More Meaning Than Meets the Eye. Robust and Scalable Applications of Pre-trained Representations for Biomedical NLP

Pieter Fivez

June 29, 2021

The research in this thesis was supported by two grants to the University of Antwerp: one (ID 150056) from the government agency Flanders Innovation & Entrepreneurship (VLAIO), and another (ID 44730) from the Research Foundation - Flanders (FWO).

© 2021 Pieter Fivez

All rights reserved. No parts of this book may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author.

Universiteit Antwerpen

Faculty of Arts
Department of Linguistics

More Meaning Than Meets the Eye**.** **Robust and Scalable Applications of Pre-trained Representations for Biomedical NLP**

Thesis submitted for the degree of doctor in Linguistics at the University of Antwerp to be defended by

Pieter Fivez

Supervisors Prof. Dr. Walter Daelemans and Dr. Simon Šuster

Antwerp, 2021

Acknowledgements

Two years into this PhD, I faced the kind of academic crisis which, according to a quick Google search, turned out to be rather archetypical. To this very day, translating novel ideas into successfully completed research remains a challenging task, which no fixed amount of motivation or hard work is guaranteed to resolve. To cope with this situation, I've started to live by the idea that a PhD is supposed to deliver a researcher rather than a book. Trusting in the process instead of fixating on the end result has allowed me to retain intellectual honesty and scientific integrity in the face of adversity.

I've had the great fortune to be mentored by two supervisors who supported this process with great patience. Walter is a supervisor who genuinely takes care of his students both practically and in matters of research direction, which, according to another quick Google search, appears to be less archetypical than my two-year dip. His patience and care was certainly matched by Simon, who remained as helpful as ever even as he was working full-time in Australia. I really owe a lot of gratitude to the both of you. I would also like to thank Prof. Mike Kestemont for being the head of my thesis committee; as well as Prof. Antal van den Bosch, Prof. Goran Nenadic, and Dr. Kim Luyckx for being part of my jury.

I've enjoyed the company of many colleagues at CLiPS, among whom Ben, Ehsan, Elyne, Enrique, Giovanni, Guy, Ilia, Jens, Jeska, Lisa, Madhumita, Maxime, Nikolay, Pietro, Robert, Stéphan, and Tim. Special thanks go out to Madhumita and Stéphan, who were always ready to enlighten me on all matters concerning biomedical NLP.

I’ve lived in three different places in Antwerp during the course of this PhD. At two of the three places, I’ve had housemates who stuck it out with me and my mood swings anytime a dreaded deadline was approaching; I’m sure the universe will repay them in generous amounts of karma. Thankfully, we’ve also had our fair share of fun to compensate. Thank you Jan, Yello, Iljen, Thor, Jasmien and Faisal. Special thanks go to Iljen for designing the cover of the PhD book.

Lastly, thanks to my family for their quiet and unquestioning belief that I would see this through.

Abstract

Pre-trained distributional representations of words and phrases have become omnipresent in natural language processing (NLP), where they have led to significant improvements in machine learning performance for a wide range of applications. Recent research has investigated to what extent these representations are effective for tackling the challenges of the biomedical text domain. However, it remains difficult to properly disentangle the interplay of model architectures, training objectives, data sources, and downstream biomedical NLP tasks for which the representations are used as input features. As a result, it is still unclear to which extent these representations can be applied to encode specific biomedical semantics for future applications which would require complex domain knowledge.

In this thesis, we specifically explore what we consider to be robust and scalable applications of pre-trained representations for biomedical NLP. These applications go against the current dominant paradigm in NLP research, which has achieved many successes by fine-tuning large and complex neural network architectures using vast amounts of data. In contrast, we explicitly try to minimize the complexity of models that use the pre-trained representations, as well as the amount of supervised data necessary for developing the models, while keeping the models transferable across various domains and applicable in unsupervised ways, e.g. using distance metrics such as cosine similarity.

While this paradigm can impose a performance ceiling on our proposed models compared to other state-of-the-art approaches, it also offers various benefits. Firstly, it helps to highlights the contribution of various aspects of a method. For instance, it can emphasize the effectiveness of training objectives which work for models with low complexity. Secondly, it minimizes the computational cost of our proposed systems, and as such aims at contributing to more equitable and democratic NLP research. Lastly, the limitations of this paradigm also challenge us to explore novel approaches that are more efficient. For example, we can compensate for less model complexity and training data by finding more effective training objectives.

In a first study, we demonstrate how we can use word and character n-gram embeddings to perform spelling correction of English and Dutch clinical free-text by ranking correction candidates according to their semantic fit in the textual context. The pre-trained embeddings are used by a cosine similarity-based ranking model which approximates semantic contexts through weighted averages of context words. We develop this model using only automatically generated spelling errors, and explicitly control for its transferability across domains such as critical care notes and reports on colon cancer. The resulting method can successfully perform context-specific clinical spelling correction, while achieving performance which is as robust as a frequency-based noisy channel ranking model.

The remaining chapters of this thesis are devoted to deep multi-task learning of biomedical name representations. We define biomedical names as those textual surface forms that represent biomedical concepts, being either official names in biomedical vocabularies or unofficial names mentioned in text. Recent research has investigated how to represent such biomedical names in a robust way for downstream NLP applications. Robustness requires that name representations should encode domain-specific knowledge, e.g. by reflecting semantic similarity between names through their closeness in the embedding space, while retaining the universal applicability and transferability of self-supervised pre-trained representations. Obvious downstream applications of robust representations include tasks such as synonym retrieval or entity linking, which links mentions of biomedical names in free-text to concept identifiers in ontologies. However, encoding specialized biomedical semantics in robust representations has the potential to also impact a wide range of other biomedical applications, such as knowledge graph completion and the discovery of relations between diseases or interactions between drugs.

Our first chapter on robust representations of biomedical names models fine-grained distinctions between biomedical concepts and introduces a novel encoder architecture for biomedical names: the Deep Averaging Network (DAN), which is a feedforward neural network (FNN) processing an unordered composition of the word embeddings in a name. While this architecture does not allow for encoding word order, it has substantially less computational overhead than more complex neural architectures such as Long Short-Term Memory Networks (LSTMs) and Transformers, and as such scales to more intensive training objectives. We exploit this tradeoff to effectively impose conceptual grounding constraints as training objectives, which enforce the similarity between the representations of names and the pre-trained prototypical representations of their biomedical concept identifiers. The resulting DAN encoder outperforms its state-of-the-art LSTM counterpart for retrieval of both literal synonyms as well as semantically related names.

The following chapter exploits the computational efficiency of our DAN encoder to train a model on higher-level biomedical conceptual distinctions, which scale to thousands of pairwise similarities within concepts. These higher-level distinctions point towards more comprehensive domain knowledge, such as grouping the names *nettle sting* and *tick-borne fever* together under the description *puncture wound of skin*. After training our model on such high-level distinctions, the resulting representations can generalise both bottom-up as well as top-down among various semantic hierarchies.

Moreover, we show how they can be used out-of-the-box for improved unsupervised detection of hypernyms using only cosine similarity.

In the last chapter of this thesis, we show that our DAN encoder can successfully model high-level conceptual distinctions using only few-shot learning on a small amount of concepts, therefore minimizing the computational cost of training. Most importantly, our approach allows for continual learning, where we accumulate information from various conceptual hierarchies to consistently improve encoder performance. This allows for efficiently estimating what conceptual distinctions are actually *relevant* to improve representations for downstream NLP applications. As of such, it provides a last clear example of the potential of robust and scalable application of pre-trained representations for biomedical NLP.

Samenvatting

Vooraf getrainde distributionele voorstellingen van woorden en zinsdelen zijn alomtegenwoordig in natuurlijke taalverwerking (NLP), waar ze hebben geleid tot aanzienlijke verbeteringen in de prestaties van *machine learning* voor een breed scala aan toepassingen. Recent onderzoek heeft onderzocht in hoeverre deze representaties effectief zijn om de uitdagingen van het biomedische tekstdomein aan te pakken. Het blijft echter moeilijk om het samenspel van modelarchitecturen, trainingsdoelstellingen, databronnen, en biomedische NLP-taken waarvoor de representaties worden gebruikt als invoer, goed te ontwarren. Als gevolg hiervan is het nog steeds onduidelijk in welke mate deze representaties kunnen worden toegepast om specifieke biomedische semantiek te coderen voor toekomstige toepassingen die complexe domeinkennis vereisen.

In dit proefschrift onderzoeken we specifiek wat we beschouwen als robuuste en schaalbare toepassingen van vooraf getrainde representaties voor biomedische NLP. Deze toepassingen druisen in tegen het huidige dominante paradigma in NLP-onderzoek, dat veel successen heeft behaald door het verfijnen van grote en complexe neurale netwerkarchitecturen met behulp van enorme hoeveelheden data. Daarentegen proberen we expliciet de complexiteit te minimaliseren van modellen die de vooraf getrainde representaties gebruiken, evenals de hoeveelheid geannoteerde data die nodig zijn voor het ontwikkelen van de modellen, terwijl we de modellen

inwisselbaar over verschillende domeinen en direct toepasbaar houden, bv. met behulp van afstandsmaten zoals cosinusgelijkenis.

Hoewel dit paradigma een prestatielimiet kan opleggen aan onze voorgestelde modellen in vergelijking met andere geavanceerde benaderingen, biedt het ook verschillende voordelen. Ten eerste helpt het om de bijdrage van verschillende aspecten van een methode te benadrukken. Het kan bijvoorbeeld de effectiviteit benadrukken van trainingsdoelen die werken voor modellen met een lage complexiteit. Ten tweede minimaliseert het de computationele kosten van onze voorgestelde systemen en beoogt het als zodanig bij te dragen aan rechtvaardiger en democratischer NLP-onderzoek. Ten slotte dagen de beperkingen van dit paradigma ons ook uit om nieuwe, efficiëntere benaderingen te onderzoeken. We kunnen bijvoorbeeld minder modelcomplexiteit en geannoteerde data compenseren door effectievere trainingsdoelen te vinden.

In een eerste studie laten we zien hoe we n-gram-representaties van woorden en karakters kunnen gebruiken om spellingcorrectie uit te voeren van Engelse en Nederlandse klinische vrije tekst door correctiekandidaten te rangschikken op basis van hun semantische geschiktheid in de tekstuele context. De vooraf getrainde representaties worden gebruikt door een rangschikkingsmodel gebaseerd op cosinusgelijkenis dat semantische contexten benadert door middel van gewogen gemiddelden van contextwoorden. We ontwikkelen dit model met alleen automatisch gegenereerde spelfouten en controleren expliciet op de overdraagbaarheid ervan tussen domeinen, zoals notities voor kritieke zorg en rapporten over darmkanker. De resulterende methode kan met succes contextspecifieke klinische spellingcorrectie uitvoeren, terwijl prestaties worden behaald die even robuust zijn als een frequentiegebaseerd *noisy channel*-model.

De overige hoofdstukken van dit proefschrift zijn gewijd aan *deep learning* van biomedische naamrepresentaties met behulp van meerdere trainingsobjectieven tegelijkertijd. We definiëren biomedische namen als die tekstuele oppervlaktevormen die biomedische concepten vertegenwoordigen, zijnde officiële namen in biomedische vocabulaires of niet-officiële namen die in de tekst worden genoemd. Recent onderzoek heeft onderzocht hoe dergelijke biomedische namen op een robuuste manier kunnen worden weergegeven voor *downstream* NLP-toepassingen. Robuustheid vereist dat naamweergaven domeinspecifieke kennis coderen, bijv. door semantische gelijkenis tussen namen te weerspiegelen door hun nabijheid in de vectorruimte, met behoud van de universele toepasbaarheid en overdraagbaarheid van vooraf getrainde representaties. Voor de hand liggende *downstream* toepassingen van robuuste representaties omvatten taken zoals het herkennen van synoniemen of het koppelen van entiteiten, waarbij vermeldingen van biomedische namen in vrije tekst worden gekoppeld aan concept-ID's in ontologieën. Het coderen van gespecialiseerde biomedische semantiek in robuuste representaties heeft echter het potentieel om ook een breed scala aan andere biomedische toepassingen te beïnvloeden, zoals het voltooien van kennisgrafieken en de ontdekking van relaties tussen ziekten of interacties tussen geneesmiddelen.

Ons eerste hoofdstuk over robuuste representaties van biomedische namen modelleert een fijnmazig onderscheid tussen biomedische concepten en introduceert een nieuwe encodeerarchitectuur voor biomedische namen: het Deep Averaging Network (DAN), een *feedforward* neuraal netwerk (FNN) dat een ongeordende samenstelling van de woordrepresentaties in een naam transformeert. Hoewel deze architectuur het coderen van woordvolgorde niet toestaat, heeft het aanzienlijk minder computationele overhead dan complexere neurale architecturen zoals Long Short-Term Memory Networks (LSTM's) en Transformers, en schaalt het als zodanig naar intensievere trainingsobjectieven. We gebruiken deze afweging om op effectieve wijze conceptuele basisbeperkingen op te leggen als trainingsdoelstellingen, die de gelijkenis versterken tussen de representaties van namen en de vooraf getrainde prototypische representaties van hun biomedische conceptidentificatoren. De resulterende DAN-encoder presteert beter dan zijn geavanceerde LSTM-tegenhanger voor het herkennen van zowel letterlijke synoniemen als semantisch gerelateerde namen.

In het volgende hoofdstuk wordt de computationele efficiëntie van onze DAN-encoder geëxploiteerd om een model te trainen voor biomedische conceptuele onderscheidingen op hoger niveau, die kunnen opschalen naar duizenden paarsgewijze overeenkomsten binnen concepten. Deze onderscheidingen op een hoger niveau duiden op meer uitgebreide domeinkennis, zoals het groeperen van de namen *brandnetelsteek* en *door teken overgedragen koorts* samen onder de beschrijving *prikwond van de huid*. Na ons model te hebben getraind in dergelijke onderscheidingen op hoog niveau, kunnen de resulterende representaties zowel bottom-up als top-down generaliseren over verschillende semantische hiërarchieën. Bovendien laten we zien hoe ze direct kunnen worden gebruikt voor verbeterde detectie van hyperoniemen door alleen cosinusgelijkenis te gebruiken.

In het laatste hoofdstuk van dit proefschrift laten we zien dat onze DAN-encoder met succes conceptuele onderscheidingen op hoog niveau kan modelleren door slechts een klein aantal concepten te leren, waardoor de computationele kosten van training worden geminimaliseerd. Het belangrijkste is dat onze aanpak voortdurend leren mogelijk maakt, waarbij we informatie verzamelen uit verschillende conceptuele hiërarchieën om de prestaties van de encoder consequent te verbeteren. Dit maakt het mogelijk om efficiënt in te schatten welke conceptuele verschillen eigenlijk relevant zijn om representaties voor *downstream* NLP-toepassingen te verbeteren. Als zodanig biedt het een laatste duidelijk voorbeeld van het potentieel van robuuste en schaalbare toepassingen van vooraf getrainde representaties voor biomedische NLP.

Contents

1 Introduction

This thesis investigates robust and scalable applications of pre-trained distributional text representations for biomedical natural language processing. Biomedical NLP is concerned with the automatic processing of biomedical and clinical text, and has grown into a large and diverse field which tackles the specific challenges of these two text genres. The language of clinical text is particularly challenging, since it communicates highly technical and precise information through often very noisy and unstructured language. Successfully processing this language for advanced text mining applications such as knowledge discovery requires consecutive processes of standardization and semantic representation. In this thesis, we address both topics using models which use pre-trained distributional text representations as input.

1.1 Pre-trained text representations

1.1.1 Distributional representations

Pre-trained representations have become omnipresent in NLP research since the popularity of the Word2Vec algorithm (Mikolov, Sutskever, et al., 2013) for learning general-purpose word embeddings from large unannotated text corpora. Compared to the sparsity of lexical representations such as TF-IDF, the distributed representations of word or sentence embeddings can allow for better interpolation between textual expressions with similar meanings but different surface forms.

Learning algorithms for pre-trained distributed representations are typically based on distributional semantics, which stipulates that the meaning of different words is defined by contrasts in the words' context of use (Firth, 1957). In order to exploit this phenomenon, artificial neural networks are trained to generate representations which optimise the associations between words and their contexts of use in large textual corpora. Such models can be described as *self-supervised*, since their learning algorithms are supervised with data that can be collected in an unsupervised way from unannotated data.

The most impactful self-supervised models since Word2Vec have all applied some variation on the masked language modelling objective, which trains an encoder to predict a masked word in a text given the other context words surrounding the masked word. Any major improvements in representation performance ever since have mainly resulted from elaborating the architecture of the neural network. Whereas Word2Vec only uses a feedforward neural network, more recent models use more complex architectures such as a Bidirectional Long Short-Term Memory network (ELMo) (Peters et al., 2018) or a Transformer encoder (BERT) (Devlin et al., 2019). This increase in modelling capacity has allowed the evolution from context-independent to context-sensitive representations.

The masked language modelling objective has remained virtually unchallenged over the years, since it has consistently outperformed suggested alternatives. These alternatives range from training on a large natural language inference task (Conneau et al., 2017), to sharing a single text encoder across weakly related tasks in a multi-task learning setting (Subramanian et al., 2018). Moreover, adding tasks to the masked language modelling objective is not guaranteed to lead to performance improvements either. For instance, while the BERT model (Devlin et al., 2019) originally included an objective for predicting the next sentence, follow-up work proved this objective to be redundant if the model is trained more intensively (Y. Liu et al., 2019), or even detrimental (Mickus et al., 2020).

Generally, advances in computational power and neural network architecture have allowed the masked language modelling objective to remain the default paradigm for distributional text representations.

