Evaluation of Contextual and Non-contextual Word Embedding Models Using Radiology Reports

By

Mirza S. Khan, M.D.

Thesis
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE
in
Biomedical Informatics

December 18, 2021

Nashville, TN

This thesis is dedicated to my wife, children, and parents.

ACKNOWLEDGEMENTS

I wish to express my gratitude and thanks to my three pillars of support: my family, my friends, and my teachers. To my lovely wife, who encouraged me to dream and pursue happiness. None of this would have been possible without her caring advice, endless support and sacrifice. My children, gentle and wide-eyed, who kept me grounded and whose smiles irradiated even the darkest of days. My parents, my first teachers, who devoted their utmost time and energy to nurturing my love for learning and success. By example, they instilled in me the value of hard work and discipline. My sister, my earliest champion and friend, who I strived to follow and whose watchful eye spared me from folly. My grandparents, their love and adoration remains palpable even in their absence. To my dear aunts and uncles who care for me as their own son. And to my cousins and friends who love me and look out for me and my family as they would their own brother. I wish to thank Drs. Matheny, Deppen, and Landman for taking a chance on me and pushing me to see this work come to fruition. I also wish to acknowledge many of my earlier teachers and mentors who helped me reach this phase in my career: Dr. Nafe Hussein, Dr. Parvez Ahmed, Mr. Iftikhar Hussain, Dr. Michael Richardson, Dr. Daniel Batlle, Dr. Peter Kaboli, Dr. Bill Silverman, Dr. Manish Suneja, and Dr. Pierre Massion. I also give my heartiest thanks and appreciation to my friends, faculty, and staff in the Tennessee Valley Healthcare System and Vanderbilt University's Department of Biomedical Informatics, including Rischelle Jenkins, Barbara Payne, Kevin Johnson, Cindy Gadd, Mark Frisse, Steven Brown, Kim Unertl, Jessica Ancker, Gretchen Jackson, Dax Westerman, Alvin Jeffery, Sharidan Parr, Ruth Reeves, Glenn Gobbel, Kaiwen Xu, Riqiang Gao, and Kim Sandler. Lastly, I wish to thank the US Department of Veterans Affairs Office of Academic Affiliations and the Medical Informatics Fellowship for their generous support and funding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Study Context

Natural language processing (NLP) refers to the set of methods for making human language accessible for computation. Traditionally, words or characters were represented using count-based methods or mapped to existing ontologies, but these approaches were limited as they failed to encode many valuable linguistic properties.

Drawing from linguistic principles, methods were developed to encode distributional representations of words. These representations are commonly referred to as *embeddings*. Although these captured important semantic and syntactic properties and improved downstream NLP task performance, early adoption was limited as training these neural network models proved to be computationally expensive. Advancements in training techniques and processing capabilities made the use of embeddings more accessible. Overcoming some of the limitations of early embedding representations, embedding models were further sub-divided into *non-contextual word embeddings* (NCWE) and *contextual word embeddings* (CWE). NCWE models refer to methods where each word is represented by a single, static word vector (See Section 2.4). Thus, NCWE representations poorly reflect words with multiple senses, such as polysemy or homonymy. By contrast, CWE methods represent words dynamically as they account for each word's context when representing each word vector (See Section 2.5.3).

Despite the gains afforded by embedding models on both intrinsic and extrinsic evaluation, rule-based NLP methods are still commonly used in practice. For example, Vanderbilt University Medical Center (VUMC) and other hospital systems use a rule-based NLP tool, De-Id, for de-identification of clinical text documents (D. Gupta, Saul, and Gilbertson 2004; Danciu et al. 2014). As is common with rule-based NLP systems, additional engineering and iterative improvements of the original De-Id software was needed to achieve suitable results at VUMC. The primary advantage of rule-based NLP methods is model interpretability, which is of critical importance given the potential health care implications using clinical NLP methods.

Phenotyping is a common application within clinical NLP as leveraging clinical text can improve phenotype classification performance. Often, NLP approaches can lead to striking improvements, e.g. where conditions, procedures, or observations are poorly coded or unaccounted for in existing medical coding systems (Wiley et al. 2013). Many existing NLP-based phenotyping algorithms rely on regular expressions or existing biomedical NLP tools, which incorporate the traditional NLP pipeline. This persists despite embedding models being far more generalizable and their potential for yielding state-of-the-art (SOTA) performance on most downstream NLP tasks. In addition, embedding models require far less text processing and engineering to achieve SOTA performance relative to earlier clinical NLP methods.

NLP methods continue to evolve, often doing so to overcome the limitations of existing techniques. Recent advances in embeddings include optimized training of spherical embeddings and the introduction of Efficient Transformer models.

1.2 Motivation for the Study

Clinical NLP tends to track many of the advancements from the general NLP domain. Early clinical NLP efforts relied on regular expressions, followed by more sophisticated NLP pipelines, which comprised a series of steps such as tokenization and dependency parsing. The output of these NLP approaches was either the object of interest itself, e.g. clinical concept extraction, or used as features for other tasks, e.g. as an input for phenotype classification models. Using advanced NLP methods, many of these tedious and intensive processing steps can be eliminated. For instance, embedding models allow for easy and efficient processing of raw text and may achieve SOTA results on a broad range of downstream NLP applications.

Despite these advancements, we have a limited understanding of how well embedding models encode clinical text information and if or how each differs on downstream clinical NLP tasks. Each word embedding model tends to differ with respect to the training objective, which may influence how the generated embeddings encode text information. Presently, few, if any, clinical intrinsic evaluation benchmarks exist to study embedding model representations against clinician judgment. Those that do exist were manually curated and developed prior to the introduction of existing embedding algorithms in use today.

Clinical text is distinct from general text corpora making it difficult to extrapolate general NLP findings to the clinical domain. Thus, it remains unclear if and how performance of many existing and recently developed embedding algorithms may differ on evaluation tasks derived from clinical text. While intuitively in-domain embeddings may be considered ideal model choices, evidence is needed to support the use of embedding models pre-trained on clinical documents over those pre-trained on general text corpora.

1.3 Aim and Scope

The aim of this study is to examine and compare embedding model representations of clinical text as compared to clinician judgment and to investigate if and how model selection may effect phenotype classification performance.

To assess this, I defined and developed intrinsic and extrinsic evaluation tasks using a custom radiology report corpus. Our intrinsic evaluation tasks allowed us to compare embedding model representations against clinician judgment to study the effect of embedding algorithm selection and choice of corpus used for pre-training. For extrinsic evaluation, I developed a phenotyping classification task, a common use case for clinical NLP practitioners, and compare performance for different models to examine the effect of algorithm type or pre-training corpus.

I hypothesized that embedding models pre-trained on our custom clinical corpus would outperform others on both intrinsic and extrinsic evaluation tasks. I also expected to find that embeddings pre-trained on a large biomedical and clinical corpus would outperform those pre-trained on general corpora for both evaluation tasks. Among the NCWE models pre-trained on our custom corpus, I hypothesized that spherical embeddings would provide the best performance on both intrinsic and extrinsic evaluation based on published findings demonstrating so on general NLP intrinsic and extrinsic evaluation benchmarks. For the extrinsic evaluation task, I hypothesized that Efficient Transformer models would perform best, followed by BERT-based models, and that Bidirectional LSTM models would perform worst for phenotype classification.

1.4 Study Significance

One intended outcome of the study is to develop a data-driven method for creation of intrinsic evaluation benchmarks by leveraging pre-trained embedding models. This method aims to improve on the time and resource intensive manual curation process used to construct existing clinically oriented intrinsic evaluation tasks. This approach is easy to implement and can be used to expedite the creation of additional clinical intrinsic evaluation benchmarks, which is important to further our understanding and trust in these models and their representations.

A seconded intended outcome of the study is to determine which embedding algorithm(s) and pre-training corpora represents words in greatest agreement with clinician judgment. The results of our intrinsic evaluation task clarify model and pre-training corpus selection most useful to optimize word vector representations to align with clinician appraisal. In addition to improving interpretability and establishing trust, informing these optimal model preferences is useful for other NLP tasks, such as term expansion and word sense disambiguation.

Further, a third intended outcome clarifies which embedding model(s) and pre-training corpora perform best on clinical phenotyping using radiology text. These results inform the effect of custom model pre-training and how model and corpus choice influence phenotype classification performance, and which may be most preferable depending on label balance and access to computing resources.

1.5 Overview of the Study

This thesis consists of 4 further chapters. In Chapter 2, I situate the current study in related literature and pre-existing work to provide appropriate background for the remaining chapters. In Chapter 3, I detail the development of intrinsic evaluation tasks and share how different embedding model configurations influence performance on these tasks. In Chapter 4, I present extrinsic evaluation tasks and establish how model selection and pre-training corpus effect training and phenotype classification performance. Lastly, I conclude with Chapter 5, tying in findings from the preceding chapters.

CHAPTER 2

BACKGROUND

2.1 Clinical Text

Free-text clinical documents in the electronic medical record (EMR) contain a wealth of useful clinical information, much of which is unavailable as structured data elements. This text may include specific information about clinical findings, patient risk factors or disease management. Moreover, this free text may contain nuance and language about diagnostic reasoning, clinical uncertainty, and other information that cannot be expressed or captured using existing structured terminologies.

Clinical text data is often subject to misspellings, abbreviations and other sources of ambiguity. Despite this, the **semantics** (or meaning) of the text is preserved as clinicians are able to comprehend these varying forms. For example, 'Cr', 'Creat', and 'creat' are all understood by clinicians to mean *creatinine*. Such heterogeneity in representation of terms or concepts presents a unique set of challenges working with clinical text. To further complicate matters, the **syntax**, or style of language, sentence construction, and grammar usage, used in clinical notes is distinct than what is encountered in other domains.

Importantly, spelling errors or frequent use of abbreviations may be less common in radiology reports compared to other clinical documents. Radiologists frequently transcribe their reports, which may be prone to other sources of error. Examples include incorrect word substitution, failure of word capture, or nonsensical text, which may be found in final reports if not carefully reviewed by dictating authors (Quint, Quint, and Myles 2008).

Moreover, the structure of clinical notes need not adhere to any standard and can also vary by practice setting, clinical note type, and author preference. For instance, clinician notes traditionally adhered to a particular format called SOAP: Subjective, Objective, Assessment, and Plan. To improve readability, some clinicians have begun to adopt the APSO format, wherein the note begins with the Assessment and Plan sections (Lin et al. 2013). Additionally, many of the documents within the EMR are often auto-populated from structured data or copied from earlier notes, contributing to increased note length and redundancies.

2.2 Ontologies and Knowledge Graphs

Significant effort and resources in biomedical and clinical natural language processing (NLP) have been devoted to developing and maintaining structured ontologies or controlled vocabularies. The Unified Medical Language System (UMLS) Metathesaurus is produced and maintained by the National Library of Medicine. This resource can be used to link concepts across several existing ontologies. Varied expressions of the same concept can be mapped to a single concept unique identifier (CUI) in the Metathesaurus. Maintaining such ontologies requires continued manual curation and refinement as language and medical understanding evolves, e.g. as new medical therapies are introduced or novel diseases are discovered.

Lexical hierarchies for the general English language, such as WordNet, have been used as part of general NLP pipelines to capture semantic properties (Herdağdelen, Erk, and Baroni 2009). In biomedical NLP, lexical hierarchies, such as the UMLS Metathesaurus and Medical Subject Headings (MeSH) thesaurus have also proven useful for NLP tasks and capturing semantics. For example, Rosario and Hearst were able to reduce their input feature space by converting text to UMLS CUIs, which they then used to convert to parent MeSH terms. MeSH term representations gave equivalent semantic relation extraction performance and generalized better than neural network models using text features as input (Rosario and Hearst 2001). Others have also leveraged the relationships in these knowledge-based taxonomies to calculate similarity, e.g. using path finding or information content measures. Garla and Brandt used the Systematized Nomenclature of Medicine - Clinical Terms (SNOMED-CT), MeSH, and the UMLS Metathesaurus as knowledge graphs to compute similarity measures for concept pairs (Garla and Brandt 2012).

Early work involving concept extraction and mapping relied on heuristics or rule-based systems and often involved manual engineering to get suitable results. Yet, such customization can often lead to poor generalization. As the field advanced, text processing in the traditional NLP pipeline began to involve one or more of a series of steps: tokenization, text normalization, part-of-speech tagging, dependency parsing, and named entity recognition. Many of these steps improved performance on downstream NLP tasks. Several NLP tools have been developed over the years to identify and extract concepts from biomedical text documents often using elements of this NLP pipeline. Available systems include: MedLEE, MetaMap, KnowledgeMap Concept Identifier (KMCI), Clinical Text Analysis and Knowledge Extraction System (cTAKES), Clinical Language Annotation, Modeling, and Processing (CLAMP) toolkit, CliNER, and scispacy (Friedman et al. 1994; Aronson 2001; Denny et al. 2003; Savova et al. 2010; Soysal et al. 2017; Boag et al. 2018; Neumann et al. 2019). Additional heuristics were also introduced to address the challenges of negation and abbreviation detection (Chapman et al. 2001; Schwartz and Hearst 2003) Still, the heterogeneity of clinical text can affect performance of these tools, e.g. misspellings and non-standard acronyms. Additional challenges include differences in representing conditions or diagnoses which can vary by region, specialty or clinical practice setting, English proficiency, typing skills, and fidelity of transcription.

New methods have shifted the paradigm of the traditional NLP pipeline yet again with the introduction of word embedding models.

2.3 Count-based word representations

Natural language processing (NLP) methods provide an opportunity to leverage unstructured text for several clinically relevant tasks, such as information extraction, word sense disambiguation, and clinical phenotyping. Count-based word representations provide one approach for natural language to be represented numerically for computation.

2.3.1 One-hot encodings

Traditionally, words were numerically represented using one-hot encodings (OHE). In this method, each word in the corpus vocabulary is transformed to a feature vector using a a one-hot representation where the vector contains all zeros except for one 1 at the word index position. Typically, the index of each word is determined by its position after each word in the corpus vocabulary is sorted. Each word represented using OHE is a vector with length equal to the size of the corpus vocabulary. An example of a one-hot representation of words is as follows:

$$\boldsymbol{w}^{\text{abacavir}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \boldsymbol{w}^{\text{abasia}} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \boldsymbol{w}^{\text{abatacept}} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots \boldsymbol{w}^{\text{Zyrtec}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

OHE generates sparse vector representations that may become computationally in-tractable as the vocabulary grows larger. Moreover, this representation fails to capture any additional linguistic properties, such as semantic or syntactic information that may be useful for NLP-related tasks.

2.3.2 n-grams

***n*-grams:** One of the limitations of OHE is that they are unable to account for neighboring terms that may provide important context and information. *n*-grams are obtained by concatenating short overlapping sequences of *n* words or characters.

As with OHE, *n*-grams also fail to capture semantic or syntactic information between words. Moreover, while it does aim to capture context from neighboring words, the learned representations may fail to generalize (Bengio et al. 2003).

2.3.3 Term-frequency and Inverse document frequency

Term frequency (TF) reflects the frequency by which a given term t appears in a document d.

$$\text{TF}_{t,d} = \text{count}(t, d)$$

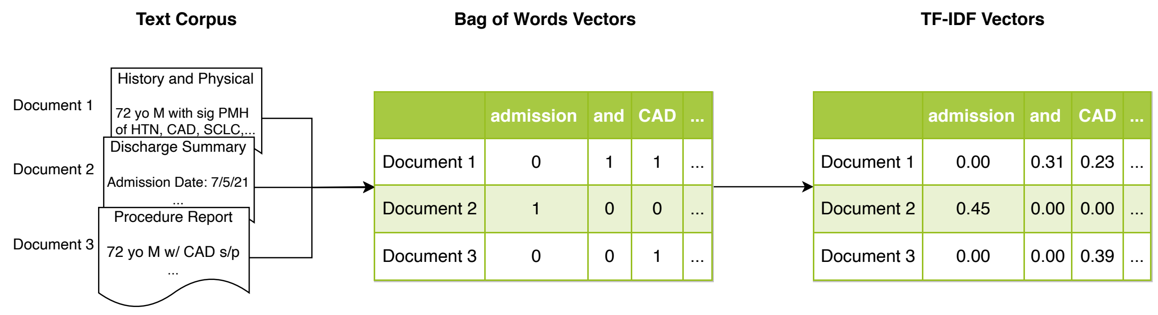
The inverse document frequency (IDF) represents the inverse form of the number of documents containing a given term (df_t) of the overall documents in the corpus (N). The IDF vector representation penalizes common words and weights rare words higher.

$$\text{IDF} = \log \frac{N}{df_t}$$

The term frequency-inverse document frequency (TF-IDF) score for each term t combines both the TF and IDF weighting schemes and is simply the product of the two:

$$\text{TF-IDF} = \text{TF}(t) \cdot \text{IDF}(t)$$

TF, IDF or TF-IDF can be used for term weighting and normalization. TF-IDF has commonly been used in information retrieval and text mining tasks. These weighting methods are each a statistical method that may be used to evaluate how important a word is to a document in a collection or corpus. Figure 2.1 shows an example of TF-IDF word vector creation extending from the bag of words vectors (See Section 2.3.1). Variations of these approaches have commonly been used to score and rank relevant documents provided a given query (Jurafsky and Martin 2020).



	admission	and	CAD	...
Document 1	0	1	1	...
Document 2	1	0	0	...
Document 3	0	0	1	...

	admission	and	CAD	...
Document 1	0.00	0.31	0.23	...
Document 2	0.45	0.00	0.00	...
Document 3	0.00	0.00	0.39	...

Figure 2.1: Term-frequency and Inverse document frequency example.

2.3.4 Dense vector representations

Most of the aforementioned count-based methods create sparse vector representations. Yet, dense vectors are more computationally efficient and provide better generalization. Dense vectors can be obtained from sparse vectors by applying dimensionality reduction techniques. One of the more common approaches of doing so in NLP is by using Singular Value Decomposition (SVD), which was purportedly popularized in this space through Latent Semantic Analysis (LSA), which involves SVD computation (Levy, Goldberg, and Dagan 2015). Additional dense vector approaches are described in Section 2.4.6.

2.3.5 Limitations of Count-based Methods

As detailed above, many of the count-based methods for word representation fail to capture information about semantic or syntactic similarity. Those that have this capability, such as LSA, suffer from other challenges that plague count-based approaches, including computational inefficiencies and poorer performance relative to embedding models (Bengio et al. 2003; Baroni, Dinu, and Kruszewski 2014; Levy, Goldberg, and Dagan 2015). While dimensionality reduction methods like SVD and principal component analysis (PCA) may be applied to generate dense vectors, these methods scale poorly when dimensions are on the order of millions or greater, which is not uncommon for dictionaries based on large corpora. Additionally, sparse vector representations fail to generalize well in handling text beyond the training corpus.

2.4 Non-contextual Word Embeddings

One pivotal NLP advancement was the introduction of non-contextual word embeddings (NCWEs), also called word vectors, to represent a vocabulary of terms. Word embedding models are based upon the distributional hypothesis, which, succinctly put, states that “a word is characterized by the company it keeps” (Firth 1957). Earlier work in linguistics supports that contextual information provides a suitable approximation for word meaning as semantically similar words often have similar contextual distributions (Faruqui and Dyer 2014). Thus, if trained well, the resulting vector space yields semantically or syntactically similar words mapped close to one another.

The concept of distributed word representations and their ability to capture word semantics in an efficient and generalizable manner was detailed decades ago (Hinton, McClelland, and Rumelhart 1986; Rumelhart, Hinton, and Williams 1986). Yet, early efforts using neural networks for language modeling relied on central processing units (CPUs) for computation and took several weeks to train (Bengio et al. 2003; Collobert and Weston 2008; Turian, Ratinov, and Bengio 2010; Mikolov, Chen, et al. 2013). For example, Bengio et al. report training their model for 3 weeks using 40 CPUs and Collobert and Weston trained for 8 weeks on a single CPU (Bengio et al. 2003; Collobert and Weston 2008). Despite lengthy training times, these early neural network-generated word vectors were transformative as they significantly outperformed other approaches on a range of NLP applications (Bengio et al. 2003; Collobert and Weston 2008; Collobert et al. 2011; Agirre et al. 2009; Mnih and Hinton 2008; Turian, Ratinov, and Bengio 2010; Socher et al. 2012).

2.4.1 word2vec

In 2013, Mikolov et al. introduced the `word2vec` algorithm, which extended these earlier methods and reduced the embedding model training time from several weeks to mere hours (Mikolov, Chen, et al. 2013). Shortly after its initial release, the authors found that additional modifications to the training procedure, specifically negative sampling and subsampling of frequent words, improved training speeds further and generated more accurate word representations based on intrinsic evaluation tasks, i.e. tasks comparing model representations against human judgment (Mikolov, Sutskever, et al.

2013). A more detailed overview of intrinsic evaluation is presented in Section 2.6.

Algorithms such as `word2vec` are trained on an unannotated corpus and use an unsupervised learning approach comprising shallow neural networks that map each word in a vocabulary to a real-valued vector ('dense vector') in an embedding space of specified dimensionality D. In effect, this allows each word to be represented by a single vector of fixed length D, e.g. the term 'hypertension' may be represented as a vector $[0.32, 0.79, 0.36, 0.5, 0.65]$ in an embedding space where $D = 5$.

This contrasts with earlier methods that relied on count-based approaches, many of which yielded sparse vector representations and captured little, if any, syntactic or semantic information. Unlike sparse vectors, dense word vectors are more computationally tractable, are capable of capturing synonymy, and less susceptible to the "curse of dimensionality" (Bengio et al. 2003; Mikolov, Chen, et al. 2013; Jurafsky and Martin 2020).

`word2vec` allows for training using either a skip-gram method or a continuous bag of words (CBOW) strategy. The CBOW model learns to predict the center word within a window of specified length using the sum of the vector representations of the words within that window. Whereas CBOW uses the context words to predict the center word, the skip-gram algorithm learns to predict the context word from the center word (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013). Conceptually, the CBOW training approach may be thought to generate better representations by combining tokens in each context window. Yet, skip-gram models have been shown to outperform CBOW on most benchmarks. Additionally, Levy and Goldberg find that the skip-gram method is also the fastest to train and requires less disk space and memory consumption (Levy, Goldberg, and Dagan 2015).

2.4.2 Beyond word2vec

In contrast to `word2vec`, the `fastText` algorithm can create word vectors for character n-grams rather than the complete words. Thus, embedding models trained using `fastText` are capable of computing word representations for words not present within the training corpus and may generalize better given this ability to handle out-of-vocabulary words (Bojanowski et al. 2017).

While initially applied to words in a text corpus, clinical embedding models have also been extended to concept unique identifiers, phrases, sentences and documents (Beam et al. 2019; Choi, Chiu, and Sontag 2016; Baumel et al. 2017; Kalyan and

Sangeetha 2020; Henry, Cuffy, and McInnes 2017). Such embedding models have also been adapted to represent patient-level longitudinal EMR data as demonstrated by Zhang et al.'s `Patient2Vec` (J. Zhang et al. 2018).

2.4.3 Properties of NCWEs

Each dimension of an embedding is thought to reflect a latent property of the word, which may help capture useful semantic and syntactic properties (Turian, Ratinov, and Bengio 2010). These real-valued word vectors occupy a point in vector space and nearby points within this space should represent semantically similar terms.

As highlighted in Figure 2.2, one can use vector similarity metrics, such a cosine similarity or cosine multiplication similarity, to compute how similar word pairs are using their word vector representations (Levy and Goldberg 2014). Word pair similarity measurement is discussed further in Section 2.6.3.

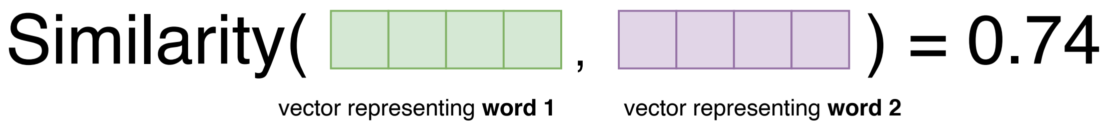

Similarity(
,
) = 0.74
vector representing **word 1**
vector representing **word 2**

Figure 2.2: Vector similarity methods can be used to compute a numeric value representing the similarity of two words based on each word's vector representation.

