input and output dimensionality are 300 (which is the dimensionality of the input fastText embeddings described in Section 3.1). All encoder architectures for which we report results performed best with a single hidden layer.

We tuned the hidden size of the DAN to 38,400 dimensions using a grid search over 300×2^n, with n starting at 1 and being increased until performance declined again. We tuned the BiLSTM for the BNE model to 4,800 dimensions using the same grid search, to make sure the architecture

	Train			Test			Zero-shot		
	mAP	Acc	MRR	mAP	Acc	MRR	mAP	Acc	MRR
Sent2Vec	0.30	0.37	0.47	0.46	0.65	0.71	0.34	0.46	0.54
BioBERT	0.28	0.40	0.47	0.41	0.64	0.68	0.25	0.43	0.49
fastText	0.41	0.51	0.61	0.51	0.70	0.76	0.43	0.61	0.68
CCA fastText	0.44	0.53	0.63	0.53	0.72	**0.77**	**0.45**	**0.62**	**0.70**
CCA+DAN	**0.88**	**0.89**	**0.93**	**0.70**	**0.73**	**0.77**	**0.45**	0.60	0.67
DAN	0.83	0.85	0.90	0.67	0.71	0.76	0.43	0.59	0.67
BNE	0.71	0.74	0.81	0.64	0.72	**0.77**	**0.45**	**0.62**	**0.70**
BNE (Phan et al., 2019)	0.40	0.52	0.60	0.50	0.68	0.74	0.40	0.58	0.66

Table 5: Synonym retrieval and concept mapping scores for the MedMentions encoders. The highest score is denoted in bold, the second highest is underlined.

ICD-10 code	**R07.1**		
Test mention	**pain provoked by breathing**		
Target synonyms	anterior pleuritic pain / breathing painful / chest pain on breathing / pleural pain / pleuritic pain		
	CCA+DAN	**BNE**	**fastText**
Top 10 ranking	*chest pain on breathing*	*chest pain on breathing*	*chest pain on breathing*
	anterior pleuritic pain	*breathing painful*	*breathing painful*
	pleuritic pain	back pain worse on sneezing	disorder characterized by back pain
	breathing painful	disorder characterized by back pain	disorder characterised by back pain
	pleural pain	disorder characterised by back pain	back pain worse on sneezing
	chest pain	*anterior pleuritic pain*	distress from pain in labor
	chronic chest pain	pain in heart	persistent pain following procedure
	pain in heart	*pleuritic pain*	chronic mouth breathing
	upper chest pain	precordial pain	chronic chest pain
	parasternal pain	chronic chest pain	dermatitis caused by sweating and friction

Table 6: A comparison of the synonym retrieval by various encoders for the ICD-10 test mention *pain provoked by breathing*. While fastText is already good at matching a few semantically related terms at the top, it retrieves no further names in its top ranks. The BNE ranking picks up on more specific biomedical semantics, but still has a limited coverage. In contrast, the conceptually grounded CCA+DAN ranks all 5 target names at the top.

was compared fairly to our model. At that point, the DAN has ±23M trainable parameters, whereas the BiLSTM already has ±200M trainable parameters. This allows us to empirically confirm that our proposed DAN model is more computationally efficient than the BNE BiLSTM.

Adam optimization (Kingma and Ba, 2015) is performed on a batch size of 64, using a learning rate of 0.001 and a dropout rate of 0.5. Input strings are first tokenized using the Pattern tokenizer (Smedt and Daelemans, 2012) and then lowercased. We use a triplet margin of 0.1 for the siamese triplet loss L_{syn} defined in Equation 3. For the prototypical constraint L_{proto} defined in Equation 4, we use a synonym dropout rate of 0.5. As stopping criterion we use the mAP of synonym retrieval for held-out validation names: we stop training once this score for the current epoch is worse than for the previous epoch.

5.4 Results and discussion

We compare the 3 baselines and the BNE reference model against 3 variants of our model. The CCA fastText model only applies the learned CCA mapping to the pretrained fastText embeddings. The CCA+DAN model applies the linear CCA constraint before training, while the DAN model leaves out the linear constraint.