1.1.2 Applying distributional representations to biomedical NLP

Prior research has investigated the application of distributional text representations to the biomedical domain. This research addresses 2 main questions. Firstly, it investigates which distributional representation methods (e.g. context-insensitive vs. context-sensitive) are most effective when used as pre-trained input for biomedical NLP models. Secondly, it observes whether pre-training on biomedical text corpora provides relative improvements compared to general-domain representations. Both questions are addressed using intrinsic and extrinsic evaluations. Intrinsic evaluations are typically performed using benchmarks of semantic similarity between two words or phrases. This similarity can be interpreted in various ways depending on the benchmark, e.g. referring only to strict synonymy or also to shared hyponymy in general (Gladkova & Drozd, 2016). Extrinsic evaluations measure how useful pre-trained representations are for specific downstream NLP tasks such as named entity recognition (NER) and text classification. While several studies observe improvements from using context-sensitive representations or domain-specific pre-training, there is not yet a consistent trend to be extracted from these results across all biomedical NLP tasks (Tawfik & Spruit, 2020a; Wang et al., 2018).

In this thesis, we mainly use 300-dimensional fastText (Bojanowski et al., 2017) word embeddings as input for our models. This embedding model combines word and character n-gram representations into a single vector, which allows the model to construct out-of-vocabulary representations for words that were absent from the fastText training data. We have trained these embeddings either on 425M words from the MIMIC-III (Johnson et al., 2016) corpus, which contains medical records from critical care units, or on 76M sentences of preprocessed MEDLINE articles released by Hakala et al. (2016). We include two other self-supervised embeddings in this thesis: the publicly released context-sensitive 728-dimensional BioBERT (Lee et al., 2019) model, which has adapted the BERT encoder to the biomedical domain, and a 600-dimensional Sent2Vec (Pagliardini et al., 2018) model, which is an extension of the fastText model to word n-gram composition, and which was trained on the same MEDLINE data.

While a recent line of work has looked into adding specific domain knowledge to pre-trained word embedings, e.g. by injecting ontological information in the training data (Zhang et al., 2019) or post-processing already trained word embeddings to fit semantic constraints (Chiu et al., 2019), we do not use such embeddings as input to our models. We explicitly want to observe how much information we can extract from generic self-supervised representations, and whether they contain more meaning than meets the eye.

1.2 Robust and scalable applications

In this thesis, we specifically explore what we consider to be robust and scalable applications of pre-trained representations for biomedical NLP. These applications go against the current dominant paradigm in NLP research, which has achieved many successes by fine-tuning large and complex neural network architectures using vast amounts of data. In contrast, we explicitly try to minimize the complexity of models that use the pre-trained representations, as well as the amount of supervised data necessary for developing the models, while keeping the models transferable across various domains and applicable in unsupervised ways, e.g. using distance metrics such as cosine similarity.

Our approach is inspired by a recent strand of research which has explored low-cost improvements for representation baselines. While earlier work has focused on supervised methods for post-processing pre-trained word embeddings, such as retrofitting (Faruqui et al., 2015) and counterfitting (Mrkšić et al., 2016), more recent work has investigated unsupervised counterparts which can be similarly effective. Such progress has been made possible by reconsidering various implicit assumptions about the relevance of specific representational information. For instance, while positional information of words in a text can evidently be important, the extent to which it contributes to model performance for various NLP tasks can be unclear or even counterintuitive. Work on Deep Averaging Networks (Iyyer et al., 2015) has shown that unordered composition of word embeddings (e.g. averaging) can rival syntactically-aware methods for text classification tasks if the composition is transformed by a sufficiently deep feedforward neural network. Moreover, in use cases where word order is more directly relevant for a specific task, using randomly initialised order-sensitive encoders can sometimes be sufficient (Wieting & Kiela, 2019). Similar questions of relevance have been raised about other aspects of textual information, including higher-level features: while the original BERT model lacks specific information about entities across token boundaries, injecting entity knowledge has not yet led to performance gains for NLP benchmarks ranging from text understanding to question answering and machine translation (Broscheit, 2019).

These observations point to the compression tradeoff in text representation: trying to compress textual expressions in a single distributed vector requires to partially suppress various low-frequency patterns in favour of robust inductive biases which are useful along many use contexts. For example, contrary to earlier assumptions, information about different senses of a word is generally represented well in a single-vector embedding of that word, as long as the senses are sufficiently frequent in the corpus for training word embeddings (Yaghoobzadeh et al., 2019). Since the distributed features of distributional representations can be useful without aligning with clearly delineated linguistic phenomena, the potential contribution of those representations to various downstream models can be unclear. While a wide range of probing tasks has been developed to investigate this potential more systematically (Adi et al., 2017; Conneau et al., 2018), effective exploitation of pre-trained representations remains a fundamentally empirical question. In this thesis, we raise the question whether more meaning than meets the eye can be extracted from those representations to work towards more comprehensive encoding of biomedical semantics.

In summary, these are the four criteria we aim to fulfil in the main chapters of this thesis, either explicitly or implicitly, to provide robust and scalable applications of pre-trained distributional text representations for biomedical NLP:

1. **Minimise the complexity of models** that use the pre-trained representations.

2. **Minimise the amount of supervised data** necessary for development while keeping models sufficiently generalisable.

3. Develop models for biomedical domains while keeping them **transferable** across domains.

4. Apply developed models to various tasks using only **unsupervised** distance metrics such as cosine similarity instead of training them end-to-end for separate tasks.

1.3 Normalization of clinical text

Compared to biomedical text, clinical text can offer additional difficulties for natural language processing. While the former is intended to distribute research results among a larger community, the latter is typically written by health care professionals to communicate among each other and to examined patients. Such clinical notes are often unstructured and heterogeneous in purpose. Moreover, their language can be particularly noisy, containing difficulties such as atypical grammar, non-standardized abbreviations and acronyms, as well as misspellings (Leaman et al., 2015). Meanwhile, they also contain highly technical language specific to medical context, which needs to be detected by downstream NLP applications with high precision. As a result, automatic processing of clinical notes can require an effective normalization component, which transforms the raw source text into a more standardized form that can serve as more suitable input for an NLP pipeline.

The research in Chapter 2 of this thesis describes our proposed model for context-sensitive spelling correction of English and Dutch clinical free-text. While strong baselines for spelling correction often leverage corpus frequencies as inductive prior, they typically do not include the textual context of the misspelling. However, the technical specificity of clinical text could provide textual contexts which are clearly indicative of correction candidates. As of such, these contexts could be even used to detect where a specific misspelling maps to different corrections in different contexts, e.g. *iron deficiency due to ~~enemia~~* → *anemia* vs. *fluid injected with ~~enemia~~* → *enema*.

In our application, we use word and character n-gram embeddings from the fastText (Bojanowski et al., 2017) embedding model to rank spelling correction candidates according to their semantic fit in the textual context. The character n-gram embeddings allow for constructing representations for correction candidate words which are out-of-vocabulary. The pre-trained embeddings are used by a cosine similarity-based ranking model which approximates semantic contexts through weighted averages of context words. We develop this model using only automatically generated spelling errors, and explicitly control for its transferability across domains such as critical care notes and reports on colon cancer. The resulting method can successfully perform context-specific clinical spelling correction, while achieving performance which is as robust as a frequency-based noisy channel ranking model.

1.4 Biomedical name representations

The remaining chapters of this PhD are devoted to multi-task learning of biomedical name representations. We define biomedical names as those textual surface forms that represent biomedical concepts, being either official names in biomedical vocabularies or unofficial names mentioned in text. Recent research has investigated how to represent such biomedical names in a robust way for downstream NLP applications. Robustness requires that name representations should encode domain-specific knowledge, e.g. by reflecting semantic similarity between names through their closeness in the embedding space, while retaining the universal applicability and transferability of self-supervised pre-trained representations. Obvious downstream applications of robust representations include tasks such as synonym retrieval or entity linking, which links mentions of biomedical names in free-text to concept identifiers in ontologies. However, encoding specialized biomedical semantics in robust representations has the potential to also impact a wide range of other biomedical applications, such as knowledge graph completion and the discovery of relations between diseases or interactions between drugs.

In all of our chapters on biomedical name representations, we use the same neural name encoder for our proposed models. Instead of using an LSTM or Transformer architecture, we use a Deep Averaging Network (DAN) (Iyyer et al., 2015), which is a feedforward neural network processing an unordered composition of the word embeddings in a name. This encoder is a core characteristic of our robust and scalable applications, and serves two main purposes. Firstly, the encoder architecture requires much less computational cost to train, and as a result can also allow for more cost-intensive training objectives that could be more effective. Secondly, we can prove the robustness and scalability of a proposed approach by using a neural architecture that has no access to word order like LSTMs have or cannot apply attention over specific word combinations like Transformers can. This emphasises the role of factors outside of encoder complexity, and thus shows to what extent domain-specific information can be better extracted from generic self-supervised representations by e.g. only improving the training objectives.

1.4.1 Conceptual grounding

Chapter 3 focuses on the effectiveness of using conceptual grounding constraints during multi-task training of biomedical name representations. Such grounding constraints tie the output of a biomedical name encoder to specific pre-trained targets which constitute a globally coherent and meaningful embedding space. In the case of conceptual grounding, these targets are prototypical representations of the biomedical concept identifiers of names. Earlier research has indicated that such grounding can be effective using pre-trained knowledge graph embeddings which are infused with textual features (Kartsaklis et al., 2018). However, later research has indicated that simple approximations of concept representations can be similarly effective. For instance, the Biomedical Name Encoder model (Phan et al., 2019) constructs concept prototypes by averaging the pre-trained name embeddings from the set of names belonging to that concept. We use such targets in our own experiments.

Our application enriches a siamese neural network encoder for biomedical names with 2 novel constraints which effectively enforce conceptual grounding. The first constraint, which we call the linear constraint, applies canonical correlation analysis (CCA) to pre-trained 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 applies additional conceptual grounding using a training objective which forces biomedical names from the same concept to form averaged concept prototypes that approximate the pre-trained embedding of their concept identifier. The low computational cost of the DAN encoder allows us to jointly optimize entire parts of the embedding space with this second constraint instead of only stochastically iterating over single names during training.

Our experimental results show that training a DAN using conceptual grounding constraints can infuse name representations with more domain-specific semantics without losing robustness, even when trained on substantially less data than previous research. These representations can help with retrieving literal synonyms as well as semantically related terms for various biomedical ontologies, and also perform well on benchmarks of relatedness between biomedical names.

1.4.2 Higher-level semantics

While Chapter 3 trains and tests biomedical name encoders on distinctions between fine-grained concepts (i.e., concepts with no child nodes in an ontological directed graph), the remaining chapters of this thesis focus on higher-level conceptual distinctions. While such distinctions have not been investigated yet in the context of biomedical name representations, we believe that they could play a crucial role in truly capturing relevant biomedical semantics for downstream NLP applications. Earlier research shares the underlying assumption that complex neural encoder architectures can learn biomedical semantics by generalising in a bottom-up fashion from large amounts of fine-grained semantic distinctions, if provided with sufficient quantities of training data.

In Chapter 4, we show that this assumption only partially holds, and propose a novel multi-task DAN model which can generalise both bottom-up as well as top-down among various semantic hierarchies. Moreover, the resulting representations can be used out-of-the-box for unsupervised detection of hypernyms and also perform well on benchmarks of relatedness between biomedical names. Our proposed framework can even be effective using only around 30 coarse-grained higher-level classes. This opens up possibilities for applying our framework to data beyond carefully curated ontologies, for instance in self-supervised or semi-supervised settings.

1.4.3 Few-shot learning

In Chapter 5, we explore the limits of robust representation learning of biomedical names by training a feedforward neural network to transform pre-trained name embeddings using only small sets of names randomly sampled from high-level biomedical concepts. This approach is effective for various types of input representations, both domain-specific or self-supervised, and generalises well to benchmarks of relatedness between biomedical names. Most importantly, our approach allows for continual learning, where we accumulate information from various conceptual hierarchies to consistently improve encoder performance. Finding and exploiting relevant distinctions can be an empirical question, and a heuristic search among various conceptual hierarchies is computationally expensive when using more complex neural encoders. Our proposed model allows for efficiently estimating what conceptual distinctions are actually *relevant* to improve representations for downstream NLP applications. As of such, it provides a last clear example of the potential of robust and scalable application of pre-trained representations for biomedical NLP.

1.5 Publications and contributions

1.5.1 Publications

Chapter 2 has been published as follows:

- Fivez, P., Šuster, S., & Daelemans, W. (2017). Unsupervised Context-Sensitive Spelling Correction of Clinical Free-Text with Word and Character N-Gram Embeddings. In *Proceedings of the BioNLP 2017 workshop* (pp. 143–148).

- Fivez, P., Šuster, S., & Daelemans, W. (2017). Unsupervised Context-Sensitive Spelling Correction of English and Dutch Clinical Free-Text with Word and Character N-Gram Embeddings. In *CLIN Journal*.

Chapter 3 has been published as follows:

- Fivez, P., Šuster, S., & Daelemans, W. (2021). Conceptual Grounding Constraints for Truly Robust Biomedical Name Representations. In *EACL 2021*.

Chapter 4 has been published as follows:

- Fivez, P., Šuster, S., & Daelemans, W. (2021). Integrating Higher-Level Semantics into Robust Biomedical Name Representations. In *Workshop on Health Text Mining and Information Analysis (LOUHI), EACL*.

Chapter 5 has been published as follows:

- Fivez, P., Šuster, S., & Daelemans, W. (2021). Scalable Few-shot Learning of Robust Biomedical Name Representations. In *Proceedings of the BioNLP 2021 workshop*.

1.5.2 Contributions

These are some of the most notable contributions of the research presented in this thesis:

1.5.2.1 Semantic composition of textual contexts in clinical text

In Chapter 2, we have proposed a spelling correction model which successfully applies simple approximations of textual contexts using weighted compositions of word embeddings. Comparable approximations have been incorporated into e.g. concept extraction models without much success (Tulkens et al., 2019). Our application serves as empirical evidence that there are use cases in which these approximations provide added value.

1.5.2.2 Deep Averaging Networks

While Deep Averaging Networks have already been proven effective in e.g. text classification tasks, the research in Chapters 3-5 has demonstrated for the first time that their potential application includes the successful encoding of highly specialized biomedical semantics.

1.5.2.3 Conceptual grounding

While prior research had already proven the potential contribution of conceptual grounding for training encoders, we have demonstrated in Chapter

3 that applying this grounding more effectively can lead to substantial improvements in the encoding of biomedical semantics.

1.5.2.4 Higher-level biomedical semantics

Whereas prior research on biomedical name representations has consistently focused on distinctions between fine-grained biomedical concepts, we have provided a framework in Chapter 4 to train and evaluate encoders for higher-level categorizations of biomedical names as well. Moreover, our proposed model can generalize biomedical semantics both bottom-up as well as top-down along semantic hierarchies.

1.5.2.5 Encoding hypernymy through cosine similarity

The biomedical name encoder which we have proposed in Chapter 4 can perform unsupervised detection of hypernyms using only cosine similarity. This contrasts with earlier approaches for encoding hypernymy which explicitly require more than cosine similarity to properly work. For example, Vulić and Mrkšić (2018) use vector norms to encode hierarchical hypernymic relations, while other research into hypernymy even requires other geometric spaces than Euclidean space, such as hyperbolic space (Dhingra et al., 2018). Our results can indicate that cosine similarity in Euclidean space still shows potential for encoding these hierarchical relations given the right training objectives.

1.5.2.6 Few-shot biomedical name representations

Prior research on biomedical name encoders trains complex neural encoder architectures to learn biomedical semantics by generalising in a bottom-up fashion from large amounts of fine-grained semantic distinctions. In Chapter 5, we have demonstrated that few-shot top-down approximations of biomedical semantics are easily achievable and could even prove to be more efficient.

1.5.2.7 Continual learning of biomedical name representations

While prior research on biomedical name encoders is mostly concerned with modeling all fine-grained distinctions within a single large ontology, we have demonstrated in Chapter 5 that our few-shot top-down approximations of biomedical semantics can be accumulated over multiple different hierarchies. This can highlight the added value of specific ontological categorizations for downstream biomedical NLP applications.

2

Unsupervised Context-Sensitive Spelling Correction of English and Dutch Clinical Free-Text with Word and Character N-Gram Embeddings

In this chapter, we present an unsupervised context-sensitive spelling correction method for clinical free-text that uses word and character n-gram embeddings. Our method generates misspelling replacement candidates and ranks them according to their semantic fit, by calculating a weighted cosine similarity between the vectorized representation of a candidate and the misspelling context. To tune the parameters of this model, we generate self-induced spelling error corpora. We perform our experiments for two languages. For English, we greatly outperform off-the-shelf spelling correction tools on a manually annotated MIMIC-III test set, and counter the frequency bias of a noisy channel model, showing that neural embeddings can be successfully exploited to improve upon the state-of-the-art. For Dutch, we also outperform an off-the-shelf spelling correction tool on manually annotated clinical records from the Antwerp University Hospital, but can offer no empirical evidence that our method counters the frequency bias of a noisy channel model in this case as well. However, both our context-sensitive model and our implementation

of the noisy channel model obtain high scores on the test set, establishing a state-of-the-art for Dutch clinical spelling correction with the noisy channel model.

2.1 Introduction

The problem of automated spelling correction has a long history, dating back to the late 1950s.[1] Traditionally, spelling errors are divided into two categories: non-word misspellings, the most prevalent type of misspellings, where the error leads to a nonexistent word, and real-word misspellings, where the error leads to an existing word, either caused by a typo (e.g. *I ~~hole~~* → *hope so*), or as a result of grammatical (e.g. *their* - *there*) or lexical (e.g. *aisle* - *isle*) confusion. The spelling correction task can be divided into three subtasks: detection of misspellings, generation of replacement candidates, and ranking of these candidate replacements to correct the misspelling. The nature of the detection subtask is dependent on the type of error: non-word misspellings are typically defined as tokens absent from a reference lexicon, while for real-word misspellings, the detection task is postponed by considering all tokens as replaceable, using the confidence of the candidate ranking module to determine which tokens should be treated as misspellings. The generation of replacement candidates is typically performed by including all items from a lexicon which fall within a predefined edit distance of the misspelling (e.g. all items within a Levenshtein distance of 3). The ranking component is the most complex of the three subtasks, and is the main topic of this paper.