The quality of embeddings depends on several factors, most of important of which are the size and quality of the training corpus, model architecture and model hyperparameter selection (Chiu et al. 2016). Embedding sizes commonly seen in the literature range from 25 to 500 dimensions (Rao and McMahan 2019). Yin and Shen conducted an in-depth study of embedding dimension size on different benchmarks, including WordSim353. They compared performance for dimensions of size 0 to 10,000 and found that with increasing dimension size, performance initially improved, then slightly decreased before plateauing (Yin and Shen 2018). A similar increase and decay was also reported by Agirre et al. in their experiments with dimension size on intrinsic evaluation performance (Agirre et al. 2009). TH et al. found that using word vectors with a dimension of at least 200 provided the best performance (TH, Sahu, and Anand 2015).

Use of word embeddings have been shown to dramatically improve NLP model and machine learning model performance (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013; Joulin et al. 2016). Furthermore, domain-specific word embeddings have been shown to significantly improve semantic relations, such as those developed using radiology reports (Z. Jiang et al. 2015; Zech et al. 2018).

2.4.4 Augmenting Non-contextual Embeddings with Knowledge Graphs

Knowledge graphs or lexical hierarchies, such as those introduced in Section 2.2, were also used to represent terms as an alternative to count-based word representations (See Section 2.3) prior to the adoption of non-contextual embedding models (Agirre et al. 2009; Rosario and Hearst 2001; Rosario and Hearst 2004; Garla and Brandt 2012). Agirre et al. showed that NCWEs trained using a web crawl corpus were comparable to knowledge graph-based approaches, specifically WordNet, on word similarity benchmarks. They also found that the combination of the two performed better than either approach alone (Agirre et al. 2009). Using a model agnostic post-processing procedure, Faruqui et al. demonstrated that knowledge graphs could be used to improve word vector representations based on word similarity and other intrinsic evaluation tasks (Faruqui et al. 2015). For a more detailed overview of intrinsic evaluation, see Section 2.6. Such methods to augment word embeddings with biomedical knowledge graphs have also shown gains in biomedical NLP task performance (Yu et al. 2017; Boag and Kané 2017; Banerjee et al. 2017; Banerjee et al. 2018).

2.4.5 Limitations of Non-contextual Embeddings

One of the major limitations of NCWEs is their inadequacy in handling words with multiple senses, such as polysemy or homonymy (Jurafsky and Martin 2020). For example, the word *calculus* may refer to the mathematics subject or mineral deposition. Non-contextual embeddings typically represent *calculus* as only one vector form, rather than the many forms it theoretically may take. Contextual embedding models (detailed in Section 2.5) use the surrounding context, e.g. that the text refers to renal anatomy to deduce that *calculus* in this context refers to a kidney stone.

2.4.6 Other weighting schemes

There are many other weighting schemes that have been introduced, among these are Pointwise Mutual Information (PMI), Latent Semantic Analysis (LSA), and Latent Dirichlet Allocation (LDA). Each of these three can create distributional representations of words, which capture word context information. Yet, LDA can become computationally expensive on large datasets. Additionally, neural networks have been shown to outperform LSA in their ability to preserve linear regularities among words (Mikolov, Chen, et al. 2013).

I introduce the reader to PMI as variants of this approach have been shown to generate word vector representations comparable to embedding models, which are to be discussed in the next section (Section 2.4) (Levy and Goldberg 2014; Levy, Goldberg, and Dagan 2015). Positive PMI (PPMI) is merely a variant of PMI where all negative values are replaced with 0. PPMI representations have previously been shown to outperform PMI on semantic similarity tasks. Levy and Goldberg find that a few modifications to the PPMI matrix, namely context distribution smoothing and singular value decomposition (SVD), can induce word vector representations that yield comparable performance to word2vec and GloVe embeddings on a series of general intrinsic evaluation tasks (Levy, Goldberg, and Dagan 2015). While Baroni et al. find that dense SVD representation were inferior to the embedding representations presented in Section 2.4 (Baroni, Dinu, and Kruszewski 2014), Levy and Goldberg find that that these inferior results were likely because the authors used the traditional SVD eigenvalue matrix with a value of 1. By simply modifying the eigenvalue to 0.5 or 0, they generated dense representations on par with word2vec and GloVe embeddings on a series of intrinsic evaluation benchmarks (Levy, Goldberg, and Dagan 2015).

2.5 Deep Learning for NLP

2.5.1 Recurrent Neural Networks

Text can be thought of a sequence of characters or words. Many neural network architectures, such as convolutional neural networks (CNN), only accept a fixed-sized input and produce a fixed-sized output. Using such models on text often involves padding or truncating inputs to a fixed length for processing. By contrast, recurrent neural networks (RNN) perform sequential processing of inputs and can handle inputs of varying length. The design of RNNs also allows for the sequence order to be preserved. Each component of the RNN architecture builds upon the previous so-called hidden output step. RNNs can perform many of the aforementioned parts of the traditional NLP pipeline (Section 2.2). For example, RNNs can be used for part-of-speech tagging and named entity recognition. With fewer pipeline processing steps, they can directly perform many downstream NLP tasks as well, e.g. text classification, summarization, and speech recognition.

In practice, RNNs can become unstable during training, particularly when handling longer input sequences (Bengio, Simard, and Frasconi 1994). The exploding gradient problem can arise if the gradient becomes too large. In effect, this may cause excessive updating of model weight parameters during training. A common solution to the exploding gradient problem is gradient clipping. Gradient clipping involves scaling down the gradient prior to applying the gradient descent optimization update during training (Goodfellow, Bengio, and Courville 2016). Another drawback of traditional RNN architectures is referred to as the vanishing gradient problem. With an increasing number of time steps such as with longer input sequences, traditional RNNs have increasing difficulty preserving information. Long short-term memory (LSTM) and the Gated recurrent unit (GRU) models are variants of the traditional RNN that improve the ability to retain long-term information thus overcoming the vanishing gradient problem (Hochreiter and Schmidhuber 1997; Cho et al. 2014; Goodfellow, Bengio, and Courville 2016).

2.5.2 Attention Mechanisms

Another solution to the vanishing gradient problem is the use of an attention mechanism. Additional benefits of attention are improved model performance and interpretability. Attention is a general deep learning technique and has even been combined with CNN and RNN architectures for image captioning (Xu et al. 2015). Attention allows the model to focus on a particular part(s) of the input sequence. There are several variants of attention, but each involves the following sequence: calculation of attention scores, applying softmax to these scores to obtain an attention distribution, and computing a weighted sum of this attention distribution to generate an attention vector. Attention variants often differ in how the attention scores are calculated. These variants include additive attention, multiplicative attention, basic dot-product attention, and scaled dot-product attention (Genthial et al. 2019). This latter type, scaled dot-product attention, is the attention mechanism used in the Bidirectional Encoder Representations from Transformers (BERT) architecture (Figure 2.3) (Devlin et al. 2019). Additionally, attention mechanisms may operate on only part of the input sequence (*local attention*) or on the entire input sequence (*global attention*).

2.5.3 Contextual Word Embeddings

Embeddings from Language Models (ELMo) ushered in contextualized word embedding (CWE) models helping to overcome some of the limitations of NCWE models (See

Section 2.4.5). Existing state-of-the-art (SOTA) methods in NLP use contextual word embedding models, such as BERT, and perform extremely well on a wide range of tasks according to established benchmarks. Each of these models demonstrates the benefits of fine-tuning the models for domain-specific tasks.

2.5.3.1 Transformer Architecture

ELMo was developed by simultaneously training two bidirectional LSTM models with a language modeling objective: a forward language model and a backward language model (Peters et al. 2018). Vaswani et al. developed the transformer architecture, which is based entirely on attention blocks as shown in Figure 2.3 (Vaswani et al. 2017). By eliminating RNN components, the transformer effectively decreases the model's path length. Additionally, unlike RNNs and their variants, which receive each token as input at each time step, transformers can take the entire sequence as input. The model dramatically outperformed other architectures and gave state-of-the-art results while also generalizing to other NLP tasks beyond machine translation (Vaswani et al. 2017). This led to the development of several transformer language models, such as Google's Bidirectional Encoder Representations from Transformers (BERT) and Facebook's Robustly Optimized BERT Pretraining Approach (RoBERTa) (Devlin et al. 2019; Liu et al. 2019).

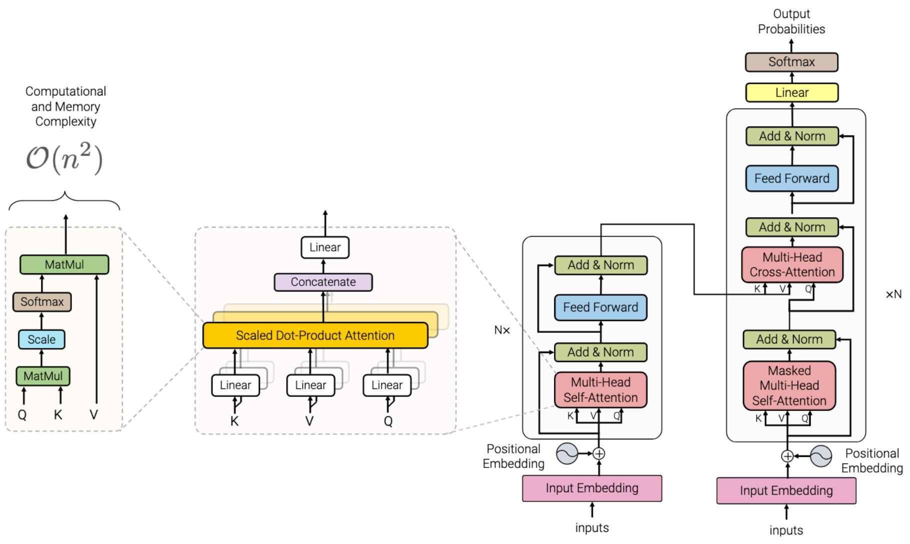


Figure 2.3: Architecture of the original Transformer model by Vaswani et al., 2017 (Source: Tay et al., 2020)

This new class of language models was referred to a *large language models* as their parameter size was orders of magnitude greater than previous models. One of the major advantages of these and other language models, such as ELMo and Universal

Language Model Fine-tuning for Text Classification (ULMFiT), is their utility for transfer learning (Peters et al. 2018; Howard and Ruder 2018). These contextual word embedding (CWE) models pre-trained on a large dataset, e.g. an English Wikipedia corpus, could then be fine-tuned on a smaller or niche dataset, e.g. clinical notes, and provide stellar results. One can fine-tune a few or all of a model's layers. Devlin et al. demonstrated that fine-tuning only the last four hidden layers for the Conference on Natural Language Learning (CoNLL) NER task performed almost as well as fine-tuning all layers (Devlin et al. 2019).

BERT and RoBERTa were trained on general text corpora, e.g. BERT was trained using BookCorpus and English Wikipedia (Devlin et al. 2019). As with non-contextual embeddings, these models can be adapted to a domain-specific context using a targeted domain corpus. Researchers have pre-trained the BERT model using biomedical and clinical documents; these include BioBERT, SciBERT, BlueBERT, and ClinicalBERT (Alsentzer et al. 2019; Beltagy, Lo, and Cohan 2019; Lee et al. 2020; Peng, Yan, and Lu 2019).

BERT pre-training involves two main tasks: masked language modeling (MLM) and next sentence prediction (NSP). By contrast, RoBERTa uses MLM and does not rely on NSP for pre-training (Liu et al. 2019). For each input sequence, BERT's MLM selects 15% of token at random; 80% of these are replaced by a `[MASK]` token, 10% are replaced by another random token, and the remaining 10% are left unchanged. During pre-training, BERT attempts to reconstruct the original token that was masked as shown in Figure 2.4 (Devlin et al. 2019). Tenney et al. probed the layers of the BERT model and found that they learn different elements of the traditional NLP pipeline, which is described in Section 2.2 (Tenney, Das, and Pavlick 2019).

2.5.3.2 Tokenization in BERT

For text tokenization, BERT relies on the WordPiece algorithm (Devlin et al. 2019). For out-of-vocabulary (OOV) words, some tokenization algorithms often use a single word vector to represent unknown words. The WordPiece algorithm can maintain a shorter vocabulary and handle OOV words by representing words as subwords. As shown in Figure 2.5, the base BERT model does not have the word *bullae* in its vocabulary, but does contain the word *bull*. The base BERT model's WordPiece tokenizer converts the word *bullae* to the subwords *bull* and *##ae*, i.e. it represents this single word using two tokens. On average, subword tokenization models such as WordPiece have approximately four characters per subword (Clark et al. 2021).

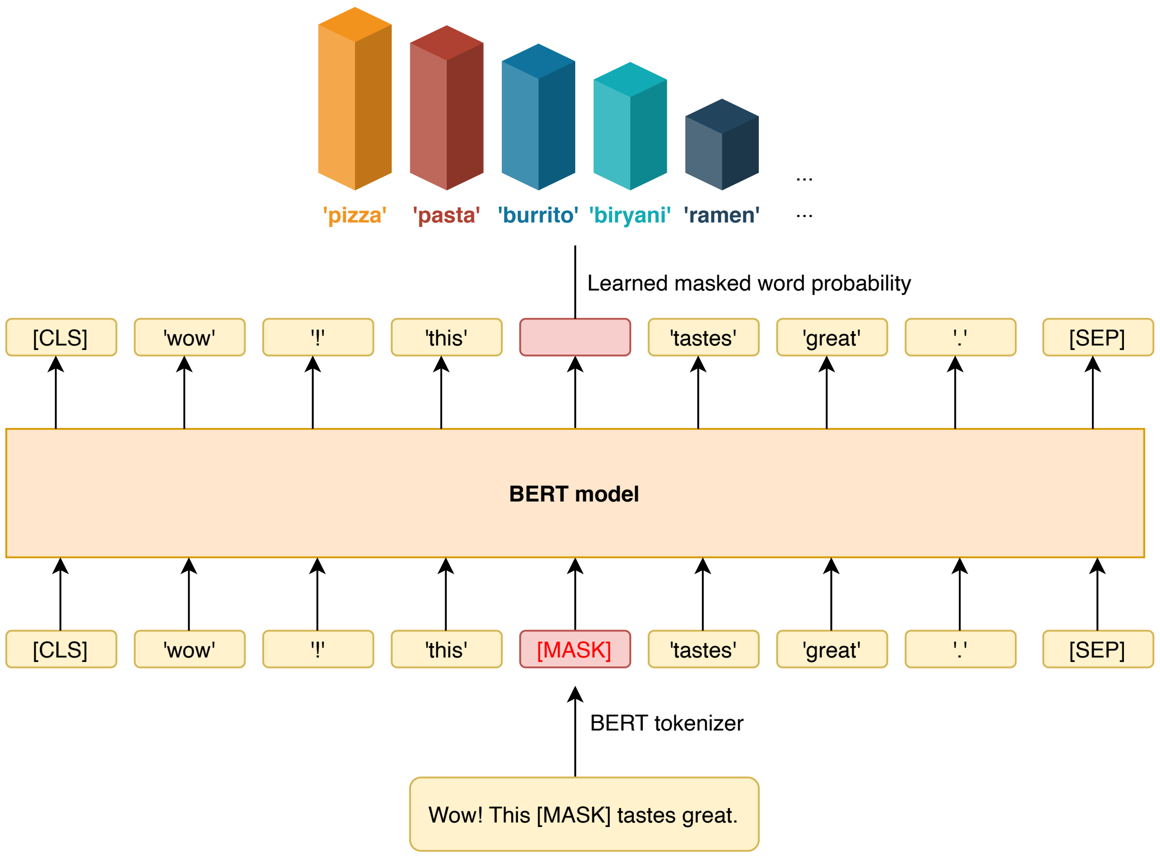
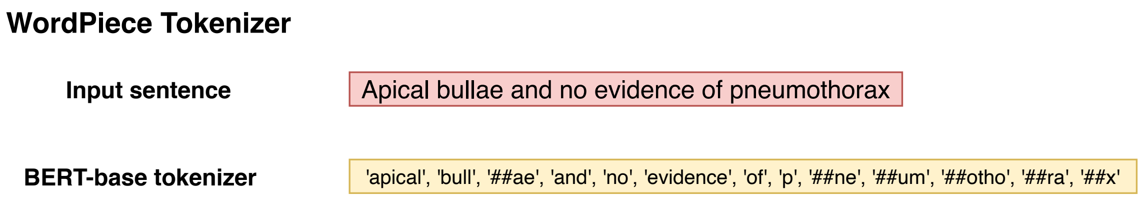


Figure 2.4: Masked language modeling involves random masking of tokens. The language model is optimized to reconstruct the masked token, and aided in doing so by using the context words.



WordPiece Tokenizer

Input sentence Apical bullae and no evidence of pneumothorax

BERT-base tokenizer 'apical', 'bull', '##ae', 'and', 'no', 'evidence', 'of', 'p', '##ne', '##um', '##otho', '##ra', '##x'

Figure 2.5: Example of BERT's WordPiece tokenization on sample clinical text.

Transformers accept the tokenized input sequence and convert it to a set, which lacks inherent order. This contrasts with RNNs, which maintain sequence order through sequential processing. Thus, to provide positional information, Vaswani et al. introduced a sinusoidal positional encoding in the transformer architecture. As shown in Figure 2.3, positional encodings are applied to the tokenized embedding vector to provide positional information before the input vector is passed to the first self-attention layer (Vaswani et al. 2017). More recent work using alternative positional encoding methods have demonstrated improvements in model convergence and performance (Su et al. 2021).

2.5.3.3 Self-Attention in BERT

BERT and other BERT-based models consist of a series of self-attention layers (often referred to as multi-head self-attention); the base BERT model has 12 such layers. Each layer transforms the input using linear layers and applies attention to the sequence. The maximum input sequence length of the BERT model is 512 tokens, two of which are the special tokens `CLS` and `SEP`. The `CLS` token of the BERT model is a hidden state representation of the entire input sequence, which is of size 768 in the base BERT model (Devlin et al. 2019). This fixed-size vector representation of the input sequence can then be represented as a vector in downstream NLP tasks, such as text classification.

2.5.3.4 The Introduction of Efficient Transformers

The matrix operations underlying BERT model's local self-attention mechanism are of $\mathcal{O}(n^2)$ time and memory complexity (Figure 2.3). The computational complexity grows quadratically with the length of the sequence input, which places an upper-bound on the length of the input BERT can operate on. This functionally limits the model's ability to operate on longer sequences of text. Researchers have recently proposed new variations of the original transformer model to address the $\mathcal{O}(n^2)$ bottleneck. These models are commonly referred to as Efficient Transformer models (Figure 2.6).

Transformer-XL is one of the first transformer models to successfully address the issue of BERT's input sequence length limitation (Dai et al. 2019). Many more recent Efficient Transformer models overcome the $\mathcal{O}(n^2)$ bottleneck by using alternative attention mechanisms. The Reformer model relies on two different self-attention layers: local self-attention and Locality Sensitive Hashing self-attention layers (Kitaev, Kaiser, and Levskaya 2020). The Longformer model replaces the BERT self-attention mechanism with a local sliding window attention and a global attention mechanism (Beltagy, Peters, and Cohan 2020). For a comprehensive review of Efficient Transformers as of 2020, I refer the reader to an excellent survey by Tay et al. (Tay et al. 2020).

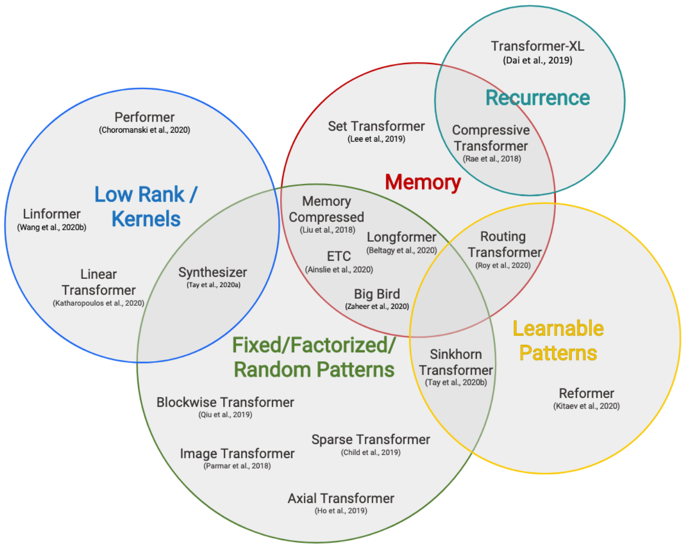


Figure 2.6: Taxonomy of Efficient Transformer Architectures (Source: Tay et al., 2020).

2.5.3.5 Limitations of Contextual Word Embeddings

One of the primary limitations of contextual word embeddings is their large size. There is also a growing trend of ever-increasing size of these large language models. Thus, working with CWE models requires access to graphics processing units (GPU) or tensor processing units (TPU) with sufficiently large memory capacity. Increasing CWE model size may also hamper the ability to share and distribute these models. Additionally, while transformer models have demonstrated SOTA or near-SOTA performance on a range of downstream NLP tasks, it is still unclear how these models encode linguistic information.

2.6 Intrinsic Evaluation

Quantitative evaluation of NLP models can generally be categorized into intrinsic and extrinsic evaluation methods. Intrinsic evaluation reflects the correlation between the algorithms and human judgment. This includes testing for syntactic or semantic relationships between words. While much emphasis in NLP-related research is on extrinsic evaluation of NLP methods, it is vital to conduct rigorous intrinsic evaluation. For instance, intrinsic evaluation using word vector analogies has highlighted gender, racial and religion-based biases in word embeddings trained using Google News or Reddit corpora (Bolukbasi et al. 2016; Manzini et al. 2019).

Commonly used approaches to collecting human judgment of word pair similarity or relatedness have relied on services such as Amazon Mechanical Turk or surveying those with domain proficiency, such as medical students, residents or clinicians in the clinical domain (Schnabel et al. 2015; Pedersen et al. 2007; Pakhomov et al. 2010; Ye and Fabbri 2018). Semantic relatedness or semantic similarity based upon human judgment is often computed as a correlation coefficient by comparing aggregate human determined scores to similarity metrics, such as cosine similarity, from word vectors. The models in greater agreement to human judgment are considered better models (Mikolov, Chen, et al. 2013; Baroni, Dinu, and Kruszewski 2014; Schnabel et al. 2015; Bakarov 2018).

2.6.1 Semantic Similarity

Semantic similarity as I use it here refers to the attributional similarity or synonymy of words; words that are synonyms have a high degree of semantic similarity.[1] Semantic similarity is considered a special case of the more general concept of semantic relatedness (Resnik 1995; Turney 2006). Humans may view terms or entities as being related despite them not being synonymous. Relatedness may be based upon features such as likeness, meronymy, antonymy, or by functional or frequent relationship (Turney 2006). For example, the terms *scalpel* and *surgeon* or *lung* and *nodule* are not synonyms but are related. The semantic relation between the words *lung* and *nodule* can be quantified using a measure of relation between these two words to provide a real-valued number, e.g. $\text{similarity}(w_{\text{lung}}, w_{\text{nodule}}) \in \mathbb{R}$. One way to assess similarity of terms or concepts is to ask humans to judge how similar the two are to one another.

2.6.2 Qualitative Evaluation

NLP practitioners working with topic models, such as the aforementioned LDA (Section 2.4.6), commonly use qualitative methods to evaluate the quality of model performance and judge their semantic properties. Qualitative evaluation alone may not be the best approach to judge the human interpretability of a model's latent space. Recognizing the limitations of qualitative evaluation methods, Chang et al. proposed quantitative methods to measure the semantic meaning of inferred topics against human judgment. The authors proposed two tasks to assess semantic coherence: word intrusion and topic intrusion. For each word, the top k nearest neighbor words was selected along with

[1]Semantic similarity also includes relational similarity, which reflects the relation between words. Words that are analogous to one another have a high degree of relational similarity (Turney 2006).

a bottom ranked word. Human annotators were then asked to identify the intruder word or topic from the random ordering of this set. Amazon Mechanical Turk was used to survey human judgment of word or topic coherence for each prompt (Chang et al. 2009).

As is seen in the topic model literature, many embedding model papers demonstrated the effectiveness of their methodology in capturing semantic relations by sharing qualitative examples of the top k nearest neighbors for select words. For example, Collobert and Weston created an embedding model using an English Wikipedia corpus and reported that for the word *France*, the top nearest neighbors were *Spain*, *Italy*, and *Russia* (Collobert and Weston 2008). As did Chang et al. for topic models, Mikolov et al. criticize the limitations of qualitative evaluation of embedding models. Alongside introducing the `word2vec` algorithm, they introduced a novel word analogy task to illustrate the semantic and syntactic properties of word2vec trained embeddings (Mikolov, Chen, et al. 2013).