ICD-10 & SNOMED-CT Table 3 and 4 show the concept mapping and synonym retrieval performance of the different encoders for the ICD-10 and SNOMED-CT data. We see that the fastText baseline consistently outperforms the other baselines. Applying the CCA transformation to the fastText baseline improves performance for every metric, including zero-shot cases. In other words, applying this linear constraint for conceptual grounding already leads to better extrapolation. The DAN model, which combines the siamese triplet loss

MedMentions CUI	C0870951		
Test mention	**cariogenesis**		
Target synonyms	caries / cavities / dental caries / mod cavities / tooth decay		
	CCA+DAN	**BNE**	**fastText**
Top 10 ranking	*dental caries*	*caries*	*caries*
	caries	biofilm formation	caries prevention
	mod cavities	formation of these biofilms	preventive treatment for dental caries
	tooth decay	*dental caries*	*dental caries*
	preventive treatment for dental caries	formation of biofilms	biofilm formation
	streptococcus mutans	caries prevention	formation of biofilms
	pellicle formation	biofilm	streptococcus mutans
	cavities	biofilm forming	anti-staphylococcal biofilm agents
	bottle tooth decay	biofilm community	formation of these biofilms
	biofilm formation	pellicle formation	dental plaque

Table 7: A comparison of the synonym retrieval by various encoders for the MedMentions test mention *cariogenesis*. While the BNE model does not improve over the fastText baseline, the conceptually grounded CCA+DAN already has complete coverage of all 5 target synonyms at rank 8.

with only the prototypical grounding loss, is able to fit the training data to near perfection without overfitting, since it generalizes well across both test and zero-shot data. Applying the CCA constraint before training increases the performance even more. These observations support the hypothesis of this paper that increasing the effectiveness of conceptual grounding can improve trained encoders.

The results also clearly confirm the robustness of our approach: synonym retrieval is dramatically improved for the test data, without any performance loss for concept mapping. In other words, the representations have encoded more domain-specific semantics while retaining the relevant lexical information. Table 6 gives an example of the impact of our conceptual grounding constraints for ICD-10 test data: the model is able to encode domain-specific semantics beyond word-level analogies for the semantically related names of the test mention *pain provoked by breathing*. Not only does the CCA+DAN model rank all semantically related names at the top: all the following top-ranked names, such as *chest pain*, also have clear semantic links to the mention. In contrast, the BNE model ranks less related names such as *back pain worse on sneezing* and *disorder characterized by back pain* higher than correct synonyms such as *pleuritic pain*.

MedMentions Table 5 shows the performance of the different encoders for the MedMentions data. Table 7 gives an example of how, similar to the disorder data, our CCA+DAN encoder is able to encode specific semantics that the BNE model is lacking: the conceptual grounding constraints have allowed our encoder to represent the semantic similarity between *cariogenesis*, *tooth decay* and *cavities*, while the BNE model does not improve over the fastText baseline.

Despite showing similar trends to the disorder data, the relative improvements of our CCA+DAN encoder over the reference BNE model are less dramatic. Interestingly, the publicly released BNE + SG_w model trained by Phan et al. (2019) performs worse out-of-the-box than our pretrained fastText embeddings. This highlights the difficulty of achieving true robustness of biomedical name encoding.

5.5 Semantic relatedness benchmarks

We also evaluate our name encoders on two biomedical benchmarks of semantic similarity, which allow to compare cosine similarity between name embeddings with human judgments of relatedness. MayoSRS (Pakhomov et al., 2011) contains multi-word name pairs of related but different concepts, and can indicate how much generalized domain knowledge has been captured by our conceptual grounding constraints. UMNSRS (Pakhomov et al., 2016) contains only single-word pairs, which also stem from different concepts. This benchmark makes a distinction between *similarity* and *relatedness*.

The correlations in Table 8 confirm the robustness of our conceptually grounded biomedical name representations. While the correlations for the BNE models barely improve over those of the fastText embeddings, our CCA+DAN encoder improves substantially over all 3 benchmarks, regard-

	MayoSRS (rel)	UMNSRS (rel)	UMNSRS (sim)
fastText	0.443	0.473	0.479
CCA+DAN, ICD-10	**0.666**	0.556	0.561
CCA+DAN, SNOMED-CT	0.648	0.537	0.540
CCA+DAN, MedMentions	0.600	0.526	0.543
Phan et al. (2019)	0.626	**0.580**	**0.606**
BNE, ICD-10	0.492	0.472	0.503
BNE, SNOMED-CT	0.415	0.510	0.527
BNE, MedMentions	0.506	0.467	0.500

Table 8: Spearman's rank correlation coefficient between cosine similarly scores of name embeddings and human judgments, reported on semantic similarity (sim) and relatedness (rel) benchmarks. The highest score is denoted in bold, the second highest is underlined.

less of the data source it was trained on. Remarkably, while the publicly released BNE model of Phan et al. (2019) was trained on 156K disease names, the CCA+DAN encoder already outperforms it on MayoSRS when trained on the ICD-10 and SNOMED-CT subsets, which contain only 30K disease names. This proves that Deep Averaging Networks can be effective even for large-scale encoding of biomedical names. Moreover, this finding suggests that future work on biomedical name encoders should not take complex neural architectures for granted. On the contrary, enforcing more relevant constraints such as our conceptual grounding constraints can boost even lightweight encoder architectures.