The genre of clinical free-text poses an interesting challenge to the spelling correction task, since it is notoriously noisy. English corpora contain observed spelling error rates which range from 0.1% (H. Liu et al., 2012) and 0.4% (Lai et al., 2015) to 4% and 7% (Tolentino et al., 2007), and even

[1] A good overview is given by Mitton (2010) and Jurafsky and Martin (2016).

10% (Ruch et al., 2003). Moreover, clinical text also has variable lexical characteristics, caused by a broad range of domain- and subdomain-specific terminology and language conventions. These properties of clinical text can render traditional spell checkers ineffective (Patrick et al., 2010). Recently, Lai et al. (2015) have achieved nearly 80% correction accuracy on a test set of clinical notes with their noisy channel model. However, their ranking model does not use any contextual information, while the context of a misspelling can provide important clues for the spelling correction process, for instance to counter the frequency bias of a context-insensitive system based on corpus frequency. As an example, consider the misspelling *~~goint~~* → *going* present in the MIMIC-III (Johnson et al., 2016) clinical corpus. While in many domains, *going* will be a relatively frequent word type and will consequently be picked by a corpus frequency-based system, it is actually outnumbered in MIMIC-III by the more prevalent word types *joint* and *point*, which are other replacement candidates for the same misspelling. In other words, corpus frequency is not a reliable metric in such cases. Flor (2012) also pointed out that ignoring contextual clues harms performance where a specific misspelling maps to different corrections in different contexts, e.g. *iron deficiency due to ~~enemia~~* → *anemia* vs. *fluid injected with ~~enemia~~* → *enema*. A noisy channel model like the one by Lai et al. (2015) will choose the same item for both corrections.

Our proposed unsupervised context-sensitive method exploits contextual clues by using neural embeddings to rank misspelling replacement candidates according to their semantic fit in the misspelling context. Neural embeddings have proven useful for a variety of related tasks, such as unsupervised normalization (Sridhar, 2015) and reducing the candidate search space for spelling correction (Pande, 2017). We hypothesize that, by using neural embeddings, our method can counter the frequency bias of a noisy channel model. We test our system on manually annotated misspellings from the MIMIC-III corpus. We also conduct experiments on Dutch data, since there is still a need for a Dutch spelling correction method for clinical free-text (Cornet et al., 2012). By replicating our English research setup for Dutch, we simultaneously examine the language adaptability of our context-sensitive model, and establish a state-of-the-art for Dutch clinical spelling correction. We test our Dutch model on manually annotated misspellings from clinical records collected at the Antwerp University Hospital (UZA). In our experiments for both English and Dutch, we focus on already detected non-word misspellings for developing and testing our spelling correction method, following Lai et al (2015). Note that our method could also be applied to real-word errors. However, since our strategy for collecting an empirical test set of misspellings, which we describe in section 2.3.4, can not be used for real-word errors, we do not address them in this article.

2.2 Approach

Since we focus on already detected non-word misspellings, our system only deals with two subtasks of the spelling correction task, namely, generating candidate replacements and ranking them.

2.2.1 Candidate Generation

We generate replacement candidates in 2 phases, using the reference lexicons described in section 2.3.1. First, we extract all items within a Damerau-Levenshtein edit distance of 2 from a reference lexicon. Secondly, to allow for candidates beyond that edit distance, we also apply the Double Metaphone matching popularized by the open source spell checker Aspell[2]. This algorithm converts lexical forms to an approximate phonetic consonant skeleton, and matches all Double Metaphone representations within a Damerau-Levenshtein edit distance of 1. The Double Metaphone representation is an intentionally approximate phonetic representation, which

[2] http://aspell.net/metaphone/

is created with an elaborate set of rules, and whose principles of design include mapping voiced/unvoiced consonant pairs to the same encoding, encoding any initial vowel with 'A', and disregarding all non-initial vowel sounds. For example, the Double Metaphone representation of *antibiotic* is *ANTPTK*.

2.2.2 Candidate Ranking

Our approach computes the cosine similarity between the vector representation of a candidate and the composed vector representations of the misspelling context, weights this score with other parameters, and uses it as the ranking criterium. This setup is similar to the contextual similarity score by Kilicoglu et al. (2015), which proved unsuccessful in their experiments. However, their experiments were preliminary. They used a limited context window of 2 tokens, could not account for candidates which are not observed in the training data, and did not investigate whether a bigger training corpus would lead to vector representations which scale better to the complexity of the task.

We undertake a more thorough examination of the applicability of neural embeddings to the spelling correction task. To tune the parameters of our context-sensitive spelling correction model in an unsupervised way, we automatically generate development corpora with artificial, randomly created spelling errors for three different scenarios following the procedures described in section 2.3.3. These three types of generated spelling error corpora, which we refer to as *setups*, are increasingly difficult for the spelling correction task. We apply the same setups to both English and Dutch. **Setup 1** is generated from the same corpus which is used to train the neural embeddings, and exclusively contains corrections which are present in the vocabulary of these neural embeddings. **Setup 2** is generated from a corpus in a different clinical subdomain, and also exclusively contains in-vector-vocabulary corrections. **Setup 3** presents the most

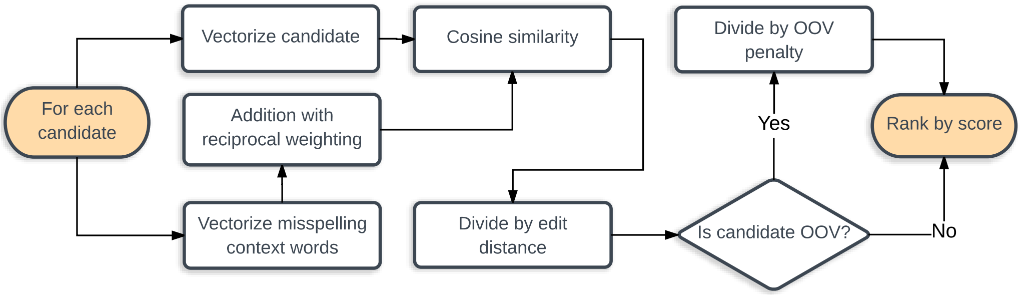

Fig. 2.1.: The final architecture of our model. Within a specified window size (9 for English, 10 for Dutch), it vectorizes every context word on each side if it is present in the vector vocabulary, applies reciprocal weighting, and sums the representations. It then calculates the cosine similarity with each candidate vector, and divides this score by the Damerau-Levenshtein edit distance between the candidate and misspelling. If the candidate is OOV, the score is divided by an OOV penalty.

difficult scenario, where we use the same corpus as for Setup 2, but only include corrections which are not present in the embedding vocabulary (OOV). In other words, here our model has to deal with both domain change and data sparsity.

Correcting OOV tokens in Setup 3 is made possible by using a combination of word and character n-gram embeddings. We train these embeddings with the fastText model (Bojanowski et al., 2017), an extension of the popular Word2Vec model (Mikolov, Chen, et al., 2013), which creates vector representations for character n-grams and sums these with word unigram vectors to create the final word vectors. FastText allows for creating vector representations for misspelling replacement candidates absent from the trained embedding space, by only summing the vectors of the character n-grams.

We report our development experiments with the different setups in section 2.4.1. The final architecture of our model for both English and Dutch is described in Figure 2.1. We evaluate this model on our test data in section 2.4.2.

Language	Corpus type	Domain	Data used	Instances
English	Development: Setup 1	critical care	mimic-iii	5,000
	Development: Setup 2	breast/colon cancer	thyme	5,000
	Development: Setup 3	breast/colon cancer	thyme	1,500
	Test	critical care	mimic-iii	873
Dutch	Development: Setup 1	critical care	uza	5,000
	Development: Setup 2	breast/colon cancer	uza	5,000
	Development: Setup 3	breast/colon cancer	uza	350
	Test	miscellaneous	uza	490

Tab. 2.1.: A comprehensive overview of our corpora described in section 2.3.3 and 2.3.4.

2.3 Materials

We tokenize all English data with the Pattern tokenizer (Smedt & Daelemans, 2012), and all Dutch data with Ucto[3]. All text is lowercased[4], and we remove all tokens that include anything different from alphabetic characters or hyphens. Table 2.1 gives a comprehensive overview of the English and Dutch development and test corpora we describe in section 2.3.3 and 2.3.4.

2.3.1 Lexicons

To construct reference lexicons, we fuse general dictionaries with specialized resources. For our English lexicon, we use a union of the general dictionary from Jazzy[5], a Java open source spell checker (47,160 items), and the UMLS® SPECIALIST lexicon[6] (304,840 items), which contains a broad range of specialized clinical terms. This amounts to 319,579 unique

[3] https://languagemachines.github.io/ucto/

[4] While this has consequences for the nature of the task, it is a salient aspect of training good embeddings. Lowercasing reduces sparsity, therefore leading to more reliable representations, especially in the case of low frequency words.

[5] http://jazzy.sourceforge.net

[6] https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lexicon/current/web/index.html

	Misspelling	**Candidates**
Setup 1	*unchanged* → *unchainged*	unchanged, unchained, uncharged, unhinged
Setup 2	*chronic* → *chornic*	chronic, choreic, cornice, chloric
Setup 3	*accrued* → *accued*	accrued, accused, accuse, accede

Tab. 2.2.: Examples of automatically generated spelling errors and some replacement candidates for the English development setups.

lexical items. For our Dutch lexicon, we use as general dictionary the publicly available word list from Stichting OpenTaal[7] (320,913 tokens), which has the official quality label of the Dutch Language Union. As specialized resource, we extract terminology from two clinical resources, namely, the Belgian Bilingual Biclassified Thesaurus (23,794 items) constructed by the universities of Ghent and Brussels, and the UMLS® Metathesaurus[8] (77,646 items). This amounts to 371,559 unique lexical items.

2.3.2 Neural embeddings

We train a fastText skipgram model using the default parameters, except for the dimensionality, which we raise to 300, since we want to make sure that the embeddings are able to capture subtle semantic relationships in a training corpus of our size. For our English experiments, we train on 425M words from the MIMIC-III corpus, which contains medical records from critical care units. For our Dutch experiments, we train on 720M words from clinical records collected at the Antwerp University Hospital (UZA). These records span a decade in time, and cover various genres (notes, letters, protocols, reports) as well as a wide range of clinical subdomains, including gastroenterology, pulmonology, and critical care.

[7] https://www.opentaal.org

[8] https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/

	Misspelling	Candidates
Setup 1	*mediane → medciane*	mediane, mediale, medianen, Mediene
Setup 2	*beperkt → beprekt*	beperkt, betrekt, verrekt, gerekt, bevlekt
Setup 3	*megacyste → megacyte*	megacyste, megabyte, megabytes

Tab. 2.3.: Examples of automatically generated spelling errors and some replacement candidates for the Dutch development setups.

2.3.3 Development corpora

In order to tune our model parameters in an unsupervised way, we automatically create self-induced error corpora. We generate these development corpora by randomly sampling lines from a reference corpus, randomly sampling a single word per line if the word is present in our reference lexicon, transforming these words with either 1 (80%) or 2 (20%) random Damerau-Levenshtein operations to a non-word, and then extracting these misspelling instances with a context window of up to 10 tokens on each side. Table 2.1 gives an overview of all the development corpora and the data used to generate them. Table 2.2 and 2.3 give examples from all development corpora for both languages. For **Setup 1**, we perform our corpus creation procedure for critical care records, a domain which is present in the data used to train our neural embeddings. We exclusively sample words present in our vector vocabulary, resulting in 5,000 tokens for both English and Dutch. For **Setup 2**, we perform our procedure for records which exclusively cover the domain of brain and colon cancer, which is not represented in our neural embedding corpora. For English, we use the THYME (Styler IV et al., 2014) corpus. For Dutch, we use data which originally belonged to our neural embeddings training data, but which was located and held out before our experiments. We once again exclusively sample in-vector-vocabulary words, resulting in 5,000 tokens for both English and Dutch. For **Setup 3**, we again perform our procedure for the cancer corpora, but this time we exclusively sample OOV words, resulting in 1,500 tokens for English and 350 for Dutch. While this last setup can seem exaggerated or overly artificial, we want to explicitly isolate these cases from the other setups, since the distribution of OOVs is entirely dependent on the vocabulary overlap between the data being corrected and the data used to train the neural embeddings. In other words, it is relative with respect to the specific use case of our model in practice. On the one hand, we use this setup to estimate how well our trained model can generalize to other subdomains and corpora with only partially overlapping vocabulary; on the other hand, we use this setup to regulate the role of OOV correction candidates, as we discuss in section 2.4.1.

2.3.4 Test corpora

No benchmark test sets are publicly available for clinical spelling correction. A straightforward annotation task is costly and can lead to small corpora, such as the one by Lai et al. (2015), which contains just 78 misspelling instances. Therefore, we adopt a more cost-effective annotation approach. In a corpus, we spot misspellings by looking at items with a frequency of 5 or lower which are absent from our lexicon.[9] We then extract and annotate instances of these misspellings along with their context. For English, we use the MIMIC-III data, resulting in 873 contextually different tokens of 357 unique error types.[10] For Dutch, we use a recent set of clinical records from the Antwerp University Hospital, which covers the same genres and domains as the neural embeddings training data. This results in 490 contextually different tokens of 359 unique error types. Tables 2.4 and 2.5 give examples from both test corpora.

[9] While this excludes frequent error types, and is therefore far from an optimal strategy, it is hard to estimate the possible deceiving effect of this strategy without knowing the frequency distribution of spelling errors in the MIMIC-III corpus.

[10] A script to extract this data can be found at https://github.com/clips/clinspell.

	Misspelling	**Candidates**
Edit distance 1	*sclerosing* → *sclerosin*	sclerosing, sclerosis, sclerotin, sclerostin
Edit distance 2	*symptoms* → *sympots*	symptoms, symptom, spots, symbols
Edit distance 3	*phlebitis* → *phebilitis*	phlebitis, cheilitis, pyelitis, phallitis

Tab. 2.4.: Examples of empirically observed misspellings and some replacement candidates from our English test set, per Damerau-Levenshtein edit distance.

	Misspelling	**Candidates**
Edit distance 1	*letsels* → *letels*	letsels, lepels, netels, zetels, zetsels
Edit distance 2	*weinig* → *wijnig*	weinig, pijnig, wijzig, tijdig, wijn
Edit distance 3	*verminderde* → *verminderderde*	verminderde, verminderende

Tab. 2.5.: Examples of empirically observed misspellings and some replacement candidates from our Dutch test set, per Damerau-Levenshtein edit distance.

2.4 Results

We first develop our model for each language by tuning the parameters with the development corpora. We then test this tuned model on the test data. We discuss the results and their implications in the next section. To evaluate the performance of our model, we use **first-best** accuracy as criterion, i.e., the percentage of misspellings which are properly corrected by the first-ranked replacement suggestion of our model. We use two variations of first-best accuracy, the terminology of which we borrow from Reynaert (2008): **true** first-best accuracy, which is the accuracy given the system's dictionary; and **upper-bound** first-best accuracy, which removes the effect of dictionary shortcomings, by adding all correct word forms for the errors to be corrected to the system's spelling dictionary. The latter criterion allows for measuring the upper bound on correction attainable by our system.

2.4.1 Development

To develop our model, we investigate a variety of parameters:

Vector composition functions

- addition
- multiplication
- max embedding by Wu et al. (2015)

Edit distance penalty

- Damerau-Levenshtein
- Double Metaphone
- Damerau-Levenshtein + Double Metaphone
- Spell score by Lai et al. (2015)

Context parameters

- Window size (1 to 10)
- Reciprocal weighting
- Removing stop words using the English stop word list from scikit-learn (Pedregosa et al., 2011) or the Dutch stop word list from Pattern (Smedt & Daelemans, 2012)
- Including a vectorized representation of the misspelling

We perform a grid search for Setup 1 and Setup 2 to discover which parameter combination leads to the highest accuracy averaged over both corpora. In this setting, we only allow for candidates which are present in the vector vocabulary. We then introduce OOV candidates for Setup 1, 2 and 3, and experiment with penalizing them, since their representations are less reliable. As these representations are only composed out of character n-gram vectors, with no word unigram vector, they are susceptible to noise caused by the particular nature of the n-grams; namely, sometimes the semantic similarity of OOV vectors to other vectors can be inflated in cases of strong orthographic overlap. OOV replacement candidates are more often redundant than necessary, as in most practical use cases of the correction model (where there is considerable vocabulary overlap between the embedding domain and the correction domain), the majority of correct misspelling replacements will be present in the trained vector space. Therefore, we try to penalize OOV representations to the extent that they do not cause noise in cases where they are redundant, but still rank first in cases where they are the correct replacement. We tune this OOV penalty by maximizing the accuracy for Setup 3 while minimizing the performance drop for Setup 1 and 2, using a weighted average of their correction accuracies.

The final architecture of our model for both English and Dutch is described in full in Figure 2.1, showing all used parameters. As the description shows, the models for both languages only differ in optimal window size (9 for English, 10 for Dutch). To compare our method against a reference noisy channel model in the most direct way, we implement the ranking component of Lai et al. (2015) in our pipeline (**Noisy Channel**). This component requires corpus frequencies, which we extract from the same data that we use to train the embeddings. Our context-sensitive model (**Context**) outperforms the noisy channel for each corpus in our development phase, for both English and Dutch, as shown in Table 2.6 and 2.7. Moreover, as the results for Setup 3 show, our method generalizes considerably better to OOV misspellings, as we explicitly intended in the development of our model.

	Setup 1	Setup 2	Setup 3
Context	90.24	88.20	57.00
Noisy Channel	85.02	85.86	39.73

Tab. 2.6.: True first-best correction accuracies for our 3 English development setups.

	Setup 1	Setup 2	Setup 3
Context	87.94	89.10	82.00
Noisy Channel	86.90	85.80	66.57

Tab. 2.7.: True first-best correction accuracies for our 3 Dutch development setups.