2.6.3 Quantitative Evaluation

Several datasets within the linguistics literature assess human judgment of term relationships, many of which pre-date the introduction of word embedding methods. Construction of such datasets has previously relied on existing lexicons or knowledge graphs, corpora, or both (Turney 2006). Several existing lexicons and knowledge graphs describe relations that may be used for quantitative evaluation of concept relationships. Early examples using existing relationship taxonomies include the use of WordNet, Medical Subject Headings (MeSH), and Unified Medical Language System (UMLS) to quantitatively assess concept relations (Resnik 1995; Rosario and Hearst 2001; Rosario and Hearst 2004). Corpus-based approaches are based on the distributional hypothesis, i.e. semantically similar words are thought to have similar distributional behavior within a corpus (Resnik 1995).

Often in the context of NLP methods, intrinsic evaluation involves judging the similarity or relatedness of pre-selected word pairs. Semantic similarity and semantic relatedness are linguistically distinct concepts. Similarity is a quantitative measure of how alike two concepts are. For example, *artery* and *vein* are not synonyms, but both terms are similar in that they are blood-carrying vessels. Relatedness of terms can also be quantified, but this is a more general concept of association. For example, *scalpel* and *surgeon* are not synonyms, but are certainly related concepts (Pedersen et al. 2007; Jurafsky and Martin 2020).

Calculating similarity between two words (w_1, w_2) for non-contextual embeddings can easily be computed using each words embedding vector representation (θ_1, θ_2). This is most commonly calculated using cosine similarity. Human judgment of term similarity is commonly calculated using ordinal-scale surveys. A correlation coefficient is then computed using the calculated cosine similarity and human assessment values.

$$\text{cosine similarity}(w_1, w_2) = \frac{\theta_1 \cdot \theta_2}{\parallel \theta_1 \parallel \parallel \theta_2 \parallel}$$

In the general NLP domain, many similarity benchmarks exist, including WordSim-353, the MEN dataset, and SimLex-999 (Finkelstein et al. 2001; Agirre et al. 2009; Bruni, Tran, and Baroni 2014; Hill, Reichart, and Korhonen 2015). Additional intrinsic evaluation tasks have also been proposed. The use of pre-specified analogies is a popular approach for intrinsic evaluation of embedding models (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013; Bolukbasi et al. 2016; Manzini et al. 2019). Schnabel et al. proposed the comparative intrinsic evaluation task and also recommend other methods for intrinsic evaluation of word embedding models: relatedness, analogy, categorization and selectional preference (Schnabel et al. 2015). Ye and Fabbri use a similar design to the comparative intrinsic evaluation task for clinical NLP assessment (Ye and Fabbri 2018).

Contextual word embedding (CWE) models, such as BERT, use a different training objective than NCWEs. Specifically, BERT-based models include a masked-language modeling (MLM) pre-training strategy as shown in Figure 2.4 (Devlin et al. 2019). Given the different training objective of CWEs, Devlin suggests that these models may not provide meaningful cosine similarity measurements ("BERT Vector Space Shows Issues with Unknown Words · Issue #164 · Google-Research/Bert," n.d.).

Given this limitation, researchers have proposed intrinsic evaluation tasks in keeping with the BERT modeling approach. Goldberg and Ettinger respectively evaluate BERT models using previously published cloze ('fill-in-the-blank') tasks and compare the 'gold-standard' masked word with BERT word predictions (Goldberg 2019; Ettinger 2020). An example of a cloze task prompt used to study BERT-based models and survey participants is shown in Figure 2.7.

Clinical Intrinsic Evaluation tasks

There are two frequently used methods to produce a set of clinically similar terms: existing ontologies and EMR-based embeddings (Ye and Fabbri 2018). Previous studies

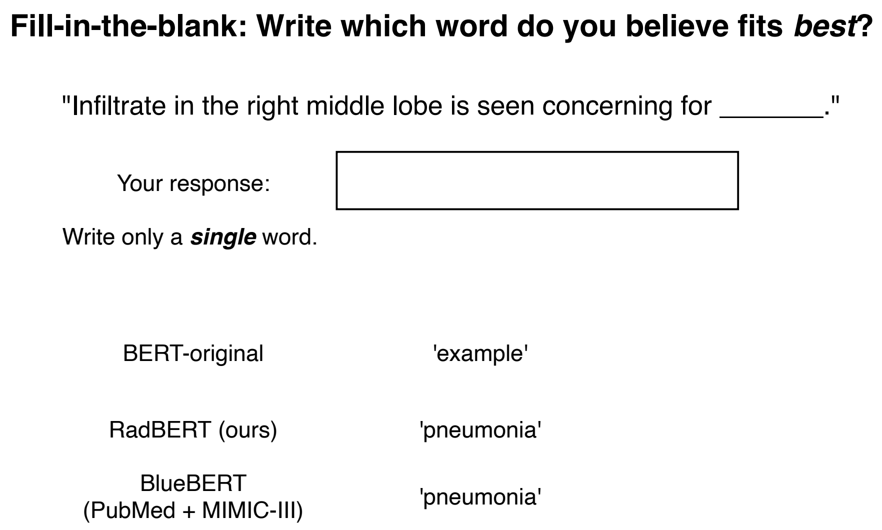

Fill-in-the-blank: Write which word do you believe fits *best*?
"Infiltrate in the right middle lobe is seen concerning for _______."
Your response:
Write only a *single* word.
BERT-original
'example'
RadBERT (ours)
'pneumonia'
BlueBERT (PubMed + MIMIC-III)
'pneumonia'

Figure 2.7: Example cloze prompt. Predictions from BERT and BERT-based models are illustrative and were not shown to survey participants.

have use knowledge-based measures of semantic similarity by leveraging existing clinical ontologies to quantify similarity. Existing clinical benchmarks comparing similarity or relatedness of clinical concepts include Hliaoutakis' work, MayoSRS, original and modified MNSRS (Pedersen et al. 2007; Hliaoutakis 2005; McInnes, Pedersen, and Pakhomov 2009; Pakhomov et al. 2010; Pakhomov et al. 2016) (See Table 2.1) . The concept pairs that comprise each of these benchmarks were manually curated by a physician from existing medical ontologies, such as Medical Subject Headings (MeSH) or Unified Medical Language System (UMLS).

Hliaoutakis examined the similarity of 36 clinical concepts that were drawn from the MeSH ontology by a physician. Evaluators were then asked to judge concept similarity on a 1-4 scale using an online survey (Hliaoutakis 2005).[2] Term pairs used to construct MayoSRS were also compiled by a physician. Human judgment for concept pairs in the MayoSRS study was assessed by surveying 13 medical coders on a 1-10 scale using a larger set of 120 pairs. Of the 120 term pairs, the 30 pairs with the highest agreement among the coders were then annotated by 9 medical coders and 3 rheumatologists on a 4 point scale (Pedersen et al. 2007). For UMNSRS, pairs of clinical concepts were collected from the Unified Medical Language System (UMLS) and then a physician manually selected single word terms to create concept pairs for each of six semantic type categories. Four medical residents were then surveyed to judge similarity of 566 concept pairs and another 4 medical residents assessed the relatedness of 587 pairs

[2]Hliaoutakis survey remains active and can be found at http://www.intelligence.tuc.gr/mesh/.

Table 2.1: Published clinical concept pairs assessed using human judgment.

Method	Metric	Data	Pair count
UMNSRS-Similarity	similarity	UMLS	566
UMNSRS-Relatedness	relatedness	UMLS	588
Modified UMNSRS-Sim	similarity	UMLS	449
Modified UMNSRS-Rel	relatedness	UMLS	458
MayoSRS	relatedness	SNOMED-CT	101
MiniMayoSRS (MayoSRS subset)	relatedness	SNOMED-CT	29/30
Hliaoutakis	similarity	MeSH	36

(Pakhomov et al. 2010).

To assess human judgment of concept pair similarity or relatedness, the researchers conducted surveys of medical coders or clinicians. Importantly, these were created before `word2vec` and subsequent non-contextual word embedding (NCWEs) models were introduced. These benchmarks are publicly available and have been used by others to study the effects of different training corpora or other training strategies (Sajadi 2014; TH, Sahu, and Anand 2015; Pakhomov et al. 2016; Z. Jiang et al. 2015; Yu et al. 2017; S. Jiang et al. 2020; Y. Wang et al. 2018; Mao and Fung 2020).

Few available clinical NLP benchmarks for CWEs exist. The Biomedical Language Understanding Evaluation (BLUE) benchmark contains 5 extrinsic evaluation tasks: sentence similarity, named entity recognition, relation extraction, document classification and inference (Peng, Yan, and Lu 2019). I was unable to find any publicly available intrinsic evaluation tasks designed to examine CWEs for the clinical or biomedical domains.

2.7 Extrinsic Evaluation

Intrinsic evaluation NLP benchmarks are scarce relative to those for extrinsic evaluation. The dearth of intrinsic evaluation benchmarks is likely due to the high cost of conducting surveys to obtain human judgments. Extrinsic evaluation of embedding models relates to the study of performance on any of a number of downstream NLP tasks. These tasks can include text classification, machine translation, named entity recognition, relation extraction, and question answering. The most significant cost towards developing extrinsic evaluation tasks is label annotation. This cost of labeling data for clinical tasks may be greater than in the general NLP domain as it may require highly specialized annotators.

Many extrinsic evaluation datasets are available for a variety of general domain NLP tasks. For example, publicly available text classification datasets include Amazon product reviews, IMDb movie reviews, and Reuters news articles; public named entity recognition datasets include CoNLL and Ontonotes. Additionally, General Language Understanding Evaluation benchmark (GLUE) and SuperGLUE are benchmarks that comprise a diverse set of general natural language understanding tasks (A. Wang et al. 2018, 2019). More recently, the Long Range Arena benchmark was introduced to evaluate and compare performance among Efficient Transformer models (Tay et al. 2021).

Much like GLUE serves for general NLP, Peng et al. developed Biomedical Language Understanding Evaluation (BLUE) benchmark (Peng, Yan, and Lu 2019). BLUE is comprised of a variety of biomedical and clinical NLP tasks. Of the 10 tasks in the BLUE benchmark set, only one evaluates document classification performance, and involves multi-label classification of PubMed abstracts for hallmarks of cancer (Baker et al. 2015).

Over the last several years, a series of clinical NLP challenges have been issued that may be used for extrinsic evaluation. Among these are the Shared Tasks for Challenges in NLP for Clinical Data by the National NLP Clinical Challenges (n2c2) and clinical or biomedical challenges from the Text REtrieval Conference (TREC) (Roberts, Voorhees, and Hersh, n.d.; Roberts et al., n.d.). The n2c2 (formerly i2b2) challenges included text de-identification, text classification, concept extraction, and assertion and relation classification tasks and often covered a range of clinical conditions (O. Uzuner, Luo, and Szolovits 2007; O. Uzuner et al. 2008; Ozlem Uzuner 2008; Ö. Uzuner et al. 2011; Stubbs et al. 2015; Stubbs, Kotfila, and Uzuner 2015). To compare select NCWE and CWE models, Si et al. use the i2b2 2010 and 2012 clinical concept extraction challenges (Si et al. 2019). Given varying desired clinical NLP objectives, the majority of clinical NLP papers performing extrinsic evaluation of NLP methods do so using custom datasets. For example, Gehrmann et al. created a custom extrinsic evaluation dataset for clinical phenotyping comprising 1,610 discharge summaries manually annotated for 10 different phenotypes. They then compared different NLP techniques on their phenotype classification performance (Gehrmann et al. 2018).

Clinical Phenotyping for Extrinsic Evaluation

Clinical phenotyping is an important application within clinical research, clinical trial recruitment, and population surveillance. Most clinical phenotyping studies involve the development of a curated dataset for identification of one or more phenotypes of interest. Shivade et al. conducted a literature review of clinical phenotyping methods and found that nearly 50% of studies relied on NLP (Shivade et al. 2014). Other phenotyping techniques included rule-based systems, statistical or machine learning-based approaches, and hybrid methods (Shivade et al. 2014; Zeng et al. 2019). They concluded that there are few robust tools or off-the-shelf models available for phenotyping (Shivade et al. 2014). This emphasizes the need to evaluate NLP models for phenotyping.

CHAPTER 3

INTRINSIC EVALUATION OF WORD EMBEDDING MODELS

Many clinical natural language processing methods rely on non-contextual word embedding (NCWE) or contextual word embedding (CWE) models.[1] Yet, few, if any, intrinsic evaluation benchmarks exist comparing embedding representations against clinician judgment. For a more detailed discussion on intrinsic evaluation, I refer the reader to Section 2.6. Below I share an approach to developing intrinsic evaluation tasks for embedding models using a corpus of radiology reports: term pair similarity for NCWEs and cloze task accuracy for CWEs. Using surveys, I collected clinicians' responses for each of these two tasks. I then quantified the agreement between clinician judgment against custom and publicly available embedding model representations.

3.1 Methods

3.1.1 Study Design

I curated a corpus of radiology reports from the Vanderbilt University Medical Center (VUMC) Research Derivative, an extract of the electronic health record from legacy and Epic record data, normalized to the OMOP common data model (Danciu et al. 2014). Radiology reports were selected from the earliest available record until October 21, 2020. I selected only those radiology reports with study descriptions corresponding to computed tomography (CT) scans inclusive of the chest using the 'Study Description' metadata. Candidate study descriptions containing comprehensive thoracic imaging were identified and confirmed with a VUMC radiologist. Study description codes used for note selection are listed in Table 3.1. Excluded study types included cardiac CT imaging that tend to lack full lung views and those considered to be unreliable, such as ones marked for billing. Report collection was limited to those for patients 18 years of age or older. This study was approved by the VUMC Institutional Review Board.

Text from the publicly available Fleischner Society Glossary of Terms for Thoracic Imaging published in 2008 was also collected to represent a corpus containing terms

[1]For more details about non-contextual word embedding models, please see Section 2.4. See Section 2.5.3 for a detailed discussion on contextual embedding models.

Table 3.1: List of CT Study Description codes with description used for inclusion.

Study Description Code	Exam Description
CT804B	CT READ RESEARCH
CTAC3	CT ANGIOGRAPHY CHEST
CTACAP1	CT ANGIO CHST/ABD/PEL W/WO CONTRAST
CTACAR	CT ANGIOGRAPHY CARDIAC
CTACG	CT ANGIO CHEST GATED
CTACOR	CTA Coronary Arteries Complete W or WO Cardiac Calcium Score
CTACPE	CT ANGIO CHEST PE PROTOCOL PEDS
CTCA2	CT CHEST W ABDOMEN W/WO CON
CTCAP	CT CHST/ABD/PEL W/IV CONTRAST
CTCAP2	CT CHEST W ABDOMEN PELVIS W/WO CON
CTCAP3	CT CHEST W ABDOMEN W/WO PELVIS W CON
CTCAPTLWO	CT CHST/ABD/PEL/TSP/LSP W/O C
CTCAPW	CT CHST/ABD/PEL/TSP/LSP WITH C
CTCAPWO	CT CHST/ABD/PEL WITHOUT CONTRAST
CTCARNC	CT Cardiac Non Coronary
CTCAW	CT CHEST AND ABDOMEN W CONTRAST
CTCAWO	CT CHEST AND ABDOMEN WO CONTRAST
CTCH1	CHEST CT WITHOUT CONTRAST
CTCH2	CHEST CT WITH CONTRAST
CTCH3	CHEST CT WITH/WITHOUT CONTRAST
CTCHEPE	CTA Chest W Embolus
CTCHR	CT CHEST HIGH RES WITHOUT CONT
CTLUNGSCR	CT LUNG SCREENING

more relevant within radiology reports (Hansell et al. 2008).

3.1.2 Text Pre-processing

Pre-processing of the raw clinical text included removal of unicode characters and HTML parsing to plain text using the Beautiful Soup library in Python. I concatenated hyphenated terms by converting '-' characters to '_', e.g. *status-post* was converted to *status_post*. Dates and times, age, medical record numbers, email addresses, phone numbers, social security numbers and location information were identified using the presidio Python package, a custom regular expression to handle numeric values of 7 or more digits, and the spaCy library tokenizer (*Microsoft Presidio* 2020; Honnibal et al. 2020). Identified terms were then replaced with the corresponding general token, e.g. `DATE_TIME`, `MRN`. Stop words for the radiology report corpus were not excluded. I pre-processed text from the Fleischner glossary using the same process above except that I also performed stop word exclusion using the spaCy English stop word vocabulary (Honnibal et al. 2020).

3.1.3 Non-contextual embedding models

Custom embeddings: I used the gensim library implementations of the `word2vec` and `fastText` models (Mikolov, Chen, et al. 2013; Bojanowski et al. 2017; Řehůřek and Sojka 2010). Joint spherical embedding models have recently been shown to generate embeddings with superior performance on general NLP benchmarks for word similarity, document clustering, and document classification. In this study, spherical embedding models were created using the source code provided by the authors (Meng et al. 2019). Each word2vec and fastText models were trained using the skip-gram method described in in Section 2.4.1 (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013).

Based on earlier studies detailed in Section 2.4.3, I constructed custom embedding models of 200 dimensions. I used similar training parameters for each custom embedding models: fixed dimension of 200, initial learning rate of 0.025, context window of 5, excluded words occurring fewer than 5 times, sampling threshold of 0.001, negative sampling rate of 5 and trained for 10 epochs.

Public embeddings: I used BioWordVec, which is a 200-dimensional fastText model trained using both a PubMed corpus and text from the Medical Information Mart for Intensive Care (MIMIC-III) dataset and made available by the authors (Y. Zhang

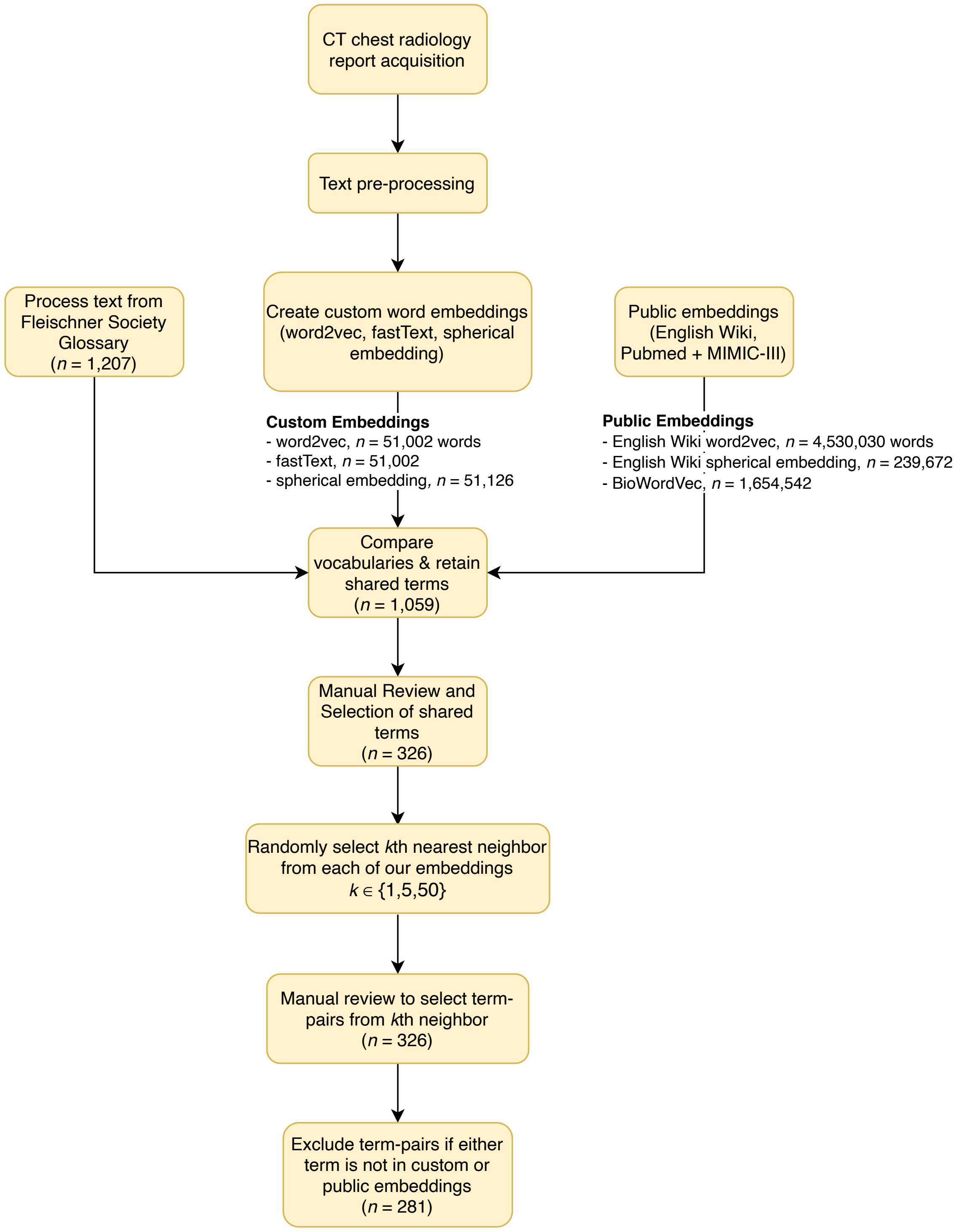

CT chest radiology report acquisition
Text pre-processing
Create custom word embeddings (word2vec, fastText, spherical embedding)
Process text from Fleischner Society Glossary (n = 1,207)
Public embeddings (English Wiki, Pubmed + MIMIC-III)
Custom Embeddings
- word2vec, n = 51,002 words
- fastText, n = 51,002
- spherical embedding, n = 51,126
Public Embeddings
- English Wiki word2vec, n = 4,530,030 words
- English Wiki spherical embedding, n = 239,672
- BioWordVec, n = 1,654,542
Compare vocabularies & retain shared terms (n = 1,059)
Manual Review and Selection of shared terms (n = 326)
Randomly select kth nearest neighbor from each of our embeddings k ∈ {1,5,50}
Manual review to select term-pairs from kth neighbor (n = 326)
Exclude term-pairs if either term is not in custom or public embeddings (n = 281)

Figure 3.1: Algorithm for data-driven generation of term pairs from a radiology report corpus using custom non-contextual embedding models.

et al. 2019).[2] Briefly, the PubMed corpus includes title, abstract, and article text data for approximately 30,000,000 open access articles at the time BioWordVec was created. The MIMIC-III database contains over 2 million clinical documents related to care of patients admitted to critical care units at a single academic medical center (Johnson et al. 2016). Additionally, I used a publicly available gensim `word2vec` model trained using the skip-gram method on the English Wikipedia corpus (February 2017) obtained from the University of Oslo's Nordic Language Processing Laboratory word vector repository (Fares et al. 2017). I also used a spherical embedding model trained on the English Wikipedia corpus provided by Meng et al (Meng et al. 2019).

3.1.4 Contextual embedding models

Custom embeddings: The pre-processed text of the radiology report corpus was exported to a txt file with each document per line. I used the transformers library to create a custom tokenizer and for pre-training on the radiology report corpus (Wolf et al. 2020). For pretraining our custom BERT-based model, referred to as RadBERT, I used the same configuration parameters as the original BERT model (Devlin et al. 2019). Pre-training the RadBERT model took approximately 2.5 days using two 12GB graphics processing units (GPU).

Public embeddings: I compared RadBERT to the original BERT model and a publicly available BERT-based model pre-trained using a PubMed and MIMIC-III corpus, BlueBERT (Devlin et al. 2019; Peng, Yan, and Lu 2019) . Each of these models was downloaded from the HuggingFace Models repository ("Hugging Face Models and Datasets," n.d.).

3.1.5 Term pair selection

I compared the pre-processed Fleischner Society Glossary tokens with the vocabulary of each of the 6 NCWE models. I retained only those terms shared between the Fleischner Society Glossary that were present in all 6 embedding vocabularies. The resulting 1,059 shared terms ('query words') were manually reviewed by a physician (MSK), and a subset of these terms (326 terms) were manually selected for possible inclusion in the survey. For each of the 326 query words, I randomly select the k^{th} most similar term using each of the 3 custom embeddings, where $k \in \{1, 5, 50\}$. This

[2]The original published BioWordVec model was trained using the PubMed corpus, but the more recent implementation is jointly trained on the PubMed corpus and MIMIC-III corpus and is publicly available for download at https://github.com/ncbi-nlp/BioSentVec.

Table 3.2: Example use of the kth most similar term from custom embeddings to inform term pair creation.