6 Conclusion and future work

In this paper, we have shown how two conceptual grounding constraints for biomedical name encoders can infuse name representations with more domain-specific semantics without losing robustness. These representations can help with retrieving literal synonyms as well as semantically related terms, and can be sufficiently expressed by a Deep Averaging Network, which is a feedforward neural network that only takes averaged word embeddings as input.

We believe future work can include a comparison of neural encoding architectures with a wider range of complexity. Decreasing the complexity of neural architectures can allow for including more comprehensive training objectives which target more effective encoding of domain-specific semantics.

Acknowledgments

We would like to thank the anonymous reviewers for their feedback. This research was carried out in the framework of the Accumulate VLAIO SBO project, funded by the government agency Flanders Innovation Entrepreneurship (VLAIO). We also like to thank Madhumita Sushil and Nicolae Banari for their comments.

References

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.

Gal Chechik, Varun Sharma, Uri Shalit, and Samy Bengio. 2010. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11:1109–1135.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Jennifer D'Souza and Vincent Ng. 2015. Sieve-based entity linking for the biomedical domain. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 297–302.

Alex Graves and Jurgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Kai Hakala, Suwisa Kaewphan, Tapio Salakoski, and Filip Ginter. 2016. Syntactic analyses and named entity recognition for PubMed and PubMed Central — up-to-the-minute. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 102–107.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Association for Computational Linguistics*.

Dimitri Kartsaklis, Mohammad Taher Pilehvar, and Nigel Collier. 2018. Mapping text to knowledge graph entities using multi-sense LSTMs. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1959–1970.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations (ICLR)*.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. DNorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Robert Leaman, Ritu Khare, and Zhiyong Lu. 2015. Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, 57:28–37.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Haodi Li, Qingcai Chen, Buzhou Tang, Xiaolong Wang, and Hua Xu. 2017. CNN-based ranking for biomedical entity normalization. *BMC Bioinformatics*, 18(Suppl 11)(385).

Nut Limsopatham and Nigel Collier. 2016. Normalising medical concepts in social media texts by learning semantic representation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1014–1023, Berlin, Germany. Association for Computational Linguistics.

Sunil Mohan and Donghui Li. 2019. Medmentions: A large biomedical corpus annotated with {umls} concepts. In *Automated Knowledge Base Construction (AKBC)*.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised learning of sentence embeddings using compositional n-gram features. 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 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Serguei V.S. Pakhomov, Greg Finley, Reed McEwan, Yan Wang, and Genevieve B. Melton. 2016. Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, 32(23):3635–3644.

Serguei V.S. Pakhomov, Ted Pedersen, Bridget McInnes, Genevieve B. Melton, Alexander Ruggieri, and Christopher G. Chute. 2011. Towards a framework for developing semantic relatedness reference standards. *Journal of Biomedical Informatics*, 44:251–265.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Minh C. Phan, Aixin Sun, and Yi Tay. 2019. Robust representation learning of biomedical names. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3275–3285, Florence, Italy. Association for Computational Linguistics.

Sameer Pradhan, Noémie Elhadad, Brett R South, David Martinez, Lee Christensen, Amy Vogel, Hanna Suominen, Wendy W Chapman, and Guergana Savova. 2015. Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *Journal of the American Medical Informatics Association*, 22(1):143–154.

Dinghan Shen, Guoyin Wang, Wenlin Wang, Martin Renqiang Min, Qinliang Su, Yizhe Zhang, Chunyuan Li, Ricardo Henao, and Lawrence Carin. 2018. Baseline needs more love: On simple word-embedding based models and associated pooling mechanisms. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, pages 440–450.

Tom De Smedt and Walter Daelemans. 2012. Pattern for Python. *Journal of Machine Learning Research*, 13:2031–2035.

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 4077–4087. Curran Associates, Inc.

Mujeen Sung, Hwisang Jeon, Jinhyuk Lee, and Jaewoo Kang. 2020. Biomedical entity representations with synonym marginalization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3641–3650, Online. Association for Computational Linguistics.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 3630–3638. Curran Associates, Inc.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *International Conference for Learning Representations (ICLR)*.

John Wieting and Douwe Kiela. 2019. No training required: Exploring random encoders for sentence classification. In *International Conference on Learning Representations*.

Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krahenbuhl. 2017. Sampling matters in deep embedding learning. In *ICCV*.

A Redundant metatags

In section 4.1.1, we mention that many names from our SNOMED-CT data are duplicates of other names, with the only difference being that they also contain the following redundant metatags (in order of frequency):

- (disorder)
- (finding)
- (nos)
- (morphologic abnormality)
- (situation)
- (event)
- (observable entity)
- (qualifier value)
- (context-dependent category)
- (procedure)
- (function)
- (attribute)
- (clinical)