2.4.2 Test

Table 2.8 shows the English correction accuracies for **Context** and **Noisy Channel** as off-the-shelf tools, compared to two existing tools. The first tool is HunSpell, a popular open source spell checker used by Google Chrome and Firefox. The second tool is the original implementation of the model by Lai et al. (2015), which they shared with us. Table 2.9 shows the Dutch correction accuracies for **Context** and **Noisy Channel** as off-the-shelf tools, as compared to HunSpell.

The performance of our models on the test sets is held back by the incomplete coverage of our reference lexicons. For English, missing corrections are mostly highly specialized medical terms, or inflections of more common terminology. For Dutch, this includes relatively infrequent compounds as well. Compounds in Dutch, as opposed to English, are mostly orthographically concatenated into one lexical item. Since Dutch language users tend to be very productive with compounding, this leads to a whole range of standard language that is hard to cover exhaustively in a lexicon. We use the upper-bound first-best correction accuracy to examine the performance

Evaluation	HunSpell	Lai et al.	Context	Noisy Channel
TRUE FIRST-BEST ACCURACY	52.69	61.97	88.21	87.85
UPPER-BOUND FIRST-BEST ACCURACY			93.02	92.66

Tab. 2.8.: The correction accuracies for our English test experiments, evaluated for two different scenarios. *True first-best accuracy*: gives the first-best accuracies of all off-the-shelf tools. *Upper-bound first-best accuracy*: gives the first-best accuracies of our implemented models for the scenario where correct replacements missing from the lexicon are included in the lexicon before the experiment.

Evaluation	HunSpell	Context	Noisy Channel
TRUE FIRST-BEST ACCURACY	64.29	76.53	79.71
UPPER-BOUND FIRST-BEST ACCURACY		87.75	92.45

Tab. 2.9.: The correction accuracies for our Dutch test experiments, evaluated for two different scenarios. *True first-best accuracy*: gives the accuracies of all off-the-shelf tools. *Upper-bound first-best accuracy*: gives the accuracies of our implemented models for the scenario where correct replacements missing from the lexicon are included in the lexicon before the experiment.

of our ranking models with disregard to such circumstances. Tables 2.8 and 2.9 show that the performance according to this metric is comparable to the true first-best correction accuracy for the development corpora.

2.5 Discussion

In terms of correction accuracy, our context-sensitive model and our own implementation of Lai et al.'s ranking model outperform off-the-shelf tools for both English and Dutch, establishing a state-of-the-art for spelling correction of clinical free-text. The salient difference in performance between Lai et al.'s system and our specific implementation of their noisy channel model highlights the influence of (lack of) training resources and development decisions on the general applicability of spelling correction models. Moreover, it shows the strength of the noisy channel model in scenarios where the scale of the resources is sufficient (in this case, 425M words for English and 720M words for Dutch) to reliably estimate prior probabilities from corpus frequencies.

However, sufficient empirical resources to estimate a fine-grained likelihood (namely, a large corpus of empirically observed errors from which a reliable error model can be extracted) are still absent for the clinical domain. Therefore, the likelihood of Lai et al.'s ranking model is estimated with a rudimentary spell score, which is a weighted combination of Damerau-Levenshtein and Double Metaphone edit distance. While this error model leads to a noisy channel model which is robust in performance, as shown by our test results, it also leads to a pragmatic performance ceiling where more heavily distorted replacement candidates are downplayed to safeguard robustness of performance, regardless of their possible empirical association with the misspelling. As a result, our noisy channel model is still prone to cases of frequency bias, including the example of frequency bias which we have provided in the introduction of this paper: our noisy channel model does not succeed in correcting the MIMIC-III misspelling *goint* to the correct form *going* due to the higher corpus frequency of, and therefore higher prior probability assigned to, the word type *point*. While the difference in frequency is salient, it is not insurmountable for a likelihood reflecting a proper error model, which in this case would typically reflect that *goint* is more probable to be a typo of *going* than of *point*. However, the rudimentary spell score does not reflect that notion. This example illustrates that, regardless of the theoretical validity of the noisy channel, we are still very much bound to the practical reality of its implementation, including the state of resources.

Our method tries to improve on the clinical spelling correction process considering the availability of actual incomplete resources. As it stands, a noisy channel model like the one by Lai et al. (2015) still occasionally

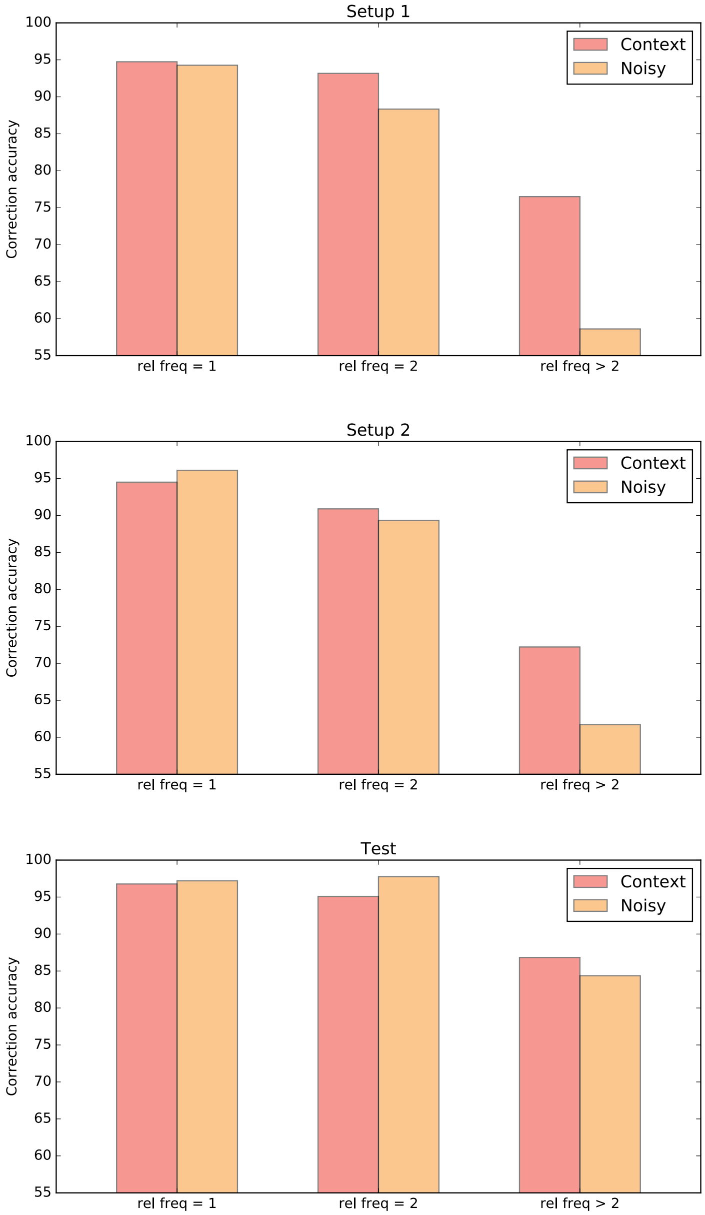

Fig. 2.2.: The English correction accuracies for **Context** and **Noisy Channel** for Setup 1, Setup 2, and the test set, grouped per relative frequency of the correct replacement compared to other replacement candidates. *rel freq* $= 1$: highest corpus frequency of all candidates. *rel freq* $= 2$: second highest corpus frequency of all candidates. *rel freq* > 2: corpus frequency lower than second highest of all candidates.

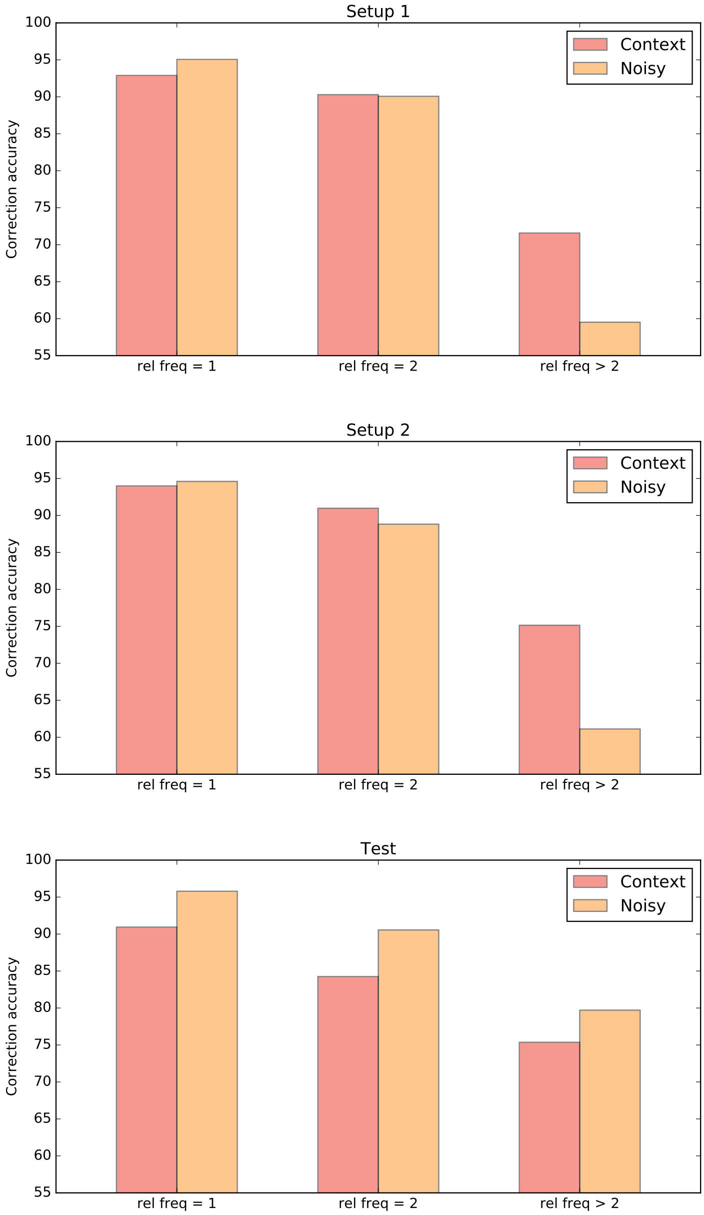

Fig. 2.3.: The Dutch correction accuracies for **Context** and **Noisy Channel** for Setup 1, Setup 2, and the test set, grouped per relative frequency of the correct replacement compared to other replacement candidates. *rel freq = 1*: highest corpus frequency of all candidates. *rel freq = 2*: second highest corpus frequency of all candidates. *rel freq > 2*: corpus frequency lower than second highest of all candidates.

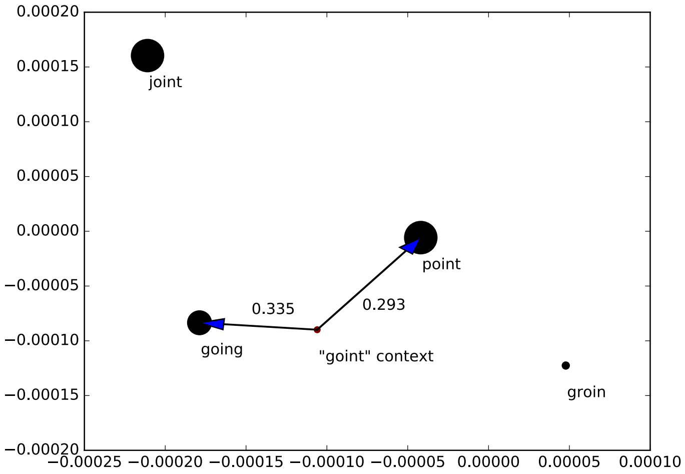

Fig. 2.4.: 2-dimensional t-SNE projection of the vectorized context of the English test misspelling *goint* and 4 replacement candidates in the trained MIMIC-III vector space. Dot size denotes corpus frequency, numbers denote cosine similarity. The English misspelling context is *new central line lower extremity bypass with sob now [goint] to [be] intubated*. While the noisy channel chooses the more frequent *point*, our model correctly chooses the most semantically fitting *going*.

suffers from frequency bias; it is not able to correct a specific misspelling type to different corrections in different contexts, and is not sufficiently equipped to deal with word types that are not observed in training data. Our unsupervised context-sensitive model targets these weaknesses. Figures 2.2 and 2.3 show the correction accuracies for three scenarios: one where the most frequent candidate is the correct one (*rel freq = 1*), one where the second most frequent candidate is the correct one (*rel freq = 2*), and one where the correct candidate has a lower relative frequency (*rel freq > 2*). Figure 2.2 confirms the hypothesis that our context-sensitive model counters the frequency bias of a noisy channel model for our English experiments. The results for our development corpora show that in cases where *rel freq = 1*, the noisy channel scores similar or slightly better, as expected. This trend is reflected in the test results. In cases where *rel freq = 2*, our model scores slightly better. This trend is not reflected in the test results. In fact, it is reversed. Lastly, in cases where *rel freq > 2*, our model scores much better. This trend is reflected in the test results, if to a smaller extent. However, the relatively small sample size (a difference of 6 correct instances on a total of 243) should be kept in mind. Figure 2.4 visualizes an example of frequency bias, where the *goint* misspelling which we discussed earlier is correctly handled by our model as opposed to the noisy channel model.

Figure 2.3 shows that the performance our context-sensitive model exhibits the same characteristics for the Dutch development corpora as for the English development corpora. However, this time none of the trends are reflected in the test results, which leads to our model being outperformed by the noisy channel model. This discrepancy raises the question to what extent the artificial nature of the development corpora leads to reliable models for empirical data. If the distributions of the several data types differ greatly, this undermines our unsupervised approach, which implicitly assumes that the distributions will not differ that greatly. To investigate this, we performed a grid search for both the English and Dutch test corpus, to examine which parameter combination leads to the best-performing model.

For the English test data, this parameter combination is similar to our actual model derived from our development experiments. In other words, the underlying assumption of our unsupervised approach is confirmed.

For the Dutch test data, however, the optimal parameter combination differs dramatically from our developed model. It includes two parameters which are absent from our developed model described in Figure 2.1: the context representation also includes a vectorized representation of the misspelling itself, and the edit distance weighting adds Double Metaphone edit distance to the Damerau-Levenshtein edit distance. Moreover, the optimal context window size is 2, which is considerably smaller than for the originally developed model. With this parameter combination, the output of the model for the Dutch test data is exactly similar to the output of the noisy channel model. These analyses suggest that the distribution of the Dutch test data differs greatly from that of the development data. This discrepancy can be caused by the sparsity of the Dutch test data, which covers the same amount of error types as the English test data, but much fewer contextually different instances. The only conclusion we can draw is that the nature of our test set is possibly skewed in a way that does not allow for a thorough comparative evaluation of our models. As it stands, however, we have no empirical evidence that our Dutch context-sensitive model actually counters the frequency bias of our noisy channel. While we want to avoid too much speculation as to the reason why, these results invite inquiry into how important context actually is for Dutch clinical spelling correction.

When we look at the output of our context-sensitive model for both English and Dutch, we can categorize the errors it makes in 3 different types. The first type of errors concerns, predictably, misspellings for which the contextual clues are too unspecific. This lack of useful contextual information is sometimes caused by occurrences of other misspellings in the context window, and poses a fundamental challenge to our method. The second type of errors concerns cases where the contextual clues are actually misguiding. This happens for instance in cases where a word type has multiple senses which are not strongly related. Our Dutch test set contains the misspelling ~~*poslen*~~ → *polsen*, where from the context it appears that *polsen* has the more infrequent sense of 'polling someone about something' instead of the prevalent sense 'wrists'. Since this word type shares one vector representation for both senses, the contextual information does not turn out to be strong enough for correcting the misspelling to the correct word type. Lastly, while our development experiments have tried to minimize the noise spread by OOV candidates, it is still noticeable in some instances.

2.6 Conclusion and future research

In this paper, we have proposed an unsupervised context-sensitive model for clinical spelling correction which uses word and character n-gram embeddings. This simple ranking model, which can be tuned to a specific language and domain by generating self-induced error corpora, tries to counter the frequency bias of a noisy channel model by exploiting contextual clues.

As an implemented spelling correction tool for English clinical free-text, our method outperforms both a broadly used and a domain-specific off-the-shelf tool for empirically observed misspellings in MIMIC-III. Moreover, a detailed analysis of its performance shows that it does in fact counter the frequency bias of a noisy channel model. However, the relatively small sample size for this analysis should be kept in mind.

As an implemented spelling correction tool for Dutch clinical free-text, our method outperforms a broadly used off-the-shelf tool for empirically observed misspellings in collected data from the Antwerp University Hospital. However, our Dutch test set offers no empirical evidence that it counters the frequency bias of a noisy channel model. It is unclear whether this is caused by the sparsity of the test set.

Future research can investigate whether our method transfers well to other genres and domains. Secondly, it can address the three problem areas we have identified at the end of our discussion in section 2.5, namely, unspecific contextual clues, multiple word senses of a single word type, and noise spread by OOV candidates. Lastly, it is worthwhile to investigate how successfully our model can be applied to real-word errors.

3 Conceptual Grounding Constraints for Truly Robust Biomedical Name Representations

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 chapter, 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.

3.1 Introduction

Biomedical and clinical free-text contain mentions of biomedical terms which can provide valuable information for text mining applications. Such

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*

Tab. 3.1.: 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*.

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 (D'Souza & Ng, 2015; Leaman et al., 2013) to dense distributed representations (Li et al., 2017; Limsopatham & Collier, 2016; 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.

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 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 3.1 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.

3.2 Related work

3.2.1 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.

3.2.2 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 (Shen et al., 2018; Wieting et al., 2016; Wieting & 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.

3.2.3 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.

3.3 Encoding model

3.3.1 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*:

$$\begin{aligned} u_n &= \frac{1}{|N_t|}\sum_{t\in N_t} u_t \\ f(n) &= enc(u_n) \end{aligned} \tag{3.1}$$

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.

3.3.2 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:

$$u_p = \frac{1}{|C_n|} \sum_{n \in C_n} u_n \tag{3.2}$$

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

3.3.2.1 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.