		Custom Embedding Models			
Query Word	k	word2vec	fastText	spherical embedding	Manual Review Selection
abscess	1	empyema	empyema	osteomyelitis	empyema
adenocarcinoma	50	peritonei	bronchoalveolar	status_post	bronchoalveolar

method was adapted from Schnabel et al. (Schnabel et al. 2015). As the examples in Table 3.2 show, for each query word, I retrieved 3 candidate terms. One of these three was then selected to construct a term pair. A physician (MSK) then manually reviewed the three candidate terms, i.e. the k^{th} most similar term to the query word for each of the custom embedding models. For each of the 326 query words, if 2 or 3 of the embedding generated candidate terms were similar, I selected this term to form a term pair. If all 3 of the proposed candidate terms was different, a candidate term was arbitrarily selected by MSK. As I used the custom embeddings to generate term pairs, some of the selected terms for each query word were not present in the publicly available embeddings. For accurate comparison and generation of correlation coefficients, I excluded those term pairs where the word was not present in each of the 6 embeddings being evaluated.

3.1.6 Cloze task generation

A physician (MSK) selected a random subset of radiology reports, from which representative selections of text were extracted and modified to reflect commonly seen radiographic descriptions and findings. A total of 20 cloze prompts were created for use in the survey. Each cloze task prompt was 1-3 sentences in length to provide sufficient context for human and models to identify the masked term. BERT-based models use WordPiece tokenization, which may generate sub-words (Devlin et al. 2019). Accurate comparison between the CWEs requires that the masked word be present in its complete word form as a token within each model's vocabulary, i.e. not as a subword (Goldberg 2019; Ettinger 2020). For example, using the BERT tokenizer and our custom tokenizer, *apical* was tokenized to (*apical*) for both. Tokenizing the word *bullae* with the BERT tokenizer yielded sub-words, (*bull*, *##ae*). Using our custom tokenizer, *bullae* was tokenized to *bullae*. In this example, *apical* would be considered eligible for masking in our cloze test because the full word is retained by each of our tokenizers. *Bullae* would be ineligible because one or more of the tokenizers returns sub-words rather than the complete word. Each cloze task prompt was tokenized with each of the 3 BERT-based models being studied using the transformers library (Wolf et al. 2020). Only overlapping complete word forms from each model were selected for masking.

3.1.7 Survey administration

I created and administered a survey of 281 term pairs and 20 cloze task prompts using the Research Electronic Data Capture (REDCap) tool hosted at VUMC (Harris et al. 2009). A convenience sample of 15 healthcare trainees or professionals were asked to participate in the survey. The instructions provided to participants was adapted from another concept similarity survey, WordSim353 (Finkelstein et al. 2001). Study data were collected and managed in REDCap.

Human judgment of term pair similarity was assessed using a 7-point ordinal scale. For the cloze task portion, survey participants were asked to enter only a single word for each prompt for accurate comparison to BERT-based models because these models have single word vocabularies (Ettinger 2020).

3.1.8 Survey analysis

Upon completion, survey results were analyzed using R (version 4.0.3). For each term pair, the mean and standard deviation was calculated. Pearson and Spearman correlation coefficient values were calculated with the gensim library using the mean values from the survey and cosine similarity for each NCWE (Řehůřek and Sojka 2010). To evaluate these models in relation to existing benchmarks and published results, I also calculated Pearson and Spearman correlation coefficients using the original and modified versions of the UMNSRS-similarity and UMNSRS-relatedness benchmarks introduced above (Section 2.6.3) (Pakhomov et al. 2010; Pakhomov et al. 2016). If either concept in a concept pair was not present in the embedding model vocabulary, that concept pair was not used to calculate the correlation coefficient.

To evaluate CWE models, I composed 20 fill-in-the-blank prompts that reflect text that may appear in a CT chest radiology report, e.g. “Infiltrate in the right middle lobe is seen concerning for ___.” To assess human judgment, survey participants were asked to input free-text for what they determined to be the most likely single word. Survey results were manually reviewed and tabulated to construct a list of the words provided for each cloze task prompt ordered by frequency. Each of the CWE models were then used to also predict the expected word for each cloze task. For word prediction accuracy, I designated the word entered with the highest frequency from the survey as the "expected" word. Accuracy was defined as the percentage of items for which the "expected" word is among the CWE model's top k predictions for $k \in \{1, 5\}$ (Ettinger 2020). I then compared the accuracy for each of the CWEs. I used the transformers library to find the top k predicted words by also providing special tokens as detailed by Goldberg (Goldberg 2019; Ettinger 2020).

3.2 Results

The VUMC radiology report corpus contained 479,850 documents and comprised a total of 124,892,727 tokens. These documents included other associated clinical documents, including critical result messages. For the NCWEs, custom models developed using the `word2vec` and `fastText` methods each had a vocabulary of 51,002 words, whereas the custom spherical embedding model vocabulary comprised 51,126 words. The English Wikipedia word2vec, English Wikipedia spherical embedding, and BioWordVec embedding models comprised of 4,530,030, 239,672, and 1,654,542 words, respectively.

3.2.1 Comparison to Fleischner glossary

Using exact matching, the overlap between each of the 3 custom embedding models and words from the Fleischner glossary was 1,207 distinct words. The intersection between the Fleischner glossary vocabulary and BioWordVec, word2vec (Wikipedia) and spherical embedding (Wikipedia) models was 1,272, 1,138 and 1,124 words, respectively. The overlap of the embedding model vocabularies with the Fleischner glossary yielded a shared 1,059 words (Figure 3.1).

3.2.2 Non-contextual embedding model similarity

The cosine similarity distribution using the described term pair selection approach returned an approximately normal distribution for the custom word2vec and fastText models and BioWordVec centered near 0.5. The custom spherical embedding model and the public models trained using an English Wikipedia corpus were skewed (Figure 3.2).

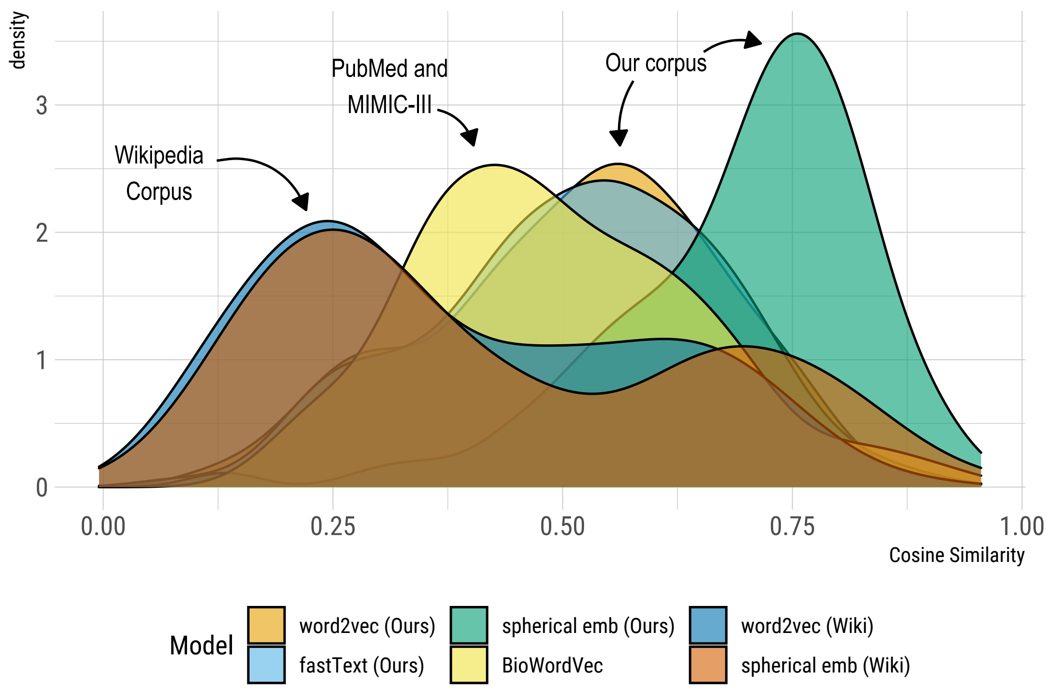


Figure 3.2: Distribution of cosine similarity scores on RadSim281. Custom fastText and word2vec models and BioWordVec are near-normal, whereas Wikipedia-trained models and custom spherical embeddings are skewed.

Table 3.3: Performance of the 6 embedding models on our custom term pair similarity benchmark, RadSim281.

Model	Pearson coefficient	Spearman coefficient
word2vec (ours)	0.34	0.35
fastText (ours)	0.40	0.41
spherical emb (ours)	0.40	0.44
BioWordVec	0.60	0.61
word2vec (Wiki)	0.31	0.32
spherical emb (Wiki)	0.26	0.27

3.2.3 Survey participant characteristics

A total of 13 participants completed the survey. Of these 13, 9 identified as attending physicians. The remaining identified as a medical student (1), resident (1), fellow (1), nurse (1) and advanced practice provider (1). Clinical background for the resident, fellow and attending physicians included anesthesiology (1), emergency medicine (1), family medicine (1), internal medicine (5) and radiology (1). The number of years in practice post-residency for the fellow and attending physicians was a median of 3 years. Participants reported currently practicing or training in different regions within the US: Northeast (1), Midwest (7), South (4) and West (1).

3.2.4 Term pair similarity

The computed Pearson correlation coefficient (ρ_p) and Spearman correlation coefficient (ρ_s) values comparing mean survey similarity scores to cosine similarity for each model are shown in Table 3.3. Of each of the 6 models studied, BioWordVec yielded the highest ρ_p and ρ_s of 0.60 and 0.61, respectively. Among the embedding models trained using our radiology corpus, the word2vec model returned the lowest correlation between human and model similarity scores (ρ_p 0.34; ρ_s 0.35). ρ_p and ρ_s values from our custom fastText model were 0.40 and 0.41; for our spherical embedding model, ρ_p and ρ_s were calculated as 0.40 and 0.44, respectively. For the English Wikipedia trained word2vec model, ρ_p and ρ_s were 0.31 and 0.32, respectively; the spherical embedding Wikipedia model returned a ρ_p of 0.26 and ρ_s of 0.27.

Table 3.4: Performance of the 6 embedding models on the original UMNSRS (Pakhomov et al., 2010) and modified UMNSRS (Pakhomov et al., 2016) benchmarks.

	Original UMNSRS			Modified UMNSRS		
Model	Pearson	Spearman	% OOV	Pearson	Spearman	% OOV
Similarity						
word2vec (ours)	0.46	0.44	61.66	0.46	0.43	51.45
fastText (ours)	0.52	0.49	61.66	0.52	0.49	51.45
spherical emb (ours)	0.35	0.34	61.66	0.37	0.35	51.45
BioWordVec	0.64	0.62	17.84	0.65	0.62	3.34
word2vec (Wiki)	0.38	0.38	60.07	0.39	0.39	48.11
spherical emb (Wiki)	0.28	0.27	59.89	0.30	0.29	47.44
Relatedness						
word2vec (ours)	0.34	0.34	64.05	0.34	0.34	53.28
fastText (ours)	0.41	0.40	64.05	0.41	0.39	53.28
spherical emb (ours)	0.34	0.32	64.05	0.35	0.34	53.28
BioWordVec	0.57	0.57	20.27	0.57	0.57	3.93
word2vec (Wiki)	0.36	0.35	61.84	0.37	0.36	48.25
spherical emb (Wiki)	0.29	0.27	61.50	0.33	0.31	47.38

Note: % OOV is the percentage of out-of-vocabulary words; excluded for correlation calculation.

Table 3.5: Comparison of word2vec model performance between ours (skip-gram) and Pakhomov et al., 2016 (CBOW) on the modified UMNSRS Similarity and Relatedness benchmark.

	Similarity		Relatedness	
Corpus Source	Pakhomov et al.	Ours	Pakhomov et al.	Ours
Clinical text	0.60	0.43	0.57	0.34
Wikipedia	0.48	0.39	0.45	0.36
PubMed corpus	0.62	NA	0.58	NA

3.2.5 Performance on UMNSRS benchmarks

ρ_p and ρ_s results from each of our 6 embedding models on the original and modified forms of UMNSRS-Similarity and UMNSRS-Relatedness are shown in Table 3.4. For comparison of word2vec model performance on different training corpora, I present previously published algorithm performance on the modified UMNSRS benchmarks relative to our own findings in Table 3.5 (Pakhomov et al. 2016).

3.2.6 Cloze task accuracy

RadBERT, our model pre-trained on a radiology report corpus, and BlueBERT each have Top-1 and Top-5 accuracy of 85% and 95%, respectively; the original BERT

Table 3.6: Top-1 and Top-5 cloze task accuracy for each masked language model.

	Accuracy (%)	
Language Model	Top-1	Top-5
BERT-original	25	30
BlueBERT	85	95
RadBERT (ours)	85	95

model had 25% Top-1 and 30% Top-5 accuracy (Table 3.6).

3.3 Discussion

The described term pair creation approach yields a near normal distribution of cosine similarity scores for the custom word2vec and fastText models and the BioWordVec model. This supports the ability of this approach to capture a normally distributed breadth of similarity scores using word2vec or fastText models trained on biomedical text. I also found that models trained using an English Wikipedia corpus are right skewed in Fig. 3.2. Additionally, these models consistently performed least well in relation to those models trained using a domain-specific corpus on the RadSim281, original UMNSRS and modified UMNSRS term similarity benchmarks. These findings suggest poor agreement between models trained using general corpora and clinical judgment.

To my knowledge, this analysis is the first to use spherical embeddings trained using clinical text documents. In contrast to the excellent performance of spherical embeddings reported on general NLP text similarity benchmarks, I found that spherical embeddings often performed less well than other models (Table 3.3 and 3.4) (Meng et al. 2019). The only exception being that the custom spherical embedding model performed better than the custom word2vec model and at least as well as the custom fastText model on RadSim281. The spherical embedding model proposed by Meng et al. uses directional similarity and jointly learns word and paragraph embeddings. This has been shown to provide improved word similarity and document clustering over other text embedding methods by leveraging both word-word and word-paragraph co-occurrence information. Yet, relative to other text documents, clinical text often contains disparate information in neighboring paragraphs, and this may be an ill-suited substrate for this training strategy. This may cause the model to consider truly dissimilar words as being somewhat similar and may explain the left skew of cosine similarity scores from the spherical embedding model relative to the other 5 models as shown in Fig. 3.2. Additional study is required to attempt to replicate the benefits of spherical embedding models for document clustering or document classification in the clinical domain.

Among the custom NCWE models, the fastText model performed about as well as the spherical embedding model on RadSim281 (Table 3.3) and achieves the highest ρ_p and ρ_s on the original and modified UMNSRS benchmarks (Table 3.4). One of the primary benefits of fastText embedding models is that each word can be represented as a bag of character n-grams, which gives these models the ability to handle out-of-vocabulary words and generalize better. Bojanowski et al. assess the correlation between human judgment and cosine similarity comparing word2vec and fastText models with and without sub-word information using 10 different benchmarks covering 7 different languages. Similar to my own findings, they found that fastText models tend to outperform word2vec models on most term pair similarity benchmarks including the English Rare Word dataset (Bojanowski et al. 2017).

I reported both Pearson and Spearman correlation coefficients for comparison to earlier studies, some of which provide only the ρ_p and others only the ρ_s. The originally proposed BioWordVec fastText model was trained using PubMed and Medical Subject Headings (MeSH) (Y. Zhang et al. 2019). I used the publicly available BioWordVec model trained on PubMed and MIMIC-III corpora. On the UMNSRS benchmarks, I found similar, but slightly lower, ρ_p and ρ_s values compared to the published values (Table 3.4). Pakhomov et al. evaluate performance on the modified UMNSRS benchmarks using a continuous bag of words (CBOW) word2vec model. Results of their analyses using different training corpora in comparison to our own are shown in Table 3.5. Our findings for ρ_s using our skip-gram word2vec model and the spherical embedding model trained using an English Wikipedia corpus are much lower on both the UMNSRS-similarity and UMNSRS-relatedness benchmarks compared to their findings (Table 3.4). Yet, our results from the BioWordVec model are comparable to the ρ_s values they computed using model trained on a PubMed corpus. Although I used the skip-gram training approach and they used the CBOW method, others have found that the skip-gram method provides better or equivalent performance on the UMNSRS benchmarks (Sajadi 2014; Chiu et al. 2016). Thus, this discrepancy is less likely explained by the use of the skip-gram modeling technique instead of CBOW.

For models trained on the VUMC radiology report corpus or Wikipedia, I was unable to account for one or both concepts for approximately 50% of the modified UMNSRS concept pairs (Table 3.4). Given that half of these embedding vocabularies are not adequately represented among the concept pairs, our results are likely suffering from bias. This highlights that intrinsic evaluation performance can suffer if a large proportion of the term pair vocabulary is not present within the embedding vocabulary. This discrepancy may also reflect sensitivity of term pair similarity on training parameter selection between the studies. Prior studies show a tradeoff between intrinsic and extrinsic evaluation based upon model and hyperparameter choice. Agirre et al. find that using a context window approach yields better results on similarity tasks, whereas embeddings trained using their bag of words method performed better on relatedness tasks. They also find that increasing the context window size improves correlation between model and human judgment (Agirre et al. 2009). Moreover, in the biomedical domain, Chiu et al. find that larger context windows lead to gains in intrinsic evaluation measures with decreased performance on certain downstream tasks (Chiu et al. 2016). Thus, one must be aware of this and select architectures and parameters best suited for the desired objective.

NCWE models can also be augmented with knowledge graphs, which have been shown to lead to better representations (Sajadi 2014; Yu et al. 2017; Boag and Kané 2017; Banerjee et al. 2017; Banerjee et al. 2018). I attempted to enrich the embedding models using the RadLex ontology, but was met with little success as I found limited RadLex term coverage in our corpus, which is a challenge noted previously (Percha et al. 2018). This difficulty may be offset with expansion of the RadLex terminology or by using other existing ontologies that provide improved term coverage.

To my knowledge, this study is the first attempt in clinical NLP towards developing intrinsic evaluation benchmarks for CWEs. I found that the original BERT model performs poorly on cloze tasks that reflect radiology report text. I also show equivalent top-1 and top-5 cloze task accuracy between BlueBERT and our custom BERT model (Table 3.6). These findings appear to suggest that the publicly available BlueBERT model pre-trained using the PubMed and MIMIC-III corpora perform well even in comparison to a model pre-trained using a targeted domain corpus of radiology reports from which the cloze task prompts were established. Further study of the generalizability of BlueBERT on other intrinsic evaluation tasks is required. The implication of a generalizable clinical BERT-based model may mitigate the need to pre-train custom BERT-based models for different clinical purposes. This becomes especially important given the sizable financial and environmental costs incurred by language model training (Strubell, Ganesh, and McCallum 2019). One of the limitations of the BlueBERT model is that it uses the original BERT model vocabulary, rather than training a custom tokenizer using the PubMed and MIMIC-III corpora (Peng, Yan, and Lu 2019).

BERT-based models can only accept a maximum sequence input length of 512 tokens (Devlin et al. 2019). Many clinical terms are not present within the original BERT vocabulary and are thus tokenized to subwords for model input. This restricts the number of words that may be used as input for the original BERT and BlueBERT models, whereas a custom tokenizer would have helped mitigate this potential for inform loss to some degree.

While I did find that models trained using the MIMIC-III corpus (BioWordVec and BlueBERT) perform admirably relative to their counterparts, the MIMIC-III corpus reflects an intensive care unit (ICU) patient population and may present an additional source of bias worth consideration (Johnson et al. 2016). The ICU patient population tends to be of higher acuity. Moreover, the conditions managed in the ICU and the document layout and language used by ICU providers may be distinct and not representative of non-ICU care or clinical documents. The limitations of this narrow clinical frame may be remedied by the additional inclusion of the PubMed corpus (PMC) for BioWordVec and BlueBERT training. However, the PMC language is arguably more technical and scientific than what is typically seen within clinical documents, which presents yet another limitation and potential source of bias. Additionally, the BioWordVec model is much larger than each of our custom embeddings, which may preclude its use for those without access to sufficient computer memory and disk resources.

It remains unclear if a tradeoff between intrinsic and extrinsic evaluation performance exists for CWEs as is reported in NCWEs. Moreover, further study is required to determine if and to what degree the differences in cloze task performance persist after model fine-tuning.

Few, if any, benchmarks exist for intrinsic evaluation of CWE and NCWE models in the clinical domain. Given the potential impact these and other models may have on medical decision making, it is vital to probe models to identify potential flaws and biases. Furthermore, we must be cognizant of existing biases inherent within the clinical data used to train these models. If our models encode biased representations, their deployment may propagate these biases forward and maintain or exacerbate existing healthcare disparities. Probing of models trained on general corpora using intrinsic evaluation methods have unearthed gender, racial and religious biases (Bolukbasi et al. 2016; Manzini et al. 2019). With limited model probing tasks within clinical NLP, I contend that this area requires further study especially given the evidence of racial and other disparities present within clinical documentation, management and outcomes (Balderston et al. 2021).

Among this study's limitations is that I restricted intrinsic evaluation tasks for NCWE and CWE models to single words. Future studies may use methods to generate multi-word terms while still allowing for comparison within the embedding space (Mikolov, Sutskever, et al. 2013; Rose et al. 2010; Campos et al. 2018). Moreover, the term pair generating method excluded term pairs that were not present in all of the embedding vocabularies. Most terms were excluded because they were not in the NCWE vocabularies trained on a Wikipedia corpus. This design choice may have biased term pair selection away from relevant medical terms and towards common English words. Another limitation is that radiology report corpus was primarily limited to CT scans inclusive of the chest. One may find improved results using a corpus of radiology reports from all imaging modalities. Likewise, the corpus was curated from a single academic medical center, which limits the size of the corpus and influences the conditions and findings identified in reports to those found within the region.

3.4 Conclusion

This work introduces two new intrinsic evaluation methods for use among clinical NLP researchers: term pair similarity to compare NCWEs and cloze task accuracy for CWE models. For NCWEs trained on a domain-specific corpus, these results highlight that fastText models tend to outperform word2vec and spherical embedding models. I also demonstrate that the gains afforded by spherical embedding models in general NLP intrinsic evaluation tasks fail to translate to the clinical domain. This emphasizes the need for caution and rigorous evaluation prior to adoption of methods that may excel in general NLP tasks. Importantly, I found that embedding models trained using PubMed and MIMIC-III corpora - BioWordVec and BlueBERT - perform at least as well and often better than models trained on a targeted domain corpus on intrinsic evaluation tasks. This provides additional evidence in support of these biomedical corpora capturing word representations in agreement with clinician judgment. Further study is needed to the establish additional benchmarks and probing tasks. These will help to facilitate language model evaluation to assess for potential biases and determine agreement with clinician judgment. Such analyses are necessary to engender trust and promote understanding of NLP models for broader clinical adoption.

CHAPTER 4

EXTRINSIC EVALUATION OF WORD EMBEDDING MODELS

The previous chapter focused on intrinsic evaluation methods comparing word embedding representations against clinical judgment. In this chapter, I present extrinsic evaluation tasks for a clinical phenotyping objective. Using each of the contextual and non-contextual embeddings, I train binary and multi-label phenotype classification models and compare classification performance between them.

4.1 Methods

4.1.1 Study Design

I curated a corpus of radiology reports from the Vanderbilt University Medical Center (VUMC) Research Derivative, an extract of the electronic health record from legacy and Epic record data, normalized to the OMOP common data model (Danciu et al. 2014). Radiology reports were selected from the earliest available record until October 21, 2020. I selected only those radiology reports with study descriptions corresponding to computed tomography (CT) scans inclusive of the chest using the 'Study Description'. Candidate study descriptions containing comprehensive thoracic imaging were then identified and confirmed with a VUMC radiologist. Study description codes used for note selection are listed in Table 3.1. Excluded films included cardiac CT imaging that tend to lack full lung views and those considered to be unreliable, such as ones marked for billing. Report collection was limited to those for patients 18 years of age or older. This study was approved by the VUMC Institutional Review Board.

4.1.2 Text processing

Pre-processing of the raw clinical text included removal of unicode characters and HTML parsing to plain text using the Beautiful Soup library in Python. I concatenated hyphenated terms by converting '-' characters to '_', e.g. *status-post* was converted to *status_post*. Dates and times, age, medical record numbers, email addresses, phone numbers, social security numbers and location information were identified using the presidio Python package, a custom regular expression to handle numeric values of 7 or more digits, and the spaCy library tokenizer (*Microsoft Presidio* 2020; Honnibal et al. 2020). Identified terms were then replaced with the corresponding general token, e.g. `DATE_TIME`, `MRN`. Stop words for our radiology report corpus were not excluded. Processed text was stored using the Apache Parquet file format (Apache Software Foundation 2020).

4.1.3 Non-contextual embedding models

Custom embeddings: I used the gensim library implementations of the `word2vec` and `fastText` models (Mikolov, Chen, et al. 2013; Bojanowski et al. 2017; Řehůřek and Sojka 2010). Joint spherical embedding models have recently been shown to generate embeddings with superior performance on general NLP benchmarks for word similarity, document clustering, and document classification. In this study, spherical embedding models were created using the source code provided by the authors (Meng et al. 2019). Each word2vec and fastText model was trained using the skip-gram method descrived in Section 2.4.1 (Mikolov, Chen, et al. 2013; Mikolov, Sutskever, et al. 2013). I used similar training parameters for each of our custom embedding models: fixed dimension of 200, initial learning rate of 0.025, context window of 5, excluded words occurring fewer than 5 times, sampling threshold of 0.001, negative sampling rate of 5 and trained for 10 epochs.

Public embeddings: I used BioWordVec, which is a 200-dimensional fastText model trained using both a PubMed corpus and text from the Medical Information Mart for Intensive Care (MIMIC-III) dataset and made available by the authors (Y. Zhang et al. 2019).[1] Briefly, the PubMed corpus includes title, abstract, and article text data for approximately 30,000,000 open access articles at the time the BioWordVec model was published. The MIMIC-III database contains over 2 million clinical documents related to care of patients admitted to critical care units at a single academic medical center (Johnson et al. 2016). Additionally, I used a publicly available gensim word2vec model trained using the skip-gram method on the English Wikipedia corpus (February 2017) obtained from the University of Oslo's Nordic Language Processing Laboratory word vector repository (Fares et al. 2017). I also used a spherical embedding model trained on the English Wikipedia corpus provided by Meng et al (Meng et al. 2019).

[1]The original published BioWordVec model was trained using the PubMed corpus, but the more recent implementation is jointly trained on the PubMed corpus and MIMIC-III corpus and is publicly available for download at https://github.com/ncbi-nlp/BioSentVec.

4.1.4 Contextual embedding models

Custom embeddings: The pre-processed text of the radiology report corpus was exported to a txt file with each document per line. I used the transformers library to create a custom tokenizer and for pre-training on the radiology report corpus (Wolf et al. 2020). For pretraining our custom BERT-based model, referred to as RadBERT, I used the same configuration parameters as the original BERT model (Devlin et al. 2019). Pre-training the RadBERT model took approximately 2.5 days using two 12GB graphics processing units (GPU).

Public embeddings: I compared RadBERT to the original BERT model and a publicly available BERT-based model pre-trained using a PubMed and MIMIC-III corpus, BlueBERT (Devlin et al. 2019; Peng, Yan, and Lu 2019) . Each of these models was downloaded from the HuggingFace Models repository ("Hugging Face Models and Datasets," n.d.).

Efficient Transformers: Each of the three CWE models studied, base BERT, BlueBERT, and RadBERT, was converted to the Longformer model architecture using modified code provided by the Longformer author. The Longformer model maximum input length was set to 4,096 tokens by copying and extending each BERT-based models original position embeddings. Unlike the RoBERTa model used by the Longformer authors, I began with BERT-based models. In the RoBERTa model, the first two position embeddings are reserved and need to be accounted for when generating new positional embeddings from the original RoBERTa-based models. The 0- and 1-indexed spaces are not reserved in BERT-based models, and I modified the authors code to account for this difference. The Longformer model architecture uses two attention mechanisms: local windowed attention and global attention. I converted each BERT-based model's self-attention layers to Longformer self-attention. Also, for each model layer, I copied the Longformer self-attention to create the global attention component as described by the authors (Beltagy, Peters, and Cohan 2020). This process was compiled as a Python script, which I ran to convert each of the BERT-based models to their Longformer equivalent.

4.1.5 Manual Document Annotation

For annotation, I used the active learning-based annotation tool Prodigy, which is authored by the developers of the spaCy NLP library (Montani and Honnibal 2018). After text processing, our corpus was converted from CSV to the JSONL format to facilitate capture of manual annotation and related metadata as recommended by the Prodigy tool authors.

CT scan report classification: In addition to CT scan reports, our corpus included other clinical documents, such as critical alert messages and chest X-ray reports. I wished to limit the scope of our annotation to only CT scan reports. To do so, I developed a binary classification model to detect CT scan reports. I loaded our complete corpus into Prodigy, which stores this information and annotation data in a SQLite database. I annotated the first 2,050 clinical documents presented using the Prodigy annotation interface and active learning model. Options for annotation include *accept*, *reject*, or *ignore*. This annotated dataset was then used to train a classification model initialized using the blank English spaCy model. I used the Prodigy command-line text classification training function; 1,640 of our 2,050 annotated documents were used for training and 410 for evaluation.

Sampling of corpus for annotation: I loaded our CT report classification model using spaCy and applied it to all documents in our corpus (Honnibal et al. 2020). Model predictions took a real-valued number from 0 to 1. I retained only those documents with predictions > 0.75 as being a CT scan report to be included for manual annotation. Using the `sample` method in the Pandas library, I selected 5,000 random notes using a random seed of 42. This sample of notes was then exported to a JSONL file for further annotation.

Term expansion: In addition to active learning, Prodigy provides additional functionality to accelerate annotation. I used our custom word2vec embeddings for term expansion using Prodigy's `terms.teach` command. For example, for *pneumothorax*, I used our custom word2vec embedding to find similar terms such as *pneumothoraces* and *hydropneumothorax*. This expanded list of terms was saved in a JSONL file and provided using the `patterns` flag to Prodigy's active learning-based annotation function `text.teach`.

Phenotype annotation: The 5,000 reports sampled for annotation were loaded into Prodigy, which stored the text and metadata in a new SQLite database. Using the Prodigy annotation interface, MSK completed binary labeling of the 5,000 notes for each phenotype of interest, i.e. each note was annotated at least once for the presence or absence of pneumothorax and again for the presence or absence of granulomatous disease.

As part of Prodigy's active learning process, some notes may be shown to the annotator on more than one occasion. This is especially true for notes where the active learning model is uncertain, e.g. when label predictions are near 0.5. After our annotations were complete, I exported this information from the Prodigy SQLite database to a JSONL file, which I loaded using the Pandas library for further processing. Given that some documents were annotated more than once, I performed de-duplication of annotations. Disagreement between annotations for documents with multiple annotations were re-examined and resolved by MSK.

4.1.6 Model Setup

For extrinsic evaluation of different contextual and non-contextual embedding models, I defined two phenotype classification tasks: binary classification and multi-label classification. Using each of the custom and public NCWE and CWE models included in this study, I developed models for binary and multi-label classification. The binary classification task evaluated for the presence or absence of *granulomatous disease*. Our multi-label classification setup looked for the presence or absence of either *granulomatous disease* or *pneumothorax*.

Cross validation: Each model in this study was trained using a 5-fold nested cross-validation (CV) procedure. Nested cross validation includes an outer loop and inner loop. In the outer loop, the data is fragmented into a test fold and non-test folds. These non-test folds are further split to training folds and a validation fold within the inner loop. Our training fold comprised of 3,000 documents and the validation and test folds consisted of 1,000 documents each.

For binary classification, I used the scikit-learn implementation of Stratified k-fold cross validation using the `StratifiedKFold` method (Pedregosa et al. 2011). For multi-label classification, I used an iterative stratification method described by Sechidis et al. using the `MultilabelStratifiedKFold` method in the iterative-stratification Python package (Sechidis, Tsoumakas, and Vlahavas 2011).

Tokenizing text for logistic regression and BiLSTM models: I use the keras `Tokenizer` method and fit this tokenizer on the training fold text (Chollet et al. 2015). This trained tokenizer was then used to tokenize text from the training, validation and test folds. The text of each document was tokenized to a maximum length of 4,096 tokens, and those documents with fewer tokens are zero-padded at the end. The exception to this is tokenization of our binary classification logistic regression model, which is tokenized using the whatlies toolkit due to ease of use and compatibility with scikit-learn (See Section 4.1.7).

Training details: Model development and training were performed using Python 3.7.2 (Van Rossum and Drake 2009). Neural network models were trained using PyTorch 1.7.1. Each of our binary and multi-label classification models were trained using binary cross entropy with logits loss implemented in PyTorch (Paszke et al. 2019). The exception being our binary classification logistic regression model, which was trained using scikit-learn (See Section 4.1.7). Unless otherwise stated, I trained using mixed precision training to reduce memory consumption and improve training speed (Micikevicius et al. 2018).

Metrics: For binary classification tasks, I computed accuracy, precision, recall, and F1-score. Multi-label classification metrics reported are accuracy and row-wise micro-averaged precision, recall, and F1-score; I also included the macro-averaged F1-score for comparison. Each of our metrics was computed using the scikit-learn methods to calculate accuracy and precision, recall and F1-score (Pedregosa et al. 2011).

4.1.7 Logistic Regression Models

Binary classification: The whatlies toolkit makes it easy to integrate use of word embedding models with different APIs, including spaCy and scikit-learn (Warmerdam, Kober, and Tatman 2020). Interfacing NCWE models with the scikit-learn modeling API using whatlies requires that the NCWE models be in a spaCy embedding model format. Each of the six NCWE models studied were originally saved in the word2vec file format. Thus, I converted each of these to the spaCy format using spaCy's command-line function `init-model` (Honnibal et al. 2020). The whatlies `SpacyLanguage` method was used to process the text, which it does by using the embedding model as a look-up table for each word in the document and returning the summed representation of the individual token vectors (Warmerdam, Kober, and Tatman 2020). In other words, each document is distilled into a summed, fixed-vector representation, which is then used by scikit-learn's `LogisticRegression` method for model training using the default settings. Importantly, the default scikit-learn implementation of logistic regression uses ℓ_2 regularization, or weight decay.

I used grid-search for hyperparameter optimization in the inner loop of our nested CV procedure over the following parameters: ℓ_2 regularization or no regularization; C parameter over the log scale from 10^{-4} to 10^4.

Multi-label classification: The scikit-learn `LogisticRegression` method does not allow for multi-label classification setups. Thus, I implemented our multi-label logistic regression model using PyTorch. Our NCWE models were loaded using the gensim package and processed to generate an embedding matrix (Řehůřek and Sojka 2010). I tokenized the input text as described in Section 4.1.6. Our logistic regression model architecture begins with an embedding layer using the PyTorch `Embedding` method. Our embedding layer weights are frozen, and are not updated during training. The tokenized text is passed to the embedding layer, which serves as a look-up table for each token in a given document. For each document, the average of all its individual token vectors is returned. This mean vector representation is then passed to a linear transformation layer. For training this model, I used the Limited-memory Broyden–Fletcher–Goldfarb–Shanno (LBFGS) optimizer implemented in PyTorch. I evaluated model performance using other optimizers (SGD, Adam, AdamW) with and without weight decay, but used LBFGS as this gave the best results during evaluation.

4.1.8 Bidirectional LSTM Models

Binary classification: Each NCWE model was loaded using the gensim package to construct an embedding matrix (Řehůřek and Sojka 2010). The text was tokenized as described in Section 4.1.6. The initial layer of our Bidirectional LSTM (BiLSTM) model is the embedding layer, a look-up table for each token in a tokenized document. The embedding layer weights are simply those of our loaded embedding matrix. These embedding weights are frozen during training, i.e. the weights remain fixed. The embedding layer output is then sequentially processed by the BiLSTM layer using the PyTorch `LSTM` method. Our BiLSTM layer consisted of 2 layers, featured a hidden dimension size of 128, and used Dropout with a drop probability of 0.3 during training as a regularization method to reduce overfitting. I then applied mean and max pooling to the BiLSTM layer output and concatenated the results of each of these pooling operations. This concatenated representation was then ultimately passed to the linear transformation layer for classification. I used the PyTorch implementation of the AdamW optimizer and the one cycle learning rate scheduler with maximum learning rate of 0.001 (Smith 2018). When using the English Wikipedia trained spherical embeddings at a maximum learning rate of 0.001, the model failed to converge. This was resolved by increasing the maximum learning rate to 0.01. Each of these models was trained for 30 epochs using a batch size of 256 and mixed precision training using 16-bit floating point weight values.

Multi-label classification: The multi-label classification setup, network architecture and training parameters are identical to that described for the binary classification task except for modifying the linear transformation layer to output a vector of length 2 to generate prediction for each of our two phenotypes. Models trained using our custom spherical embeddings and English Wikipedia trained spherical embeddings failed to converge when trained using a maximum learning rate of 0.001. Increasing the maximum learning rate to 0.01, I found that the model trained with English Wikipedia trained spherical embeddings converges for all 5 folds. Folds 1 and 4 trained using our custom spherical embedding model failed to converge appropriately despite trial of different learning rates (0.001, 0.01, 0.05), stochastic weight averaging, or use of 32-bit floating point values. Of note, when training with 32-bit floating point values, I reduced the batch size to 128 as training with a batch size of 256 exceeded our available GPU memory.

4.1.9 BERT-based Models

Binary classification: The base, uncased BERT model, BlueBERT, our custom pre-trained model RadBERT and each model's corresponding tokenizer were loaded using the `transformers` Python library (Wolf et al. 2020). The BlueBERT model uses the same tokenizer as the uncased BERT base model, whereas RadBERT uses a custom trained tokenizer (Peng, Yan, and Lu 2019). Tokenization using BERT-based models relies on the WordPiece algorithm as detailed in Section 2.5.3. More details about BERT-based models and their tokenization methodology can be found in Sections 2.5.3 and 3.1.6. For each model, input text was tokenized using the models' corresponding tokenizer. Rather than tokenizing all input text to the maximum BERT-based model input length of 512 tokens, I use dynamic padding and uniform length batching. Documents with tokenized output in excess of 512 tokens were truncated to the maximum model input length. The tokenized output was then passed to the embedding layer, which functions as a look-up table, before being passed to each model. The hidden state output from each BERT-based model is a tensor with size 768. I then applied a linear transformation layer to the model's hidden state representation for the phenotype classification task. Models were developed using PyTorch and the transformers library (Paszke et al. 2019; Wolf et al. 2020). For fine-tuning, I used the AdamW optimizer with the default learning rate of $2e^{-5}$ and a learning rate scheduler with linear warmup followed by linear decay using weight decay of 0.01 and the default 0 warmup steps. I used a batch size of 32 and trained for 10 epochs. I also trained each using mixed-precision training and gradient checkpointing.

Multi-label classification: Network architecture, tokenization and model setup are identical to that described for the binary classification task. I use similar parameters for model fine-tuning as well, except here I train with a reduced batch size of 16 due to GPU memory constraints.

4.1.10 Efficient Transformer Models

Binary classification: As with our BERT-based models (Section 4.1.9), I loaded the Longformer equivalent of each of these three models and their tokenizers using the transformers library (Wolf et al. 2020). I used dynamic padding and uniform length batching for tokenization and batch loading, respectively. The maximum model input length for the Longformer variants of our BERT-based models was 4,096 tokens. Tokenized output exceeding this maximum model sequence length were truncated to 4,096 tokens. Tokens are fed into the embedding layer, and this output is then passed to the Longformer model. Lastly, I applied a linear transformation to the Longformer model hidden state output for phenotype classification. I used PyTorch and the transformers library for model fine-tuning. Similar to with our BERT-based models, I used the AdamW optimizer with the default learning rate of $2e^{-5}$ and a learning rate scheduler with linear warmup followed by linear decay using weight decay of 0.01 and the default 0 warmup steps. I trained using a batch size of 8 for 10 epochs. Mixed-precision training and gradient checkpointing were also used to expedite training and overcome memory constraints.

Multi-label classification: Our method for multi-label classification was identical to that used for binary classification except I reduced the batch size to 4 and applied gradient accumulation.

4.1.11 Statistical Analyses

For each model, I computed the mean and standard deviation for each metric based on the results from our 5-fold nested cross validation procedure. When comparing the results from two models, I computed the Wilcoxon signed rank test. I used the Kruskal-Wallis test as our omnibus test to compare three or more models. I specified an *a priori* significance threshold of 0.05. Statistical analyses were conducted using R version 4.0.3 (R Core Team 2020).

Table 4.1: Label Distribution by Cross-validation Fold. 'Negative' indicates the absence of a condition, and 'Positive' indicates that the condition is present.

	Condition (Granulomatous Disease, Pneumothorax)			
Fold	Negative, Negative	Positive, Negative	Negative, Positive	Positive, Positive
1	526	439	22	13
2	525	440	23	12
3	523	442	25	10
4	523	443	25	9
5	528	437	19	16

4.2 Results

4.2.1 Annotation

To limit annotation to CT scan reports, I created a binary classifier. Baseline accuracy prior to model training was 0.412. Upon training, our final classification model achieved a receiver operating characteristic curve-area under the curve value of 0.999. I conducted manual annotation of 5,000 randomly selected documents and found that all 5,000 were CT scan reports.

4.2.2 Label Distribution

Of the 5,000 total annotated reports, 2,261 (45.2%) of the reports had evidence of granulomatous disease whereas 2,739 (54.8%) did not. Pneumothorax was present in 174 (3.5%) of reports and absent in the remaining 4,826 (96.5%).

As is shown in Table 4.1, 60 documents had evidence of both granulomatous disease and pneumothorax and 2,625 reports had no evidence of granulomatous disease or pneumothorax.

4.2.3 Binary Classification

Logistic Regression models: Results for each of our binary classification models are shown in Table 4.2 and Figure 4.1. Mean accuracy, F1-score, precision, and recall was highest for the logistic regression models trained using our custom spherical embedding (JoSE) model. Models trained using the English Wikipedia trained spherical embedding (JoSE) model yielded the lowest mean accuracy, F1-score, precision, and recall.

I also compared the 4 word2vec and fastText models. There was no significant difference in model performance for accuracy, F1-score, or precision ($p = 0.057$, $p =$

Table 4.2: Binary classification performance for Logistic Regression models.

Metric	word2vec (ours)[1]	fastText (ours)[1]	JoSE (ours)[1]	word2vec (Wiki)[1]	JoSE (Wiki)[1]	BioWordVec[1]	p-value[2]
Accuracy	0.777 (0.012)	0.794 (0.010)	0.801 (0.015)	0.785 (0.015)	0.723 (0.025)	0.766 (0.014)	0.001
F1-score	0.745 (0.012)	0.766 (0.014)	0.772 (0.016)	0.754 (0.019)	0.683 (0.027)	0.735 (0.019)	0.002
Precision	0.722 (0.015)	0.746 (0.022)	0.748 (0.014)	0.728 (0.026)	0.661 (0.027)	0.715 (0.025)	0.004
Recall	0.771 (0.022)	0.788 (0.012)	0.799 (0.024)	0.781 (0.019)	0.708 (0.037)	0.755 (0.014)	0.004

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

0.101, $p = 0.168$, respectively) but recall was found to be significantly different ($p = 0.043$).

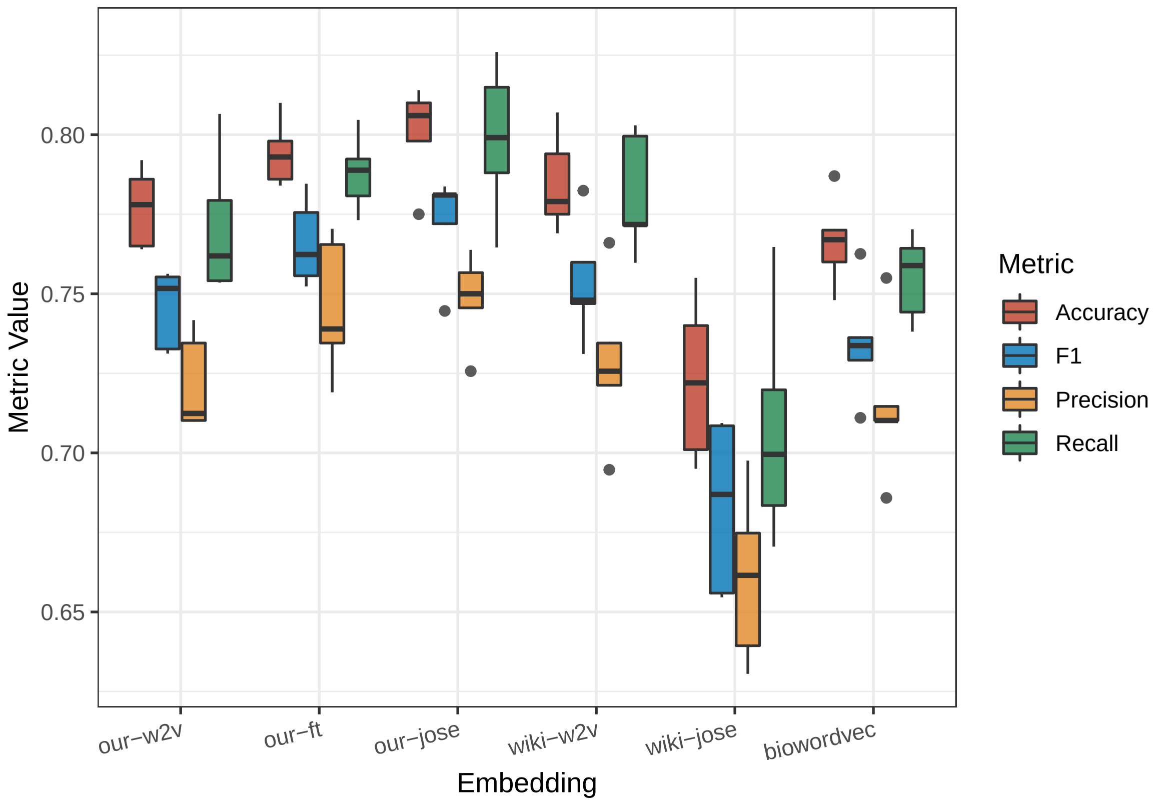


Figure 4.1: Binary Classification: Performance metrics for each of our Logistic Regression models by non-contextual embedding model type based on 5-fold cross validation.

Bidirectional LSTM models: Model performance for each of our Bidirectional LSTM (BiLSTM) models is shown in Table 4.3 and Figure 4.2. Mean accuracy, F1-score, and recall was highest for our BiLSTM model trained using the BioWordVec non-contextual word embedding model. Mean precision was highest for the BiLSTM model trained using the English Wikipedia trained word2vec model. The BiLSTM model trained using our custom spherical embedding (JoSE) model yielded the lowest mean accuracy, F1-score, and recall values compared to the other models. Our BiLSTM model trained using the English Wikipedia spherical embedding model had the lowest

Table 4.3: Binary classification performance for Bidirectional LSTM models.

Metric	word2vec (ours)[1]	fastText (ours)[1]	JoSE (ours)[1]	word2vec (Wiki)[1]	JoSE (Wiki)[1]	BioWordVec[1]	p-value[2]
Accuracy	0.991 (0.003)	0.982 (0.007)	0.968 (0.030)	0.991 (0.003)	0.980 (0.007)	0.992 (0.005)	0.034
F1-score	0.990 (0.004)	0.980 (0.008)	0.965 (0.034)	0.990 (0.003)	0.977 (0.008)	0.991 (0.005)	0.025
Precision	0.995 (0.005)	0.989 (0.008)	0.976 (0.040)	0.996 (0.004)	0.971 (0.012)	0.995 (0.003)	0.042
Recall	0.985 (0.004)	0.971 (0.010)	0.955 (0.042)	0.985 (0.005)	0.984 (0.005)	0.986 (0.010)	0.120

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

mean precision. The variance in metric values was greatest for our BiLSTM model trained using our custom spherical embedding (JoSE) model.

I compared each of our four word2vec and fastText models. Accuracy and F1-score were significantly different between the four models ($p = 0.048,\ p = 0.039$, respectively), and precision and recall were not significantly different ($p = 0.381,\ p = 0.056$, respectively).