3.3.2.2 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:

$$\begin{aligned} pos &= d(f(CCA(n)), f(CCA(n_{pos}))) \\ neg &= d(f(CCA(n)), f(CCA(n_{neg}))) \\ L_{syn} &= max(pos - neg + margin, 0) \end{aligned} \tag{3.3}$$

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 (C.-Y. Wu et al., 2017) over all training names.

3.3.2.3 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:

$$\begin{aligned} f(p) &= \frac{1}{|C_n|} \sum_{n \in C_n} f(CCA(n)) \\ L_{proto} &= d(f(p), CCA(u_p)) \end{aligned} \tag{3.4}$$

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 & Kiela, 2019).

3.3.2.4 Multi-task setup

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

$$L = L_{syn} + L_{proto} \tag{3.5}$$

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.

3.4 Data

3.4.1 Disorder names

3.4.1.1 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. Since they do not reflect relevant synonymy, we leave out such duplicates.[2]. 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.

3.4.1.2 ICD-10

The SNOMED-to-ICD-10 mapping, which has been officially provided by the U.S. National Library of Medicine[3], groups multiple SNOMED-CT concepts together under more coarse-grained ICD-10 codes, using concept

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

[2] We list all redundant metatags in the Supplementary Materials

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

	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

Tab. 3.2.: An overview of all the data used in our experiments.

unique identifiers (CUIs) from the UMLS[4] 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 3.1 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*).

3.4.2 Heterogeneous names: MedMentions

The recently released MedMentions corpus (Mohan & 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

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

to UMLS concepts covering 21 different semantic 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.

3.5 Experiments and results

3.5.1 Ranking tasks and data distributions

3.5.1.1 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.

3.5.1.2 Data distributions

Table 3.2 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.

3.5.2 Reference model and baselines

3.5.2.1 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 & Schmidhuber, 2005) with max pooling and a linear transformation:

$$\begin{aligned} h_n = max(BiLSTM(u_{[}t_1], .., u_{[}t_n])) \\ f(n) = W(h_n) + b \end{aligned} \tag{3.6}$$

We also include the publicly released BNE model with skipgram word embeddings, BNE + $\mathrm{SG_w}$, [5] 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.

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

3.5.2.2 Baselines

As baseline encoder we use the 300-dimensional **fastText** name embeddings which are used as input for the DAN (defined in Equation 3.1 in Section 3.3.1). 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).

3.5.3 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 3.3.1). 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 was compared fairly to our model. At that point, the DAN has ±23M trainable parameters, whereas the BiLSTM already has ±200M trainable parameters.

	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**

Tab. 3.3.: Synonym retrieval and concept mapping scores for the ICD-10 encoders. The highest score is denoted in bold, the second highest is underlined.

This allows us to empirically confirm that our proposed DAN model is more computationally efficient than the BNE BiLSTM.

Adam optimization (Kingma & 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 & Daelemans, 2012) and then lowercased. We use a triplet margin of 0.1 for the siamese triplet loss L_{syn} defined in Equation 3.3. For the prototypical constraint L_{proto} defined in Equation 3.4, 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.

3.5.4 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.

	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

Tab. 3.4.: Synonym retrieval and concept mapping scores for the SNOMED-CT 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	0.30	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

Tab. 3.5.: 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

Tab. 3.6.: 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.

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

Tab. 3.7.: 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.

3.5.4.1 ICD-10 & SNOMED-CT

Table 3.3 and 3.4 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 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 3.6 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*.

3.5.4.2 MedMentions

Table 3.5 shows the performance of the different encoders for the MedMentions data. Table 3.7 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.

3.5.5 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 3.8 confirm the robustness of our conceptually grounded biomedical name representations. While the correlations for

	MayoSRS	UMNSRS	UMNSRS
	(rel)	(rel)	(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

Tab. 3.8.: 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.

the BNE models barely improve over those of the fastText embeddings, our CCA+DAN encoder improves substantially over all 3 benchmarks, regardless 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.

3.6 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.

4

Integrating Higher-Level Semantics into Robust Biomedical Name Representations

Neural encoders of biomedical names are typically considered robust if representations can be effectively exploited for various downstream NLP tasks. To achieve this, encoders need to model domain-specific biomedical semantics while rivaling the universal applicability of pretrained self-supervised representations. Previous work on robust representations has focused on learning low-level distinctions between names of fine-grained biomedical concepts. These fine-grained concepts can also be clustered together to reflect higher-level, more general semantic distinctions, such as grouping the names nettle sting and tick-borne fever together under the description puncture wound of skin. It has not yet been empirically confirmed that training biomedical name encoders on fine-grained distinctions automatically leads to bottom-up encoding of such higher-level semantics. In this paper, we show that this bottom-up effect exists, but that it is still relatively limited. As a solution, we propose a scalable multi-task training regime for biomedical name encoders which can also learn robust representations using only higher-level semantic classes. These representations can generalise both bottom-up as well as top-down among various semantic hierarchies. Moreover, we show how they can be used out-of-the-box for improved unsupervised detection of hypernyms, while retaining robust performance on various semantic relatedness benchmarks.

<table>
<tr><td>Level 1</td><td colspan="4"><b>C0564444</b>
<i>wound of skin</i></td></tr>
<tr><td>Level 2</td><td colspan="4"><b>C0561369</b>
<i>puncture wound of skin</i></td></tr>
<tr><td>Level 3</td><td colspan="2"><b>C0561546</b>
<i>bite wound</i></td><td colspan="2"><b>C0576723</b>
<i>sting of skin</i></td></tr>
<tr><td>Level 4</td><td><b>C1302713</b>
<i>animal bite wound</i></td><td><b>C0275134</b>
<i>poisoning due to lizard venom</i></td><td><b>C0576722</b>
<i>animal sting</i></td><td><b>C0576724</b>
<i>plant sting</i></td></tr>
<tr><td>Example name</td><td><b>tick-borne fever</b></td><td><b>poisoning caused by gila monster venom</b></td><td><b>poisoning by bombus</b></td><td><b>nettle sting</b></td></tr>
</table>

Tab. 4.1.: Examples of how names from the SNOMED-CT ontology can be grouped into larger classes using parent concepts in the ontological graph. This allows us to investigate higher-level semantic relations, such as grouping *poisoning by bombus* and *nettle sting* under the concept of *sting of skin*, or e.g. grouping them together with *tick-borne fever* under *puncture wound of skin*.

4.1 Introduction

Recent work on representation learning for biomedical names has mainly involved the training of neural encoder architectures such as LSTMs (Kartsaklis et al., 2018) or Transformers (Kalyan & Sangeetha, 2020; Sung et al., 2020) to finetune name representations for biomedical normalization tasks. Such representations are often tailored towards normalization tasks (e.g. linking names to corresponding concept identifiers), without providing explicit guarantees about their transferability to other use contexts and applications. As a solution for this issue, the Biomedical Name Encoder (BNE) model (Phan et al., 2019) has been proposed as a comprehensive framework for robust and transferable representations.

According to this framework, the robustness of biomedical name representations is characterized along three dimensions. Firstly, semantic similarity between names should be reflected by their closeness in the embedding space. Secondly, the variety of textual contexts in which a name appears should be somehow represented in the encoding. Lastly, a name embedding should be sufficiently close to a pretrained prototypical representation of its conceptual meaning, e.g. a representation of its corresponding concept identifier from a biomedical ontology.

Such a multi-task model can be effectively trained using synonym sets extracted from ontologies such as the UMLS or SNOMED-CT. However, these synonym sets typically reflect only fine-grained distinctions between the lowest-level concepts from ontologies. If robust name representations should truly reflect semantic similarity in general, then the assumption is being made that training on such fine-grained synonym sets learns biomedical semantics in a bottom-up way, expecting names of lower-level concepts to spontaneously form relevant higher-level clusters.

However, such assumptions have not yet been empirically validated, for instance by showing that an encoder not only learns the differences between names such as *nettle sting* and *tick-borne fever*, but also simultaneously learns that they can be grouped together under the more general description *puncture wound of skin*. Moreover, research on representation learning and hierarchical classification for e.g. computer vision has indicated that neural models can leverage substantially different discriminative information for higher, more general levels of categorization than for more fine-grained lower levels (Hase et al., 2019). Such hierarchical differences can be exploited to generalize from higher to lower levels (Guo et al., 2017; Taherkhani et al., 2019), but they can also be difficult to integrate consistently into a single neural model (C. Wu et al., 2019).

In this paper, we investigate to what extent robust biomedical name representations can encode higher-level semantics while retaining relevant lower-level fine-grained information as well. To address this research question, we group synonym sets under increasingly coarse-grained semantic categories, using parent-child relations in the ontological graph. Table 4.1 gives an example of how names from the SNOMED-CT ontology can be grouped into larger classes. Such a hierarchy can be used to train and test a variety of semantic relations between names. For instance, a model might be able to encode that the names *poisoning by bombus* and *nettle sting* can be both described as *sting of skin*, but fail to represent their similarity to *poisoning caused by gila monster venom* as a *puncture wound of skin*. We believe that an evaluation of this nature is a crucial step towards achieving truly robust biomedical name representations, since it clearly requires more semantic inference from the encoder than merely resolving synonyms.

Apart from introducing this evaluation to the field of biomedical NLP, we also show that we can effectively adapt the BNE framework (Phan et al., 2019) to be trained using such large higher-level semantic classes. Most importantly, we replace the BiLSTM (Graves & Schmidhuber, 2005) encoder architecture of the BNE model with a lightweight Deep Averaging Network (DAN) (Iyyer et al., 2015). This allows us to easily scale to large amounts of training data, caused by the explosive amount of possible pairwise combinations between semantically similar names as classes grow larger.

Training on higher-level classes involves additional challenges such as handling imbalanced data distributions as well as implicit hierarchical and semantic differences among names grouped under the same class. Our aim is not to tailor the proposed approach to such artefacts. Rather, the main contribution of this paper is to show that our simple modification of the BNE model is generally applicable to a range of coarse-grained biomedical categorizations, without any finetuning apart from the size of the DAN encoder. As of such, it can be used as a low-cost but effective benchmark for future models that are more specialized.

Our experimental results for hierarchical SNOMED-CT data show that our DAN model improves semantic similarity ranking both in a bottom-up as well as top-down manner along various hierarchies. Interestingly, this observation holds even when we train on a few dozens of very broad categories. We also apply extrinsic evaluations to investigate the transferability of our DAN model. Firstly, we validate the robustness of higher-level representations on semantic relatedness benchmarks. Secondly, we perform unsupervised detection of SNOMED-CT hypernym disorder names which were not observed during training. For this task, our DAN model scores substantially better than the publicly released pretrained BNE model, which was trained on a large amount of fine-grained disorder concepts from SNOMED-CT using an elaborate BiLSTM architecture. These results provide tangible evidence that training name representations on large coarse-grained categories can help to encode exploitable higher-level semantics.

4.2 Related work

While context-dependent self-supervised representations usually outperform other text representations on a variety of BioNLP problems, such as semantic similarity and question answering, there is no single embedding model for biomedical and clinical texts that is consistently superior and thus can serve as a generally suitable bio-encoder (Tawfik & Spruit, 2020b). To this date, the BNE model by Phan et al. (2019) is the most prominent attempt at developing a supervised resource for encoding biomedical names. It uses a multi-task training regime in which it combines objectives from different aspects of deep representation learning, such as a contrastive loss (Le-Khac et al., 2020), conceptual grounding (see e.g. (Kartsaklis et al., 2018)), and explicit regularization of the learned representations (e.g. used by Vulić and Mrkšić (2018)). Our modifications to the original BNE model are informed by such literature.

Our application of a Deep Averaging Network (DAN) (Iyyer et al., 2015) is inspired by a recent subfield of NLP research which has emphasized the effectiveness of random encoders (Wieting & Kiela, 2019) and simple pooling mechanisms of word embeddings. The fastText encoder which we use as a baseline and as input for the DAN is an example of a Simple Word-Embedding-based Model (SWEM) with average pooling (Shen et al., 2018).

4.3 Encoding model

4.3.1 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*:

$$\begin{aligned} u_n &= \frac{1}{|N_t|} \sum_{t \in N_t} u_t \\ f(n) &= enc(u_n) \end{aligned} \tag{4.1}$$

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.

4.3.2 Training objectives

Our proposed approach is a simple modification of the multi-task training regime of the BNE model. We use cosine distance as distance function d for all three training objectives.

4.3.2.1 Semantic similarity

The *semantic similarity* objective is a generalization from the synonym similarity objective of the BNE model to any level of relevant semantic similarity. To enforce embedding similarity between names that are semantically related, we use a siamese triplet loss (Chechik et al., 2010). This loss forces the encoding of a biomedical name $f(n)$ to be closer to the encoding of a semantically similar name $f(n_{pos})$ than that of an encoded negative sample name $f(n_{neg})$, within a specified (possibly tuned) margin:

$$\begin{aligned} pos &= d(f(n), f(n_{pos})) \\ neg &= d(f(n), f(n_{neg})) \\ L_{sem} &= max(pos - neg + margin, 0) \end{aligned} \tag{4.2}$$

To select negative names during training we apply distance-weighted negative sampling (C.-Y. Wu et al., 2017) over all training names, since this has been proven more effective than hard or random negative sampling.

4.3.2.2 Contextual meaningfulness

The *contextual meaningfulness* objective forces the encoding of a biomedical name to be similar to its local contexts. The summary of these local contexts is approximated by taking the pretrained embedding representation u_n of the name:

$$L_{cont} = d(f(n), u_n) \tag{4.3}$$

This constraint implies that the dimensionality of the encoder output should be the same as that of the input. However, if the input dimensionality 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 & Kiela, 2019).

4.3.2.3 Conceptual grounding

The *conceptual grounding* objective is a modification of the conceptual meaningfulness objective of the BNE model. The conceptual meaningfulness objective forces the encoding of a biomedical name to be similar to a prototypical representation of its concept. This concept representation is approximated by averaging the pretrained embedding representations of all the names belonging to the concept:

$$u_p = \frac{1}{|C_n|} \sum_{n \in C_n} u_n \tag{4.4}$$

While converging to this pretrained target is feasible for small synonym sets, such convergence is unnecessary and overfitting for larger classes of names with graded differences in semantic similarity among the class members. To retain the robustness of the encodings, we only want to pull the names in the direction of their pretrained concepts, rather than minimizing their distance entirely. To this end, we simply take the average of the pretrained name representation and the pretrained concept representation:

$$\begin{aligned} v_{ground} &= \frac{u_p + u_n}{2} \\ L_{ground} &= d(f(n), v_{ground}) \end{aligned} \tag{4.5}$$

4.3.2.4 Multi-task setup

Our multi-task setup sums the losses of the 3 training objectives:

$$L = \alpha L_{sem} + \beta L_{cont} + \gamma L_{ground} \tag{4.6}$$

where α, β, and γ are possible weights for the individual losses. Since the 3 losses all directly reflect cosine distances, they are similarly scaled and don't require weighting to work properly. In our experiments, $\alpha = \beta = \gamma = 1$ showed the most robust performance along all settings.

4.4 Data and task setup

4.4.1 Extracting hierarchical data

Following previous research (Camacho-Collados et al., 2018; Kotitsas et al., 2019), we use IS-A relations between concepts from the SNOMED-CT[1] ontology as biomedical hypo-hypernymy relations. For direct comparison with the publicly released BNE embeddings, which were trained on all disorder concepts of SNOMED-CT, we use the 2018AB release of the UMLS[2] to extract only those SNOMED-CT concepts which are included in the semantic group of disorders[3], and extract their reference terms as disorder names. While the resulting directed graph should be acyclic, there are many inconsistencies, which we resolve by removing all cyclic edges, similar to the naive approach used by Mougin and Bodenreider (2005).

For our experiments, we select 3 different (yet slightly overlapping) subgraphs of IS-A relations by sampling 3 high-level concepts which have around 10K child concepts in our cleaned graph. We extract consistent taxonomies from these subgraphs by removing relations which form shortcuts between otherwise non-consecutive levels of the taxonomy, and by leaving out dead-end concepts which don't have a path to the required level of specification down the taxonomy. Child concepts can have mutually inclusive relations to multiple higher-level concepts on the same level of categorization.

4.4.2 Data setup

For each subgraph, we select 4 consecutive levels of parent concepts (level 1 is highest, level 4 is lowest). The concepts on these 4 levels are used as class labels for the names from all concepts below level 4. In other words, names belonging to the parent concepts themselves are not used

[1] https://www.snomed.org
[2] https://uts.nlm.nih.gov/home.html
[3] https://metamap.nlm.nih.gov/SemanticTypesAndGroups.shtml

C1290864	min	max	mean	stdev
Level 1	1	10203	1015	2053
Level 2	1	10203	291	1101
Level 3	1	3840	118	411
Level 4	1	2607	48	195

Tab. 4.2.: Descriptive statistics about the number of names per class for the different levels sampled from the subgraph with parent concept C1290864 (*disorder of abdomen*). These statistics show that lower levels have less extreme imbalances between classes.

during training: the parent concepts are only used as reference to cluster the names from the lower levels. Table 4.1 visualizes an example of this process.

This method of aggregating names can lead to very imbalanced classes. Table 4.2 shows how large this imbalance can get as we go up the hierarchy. While the training regime of our proposed model should be robust against such data artefacts, we want to take a representative test sample across all classes to empirically validate our approach. Therefore, for multiple iterations, we sample one held-out test name for each class on level 4. This test name is then also used for levels 1-3. Afterwards, we carry out the same procedure to sample validation data for calculating the stopping criterion during training. Table 4.3 shows the distributions of concepts and names used during training, validation, and testing.