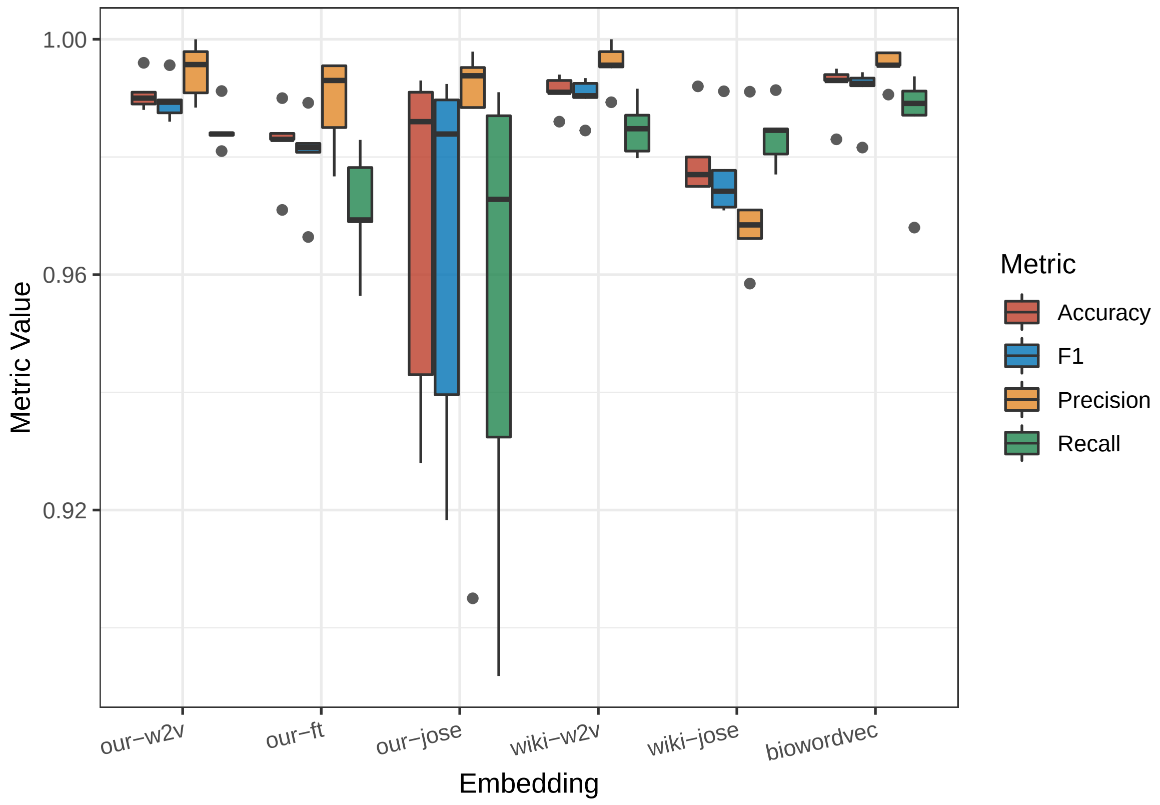


Figure 4.2: Binary Classification: Performance metrics for each of our Bidirectional LSTM models by non-contextual embedding model type based on 5-fold cross validation.

BERT-based models: Performance metrics for each of our three studied BERT-based models are shown in Table 4.4 and Figure 4.3. Mean accuracy, F1-score, precision, and recall was highest for our custom RadBERT model. The omnibus test indicates a significant difference in accuracy, F1-score, and recall between the models

Table 4.4: Binary classification performance for BERT-based models.

Metric	BERT-base[1]	BlueBERT[1]	RadBERT (ours)[1]	p-value[2]
Accuracy	0.956 (0.008)	0.945 (0.021)	0.982 (0.002)	0.007
F1-score	0.949 (0.010)	0.938 (0.022)	0.980 (0.002)	0.007
Precision	0.989 (0.003)	0.966 (0.045)	0.987 (0.004)	0.500
Recall	0.912 (0.020)	0.912 (0.018)	0.974 (0.003)	0.009

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

($p = 0.007,\ p = 0.007,\ p = 0.009$, respectively). The results of the Wilcoxon-rank sum test show no significant difference between the base BERT and BlueBERT models in terms of accuracy, F1-score, precision, and recall ($p = 0.344,\ p = 0.310,\ p = 0.310, p > 0.9$, respectively).

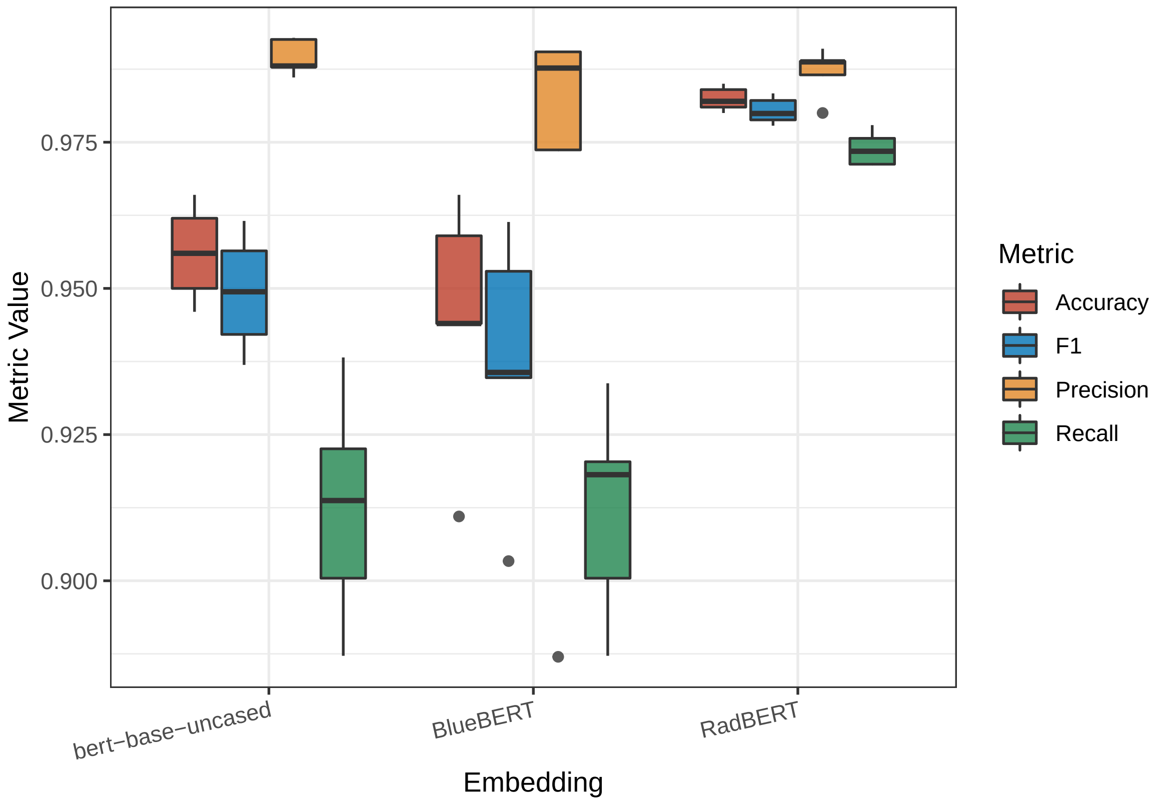


Figure 4.3: Binary Classification: Performance metrics for each of our BERT-based models based on 5-fold cross validation.

Efficient Transformer models: Performance metrics for each of our Efficient Transformer models is shown in Table 4.5. Results of our omnibus test show no significant difference for each of our model metrics.

Table 4.5: Binary classification performance for Efficient Transformer models.

Metric	BERT-base[1]	BlueBERT[1]	RadBERT (ours)[1]	p-value[2]
Accuracy	0.987 (0.004)	0.987 (0.004)	0.990 (0.002)	0.6
F1-score	0.986 (0.005)	0.986 (0.004)	0.989 (0.002)	0.7
Precision	0.977 (0.006)	0.977 (0.007)	0.980 (0.005)	0.8
Recall	0.996 (0.005)	0.995 (0.006)	0.998 (0.002)	0.8

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

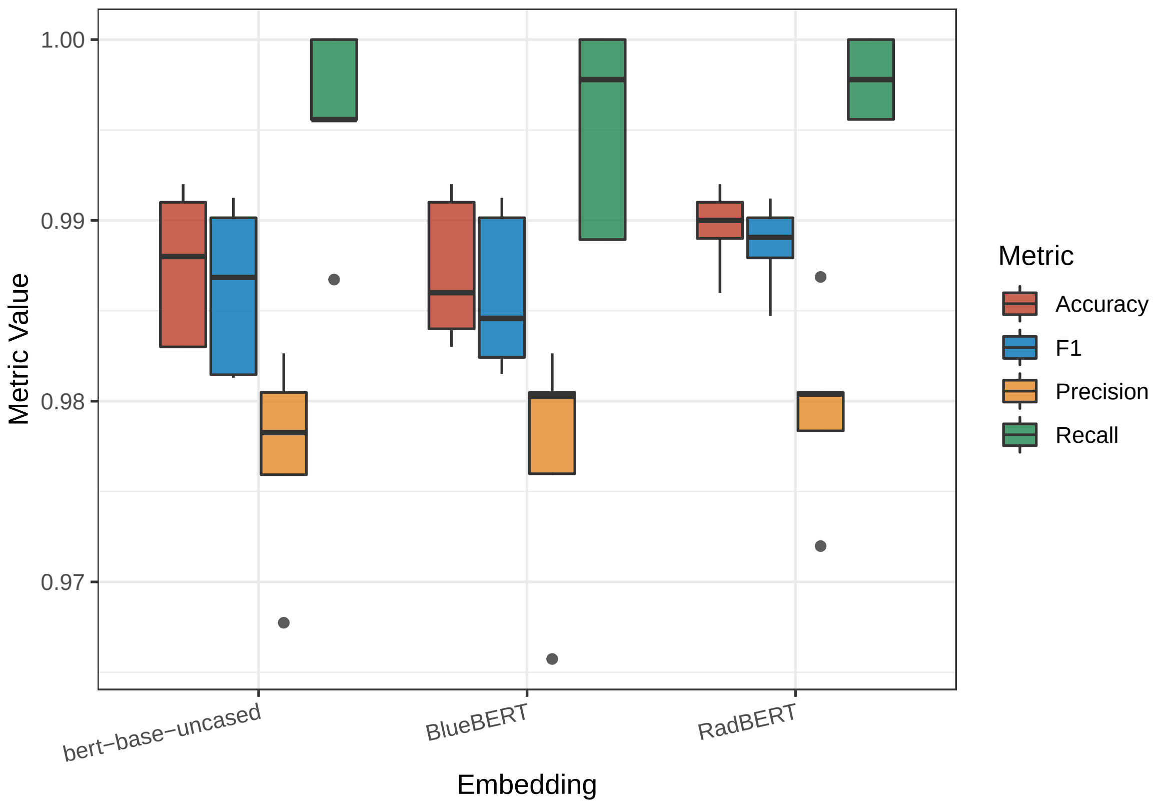


Figure 4.4: Binary Classification: Performance metrics for each of our Efficient Transformer models based on 5-fold cross validation.

Model Comparison: Figure 4.5 shows the distribution of accuracy, F1-score, precision, and recall for each of our BiLSTM, BERT-based, and Efficient Transformer models. I also compared performance metrics between BERT-based and Efficient Transformer models in Figure 4.6.

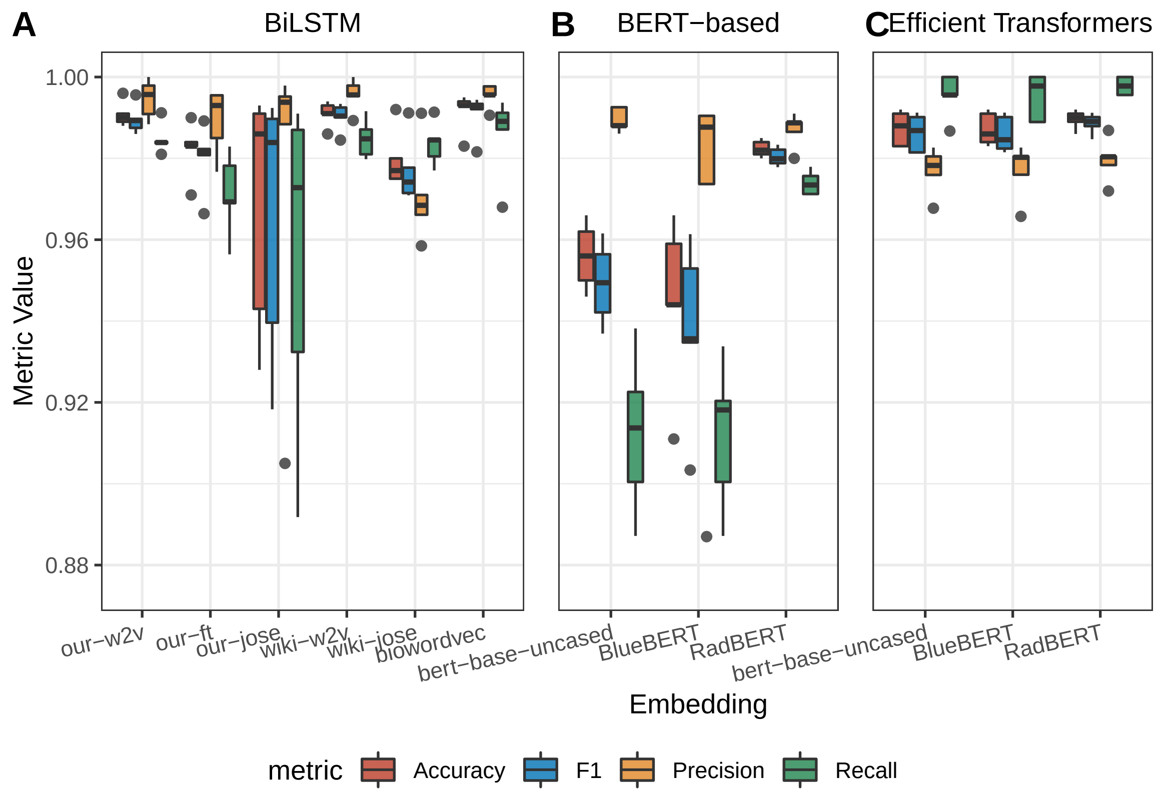


Figure 4.5: Binary Classification: Comparison of performance metrics between (A) Bidirectional LSTM models, (B) BERT-based models, and (C) Efficient Transformer models.

4.2.4 Multi-label Classification

Logistic Regression models: Results for each of our logistic regression models are shown in Table 4.6 and Figure 4.7. Logistic regression models trained using our custom fastText embedding model had the highest mean accuracy, precision, recall, and micro- and macro-F1 score. Our omnibus test results indicates a significant different between the models for each of the metrics studied. The model using English Wikipedia trained spherical embeddings (JoSE) had the poorest performance across all metrics.

I also compared each of our four word2vec and fastText models. Accuracy, micro- and macro-F1 score, precision, and recall were not significantly different between the models ($p = 0.176$, $p = 0.453$, $p = 0.361$, $p = 0.412$, $p = 0.426$, respectively).

Bidirectional LSTM models: Performance metrics for each of our BiLSTM models

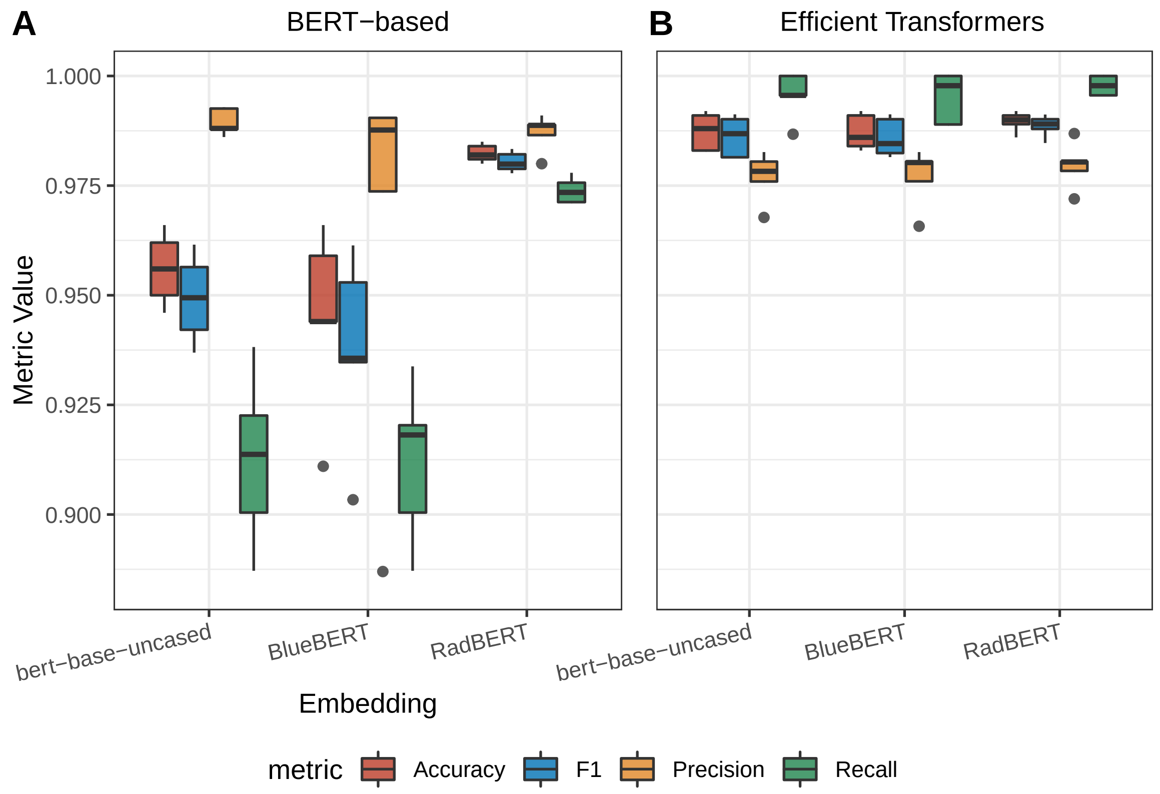


Figure 4.6: Binary Classification: Comparison of performance metrics between (A) BERT-based models and (B) Efficient Transformer models.

Table 4.6: Multi-label classification performance for Logistic Regression models.

Metric	word2vec (ours)[1]	fastText (ours)[1]	JoSE (ours)[1]	word2vec (Wiki)[1]	JoSE (Wiki)[1]	BioWordVec[1]	p-value[2]
Accuracy	0.747 (0.007)	0.766 (0.016)	0.757 (0.016)	0.752 (0.018)	0.691 (0.011)	0.744 (0.016)	0.007
Micro F1	0.333 (0.014)	0.342 (0.010)	0.330 (0.013)	0.333 (0.014)	0.301 (0.005)	0.329 (0.004)	0.013
Macro F1	0.620 (0.040)	0.644 (0.028)	0.598 (0.029)	0.589 (0.061)	0.520 (0.032)	0.607 (0.031)	0.019
Precision	0.332 (0.014)	0.341 (0.011)	0.330 (0.014)	0.333 (0.015)	0.301 (0.004)	0.329 (0.004)	0.013
Recall	0.334 (0.013)	0.343 (0.010)	0.333 (0.013)	0.336 (0.014)	0.303 (0.005)	0.330 (0.004)	0.012

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

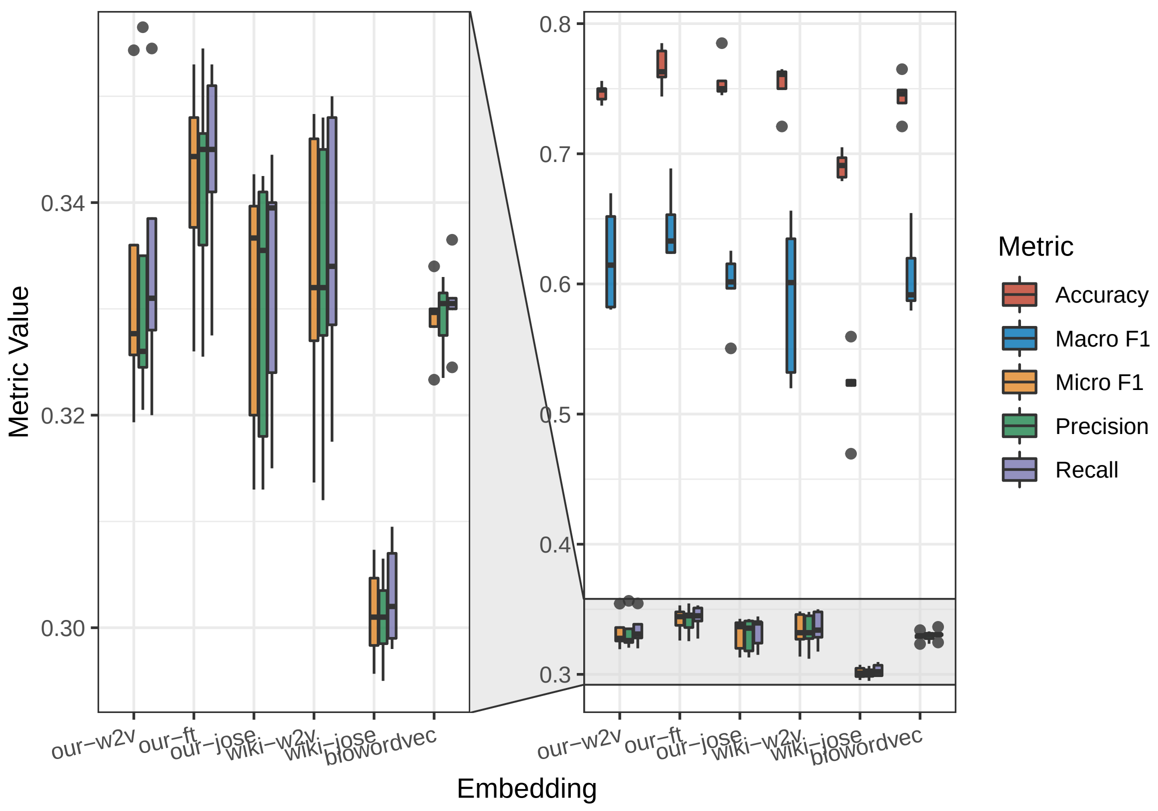


Figure 4.7: Multi-label Classification: Performance metrics for each of our Logistic Regression models by non-contextual embedding model type based on 5-fold cross validation. The left-most graph is a zoomed in view of the micro-F1, precision, and recall values.

Table 4.7: Multi-label classification performance for Bidirectional LSTM models.

Metric	word2vec (ours)[1]	fastText (ours)[1]	JoSE (ours)[1]	word2vec (Wiki)[1]	JoSE (Wiki)[1]	BioWordVec[1]	p-value[2]
Accuracy	0.948 (0.016)	0.944 (0.015)	0.785 (0.235)	0.930 (0.043)	0.948 (0.005)	0.952 (0.007)	0.8
Micro F1	0.443 (0.005)	0.443 (0.007)	0.300 (0.201)	0.442 (0.003)	0.440 (0.004)	0.445 (0.004)	0.4
Macro F1	0.490 (0.009)	0.488 (0.009)	0.348 (0.202)	0.481 (0.022)	0.496 (0.011)	0.492 (0.004)	0.9
Precision	0.442 (0.005)	0.441 (0.007)	0.299 (0.200)	0.440 (0.003)	0.438 (0.004)	0.443 (0.004)	0.5
Recall	0.447 (0.005)	0.447 (0.007)	0.302 (0.203)	0.446 (0.002)	0.444 (0.003)	0.449 (0.003)	0.2

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

shown in Table 4.7 and Figure 4.8. These include results from folds with poor model convergence using our custom spherical embedding model (JoSE) as described in Section 4.1.8. Excluding the folds that failed to converge using our custom spherical embedding model, I achieved mean accuracy 0.956, mean micro-F1 score 0.446, mean macro-F1 score of 0.495, mean precision 0.444, and mean recall of 0.450. Models trained using the BioWordVec embedding model yielded the highest mean accuracy, micro-F1, precision, and recall values. The highest macro-F1 score was seen for BiLSTM models trained using English Wikipedia trained spherical embedding (JoSE) models. As shown in Table 4.7, the omnibus test reveals no significant difference between the models across all metrics.

I also compared each of our four word2vec and fastText models. Accuracy, micro- and macro-F1 score, precision, and recall were not significantly different between the models ($p = 0.570,\ p = 0.378,\ p = 0.531,\ p = 0.505,\ p = 0.135$, respectively).

BERT-based models: Results from our multi-label classification task for each of the three BERT-based models are shown in Table 4.8. Our custom model, RadBERT, had the highest mean accuracy, precision, recall, and micro- and macro-F1 scores. BluBERT had the lowest performance across all metrics. As shown in Table 4.8, our omnibus test shows a significant difference across all metrics.

Comparing multi-label classification performance between the base BERT and BlueBERT models using the Wilcoxon rank-sum test, I found that macro-F1 score is significantly higher for the base BERT model (mean 0.929, s.d. 0.005 vs. mean 0.761, s.d. 0.087; $p = 0.008$) but there was no significant difference for the other studied metrics.

Efficient Transformer models: Performance metrics for each of our Efficient Transformer models are shown in Table 4.9. Mean accuracy, precision, recall, and micro- and macro-F1 scores are highest for our RadBERT model and lowest for the BlueBERT model. Accuracy and macro-F1 scores were found to be significantly

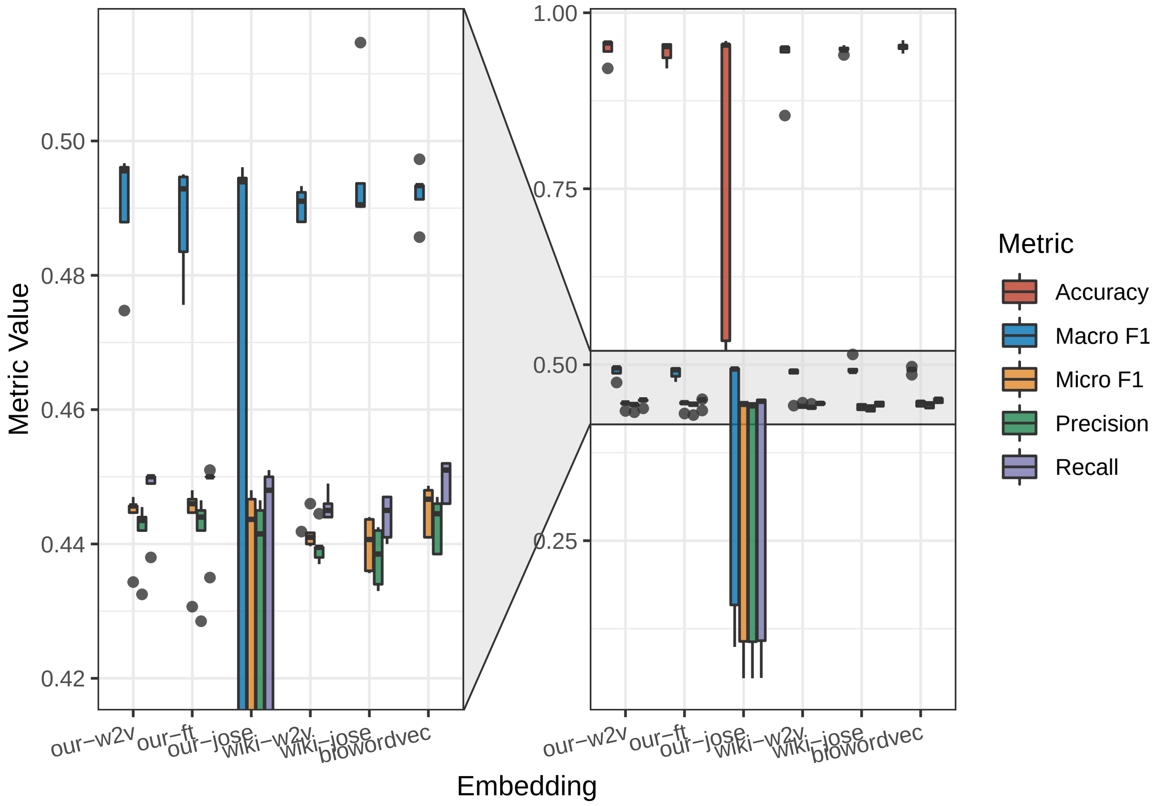


Figure 4.8: Multi-label Classification: Performance metrics for each of our Bidirectional LSTM models by non-contextual embedding model type based on 5-fold cross validation. The left-most graph is a zoomed in view of the macro- and micro-F1, precision, and recall values.

Table 4.8: Multi-label classification performance for BERT-based models.

Metric	BERT-base[1]	BlueBERT[1]	RadBERT (ours)[1]	p-value[2]
Accuracy	0.941 (0.010)	0.928 (0.014)	0.981 (0.006)	0.005
Micro F1	0.440 (0.006)	0.432 (0.011)	0.465 (0.005)	0.008
Macro F1	0.929 (0.005)	0.761 (0.087)	0.962 (0.017)	0.002
Precision	0.441 (0.006)	0.434 (0.010)	0.465 (0.005)	0.007
Recall	0.440 (0.007)	0.431 (0.012)	0.465 (0.005)	0.007

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

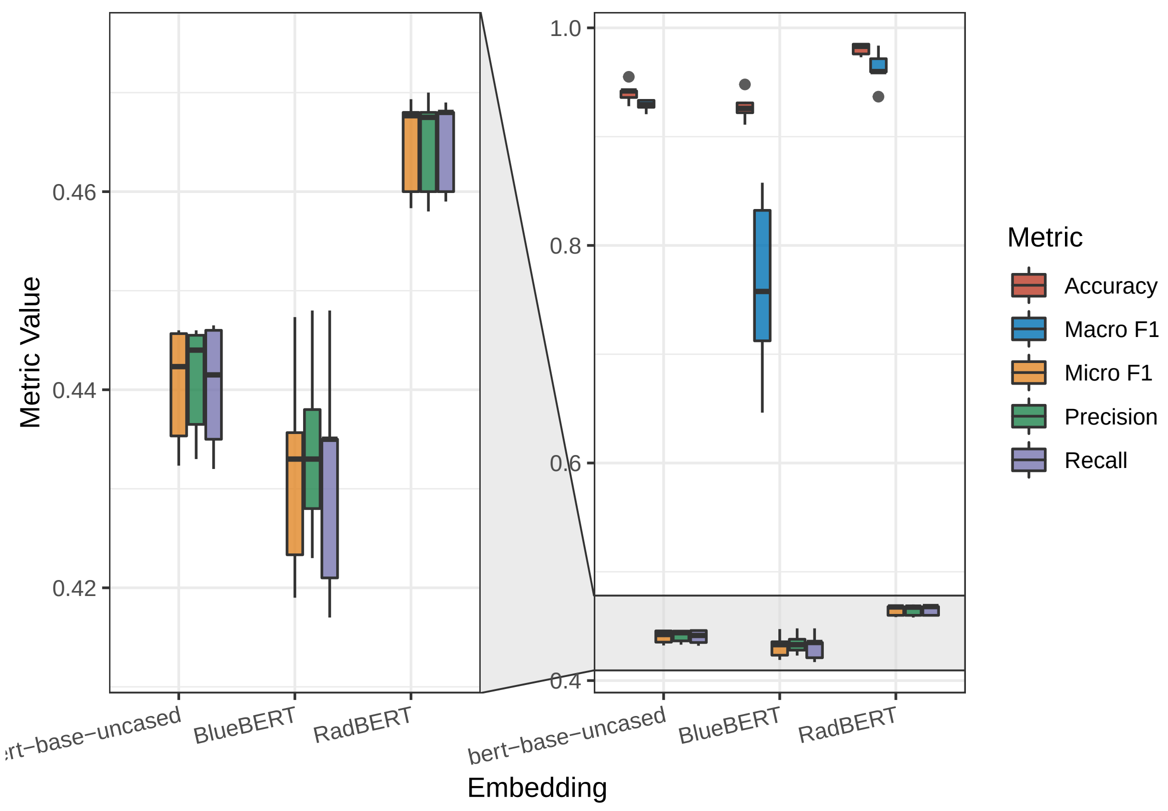


Figure 4.9: Multi-label Classification: Performance metrics for each of our BERT-based models based on 5-fold cross validation. The left-most graph is a zoomed in display of the micro-F1, precision, and recall values.

Table 4.9: Multi-label classification performance for Efficient Transformer models.

Metric	BERT-base	BlueBERT	RadBERT (ours)	p-value
Accuracy	0.975 (0.009)	0.965 (0.007)	0.990 (0.002)	0.004
Micro F1	0.466 (0.005)	0.464 (0.004)	0.472 (0.004)	0.072
Macro F1	0.866 (0.075)	0.823 (0.043)	0.964 (0.019)	0.007
Precision	0.467 (0.004)	0.465 (0.004)	0.472 (0.004)	0.063
Recall	0.466 (0.005)	0.465 (0.005)	0.472 (0.005)	0.200

[1] Data are presented as mean (SD). [2] p-value from Kruskal-Wallis test.

different as shown in Table 4.9.

Model comparison: Figure 4.11 shows micro-F1, precision, and recall scores for the BiLSTM, BERT-based, and Efficient Transformer models. I compare micro-F1, precision, and recall performance between BERT-based and Efficient Transformer models in Figure 4.12.

4.3 Discussion

I developed two extrinsic evaluation tasks to compare contextual and non-contextual embedding models: binary and multi-label classification. I then trained models using each CWE and NCWE model and compared their performance on these extrinsic evaluation tasks.

Label distribution for the binary classification task was well-balanced as 45.2% of the reports suggested evidence of granulomatous disease and the remaining 54.8% did not. The radiology reports used in this study were sourced from a hospital system in an endemic fungal region. As such, this high prevalence of granulomas noted on CT reports is to be expected. *Histoplasma capsulatum* is a fungus that typically causes benign and asymptomatic disease in immunocompetent hosts. In endemic areas, up to 90% of the population may be exposed to this species during their lifetime. A subset of these patients may develop granulomas, which can appear as incidental pulmonary nodules (IPN) and may complicate IPN evaluation and management (Azar et al. 2020).

The *pneumothorax* labels used in this study are heavily imbalanced with only 3.5% of the reports highlighting the presence of this condition. Many of the documents do contain language about pneumothoraces, but this is most often to indicate the absence of this condition. This information is commonly reported, even if findings are negative,

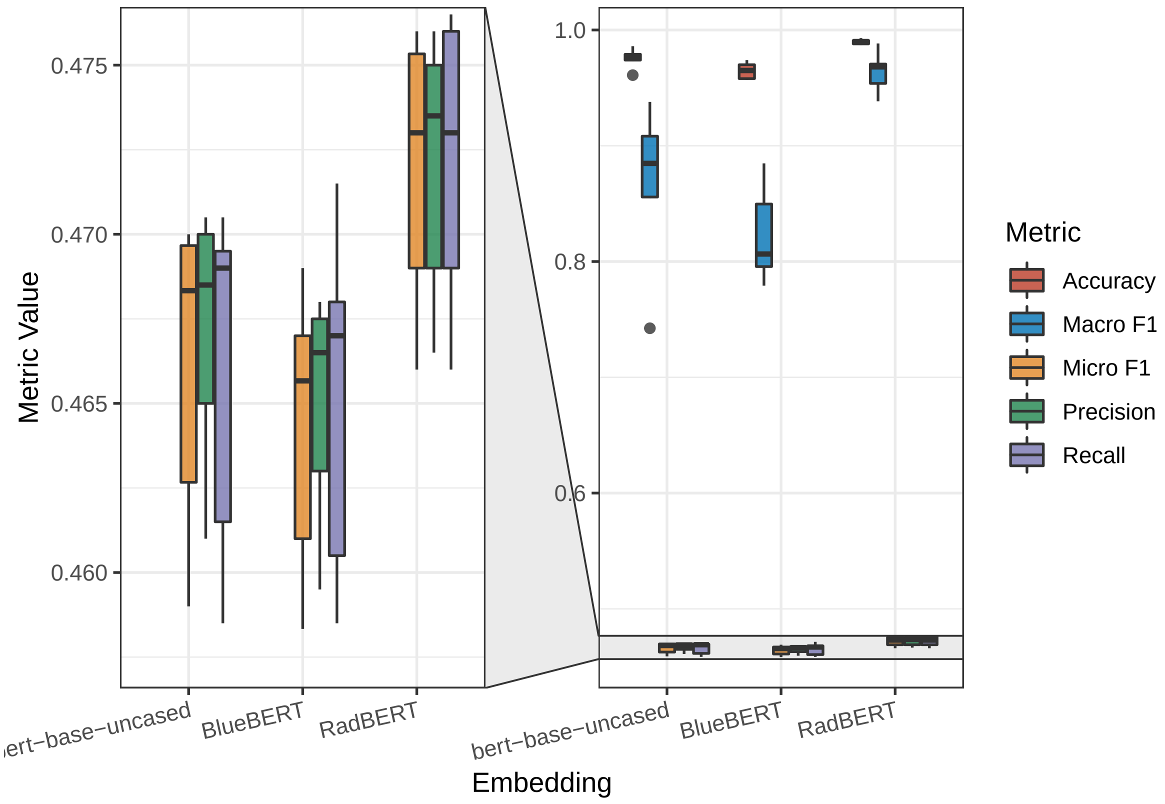


Figure 4.10: Multi-label Classification: Performance metrics for each Efficient Transformer model based on 5-fold cross validation. The left-most graph is a zoomed in display of the micro-F1, precision, and recall values.

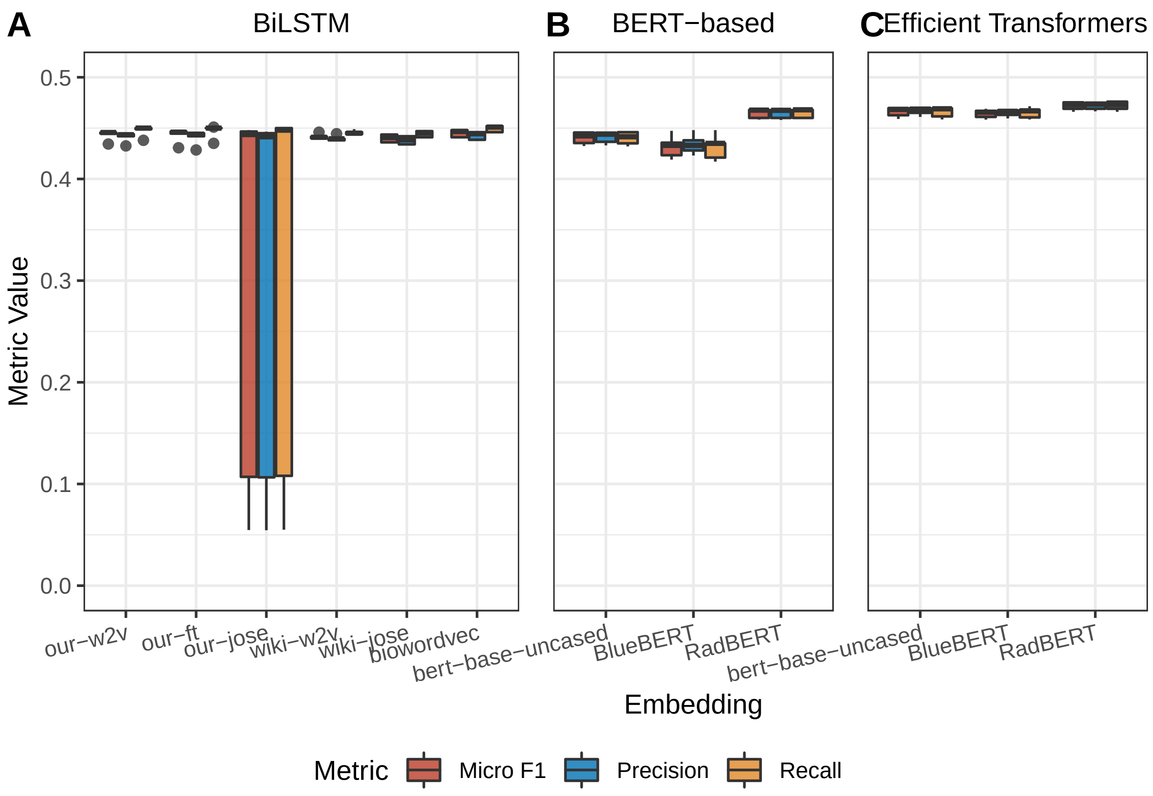


Figure 4.11: Multi-label Classification: Comparison of micro-F1 score, precision, and recall between (A) Bidirectional LSTM models, (B) BERT-based models, and (C) Efficient Transformer models.

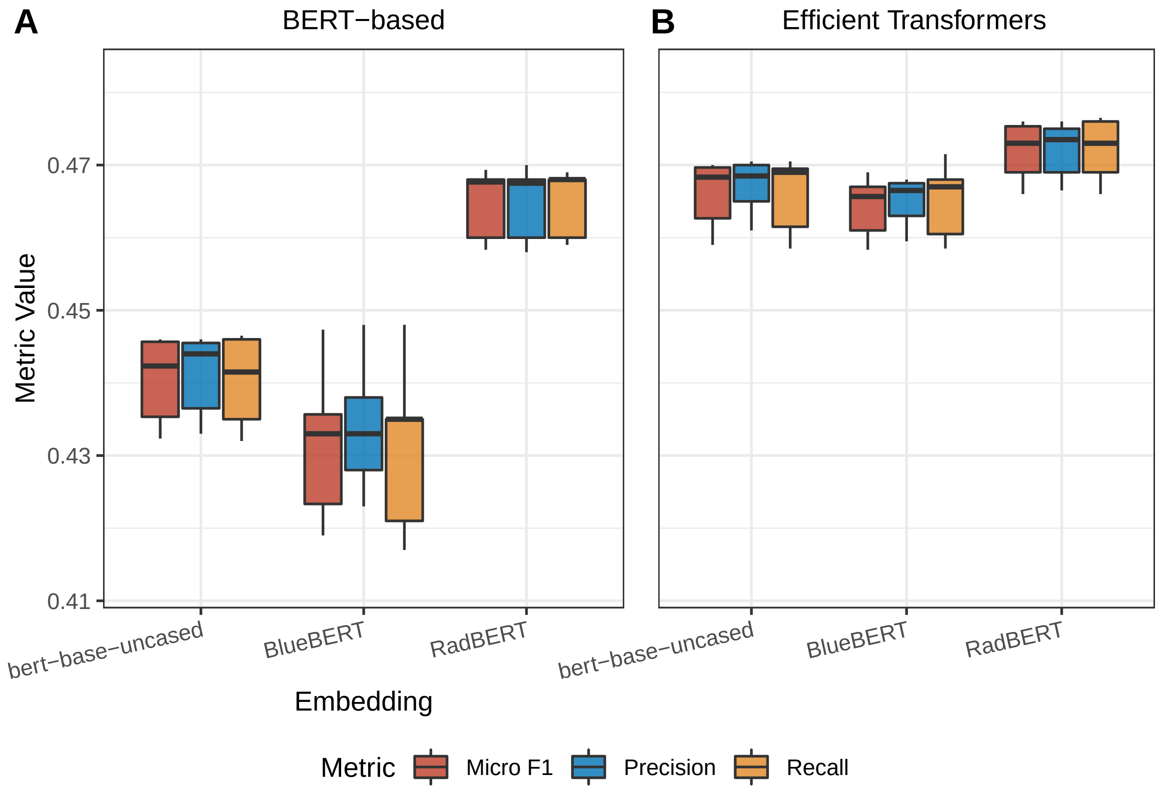


Figure 4.12: Multi-label Classification: Comparison of micro-F1 score, precision, and recall between (A) BERT-based models and (B) Efficient Transformer models.

in CT scan reports as it may warrant careful monitoring and often urgent intervention. Thus, our multi-label classification task includes two labels; one with well-balanced classes (*granulomatous disease*) and the other imbalanced (*pneumothorax*).

Non-contextual embeddings

Interestingly, the general domain word2vec model trained on an English Wikipedia corpus performed at least as well as, if not better than, our custom NCWE models trained using the `word2vec` or `fastText` algorithms and BioWordVec on both extrinsic evaluation tasks. This finding suggests that there may not be a clear or sizable advantage in using an in-domain non-contextual embedding model on extrinsic evaluation performance. For instance, I found that logistic regression and BiLSTM models using the English Wikipedia trained word2vec embedding were equivalent or marginally outperformed models trained with our custom word2vec model and BioWordVec on all studied performance metrics (Tables 4.2, 4.3, 4.6, 4.7).

Similarly, our binary and multi-label classification results show that models trained using either the `word2vec` or `fastText` algorithm do not reveal a consistent or demonstrable advantage over one another. Comparing only word2vec and fastText embeddings, the logistic regression model trained using our custom fastText embeddings performed best. Yet, the BioWordVec embeddings, which were trained using the `fastText` algorithm, performed less well in all metrics compared to our custom and English Wikipedia trained word2vec embeddings (Table 4.2 and 4.6). Comparing BiLSTM models trained only on word2vec and fastText NCWE models, the opposite to what was found with logistic regression models was seen. Among BiLSTM models, BioWordVec performed better than or as well as both custom and Wikipedia trained word2vec embeddings, and our custom fastText embeddings ostensibly performed less well, but this difference was not significantly different (Table 4.3).

Of note, I trained each of our logistic regression and BiLSTM models by freezing the embedding layer weights, i.e. the embedding model weights remained fixed to their original values. This strategy was selected as our objective was to perform extrinsic evaluation of each embedding models' learned representations based on the trained corpus and algorithm. Unfreezing these weights and allowing them to be updated during fine-tuning may yield different results.

The special case of Spherical embeddings

Our results highlight the inconsistency and instability in extrinsic evaluation performance using spherical embedding (JoSE) models. I found that logistic regression and

BiLSTM models trained using spherical embeddings (JoSE) provided either the best or worst performance on different metrics. Using logistic regression, our custom spherical embeddings trained performed best across all metrics (Table 4.2; Figure 4.1). By contrast, the BiLSTM model trained using our custom spherical embeddings yielded the lowest mean accuracy, F1-score, and recall compared to all other embeddings (Table 4.3; Figure 4.2). With the exception of the logistic regression model trained using our custom spherical embeddings, there is larger variance in performance for each model trained with spherical embeddings as evidenced in Figures 4.2 and 4.8. The instability of spherical embeddings is further evidenced by our experience with BiLSTM model fine-tuning; model convergence issues were only noted with BiLSTM models using either our custom or English Wikipedia trained spherical embeddings. As described in Section 4.1.8, BiLSTM models using our custom spherical embeddings and English Wikipedia trained spherical embeddings only converged after increasing the learning rate. Table 4.1 shows that the label distribution in each of our 5 folds is similar, which makes this unlikely to have contributed to training issues. The spherical embedding training algorithm jointly learns word and paragraph embeddings, and has been shown to outperform other embedding models on a variety of general-domain NLP tasks: intrinsic evaluation, document clustering, and document classification (Meng et al. 2019). Clinical text is quite distinct from other general text corpora as described in Section 2.1. In clinical documents, neighboring paragraphs can contain very different information. This idiosyncratic document structure may make the spherical embedding joint learning strategy ill-suited for use in the clinical domain.

I found in Section 3.2.2 that term pair cosine similarity judged by our custom spherical embedding model was skewed relative to the equivalent models trained using the `word2vec` and `fastText` algorithms (See Figure 3.2). Although it is plausible that this skewed representation may have influenced the need for a higher learning rate, the English Wikipedia trained word2vec embedding model was also found to be skewed and did not require similar modification. Thus, these challenges seem to be unique to spherical embedding models and further evaluation of spherical embedding representations and training strategies is needed.

Contextual embeddings

BERT-based models: For both binary and multi-label classification, I consistently found that our custom RadBERT model outperformed the base BERT model and BlueBERT model. As shown in Tables 4.4 and 4.8, our omnibus test shows a significant difference between the models on all metrics except for precision in the case of binary classification ($p = 0.53$). Additionally, the variance of the RadBERT model across all studied metrics is smaller than seen in the other models (Tables 4.4 and 4.8; Figures 4.3 and 4.9). Overall, these findings demonstrate the consistent gains in performance afforded by pre-training a BERT-based model on an in-domain corpus a general model (base BERT model) or biomedical model (BlueBERT).

Despite being pre-trained on a large biomedical and sizable clinical text corpus, the BlueBERT model surprisingly performed no better than the base BERT model in either the binary or multi-label classification tasks. In fact, I found that the multi-label classification macro-F1 score for the base BERT model was significantly higher than that for the BlueBERT model. The lack of improved classification performance using BlueBERT compared to the base BERT model runs counter to previously published findings of BlueBERT and other clinical or biomedical BERT-based models (Peng, Yan, and Lu 2019; Beltagy, Lo, and Cohan 2019; Alsentzer et al. 2019; Lee et al. 2020). BlueBERT was pre-trained on the PubMed corpus, which includes title, abstract, and article text for tens of millions covering a breadth of scientific literature, and an intensive care clinical text corpus. This combined biomedical and clinical text corpus may contain relatively few radiology reports as a subset of the MIMIC-III corpus. As described in Section 2.1, clinical text is quite distinct from other domains and may vary by specialty, clinical setting, region of practice, and other factors. Thus, BlueBERT and many other biomedical BERT-based models trained on similar corpora may not be considered an in-domain model for processing radiology report text.