4.4.3 Task setup

We perform 2 tasks on the held-out SNOMED-CT test data to validate our approach. Evidently, we always evaluate on individual levels of categorization. As intrinsic evaluation, we evaluate trained encoders on semantic similarity ranking. We also include the task of unsupervised hypernym detection as extrinsic evaluation. As we don't use the names of higher-level

	C1290864 *disorder of abdomen*	**C0560169** *osteoarthropathy*	**C0263661** *dermatological finding*
Level 1	27	30	35
Level 2	98	86	80
Level 3	248	236	231
Level 4	610	536	602
Lower-level names	24737 / 1557 / 763	20574 / 1335 / 649	25659 / 1567 / 814

Tab. 4.3.: An overview of the distribution of higher-level classes for the 3 subgraphs used in our experiments. The lower-level names are divided into train / test / validation.

concepts during training, we can exploit them as previously unobserved hypernymic data to show how much higher-level semantics are being modeled by encoders. If the encoder has learned to represent biomedical semantics more effectively, then the name embedding space can reflect that by being more suited for unsupervised detection of hypernyms.

Table 4.1 gives examples of hypernym names on all 4 levels. Successful hypernym detection for this data implies e.g. that we rank the previously unobserved hypernym *bite wound* over another previously unobserved hypernym *sting of skin* for the name *tick-borne fever*. This task clearly requires more semantic inference than merely resolving synonyms. In this case, the encoder has to represent that ticks are insects that bite instead of sting.

4.4.3.1 Semantic similarity ranking

We evaluate encoders on the ability to reflect semantic similarity between names by their cosine similarity. Given a mention *m* of a biomedical name which belongs to the higher-level class *c*, we have to rank the set of all training names *S* which includes $C_n \subset S$, a set of training names which belong to the same class *c* as the test mention. 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 $f(n)$. We then calculate the Mean Average Precision (mAP) over all test mentions for retrieving training names from the same higher-level class.

4.4.3.2 Unsupervised hypernym detection

Given a test mention *m* of a biomedical name which belongs to the higher-level class *c*, we have to rank the set of all hypernym names *H* belonging to a specific level of categorization. This set includes $C_h \subset H$, the set of hypernym names which belong to the same class *c* as the test mention. To rank the biomedical names according to their similarity to the mention, we first encode both the mention *m* as well as every hypernym name $h \in H$, and then rank every hypernym name h using the cosine similarity between the encoded mention $f(m)$ and the encoded hypernym $f(h)$. We then calculate the Mean Reciprocal Rank (MRR) over all test mentions for retrieving hypernym names from the same higher-level class.

4.5 Experiments and results

4.5.1 Reference model and baselines

We compare our DAN model against the the publicly released **pretrained 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 also include 2 baselines: our

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

300-dimensional **fastText** name embeddings (defined in Equation 4.1 in Section 4.3.1), and averaged 728-dimensional context-specific token activations extracted from the publicly released **BioBERT** model (Lee et al., 2019).

4.5.2 Training and implementation details

The DAN model is 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 4.3.1). All encoders for which we report results are finetuned to one hidden layer, which has 76,800 dimensions. Adam optimization (Kingma & 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 & Daelemans, 2012) and then lowercased. We use a triplet margin of 0.1 for the siamese triplet loss L_{sem} defined in Equation 4.2.

To train the model, we iterate over all names in the training data and apply the 3 training objectives for each name in a batch. To avoid overfitting on the largest classes, we always sample one siamese triplet per name, using random sampling for the positive name and distance-weighted sampling for the negative name. As stopping criterion we use the mAP of semantic similarity ranking (as defined in Section 4.4.3) for held-out validation names: we stop training once this score hasn't improved anymore over 10 epochs. This relaxed stopping criterion allows the model to optimize the subsampled siamese triplet loss in a balanced stochastic way over many epochs without quitting too early.

4.5.3 Results and discussion

4.5.3.1 Semantic similarity ranking

Table 4.4 shows the test performance for semantic similarity ranking. First and foremost, the robustness of the Level 1 DAN models is consistently great for all 3 subgraphs. For instance, in the case of the subgraph C1290864 (*disorder of abdomen*), the DAN is trained on only 27 large classes but outperforms the fastText baseline for the 610 classes on Level 4. Secondly, all DAN models generalize both bottom-up and top-down along the hierarchical levels to the extent that they consistently outperform the fastText baseline by a substantial margin.

Thirdly, the slight superiority of BioBERT over fastText for this task is most pronounced for the lowest levels. As we go up in the hierarchy, the difference grows smaller, which leads us to believe that the improvements are not so much of a semantic nature. Interestingly, the pretrained BNE model is competitive with our DAN models for the lower levels, which are still more coarse-grained than the fine-grained distinctions on which the BNE was trained. However, such a bottom-up effect is lacking for the highest levels of categorization. These observations reinforce the notion that both the size (the BNE was trained on 156K disorder names, our models on 20-25K) and the granularity of the data matter for deep representation learning.

4.5.3.2 Unsupervised hypernym detection

Table 4.5 shows the test performance for unsupervised hypernym detection. These results clearly show trends which are similar to the semantic similarity ranking. Most remarkably, the bottom-up and bottom-down effects are almost as consistent here: the highest-level DAN still outperforms the baselines for the lowest levels and vice versa. One major difference with the

	C1290864				C0560169				C0263661			
	1	2	3	4	1	2	3	4	1	2	3	4
DAN level 1	**0.57**	0.50	0.39	0.43	**0.70**	0.44	0.36	0.37	**0.64**	0.55	0.36	0.36
DAN level 2	0.49	**0.58**	0.46	0.48	0.55	**0.58**	0.44	0.44	0.58	**0.59**	0.40	0.39
DAN level 3	0.43	0.51	**0.56**	0.54	0.51	0.51	**0.52**	0.54	0.52	0.52	**0.51**	0.48
DAN level 4	0.38	0.43	0.47	**0.60**	0.45	0.45	0.48	0.58	0.45	0.44	0.41	**0.54**
fastText	0.26	0.27	0.25	0.33	0.36	0.29	0.28	0.32	0.33	0.30	0.24	0.30
BioBERT	0.27	0.29	0.29	0.39	0.38	0.32	0.31	0.37	0.36	0.33	0.27	0.35
BNE	0.35	0.41	0.42	0.57	0.43	0.41	0.45	**0.59**	0.44	0.44	0.39	0.51

Tab. 4.4.: Test performance of semantic similarity ranking per level, as measured by mAP. The highest score per level of each subgraph is denoted in bold; the second highest score is underlined.

	C1290864				C0560169				C0263661			
	1	2	3	4	1	2	3	4	1	2	3	4
DAN level 1	**0.60**	0.58	0.59	0.68	**0.48**	0.54	0.52	0.63	**0.52**	0.57	0.55	0.62
DAN level 2	0.52	**0.59**	0.62	0.70	0.45	**0.58**	0.56	0.67	0.50	**0.60**	0.57	0.63
DAN level 3	0.55	0.57	**0.66**	0.73	0.41	0.54	**0.58**	0.70	0.48	0.57	**0.62**	0.67
DAN level 4	0.53	0.52	0.63	**0.74**	0.39	0.53	**0.58**	**0.74**	0.46	0.54	0.59	**0.71**
fastText	0.46	0.44	0.53	0.65	0.34	0.47	0.49	0.63	0.38	0.45	0.50	0.59
BioBERT	0.41	0.41	0.50	0.62	0.28	0.41	0.46	0.58	0.39	0.47	0.48	0.59
BNE	0.43	0.50	0.60	0.71	0.42	0.48	0.57	0.70	0.49	0.49	0.54	0.68

Tab. 4.5.: Test performance for unsupervised hypernym detection per level, as measured by MRR. The highest score per level of each subgraph is denoted in bold; the second highest score is underlined.

results for semantic similarity ranking is the relatively worse performance from BioBERT here compared to fastText. This is in line with the findings by Yu et al. (2020), who report that BERT does not yield considerable improvement for hypernymy detection in their experiments. It also puts into perspective to what extent we can expect higher-level semantics to be encoded solely through self-supervised methods.

Table 4.6 gives an example of hypernym rankings for the test mention *poisoning caused by mexican beaded lizard bite*. By clustering similar names together with other bite wounds during training, the DAN model has learned to recognize the test mention as a bite wound. The BNE has failed to do so.

Subgraph	**C0560169**	
Level	3	
Test mention	**poisoning caused by mexican beaded lizard bite**	
Matching hypernyms	bite wound / bite wound (disorder)	
	DAN Level 1	**BNE**
Top 5 ranking	*bite wound (disorder)*	infestation caused by fly larvae (disorder)
	bite wound	fly larva infestation
	open traumatic dislocation of hip, unspecified	infestation caused by fly larvae
	open traumatic dislocation of hip, unspecified (disorder)	infestation by fly larvae (disorder)
	open dislocation of phalanx of foot (disorder)	infestation by fly larvae

Tab. 4.6.: A comparison between our DAN encoder and the BNE reference model for unsupervised hypernym ranking of the Level 3 test mention *poisoning caused by mexican beaded lizard bite*. The DAN model generalizes from the training data to associate the test mention correctly with bite wounds. In the training process, it seems to have clustered bite wounds together with open dislocations. The BNE model apparently associates lizards with infestations by fly larvae, but fails to recognize that there is a bite wound mentioned in the test mention.

.

The effectiveness of our unsupervised method using only cosine similarity contrasts with earlier approaches which explicitly require more than cosine similarity to properly work. For example, Vulić and Mrkšić (2018) use vector norms to encode hierarchical hypernymic relations, while other research into hypernymy even requires other geometric spaces than Euclidean space, such as hyperbolic space (Dhingra et al., 2018). Our results can indicate that cosine similarity in Euclidean space still shows potential for encoding these hierarchical relations given the right training objectives.

4.5.4 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 fine-grained concepts. UMNSRS (Pakhomov et al., 2016) contains only single-word pairs, which also stem from different fine-grained concepts. This benchmark makes a distinction between *similarity* and *relatedness*.

The correlations in Table 4.7 show that the majority of our trained encoders remain robust out-of-the box, with a large portion of them outperforming the fastText baseline which they use as input. The highest-level model trained on the C0560169 subgraph (*dermatological finding*) is even competitive with the pretrained BNE, having been trained on only 30 classes. All in all, these results confirm that our proposed model is relatively robust against variable granularity of clustering, and is not overly tailored to the data artefacts of one specific subgraph.

4.5.5 Discussion

While our empirical results are certainly encouraging, the true robustness of our proposed framework remains an open question. Whereas our proposed DAN model remains robust over entire hierarchies for semantic similarity ranking and unsupervised hypernym detection, its relative performance for the semantic relatedness benchmarks is not entirely predictable from those tasks. One the one hand, this likely has to do with the modest sizes of the benchmarks, for which small to very small margins in performance are not very reliable or indicative.

On the other hand, we also have to consider that our finetuned DAN only contains a single, yet very wide, hidden layer. This implies that the encoder network relies more on what can considered to be an elaborate weighted average than a deep multi-layer transformation of the input. While this is not very surprising in the context of transferable representations (and emphasizes the effectiveness of exploiting word embeddings according to their full potential in simple ways, as suggested by Wieting and Kiela (2019)), it still raises the question whether there are straightforward regu-

	MayoSRS (rel)	UMNSRS (rel)	UMNSRS (sim)
fastText	0.44	0.47	0.48
Level 1 C0560169	0.42	<u>0.55</u>	<u>0.54</u>
Level 2 C0560169	0.47	0.51	0.50
Level 3 C0560169	0.50	0.51	0.50
Level 4 C0560169	0.50	0.51	0.50
Level 1 C1290864	0.52	0.42	0.46
Level 2 C1290864	0.55	0.46	0.40
Level 3 C1290864	0.53	0.46	0.50
Level 4 C1290864	<u>0.56</u>	0.45	0.50
Level 1 C0263661	0.46	0.49	0.51
Level 1 C0263661	0.51	0.47	0.50
Level 3 C0263661	0.55	0.50	0.53
Level 4 C0263661	0.52	0.50	0.50
Phan et al. (2019)	**0.63**	**0.58**	**0.61**

Tab. 4.7.: Spearman's rank correlation coefficient between cosine similarity 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.

larization alternatives to the contextual meaningfulness objective which can allow for deep transformations with the DAN.

4.6 Conclusion and future work

In this paper, we have introduced the challenge of integrating higher-level semantics into robust biomedical name representations. We provide a framework to both train and evaluate encoders for this task. Moreover, we have proposed a modification of the Biomedical Name Encoder model which is directly applicable to a variety of coarse-grained categorizations. This modification replaces more complex neural architectures with a lightweight Deep Averaging Network encoder, which is easily scalable to the large amounts of required training data, while remaining sufficiently robust. The only important hyperparameter to tune for this encoder is the size of the Feedforward Neural Network.

Experiments indicate that our proposed framework can even be effective using only around 30 coarse-grained classes. This opens up possibilities for applying our framework to data beyond carefully curated ontologies, for instance in self-supervised or semi-supervised settings. Future work will try to understand and define the limits of applying our framework to such settings.

5 Scalable Few-Shot Learning of Robust Biomedical Name Representations

Recent research on robust representations of biomedical names has focused on modeling large amounts of fine-grained conceptual distinctions using complex neural encoders. In this paper, we explore the opposite paradigm: training a simple encoder architecture using only small sets of names sampled from high-level biomedical concepts. Our encoder post-processes pretrained representations of biomedical names, and is effective for various types of input representations, both domain-specific or unsupervised. We validate our proposed few-shot learning approach on multiple biomedical relatedness benchmarks, and show that it allows for continual learning, where we accumulate information from various conceptual hierarchies to consistently improve encoder performance. Given these findings, we propose our approach as a low-cost alternative for exploring the impact of conceptual distinctions on robust biomedical name representations.

5.1 Introduction

Recent research in biomedical NLP has focused on learning robust representations of biomedical names. To achieve robustness, an encoder should represent the semantic similarity and relatedness between different names (e.g. by their closeness in the embedding space), while its embeddings should also remain as transferable and generally applicable as self-supervised pretrained representations.

Prior research into robust representations has shown three distinct tendencies. Firstly, research typically focuses on encoders with complex neural architectures and a large amount of parameters. As compensation for this complexity, such models can be heavily regularized during training, e.g. by tying the output of a nested LSTM to a pooled embedding of its input representations (Phan et al., 2019), or by integrating a finetuned BERT model with sparse lexical representations (Sung et al., 2020).

Secondly, encoders are typically trained on fine-grained concepts from biomedical ontologies such as the UMLS, i.e., concepts with no child nodes in the ontological directed graph. Small synonym sets of such fine-grained concepts are readily available as training data, and often serve as evaluation data for normalization tasks to which trained encoders can be applied.

Lastly, as a result of using fine-grained concepts, vast amounts of biomedical names are needed to model the large collection of fine-grained distinctions present in ontologies. For instance, Phan et al. (2019) train their encoder on 156K disorder names. These three tendencies share an underlying assumption: complex neural encoder architectures can learn biomedical semantics by generalizing in a bottom-up fashion from large amounts of fine-grained semantic distinctions, if provided with sufficient quantities of training data.

However, it is not self-evident that such an approach is the most effective way to achieve general-purpose biomedical name representations. For instance, it does not directly address what conceptual distinctions are actually *relevant* to improve representations for downstream NLP applications. Finding and exploiting relevant distinctions can be an empirical question, and as such require low-cost exploration of various conceptual hierarchies. Such a heuristic search is expensive in the current paradigm.

Chapter V: Mental and behavioural disorders	
F34	**F63**
Persistent mood disorders	Habit and impulse disorders
F34.0	F63.0
Cyclothymia	*Pathological gambling*
F34.1	F63.1
Dysthymia	*Pyromania*

Tab. 5.1.: Example of how reference names are grouped together within the ICD-10 hierarchy of disorders.

In this paper, we explore a scalable few-shot learning approach for robust biomedical name representations which is orthogonal to this paradigm. We investigate to what extent we can fit a simple encoder architecture using only a small selection of data, with a limited amount of concepts containing only a few samples each (i.e., few-shot learning). To this end, we don't use fine-grained concepts for training, but more general higher-level concepts which span a large range of fine-grained concepts. Table 5.1 gives an example of such a larger grouping of biomedical names.

This paper offers two main contributions. Firstly, our proposed approach offers an alternative for training biomedical name encoders with much lower computational cost, both for training and inference at test time. It is applicable to large-scale hierarchies containing at least ten thousands of names and is equally effective for different types of pretrained representations when tested on various biomedical relatedness benchmarks. Secondly, we show that this approach allows for low-cost continual learning from multiple concept hierarchies, and as such can help with the accumulation of relevant domain-specific information for downstream biomedical NLP tasks.

5.2 Approach

Our approach is similar to supervised post-processing techniques of word embeddings such as retrofitting and counterfitting (Faruqui et al., 2015; Mrkšić et al., 2016), but instead post-processes pretrained representations of biomedical names.

5.2.1 Encoder architecture

Our encoder architecture is a feedforward neural network with Rectified Linear Unit (ReLU) as non-linear activation function. This neural network transforms a pretrained representation of a biomedical name, after which this transformation is averaged with the pretrained representation:

$$f(n) = \frac{enc(u_n) + u_n}{2} \tag{5.1}$$

where $f(n)$ is the output representation for a biomedical name, u_n is its pretrained input representation, and enc is the feedforward neural network which transforms the input representation. The averaging step ensures that the encoder architecture learns to update the pretrained input representation rather than create an entirely new representation. This makes our model more robust against overfitting in few-shot learning settings.

5.2.2 Training objectives

Our training objectives are based on the state-of-the-art BNE model by Phan et al. (2019) and the DAN model by Fivez et al. (2021b), which generalizes the BNE model to any hierarchical level of biomedical concepts. Our framework requires a set of concepts C, where each concept $c \in C$ contains a set of concept names C_n. The set of biomedical names N contains the union of all those sets of concept names. We propose a simple multi-task training regime which applies two training objectives to each biomedical name $n \in N$. We use cosine distance as distance function d for both objectives.

5.2.2.1 Semantic similarity

We enforce embedding similarity between names that are from the same concept by using a siamese triplet loss (Chechik et al., 2010). This loss forces the encoding of a biomedical name $f(n)$ to be closer to the encoding of a semantically similar name $f(n_{pos})$ than that of an encoded negative sample name $f(n_{neg})$, within a specified (possibly tuned) margin:

$$\begin{aligned} pos &= d(f(n), f(n_{pos})) \\ neg &= d(f(n), f(n_{neg})) \\ L_{sem} &= max(pos - neg + margin, 0) \end{aligned} \tag{5.2}$$

To select negative names during training we apply distance-weighted negative sampling (C.-Y. Wu et al., 2017) over all training names, since this has been proven more effective than hard or random negative sampling.

5.2.2.2 Conceptually grounded regularization

To prevent the model from overfitting on the semantic similarity objective, we regularize it by grounding the output representations to a stable and meaningful target. Simple approximations of prototypical concept representations can already be very effective as targets (Fivez et al., 2021a). Following the model by Fivez et al. (2021b), we use a grounding target which is applicable to any level of categorization, from fine-grained concept distinctions to higher-level groupings of names. This target is a compromise between the *contextual meaningfulness* and *conceptual meaningfulness* objectives of the BNE model. Rather than constraining a name encoding either to its pretrained name representation or to a pretrained representation of its concept, we minimize the distance to the average of both pretrained representations:

$$\begin{aligned} u_c &= \frac{1}{|C_n|} \sum_{n \in C_n} u_n \\ u_{ground} &= \frac{u_c + u_n}{2} \\ L_{ground} &= d(f(n), u_{ground}) \end{aligned} \tag{5.3}$$

where the concept representation u_c is approximated by averaging each pretrained embedding representation u_n from the set of names C_n belonging to the concept.

This constraint implies that the dimensionality of the encoder output should be the same as that of the input. However, if the input dimensionality 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 & Kiela, 2019).

5.2.2.3 Multi-task loss

Our multi-task loss sums the losses of the 2 training objectives:

	min	max	mean	stdev
ICD-10	247	40,519	3,414	8,693
SNOMED-CT	397	19,114	3,532	4,094
(+ ambiguous	1,108	23,915	4,990	5,134)

Tab. 5.2.: Descriptive statistics about the number of names per concept for our training data.

$$L = \alpha L_{sem} + \beta L_{ground} \tag{5.4}$$

where α and β are possible weights for the individual losses. Since both losses directly reflect cosine distances, they are similarly scaled and don't require weighting to work properly. In our experiments, $\alpha = \beta = 1$ showed the most robust performance along all settings.

5.2.3 Training data

We extract sets of high-level concepts and their constituent names from 2 large-scale hierarchies of disorder concepts, ICD-10 and SNOMED-CT. Table 5.2 gives an overview of our data distributions.

5.2.3.1 ICD-10

We use the 2018 version of the ICD-10 coding system.[1] We select the 21 chapters as concept labels, and assign the reference name of each code in a chapter to its concept label. Table 5.1 gives an example of how such a grouping includes diverse semantic relations.

[1] https://www.cdc.gov/nchs/icd

5.2.3.2 SNOMED-CT

We use the 2018AB release of the UMLS ontology[2] to extract a directed ontological graph of SNOMED-CT concepts. We then select the first-degree child nodes of concept *C0012634*, which is the parent concept for all disorders. We then remove those children which are direct parents to other selected children, since they are redundant for our purpose.

This leaves us with 87 concepts, to which we assign the reference terms of all their child concepts in the ontological graph as biomedical names. To make this setup directly comparable to our ICD-10 setup, we select the 21 largest concepts. Finally, we leave out ambiguous names which belong to multiple concepts. Table 5.2 shows the impact on the data distribution.

5.3 Experiments and discussion

5.3.1 Pretrained representations

We experiment with 3 pretrained name representations. As a first baseline, we use 300-dimensional **fastText** (Bojanowski et al., 2017) word embeddings which we train on 76M sentences of preprocessed MEDLINE articles released by Hakala et al. (2016). We use average pooling (Shen et al., 2018) to extract a 300-dimensional name representation. As a second baseline, we average the 728-dimensional context-specific token activations of a name extracted from the publicly released **BioBERT** model (Lee et al., 2019).

As state-of-the-art reference, we extract 200-dimensional name representations using the publicly released pretrained **BNE** model with skipgram

[2] https://uts.nlm.nih.gov/home.html

word embeddings, BNE + SG$_\mathrm{w}$,[3] which was trained on approximately 16K synonym sets of disease concepts in the UMLS, containing 156K disease names.

5.3.2 Training details

We randomly sample a small fixed amount of names from each concept in our training data as actual few-shot training names. We then randomly sample the same amount of names as validation data to calculate the multi-task loss as stopping criterion. This criterion is also used to finetune the size of the encoder network. Using only 1 hidden layer proved best in all settings, which leaves only the dimensionality of this layer to be tuned.

Our encoder network is implemented in PyTorch (Paszke et al., 2019). Adam optimization (Kingma & Ba, 2015) is performed on a batch size of 16, using a learning rate of 0.001 and a dropout rate of 0.5. Input strings are first tokenized using the Pattern tokenizer (Smedt & Daelemans, 2012) and then lowercased. We use a triplet margin of 0.1 for the siamese triplet loss L_{sem} defined in Equation 5.2.

5.3.3 Results

We evaluate our trained encoders on 3 biomedical benchmarks of semantic relatedness and similarity, which allow to compare similarity scores between name embeddings with human judgments of relatedness. MayoSRS (Pakhomov et al., 2011) contains multi-word name pairs of related but different fine-grained concepts. UMNSRS (Pakhomov et al., 2016) contains only single-word pairs, and makes a distinction between *relatedness* and *similarity*, which is a more narrow form of relatedness. Finally, EHR-RelB (Schulz et al., 2020) is much larger than the other benchmarks, and con-

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

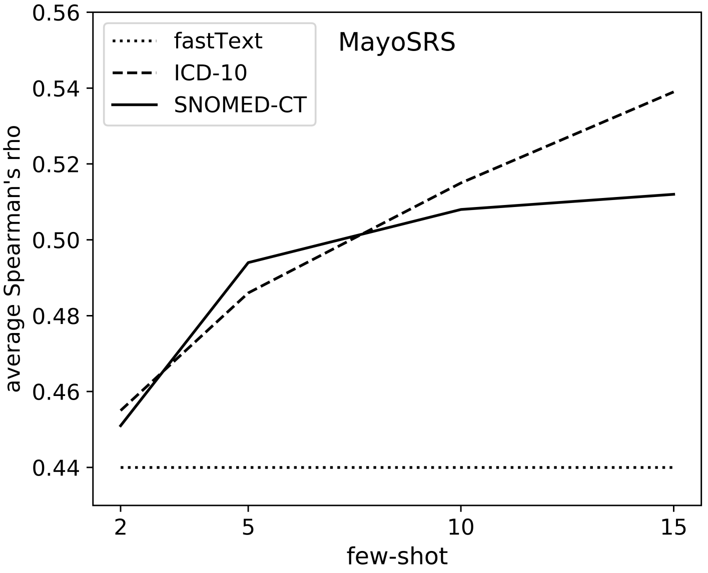

Fig. 5.1.: Few-shot performance for fastText encoders on MayoSRS, averaged over 5 random samples.

tains multi-word concept pairs which are chosen based on co-occurrence in electronic health records. This ensures that the evaluated concept pairs are actually relevant in function of downstream applications such as information retrieval.

We average all test results over 5 different random training samples. We use cosine similarity as similarity score for all baseline representations and trained encoders. Figure 5.1 shows the impact of the amount of few-shot training names on performance when using fastText representations. Our model already substantially improves over the baseline with only 5 names per concept (105 in total), and maintains consistent improvement up to 15 few-shot names. This confirms that our approach is well-suited to anticipate expected improvements from training on large-scale hierarchies.

Table 5.3 shows the results on all benchmarks for 15-shot learning. All encoders were tuned to 9,600 hidden dimensions. We include two state-of-the-art biomedical name encoders in our comparison. Firstly, BioSyn (Sung

	EHR-RelB	**MayoSRS**	**UMNSRS**	
	(rel)	(rel)	(rel)	(sim)
BioSyn	0.45	0.50	0.40	0.42
Fivez et al. (2021a)		**0.67**	**0.56**	0.56
fastText	0.39	0.44	0.47	0.48
BioBERT	0.34	0.23	0.18	0.26
BNE	0.47	0.63	0.54	0.58
SNOMED				
fastText	0.43	0.51	0.46	0.51
BioBERT	0.40	0.31	0.32	0.38
BNE	0.53	0.63	0.55	**0.60**
ICD-10				
fastText	0.43	0.55	0.52	0.56
BioBERT	0.35	0.34	0.32	0.38
BNE	0.51	0.65	**0.56**	**0.60**
S → I				
fastText	0.44	0.55	0.46	0.52
BioBERT	0.39	0.33	0.35	0.42
BNE	**0.54**	**0.67**	0.52	0.58
I → S				
fastText	0.45	0.54	0.46	0.51
BioBERT	0.39	0.33	0.37	0.42
BNE	**0.54**	**0.67**	0.53	0.58

Tab. 5.3.: Spearman's rank correlation coefficient between human judgments and similarity scores of name embeddings, reported on semantic similarity (sim) and relatedness (rel) benchmarks. The highest score is denoted in bold; the second highest is underlined.

et al., 2020) sums the weighted inner products of fine-tuned BioBERT representations and sparse TF-IDF representations into one similarity score between two names. The pre-trained model[4] for which we report results was trained on the NCBI disease benchmark (Doğan et al., 2014) for biomedical entity normalization. Secondly, we include the results of the conceptually grounded Deep Averaging Network by Fivez et al. (2021a), which was trained on SNOMED-CT synonym sets mapped into larger ICD-10 categories.

The results show various trends. Firstly, almost all trained encoders improve over their input baselines for all benchmarks, regardless of the type of input representation. Secondly, the performance increase is consistent for both ICD-10 and SNOMED-CT, even as their conceptual hierarchies are substantially different. Lastly, we also look at continual learning from SNOMED-CT to ICD-10 ($\mathbf{S} \rightarrow \mathbf{I}$) or vice versa ($\mathbf{I} \rightarrow \mathbf{S}$), where we use the output of the first model as input representations to train the second model. This approach leads to systematic improvements for all representation types, including the state-of-the-art BNE representations. In other words, we provide tangible empirical evidence that few-shot robust representations can allow for continual specialization in biomedical semantics.

To better understand how our few-shot learning approach can have a visible impact on various relatedness benchmarks, Table 5.4 gives an example of nearest neighbor names from the training set of SNOMED-CT names for the validation mention *urinary hesitancy*. While the pretrained BNE model makes various topical associations, our 15-shot model using the BNE representations as input has learned to cluster around the semantics of urinary tract disorders. As this already generalizes to validation mentions, we can expect the model to transfer this information to downstream applications wherever urinary tract disorders are relevant. This applies to

[4] https://github.com/dmis-lab/BioSyn

Parent concept	C0042075	
Parent concept name	*disorder of the urinary system*	
Validation mention	**urinary hesitancy**	
	15-shot BNE	**BNE**
Top 10 ranking	nebulous urine	nebulous urine
	calculus of lower urinary tract (disorder)	calculus of lower urinary tract (disorder)
	urinary obstruction due to nodular prostate (disorder)	urinary obstruction due to nodular prostate (disorder)
	double kidney and/or pelvis	double kidney and/or pelvis
	covered exstrophy of bladder (disorder)	genital oedema
	nephropathy caused by aminoglycoside (disorder)	perineal laceration during delivery , nos
	renal vein thrombosis	abdominal hernia
	benign tumour of urethra	covered exstrophy of bladder (disorder)
	injury of male urethra	heart :[weak] or [failure nos] (disorder)
	postprocedural bulbous urethral stricture	hourglass contraction of uterus

Tab. 5.4.: A comparison between the rankings of 315 SNOMED-CT training names for the validation mention *urinary hesitancy*. Non-matching names are underlined. While the pretrained BNE model makes various topical associations, our 15-shot model using the BNE representations as input has learned to cluster around the semantics of urinary tract disorders.

all 21 high-level topics which were simultaneously encoded for both the ICD-10 and SNOMED-CT ontologies.

5.4 Conclusion and future work

We have proposed a novel approach for scalable few-shot learning of robust biomedical name representations, which trains a simple encoder architecture using only small subsamples of names from higher-level concepts of large-scale hierarchies. Our model works for various pretrained input embeddings, including already specialized name representations, and can accumulate information over various hierarchies to systematically improve performance on biomedical relatedness benchmarks. Future work will investigate whether such improvements trickle down properly to downstream biomedical NLP tasks.

6

Conclusions and Future Work

This PhD thesis has investigated robust and scalable applications of pre-trained text representations for biomedical natural language processing. We have primarily focused on representations which are pre-trained using the masked language modelling objective. While such representations encode a variety of potentially relevant information for downstream NLP applications, successfully exploiting that information remains a fundamentally empirical question. Each of the chapters of this thesis has examined a particular biomedical NLP application and has offered a novel and empirically effective approach to leverage pre-trained representations.

In Chapter 2, we have shown that simple composition of word and character n-gram embeddings can represent textual contexts of clinical spelling errors sufficiently well to help estimate the semantic fit of spelling correction candidates. This contextual estimation is integrated with edit distance measures in a cosine similarity-based ranking model which is developed using only automatically generated supervised data. Simple compositions of context words have also been proven effective in other biomedical NLP applications, such as word sense disambiguation (Tulkens et al., 2016) and concept extraction from clinical text (Tulkens et al., 2019). Their application to spelling correction is most convenient for instances where various correction candidates are orthographically similar but semantically different. The approximations of simple embedding compositions can be used to detect where a specific misspelling maps to different corrections in different contexts, e.g. *iron deficiency due to ~~enemia~~* → *anemia* vs. *fluid injected with ~~enemia~~* → *enema*.

The remaining chapters of this PhD thesis have looked into various methods of training neural encoders for robust representation of biomedical names. Robustness requires that name representations should encode domain-specific knowledge, e.g. by reflecting semantic similarity between names through their closeness in the embedding space, while retaining the universal applicability and transferability of self-supervised pre-trained representations. Prior research on robust representations shares the underlying assumption that complex neural encoder architectures can learn biomedical semantics by generalising in a bottom-up fashion from large amounts of fine-grained semantic distinctions, if provided with sufficient quantities of training data. We have investigated empirically effective approaches which put into question this paradigm and offer robust and scalable alternatives.

In Chapter 3, we have introduced the use of a Deep Averaging Network (DAN) as biomedical name encoder, which is a feedforward neural network processing an unordered composition of the word embeddings in a name. Using such a minimalist encoder architecture, as opposed to using more elaborate architectures such as LSTMs or Transformers, allows for considering a range of training objectives which would otherwise be too computationally costly. In this chapter, we have exploited this tradeoff to enforce conceptual grounding constraints during training of biomedical name representations. Such grounding constraints tie the output of an encoder to specific pre-trained targets which constitute a globally coherent and meaningful embedding space. Inspired by previous work on prototypical networks (Snell et al., 2017), we explicitly control the joint behaviour of different names from the same biomedical concept to better match the target embedding space. This expansive training objective is effective even when we use simple approximations of prototypical concept representations. To confirm that such approximations are sufficiently robust, we have shown that linear transformations of pre-trained embeddings using Canonical Correlation Analysis (CCA) can already extrapolate to zero-shot learning scenarios such as synonym retrieval for previously unobserved concepts.

While Chapter 3 has focused on fine-grained concepts (i.e., concepts with no child nodes in an ontological directed graph), its results raise the question whether a DAN encoder can also be effective for higher-level conceptual distinctions. Chapter 4 not only confirms this, but shows that the DAN can generalise both bottom-up as well as top-down among various semantic hierarchies. In other words, the DAN can extract both high-level and low-level biomedical semantics from an unordered composition of word embeddings as well as simultaneously represent them in the same low-dimensional distributed vector. Moreover, the encoder is applicable out-of-the-box for improved unsupervised detection of hypernyms. This implies that explicitly modelling high-level distinctions is a useful complement to fine-grained approaches for encoding biomedical semantics, even when we scale down the complexity of the encoder architecture. All of these findings put into question whether more elaborate architectures such as LSTMs or Transformers should really be considered the default choice for biomedical name encoders in cases where robust semantic composition is more important than some details concerning e.g. word order.

Chapter 5 has made a compelling argument for this position. It provides a proof-of-concept that a feedforward neural network can be used for few-shot learning of biomedical name representations on top of various pre-trained representations, ranging from self-supervised to already domain-specialised. Most importantly, this approach allows for continual learning, where information from various conceptual hierarchies is accumulated to consistently improve encoder performance. While the approach offers various obvious benefits such as a substantial decrease in computational cost and greatly reduced demand for annotated data, it also raises the issue of what are considered *informative* distinctions in conceptualisations of biomedical semantics. Various low-level distinctions in biomedical ontologies have a practical rather than a semantic origin, e.g. for practices such as assigning billing codes to clinical notes. In contrast, the most high-level distinctions in ontologies such as ICD-10 and SNOMED-CT have a much higher informational value, since they reflect large-scale conceptualisations of biomedical relevance. In this sense, the default paradigm of training encoders on fine-grained distinctions only is somewhat counterintuitive.

6.1 Future work

6.1.1 Spelling correction for clinical free-text

Flor et al. (2019) have used our manually constructed test set of English clinical spelling errors described in Chapter 2, and obtained results which are competitive with our proposed approach using a general-domain minimally-supervised model which incorporates word embeddings and contextual features. S. Wu et al. (2019) have characterized our application of word and character n-gram embeddings for spelling correction as part of a larger trend towards deep learning baselines for the medical NLP community. Hladek et al. (2020) provide a comprehensive review of spelling correction models, which confirms the increasing prevalence of deep learning methods and contrasts our approach with other recent models.

The main challenge for future work in spelling correction remains the robust application of deep learning methods across languages different from English. While we have demonstrated in Chapter 2 that techniques which work well for English clinical text can be less suitable for Dutch, later work by Beeksma et al. (2018) has also confirmed difficulties for high-quality Dutch Wikipedia text, especially for error detection.

6.1.2 Deep Averaging Networks

We believe future work could investigate Deep Averaging Networks as general-purpose text encoders outside of our specific application to biomedical name representations. As we have demonstrated in Chapters 3-4, the decreased computational cost for training a DAN encoder can allow for exploration towards more intensive training objectives which would be considered computationally infeasible for LSTMs or Transformers.

6.1.3 Effective grounding

We have demonstrated in Chapter 3 that increasing the effectiveness of pre-existing techniques for conceptual grounding can still lead to substantial performance improvements. Future work could investigate this for other domains than the biomedical domain, or for other grounding targets than biomedical concepts.

6.1.4 Hierarchical encoding of hypernymy

While prior work has pointed out difficulties for encoding multiple related levels of hierarchy into one neural network and reflecting hypernymic relations through cosine similarity, our biomedical name encoder in Chapter 4 is able to do both. Future work could use our findings as a precedent to reinvestigate the limitations of Eucledian space for encoding hierarchical relations.

6.1.5 Few-shot representation learning

Whereas the field of deep representation learning is predisposed to complex encoder architectures and large amounts of training data, our results in

Chapter 5 confirm that there are use cases where few-shot approaches with simple encoders can provide more efficient approximations. Future work could investigate this more systematically, outside of biomedical NLP as well.

6.1.6 Impact of semantic representation on downstream biomedical NLP tasks

Our work on biomedical name representations in Chapters 3-5 has succeeded in improving the encoding of biomedical semantics from concept names. A crucial direction for future work consists of estimating for which downstream tasks such improvements can make a tangible impact.

Supplementary Materials

S

S.1 Conceptual Grounding Constraints for Truly Robust Biomedical Name Representations

S.1.1 Redundant metatags

In section 3.4.1.1, 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)

Bibliography

Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., & Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *International Conference for Learning Representations (ICLR)* (cit. on p. 6).

Beeksma, M., van Gompel, M., Kunneman, F., Onrust, L., Regnerus, B., Vinke, D., Brito, E., Bauckhage, C., & Sifa, R. (2018). Detecting and correcting spelling errors in high-quality Dutch Wikipedia text. *Computational Linguistics in the Netherlands Journal (CLIN)*, *8* (cit. on p. 100).

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, *5*, 135–146 (cit. on pp. 3, 7, 22, 46, 68, 90).

Broscheit, S. (2019). Investigating entity knowledge in BERT with simple neural end-to-end entity linking. *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, 677–685 (cit. on p. 5).

Camacho-Collados, J., Delli Bovi, C., Espinosa-Anke, L., Oramas, S., Pasini, T., Santus, E., Shwartz, V., Navigli, R., & Saggion, H. (2018). SemEval-2018 task 9: Hypernym discovery. *Proceedings of The 12th International Workshop on Semantic Evaluation*, 712–724 (cit. on p. 72).

Chechik, G., Sharma, V., Shalit, U., & Bengio, S. (2010). Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, *11*, 1109–1135 (cit. on pp. 47, 69, 87).

Chiu, B., Baker, S., Palmer, M., & Korhonen, A. (2019). Enhancing biomedical word embeddings by retrofitting to verb clusters. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 125–134 (cit. on p. 4).

Conneau, A., Kiela, D., Schwenk, H., Barrault, L., & Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 670–680 (cit. on p. 2).

Conneau, A., Kruszewski, G., Lample, G., Barrault, L., & Baroni, M. (2018). What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2126–2136 (cit. on p. 6).

Cornet, R., van Eldik, A., & Keizer, N. D. (2012). Inventory of tools for Dutch clinical language processing. *Proceedings of the 24th European Medical Informatics Conference* (cit. on p. 19).

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *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)*, 4171–4186 (cit. on p. 2).

Dhingra, B., Shallue, C., Norouzi, M., Dai, A., & Dahl, G. (2018). Embedding text in hyperbolic spaces. *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, 59–69 (cit. on pp. 14, 79).

Doğan, R. I., Leaman, R., & Lu, Z. (2014). NCBI disease corpus: A resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, *47*, 1–10 (cit. on pp. 50, 94).

D'Souza, J., & Ng, V. (2015). Sieve-based entity linking for the biomedical domain. *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)*, 297–302 (cit. on p. 42).

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., & Smith, N. A. (2015). Retrofitting word vectors to semantic lexicons. *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1606–1615 (cit. on pp. 5, 86).

Firth, J. R. (1957). A synopsis of linguistic theory 1930-1955. *Studies in Linguistic Analysis*, 1–32 (cit. on p. 2).

Fivez, P., Suster, S., & Daelemans, W. (2021a). Conceptual grounding constraints for truly robust biomedical name representations. *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2440–2450 (cit. on pp. 88, 93, 94).

Fivez, P., Suster, S., & Daelemans, W. (2021b). Integrating higher-level semantics into robust biomedical name representations. *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, 49–58 (cit. on pp. 86, 88).

Flor, M., Fried, M., & Rozovskaya, A. (2019). *A benchmark corpus of English misspellings and a minimally-supervised model for spelling correction*. Association for Computational Linguistics. (Cit. on p. 100).

Gladkova, A., & Drozd, A. (2016). Intrinsic evaluations of word embeddings: What can we do better? *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, 36–42 (cit. on p. 3).

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, *18*(5-6), 602–610 (cit. on pp. 54, 66).

Guo, Y., Liu, Y., Bakker, E. M., Guo, Y., & Lew, M. S. (2017). CNN-RNN: A large-scale hierarchical image classification framework. *Multimedia Tools and Applications*, *77*, 10251–10271 (cit. on p. 65).

Hakala, K., Kaewphan, S., Salakoski, T., & Ginter, F. (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*, 102–107 (cit. on pp. 4, 46, 50, 68, 90).

Hase, P., Chen, C., Li, O., & Rudin, C. (2019). Interpretable image recognition with hierarchical prototypes. *The Seventh AAAI Conference on Human Computation and Crowdsourcing (HCOMP-19)* (cit. on p. 65).

Hladek, D., Stas, J., & Pleva, M. (2020). Survey of automatic spelling correction. *Electronics*, *9*(10) (cit. on p. 100).

Iyyer, M., Manjunatha, V., Boyd-Graber, J., & Daumé III, H. (2015). Deep unordered composition rivals syntactic methods for text classification. *Association for Computational Linguistics* (cit. on pp. 5, 8, 44–46, 66, 68).

Johnson, A. E., Pollard, T. J., Shen, L., Wei, L., Lehman, H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., & Mark, R. G. (2016). MIMIC-III, a freely accessible critical care database. *Scientific Data*, *3* (cit. on pp. 4, 19).

Jurafsky, D., & Martin, J. H. (2016). *Spelling correction and the noisy channel* [Draft of November 7, 2016]. (Cit. on p. 18).

Kalyan, K. S., & Sangeetha, S. (2020). Medical concept normalization in user-generated texts by learning target concept embeddings. *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, 18–23 (cit. on p. 64).

Kartsaklis, D., Pilehvar, M. T., & Collier, N. (2018). Mapping text to knowledge graph entities using multi-sense LSTMs. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1959–1970 (cit. on pp. 9, 43, 44, 50, 64, 67).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *3rd International Conference for Learning Representations (ICLR)* (cit. on pp. 56, 76, 91).

Kotitsas, S., Pappas, D., Androutsopoulos, I., McDonald, R., & Apidianaki, M. (2019). Embedding biomedical ontologies by jointly encoding network structure and textual node descriptors. *Proceedings of the 18th BioNLP Workshop and Shared Task*, 298–308 (cit. on p. 72).

Lai, K. H., Topaz, M., Goss, F. R., & Zhou, L. (2015). Automated misspelling detection and correction in clinical free-text records. *Journal of Biomedical Informatics*, *55*, 188–195 (cit. on pp. 18, 19, 26, 28–30, 32).

Leaman, R., Doğan, R. I., & Lu, Z. (2013). DNorm: Disease name normalization with pairwise learning to rank. *Bioinformatics*, *29*(22), 2909–2917 (cit. on p. 42).

Leaman, R., Khare, R., & Lu, Z. (2015). Challenges in clinical natural language processing for automated disorder normalization. *Journal of Biomedical Informatics*, *57*, 28–37 (cit. on pp. 7, 42).

Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, *36*(4), 1234–1240 (cit. on pp. 4, 45, 55, 76, 90).

Le-Khac, P. H., Healy, G., & Smeaton, A. F. (2020). Contrastive representation learning: A framework and review. *IEEE Access*, *8*, 193907–193934 (cit. on p. 67).

Li, H., Chen, Q., Tang, B., Wang, X., & Xu, H. (2017). CNN-based ranking for biomedical entity normalization. *BMC Bioinformatics*, *18(Suppl 11)*(385) (cit. on p. 42).

Limsopatham, N., & Collier, N. (2016). Normalising medical concepts in social media texts by learning semantic representation. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1014–1023 (cit. on p. 42).

Liu, H., Wu, S. T., Li, D., Jonnalagadda, S., Sohn, S., Wagholikar, K., Haug, P. J., Huff, S. M., & Chute, C. G. (2012). Towards a semantic lexicon for clinical natural language processing. *AMIA Annual Symposium Proceedings* (cit. on p. 18).

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A robustly optimized BERT pretraining approach. *arXiv:1907.11692* (cit. on p. 2).

Mickus, T., Paperno, D., Constant, M., & van Deemter, K. (2020). What do you mean, BERT? *Proceedings of the Society for Computation in Linguistics 2020*, 279–290 (cit. on p. 2).

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *Proceedings of Workshop at International Conference on Learning Representations* (cit. on p. 22).

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 26* (pp. 3111–3119). Curran Associates, Inc. (Cit. on p. 1).

Mitton, R. (2010). Fifty years of spellchecking. *Writing Systems Research*, *2*(1), 1–7 (cit. on p. 18).

Mohan, S., & Li, D. (2019). Medmentions: A large biomedical corpus annotated with {umls} concepts. *Automated Knowledge Base Construction (AKBC)* (cit. on p. 51).

Mougin, F., & Bodenreider, O. (2005). Approaches to eliminating cycles in the UMLS metathesaurus: Naïve vs. formal. *AMIA Annual Symposium Proceedings* (cit. on p. 72).

Mrkšić, N., Ó Séaghdha, D., Thomson, B., Gašić, M., Rojas-Barahona, L. M., Su, P.-H., Vandyke, D., Wen, T.-H., & Young, S. (2016). Counter-fitting word vectors to linguistic constraints. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 142–148 (cit. on pp. 5, 86).

Pagliardini, M., Gupta, P., & Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 528–540 (cit. on pp. 4, 55).

Pakhomov, S. V., Finley, G., McEwan, R., Wang, Y., & Melton, G. B. (2016). Corpus domain effects on distributional semantic modeling of medical terms. *Bioinformatics*, *32*(23), 3635–3644 (cit. on pp. 60, 79, 91).

Pakhomov, S. V., Pedersen, T., McInnes, B., Melton, G. B., Ruggieri, A., & Chute, C. G. (2011). Towards a framework for developing semantic relatedness reference standards. *Journal of Biomedical Informatics*, *44*, 251–265 (cit. on pp. 60, 79, 91).

Pande, H. (2017). Effective search space reduction for spell correction using character neural embeddings. *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 170–174 (cit. on p. 19).

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., . . . Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, & R. Garnett (Eds.), *Advances in neural information processing systems 32* (pp. 8024–8035). Curran Associates, Inc. (Cit. on pp. 55, 76, 91).

Patrick, J., Sabbagh, M., Jain, S., & Zheng, H. (2010). Spelling correction in clinical notes with emphasis on first suggestion accuracy. *2nd Workshop on Building and Evaluating Resources for Biomedical Text Mining*, 2–8 (cit. on p. 19).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., & et al. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research*, *12*, 2825–2830 (cit. on p. 28).

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237 (cit. on p. 2).

Phan, M. C., Sun, A., & Tay, Y. (2019). Robust representation learning of biomedical names. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3275–3285 (cit. on pp. 9, 42–44, 47, 50, 54, 57, 60, 61, 64, 66, 67, 81, 84, 86).

Pradhan, S., Elhadad, N., South, B. R., Martinez, D., Christensen, L., Vogel, A., Suominen, H., Chapman, W. W., & Savova, G. (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 (cit. on p. 50).

Ruch, P., Baud, R., & Geissbühler, A. (2003). Using lexical disambiguation and named-entity recognition to improve spelling correction in the electronic patient record. *Artificial Intelligence in Medicine*, *29*, 169–184 (cit. on p. 19).

Schulz, C., Levy-Kramer, J., Van Assel, C., Kepes, M., & Hammerla, N. (2020). Biomedical concept relatedness – a large EHR-based benchmark. *Proceedings of the 28th International Conference on Computational Linguistics*, 6565–6575 (cit. on p. 91).

Shen, D., Wang, G., Wang, W., Min, M. R., Su, Q., Zhang, Y., Li, C., Henao, R., & Carin, L. (2018). Baseline needs more love: On simple word-embedding based models and associated pooling mechanisms. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Long Papers)*, 440–450 (cit. on pp. 45, 55, 68, 90).

Smedt, T. D., & Daelemans, W. (2012). Pattern for Python. *Journal of Machine Learning Research*, *13*, 2031–2035 (cit. on pp. 23, 28, 56, 76, 91).

Snell, J., Swersky, K., & Zemel, R. (2017). Prototypical networks for few-shot learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems 30* (pp. 4077–4087). Curran Associates, Inc. (Cit. on pp. 45, 98).

Sridhar, V. K. R. (2015). Unsupervised text normalization using distributed representations of words and phrases. *Proceedings of NAACL-HLT 2015*, 8–16 (cit. on p. 19).

Styler IV, W. F., Bethard, S., Finan, S., Palmer, M., Pradhan, S., de Groen, P. C., Erickson, B., Miller, T., Lin, C., Savova, G., & Pustejovsky, J. (2014). Temporal annotation in the clinical domain. *Transactions of the Association for Computational Linguistics*, *2*, 143–154 (cit. on p. 25).

Subramanian, S., Trischler, A., Bengio, Y., & Pal, C. J. (2018). Learning general purpose distributed sentence representations via large scale multi-task learning. *International Conference for Learning Representations (ICLR)* (cit. on p. 2).

Sung, M., Jeon, H., Lee, J., & Kang, J. (2020). Biomedical entity representations with synonym marginalization. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 3641–3650 (cit. on pp. 42, 45, 64, 84, 92).

Taherkhani, F., Kazemi, H., Dabouei, A., Dawson, J. E., & Nasrabadi, N. M. (2019). A weakly supervised fine label classifier enhanced by coarse supervision. *ICCV* (cit. on p. 65).

Tawfik, N. S., & Spruit, M. R. (2020a). Evaluating sentence representations for biomedical text: Methods and experimental results. *Journal of Biomedical Informatics*, *104* (cit. on p. 3).

Tawfik, N. S., & Spruit, M. R. (2020b). Evaluating sentence representations for biomedical text: Methods and experimental results. *Journal of Biomedical Informatics*, *104* (cit. on p. 67).

Tolentino, H. D., Matters, M. D., Walop, W., Law, B., Tong, W., Liu, F., Fontelo, P., Kohl, K., & Payne, D. C. (2007). A UMLS-based spell checker for natural language processing in vaccine safety. *BMC Medical Informatics and Decision Making*, *7*(3) (cit. on p. 18).

Tulkens, S., Suster, S., & Daelemans, W. (2016). Using distributed representations to disambiguate biomedical and clinical concepts. *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, 77–82 (cit. on p. 97).

Tulkens, S., Šuster, S., & Daelemans, W. (2019). Unsupervised concept extraction from clinical text through semantic composition. *Journal of Biomedical Informatics*, *91*, 103120 (cit. on pp. 13, 97).

Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., & Wierstra, D. (2016). Matching networks for one shot learning. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, & R. Garnett (Eds.), *Advances in neural information processing systems 29* (pp. 3630–3638). Curran Associates, Inc. (Cit. on p. 45).

Vulić, I., & Mrkšić, N. (2018). Specialising word vectors for lexical entailment. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1134–1145 (cit. on pp. 14, 67, 79).

Wang, Y., Liu, S., Afzal, N., Rastegar-Mojarad, M., Wang, L., Shen, F., Kingsbury, P., & Liu, H. (2018). A comparison of word embeddings for the biomedical natural language processing. *Journal of Biomedical Informatics*, *87*, 12–20 (cit. on p. 3).

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2016). Towards universal paraphrastic sentence embeddings. *International Conference for Learning Representations (ICLR)* (cit. on p. 45).

Wieting, J., & Kiela, D. (2019). No training required: Exploring random encoders for sentence classification. *International Conference on Learning Representations* (cit. on pp. 5, 45, 49, 68, 70, 80, 88).

Wu, C.-Y., Manmatha, R., Smola, A. J., & Krahenbuhl, P. (2017). Sampling matters in deep embedding learning. *ICCV* (cit. on pp. 48, 69, 87).

Wu, C., Tygert, M., & LeCun, Y. (2019). A hierarchical loss and its problems when classifying non-hierarchically. *PLOS ONE, 14*(12), 1–17 (cit. on p. 65).

Wu, S., Roberts, K., Datta, S., Du, J., Ji, Z., Si, Y., Soni, S., Wang, Q., Wei, Q., Xiang, Y., Zhao, B., & Xu, H. (2019). Deep learning in clinical natural language processing: A methodical review. *Journal of the American Medical Informatics Association*, *0*(0), 1–14 (cit. on p. 100).

Yaghoobzadeh, Y., Kann, K., Hazen, T. J., Agirre, E., & Schütze, H. (2019). Probing for semantic classes: Diagnosing the meaning content of word embeddings. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5740–5753 (cit. on p. 5).

Yu, C., Han, J., Wang, P., Song, Y., Zhang, H., Ng, W., & Shi, S. (2020). When hearst is not enough: Improving hypernymy detection from corpus with distributional models. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 6208–6217 (cit. on p. 78).

Zhang, Y., Chen, Q., Yang, Z., Lin, H., & Lu, Z. (2019). BioWordVec, improving biomedical word embeddings with subword information and MeSH. *Scientific Data*, *6* (cit. on p. 4).

List of Figures

List of Tables