An additional limitation of the BlueBERT model is that it relies on the original BERT tokenizer, rather than a custom tokenizer trained on PubMed and MIMIC-III corpora (Peng, Yan, and Lu 2019). One of the advantages of using a custom tokenizer is the ability to capture more information given the input sequence length limitation of BERT-based models of 512 tokens. As depicted in Figure 4.13, our custom RadBERT tokenizer can capture more information in fewer tokens than the original BERT tokenizer. This may help limit the possibility of information loss and could partly explain RadBERT's superior performance. SciBERT is one example of a publicly available biomedical BERT-based model pre-trained on a large biomedical corpus that uses a custom tokenizer (Beltagy, Lo, and Cohan 2019). A novel transformer model was recently introduced to overcome some of these limitations of tokenization and maintaining a vocabulary. Character Architecture with No tokenization In Neural Encoders, or CANINE, performs tokenization-free language modeling by operating directly on unicode character sequences rather than tokens (Clark et al. 2021).

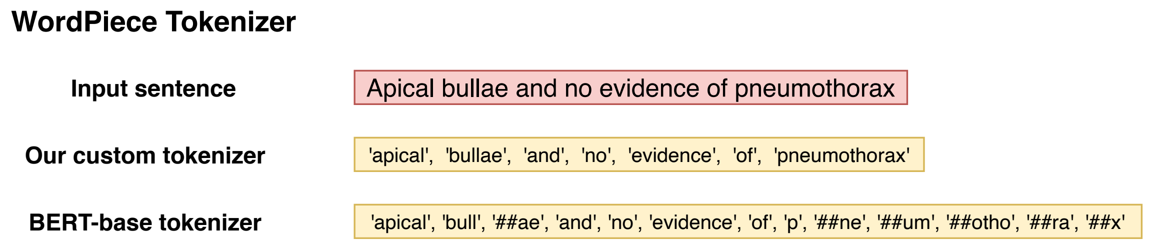

WordPiece Tokenizer
Input sentence
Apical bullae and no evidence of pneumothorax
Our custom tokenizer
'apical', 'bullae', 'and', 'no', 'evidence', 'of', 'pneumothorax'
BERT-base tokenizer
'apical', 'bull', '##ae', 'and', 'no', 'evidence', 'of', 'p', '##ne', '##um', '##otho', '##ra', '##x'

Figure 4.13: Example of text tokenization comparing our custom tokenizer and the BERT-base tokenizer. The input sequence is 7 words in length. Our custom tokenizer returns 7 tokens; BERT-original returns 13 tokens.

One of the limitations of BERT and BERT-based language models is their large size. Training of these models requires access to sufficiently large GPU memory and can take several hours to days for pre-training depending on the GPU or tensor processing capacity (TPU) one has available. Despite this large upfront training cost, the advantage includes the ability to apply these models to achieve state-of-the-art (SOTA) or near-SOTA performance on a wide range of downstream NLP tasks with relatively little effort.

Researchers are actively developing methods to reduce the model size and training efficiency without significantly compromising model performance, such as model distillation (M. Gupta and Agrawal 2021). Sanh et al. used model distillation to reduce the size of a BERT model by 40% while maintaining 97% of the BERT model natural language understanding ability. The DistilBERT model also 60% faster than the base BERT model (Sanh et al. 2020).

Efficient Transformer models: Our efficient transformer models are the Longformer model equivalent of the BERT-based models: base, uncased BERT, BlueBERT, and our custom RadBERT model. For binary classification, the performance across all studied metrics is comparable (Table 4.5 and Figure 4.4). For multi-label classification, our results indicate that the Longformer variant of RadBERT provides a modest improvement over the other models at least in terms of accuracy and macro-F1 score. Beltagy et al. show that pre-training Longformer models adapted from BERT-based model further can provide additional performance improvements (Beltagy, Peters, and Cohan 2020). I did not perform additional pre-training of each of our Longformer models, but this may be considered in future work to evaluate for gains in model performance.

Each of our Longformer models uses the same tokenizers as their BERT-based model counterparts. Thus, our Efficient Transformer models are equally subject to the same limitations of these tokenizers as outlined above. The Longformer models scale linearly with input sequence length, whereas the BERT-based models scale quadratically. This reduced complexity, allows the Longformer models to accept longer input sequence lengths. This may partly mitigate the potential for information loss seen with the equivalent BERT-based models. Despite the improvement in computational complexity, the additional parameters to account for the local sliding window attention and global attention renders these models larger than their BERT-based counterparts. This even larger model size may again serve as an impediment for use among those without access to adequate GPU or TPU resources. Methods for language model compression, such as knowledge distillation and pruning, may also be effective at reducing model size and speed while maintaining performance (M. Gupta and Agrawal 2021). As efficient transformer methods have only recently been introduced, the effects of such compression methods on these novel models remains to be explored.

Comparison of Model Architectures

Overall, each of our logistic regression models consistently perform less well than the other model architectures studied: BiLSTM, BERT-based, and Efficient Transformer models. For the binary classification task, each of our BiLSTM models tended to outperform the base BERT and BlueBERT models. The exception being our BiLSTM model trained using our custom spherical embedding model. BiLSTM model performance was also comparable to RadBERT and each of our three Efficient Transformer models (Figure 4.5). Ezen-Can et al. compared classification performance between BiLSTM and base BERT models fine-tuned on a small corpus. Similar to our binary classification findings, they found that their BiLSTM model significantly outperformed the BERT model (Ezen-Can 2020). As expected, BiLSTM models trained more quickly than the BERT-based and Efficient Transformer models. Additionally, BiLSTM models typically comprise far fewer parameters and have a much smaller memory footprint than BERT models. This makes BiLSTM models much more accessible, especially in settings with GPU constrained resources.

On the multi-label classification task, our BiLSTM models were similar in performance to the BlueBERT and base BERT models and performed less well than RadBERT and the Efficient Transformer models (Figure 4.11). In other words, the performance gains of BiLSTM models over base BERT and BlueBERT observed in the binary classification task fail to hold in the multi-label classification setting. Recall that multi-label classification includes the *pneumothorax* label, which is heavily imbalanced (See Section 4.2.2). Thus, this difference in performance may be due to BiLSTM models being less resilient to class imbalance than BERT-based models. One of the advantages of transformer models is they perform well even on imbalanced classification tasks (Tayyar Madabushi, Kochkina, and Castelle 2019). For training, I used the binary cross entropy loss function, but one may find improvement in BiLSTM model performance using other loss functions that may be better-suited for imbalanced classification, such as Dice loss (Li et al. 2020).

As shown in Figures 4.6 and 4.12, each of the Longformer models often performs better than their BERT-based model counterparts. The exception being equivalent or slightly reduced precision for each of our Longformer variants in the binary classification setting. This improvement in model performance is in accordance with Beltagy et al.'s findings, which showed that the Longformer variant of RoBERTa outperformed the base RoBERTa model (Beltagy, Peters, and Cohan 2020). These gains are likely attributable to the Longformer model's ability to accept longer text input sequences (4096 tokens, compared to 512 tokens for BERT-based models) and avoid information loss. Additionally, Beltagy et al.'s ablation studies evaluating the utility of the local sliding window attention and global attention suggest that the Longformer model's dual attention scheme may provide an additional advantage over BERT's singular local self-attention mechanism (Beltagy, Peters, and Cohan 2020).

During annotation, I commonly found that radiology reports often used capitalization to emphasize atypical findings, e.g. "LEFT-sided pneumothorax." Although I use the uncased BERT models in this study, use of the cased model may improve classification performance for conditions where capitalization is used to emphasize abnormal radiographic findings. Future studies should also examine if these findings are upheld across other transfomer or efficient transformer architectures, such as OpenAI's Generative Pre-trained Transformer 2 (GPT-2) or Google Research's efficient transformer models Reformer and Big Bird (Radford et al. 2019; Kitaev, Kaiser, and Levskaya 2020; Zaheer et al. 2020). Other transformer and efficient transformer models rely on alternative pre-training strategies or use different attention mechanisms, and it remains to be seen how these may impact clinical NLP performance.

4.4 Conclusion

This study explored extrinsic evaluation performance of different word embedding models using binary and multi-label phenotype classification tasks. Overall, neural network frameworks yield significant gains over logistic regression models. In select circumstances, Bidirectional LSTM models can perform as well as, if not better than, publicly available BERT-based models, and may be most preferred if computing resources are constrained. Among non-contextual embedding models, no one embedding illustrated a clear and consistent advantage over the others using either logistic regression or BiLSTM frameworks. The word2vec or fastText-based embeddings appear to be equally performant and may be preferred over spherical embeddings for phenotype classification. Moreover, the use of in-domain non-contextual word embeddings may not be as vital for downstream NLP classification performance as would be expected. BERT-based models provide a robust baseline and pre-training a custom BERT model and tokenizer offers a demonstrable performance improvement over BiLSTM and publicly available BERT-based models. Efficient Transformer models are as robust and tend to provide the best phenotype classification results compared to their BERT model counterparts. Efficient Transformer models provide many of the advantages of BERT-based models while also being capable of processing long text sequences making it ideal for clinical NLP tasks involving lengthy clinical documents, such as radiology reports.

CHAPTER 5

CONCLUSIONS

Given the paucity and limitations of existing clinical intrinsic evaluation (IE) benchmarks, I developed two new IE methods: term pair similarity to assess non-contextual word embeddings and cloze task accuracy for contextual word embedding models. Using this method, I was able to quantitatively compare embedding model representations against clinician judgment. Each of our IE tasks revealed that models pre-trained on a combined biomedical and clinical text corpus were better or equivalent to models pre-trained on our domain-specific clinical corpus. As expected, our custom embedding models outperformed publicly available embeddings trained on general corpora, such as English Wikipedia. I had hypothesized that spherical embedding models would provide the best IE performance, mirroring the results from published findings on general domain benchmarks. Yet, our findings suggest that the training objective for the spherical embedding algorithm, albeit well-suited for general corpora, may be improper for use with many clinical documents due to their idiosyncratic structure.

We detail our data-driven approach for more rapid development of term pair similarity IE tasks. This may serve as a framework for other researchers to efficiently develop additional IE benchmarks and probing tasks. Developing and conducting intrinsic evaluation of these and other clinical NLP models is necessary to promote trust and clinical adoption of these tools.

While ensuring that model representations are well aligned with clinician judgment is pivotal, such models must also demonstrate effective performance on downstream clinical NLP tasks. I curated a set of 5,000 radiology reports and annotated each for the presence or absence of 2 clinical conditions for our extrinsic evaluation (EE) tasks. Comparing performance using different algorithms, there is a clear advantage using neural network methods over logistic regression models. Embedding models pre-trained on a domain-specific corpus appeared to provide improved classification performance for transformer (BERT-based) and Efficient Transformer models. Yet, the advantage from domain-specific pre-training did not extend to non-contextual word embedding models. These findings suggest that the use of in-domain non-contextual word embeddings may not be as vital for phenotype classification performance as originally hypothesized. As hypothesized, the Efficient Transformer models overall were robust and outperformed their BERT model counterparts. Contrary to our hypothesis, BiLSTM models trained using word2vec of fastText embeddings were competitive with our custom RadBERT model and each of Efficient Transformer models in our binary classification task and competitive with the publicly available BERT-base models on multi-label classification. Although transformer models are often considered state-of-the-art, I found that BiLSTM models work just as well and should be a viable option for clinical NLP involving long text documents or where computing resources are limited.

Overall, intrinsic evaluation performance fails to correlate with extrinsic evaluation findings; our models that perform best on IE tasks often do not perform best on EE tasks. For instance, on cloze task evaluation our custom RadBERT model was equivalent to the publicly available BlueBERT model, but on EE, RadBERT consistently outperforms the BlueBERT model. Moreover, among our non-contextual embeddings, the publicly available word2vec model pre-trained on the English Wikipedia corpus was among the poorest models in our IE task, but was comparable in performance with our custom NCWEs and BioWordVec. These findings illustrate that different word representations may perform better for different tasks.

Extending from this work, I aim to develop additional model evaluation tasks to investigate for bias and protected health information (PHI) retention. Lehman et al. used a series of probing tasks in an effort to extract PHI from a BERT model pre-trained on the MIMIC-III clinical text corpus and were unsuccessful in doing so (Lehman et al. 2021). It remains unclear if and how successful such probing efforts would be on a BERT model pre-trained using other clinical text corpora, such as VUMC's de-identified or PHI-containing clinical documents. Further study in this area may help facilitate development and distribution of useful language models for clinical NLP research. I also wish to extend our work with clinical embeddings to develop clinical text data augmentation methods. In the general NLP domain, text data augmentation, akin to image augmentation procedures, is an effective training strategy shown to improve classification performance and generalizability. Yet, clinical text fails to adhere to traditional grammar rules and other general linguistic properties, necessitating development of text data augmentation specific to clinical text. Furthermore, I wish to compare phenotyping performance using other Efficient Transformer models and other novel transformer architectures, such as the tokenization-free CANINE model. Lastly, I wish to augment our non-contextual embedding models using biomedical knowledge graphs to see how this may impact intrinsic and extrinsic evaluation performance.

REFERENCES

Agirre, Eneko, Enrique Alfonseca, Keith Hall, Jana Kravalova, Marius Paşca, and Aitor Soroa. 2009. "A Study on Similarity and Relatedness Using Distributional and WordNet-Based Approaches." In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 19–27. NAACL '09. USA: Association for Computational Linguistics.

Alsentzer, Emily, John R. Murphy, Willie Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew B. A. McDermott. 2019. "Publicly Available Clinical BERT Embeddings." *arXiv:1904.03323 [Cs]*, June. http://arxiv.org/abs/1904.03323.

Apache Software Foundation. 2020. *Apache Parquet* (version 2.8.0). https://parquet.apache.org.

Aronson, AR. 2001. "Effective Mapping of Biomedical Text to the UMLS Metathesaurus: The MetaMap Program." *Proceedings. AMIA Symposium*, 17—21.

Azar, Marwan M., James L. Loyd, Ryan F. Relich, L. Joseph Wheat, and Chadi A. Hage. 2020. "Current Concepts in the Epidemiology, Diagnosis, and Management of Histoplasmosis Syndromes." *Seminars in Respiratory and Critical Care Medicine* 41 (01): 013–30. https://doi.org/10.1055/s-0039-1698429.

Bakarov, Amir. 2018. "A Survey of Word Embeddings Evaluation Methods." *arXiv:1801.09536 [Cs]*, January. http://arxiv.org/abs/1801.09536.

Baker, Simon, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2015. "Automatic Semantic Classification of Scientific Literature According to the Hallmarks of Cancer." *Bioinformatics* 32 (3): 432–40. https://doi.org/10.1093/bioinformatics/btv585.

Balderston, Jessica R., Zachary M. Gertz, Raees Seedat, Jackson L. Rankin, Amanda W. Hayes, Viviana A. Rodriguez, and Gregory J. Golladay. 2021. "Differential Documentation of Race in the First Line of the History of Present Illness." *JAMA Internal Medicine* 181 (3): 386. https://doi.org/10.1001/jamainternmed.2020.5792.

Banerjee, Imon, Matthew C. Chen, Matthew P. Lungren, and Daniel L. Rubin. 2018. "Radiology report annotation using intelligent word embeddings: Applied to multi-institutional chest CT cohort." *Journal of Biomedical Informatics* 77 (January): 11–20. https://doi.org/10.1016/j.jbi.2017.11.012.

Banerjee, Imon, Sriraman Madhavan, Roger Eric Goldman, and Daniel L. Rubin. 2017. "Intelligent Word Embeddings of Free-Text Radiology Reports." *arXiv:1711.06968*

[Cs], November. http://arxiv.org/abs/1711.06968.

Baroni, Marco, Georgiana Dinu, and Germán Kruszewski. 2014. "Don't Count, Predict! A Systematic Comparison of Context-Counting Vs. Context-Predicting Semantic Vectors." In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 238–47. Baltimore, Maryland: Association for Computational Linguistics. https://doi.org/10.3115/v1/P14-1023.

Baumel, Tal, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and No'emie Elhadad. 2017. "Multi-Label Classification of Patient Notes a Case Study on ICD Code Assignment." *arXiv:1709.09587 [Cs]*, November. http://arxiv.org/abs/1709.09587.

Beam, Andrew L., Benjamin Kompa, Allen Schmaltz, Inbar Fried, Griffin Weber, Nathan Palmer, Xu Shi, Tianxi Cai, and Isaac S. Kohane. 2019. "Pacific Symposium on Biocomputing 2020." In, 295–306. Kohala Coast, Hawaii, USA: WORLD SCIENTIFIC. https://doi.org/10.1142/9789811215636_0027.

Beltagy, Iz, Kyle Lo, and Arman Cohan. 2019. "Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)." In, 3613–18. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-1371.

Beltagy, Iz, Matthew E. Peters, and Arman Cohan. 2020. "Longformer: The Long-Document Transformer." *arXiv:2004.05150 [Cs]*, April. http://arxiv.org/abs/2004.05150.

Bengio, Yoshua, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. "A Neural Probabilistic Language Model." *The Journal of Machine Learning Research* 3 (null): 11371155.

Bengio, Yoshua, P. Simard, and P. Frasconi. 1994. "Learning Long-Term Dependencies with Gradient Descent Is Difficult." *IEEE Transactions on Neural Networks* 5 (2): 157–66. https://doi.org/10.1109/72.279181.

"BERT Vector Space Shows Issues with Unknown Words · Issue #164 · Google-Research/Bert." n.d. https://github.com/google-research/bert/issues/164.

Boag, Willie, and Hassan Kané. 2017. "AWE-CM Vectors: Augmenting Word Embeddings with a Clinical Metathesaurus." *arXiv:1712.01460 [Cs]*, December. http://arxiv.org/abs/1712.01460.

Boag, Willie, Elena Sergeeva, Saurabh Kulshreshtha, Peter Szolovits, Anna Rumshisky, and Tristan Naumann. 2018. "CliNER 2.0: Accessible and Accurate Clinical Concept Extraction," March. http://arxiv.org/abs/1803.02245v1.

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. “Enriching Word Vectors with Subword Information.” *Transactions of the Association for Computational Linguistics* 5: 135146. https://doi.org/10.1162/tacl_a_00051.

Bolukbasi, Tolga, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam Kalai. 2016. “Man Is to Computer Programmer as Woman Is to Homemaker? Debiasing Word Embeddings.” *arXiv:1607.06520 [Cs, Stat]*, July. http://arxiv.org/abs/1607.06520.

Bruni, E., N. K. Tran, and M. Baroni. 2014. “Multimodal Distributional Semantics.” *Journal of Artificial Intelligence Research* 49 (January): 1–47. https://doi.org/10.1613/jair.4135.

Campos, Ricardo, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. “YAKE! Collection-Independent Automatic Keyword Extractor.” In, 806–10. Springer International Publishing. https://doi.org/10.1007/978-3-319-76941-7_80.

Chang, Jonathan, Sean Gerrish, Chong Wang, Jordan L. Boyd-graber, and David M. Blei. 2009. “Reading Tea Leaves: How Humans Interpret Topic Models.” In, edited by Y. Bengio, D. Schuurmans, J. D. Lafferty, C. K. I. Williams, and A. Culotta, 288296. Curran Associates, Inc. http://papers.nips.cc/paper/3700-reading-tea-leaves-how-humans-interpret-topic-models.pdf.

Chapman, Wendy W., Will Bridewell, Paul Hanbury, Gregory F. Cooper, and Bruce G. Buchanan. 2001. “A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries.” *Journal of Biomedical Informatics* 34 (5): 301–10. https://doi.org/10.1006/jbin.2001.1029.

Chiu, Billy, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. “How to Train Good Word Embeddings for Biomedical NLP.” In, 166174. Berlin, Germany: Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-2922.

Cho, Kyunghyun, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. “Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).” In. Association for Computational Linguistics. https://doi.org/10.3115/v1/d14-1179.

Choi, Youngduck, Chill Yi-I Chiu, and David Sontag. 2016. “Learning Low-Dimensional Representations of Medical Concepts.” *AMIA Summits on Translational Science Proceedings* 2016 (July): 41–50. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5001761/.

Chollet, François et al. 2015. *Keras.* https://keras.io.

Clark, Jonathan H., Dan Garrette, Iulia Turc, and John Wieting. 2021. “CANINE: Pre-Training an Efficient Tokenization-Free Encoder for Language Representation.” https://arxiv.org/abs/2103.06874.

Collobert, Ronan, and Jason Weston. 2008. “A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning.” In, 160167. ICML ’08. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/1390156.1390177.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. “Natural Language Processing (Almost) from Scratch.” *Journal of Machine Learning Research* 12 (76): 2493–2537. http://jmlr.org/papers/v12/collobert11a.html.

Dai, Zihang, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. “Transformer-XL: Attentive Language Models Beyond a Fixed-Length Context.” In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978–88. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1285.

Danciu, Ioana, James D. Cowan, Melissa Basford, Xiaoming Wang, Alexander Saip, Susan Osgood, Jana Shirey-Rice, Jacqueline Kirby, and Paul A. Harris. 2014. “Secondary Use of Clinical Data: The Vanderbilt Approach.” *Journal of Biomedical Informatics* 52 (December): 28–35. https://doi.org/10.1016/j.jbi.2014.02.003.

Denny, Joshua C., Plomarz R. Irani, Firas H. Wehbe, Jeffrey D. Smithers, and Anderson Spickard. 2003. “The KnowledgeMap project: development of a concept-based medical school curriculum database.” *AMIA ... Annual Symposium proceedings. AMIA Symposium*, 195–99.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. “BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding.” *arXiv:1810.04805 [Cs]*, May. http://arxiv.org/abs/1810.04805.

Ettinger, Allyson. 2020. “What BERT Is Not: Lessons from a New Suite of Psycholinguistic Diagnostics for Language Models.” *Transactions of the Association for Computational Linguistics* 8 (July): 34–48. https://doi.org/10.1162/tacl_a_00298.

Ezen-Can, Aysu. 2020. “A Comparison of LSTM and BERT for Small Corpus.” *arXiv:2009.05451 [Cs]*, September. http://arxiv.org/abs/2009.05451.

Fares, Murhaf, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. “Word Vectors, Reuse, and Replicability: Towards a Community Repository of Large-Text Resources.” In, 271276. Gothenburg, Sweden: Association for Computational Linguistics.

Faruqui, Manaal, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. “NAACL-HLT 2015.” In, 16061615. Denver, Colorado: Association for Computational Linguistics. https://doi.org/10.3115/v1/N15-1184.

Faruqui, Manaal, and Chris Dyer. 2014. “EACL 2014.” In, 462471. Gothenburg, Sweden: Association for Computational Linguistics. https://doi.org/10.3115/v1/E14-1049.

Finkelstein, Lev, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. “Placing Search in Context: The Concept Revisited.” In *Proceedings of the 10th International Conference on World Wide Web*, 406–14. WWW ’01. New York, NY, USA: Association for Computing Machinery. https://doi.org/10.1145/371920.372094.

Firth, J. R. 1957. “A Synopsis of Linguistic Theory 1930-55.” 1952-59: 1–32.

Friedman, C., P. O. Alderson, J. H. M. Austin, J. J. Cimino, and S. B. Johnson. 1994. “A General Natural-Language Text Processor for Clinical Radiology.” *Journal of the American Medical Informatics Association* 1 (2): 161–74. https://doi.org/10.1136/jamia.1994.95236146.

Garla, Vijay N., and Cynthia Brandt. 2012. “Semantic Similarity in the Biomedical Domain: An Evaluation Across Knowledge Sources.” *BMC Bioinformatics* 13 (1): 261. https://doi.org/10.1186/1471-2105-13-261.

Gehrmann, Sebastian, Franck Dernoncourt, Yeran Li, Eric T. Carlson, Joy T. Wu, Jonathan Welt, John Foote, et al. 2018. “Comparing Deep Learning and Concept Extraction Based Methods for Patient Phenotyping from Clinical Narratives.” Edited by Jen-Hsiang Chuang. *PLOS ONE* 13 (2): e0192360. https://doi.org/10.1371/journal.pone.0192360.

Genthial, Guillaume, Lucas Liu, Barak Oshri, Kushal Ranjan, Richard Socher, and Christopher Manning. 2019. “CS224n: Natural Language Processing with Deep Learning Lecture Notes: Part VI Neural Machine Translation, Seq2seq and Attention,” 17. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/readings/cs224n-2019-notes06-NMT_seq2seq_attention.pdf.

Goldberg, Yoav. 2019. “Assessing BERT’s Syntactic Abilities.” *arXiv:1901.05287 [Cs]*, January. http://arxiv.org/abs/1901.05287.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT Press.

Gupta, Dilip, Melissa Saul, and John Gilbertson. 2004. “Evaluation of a Deidentification (De-Id) Software Engine to Share Pathology Reports and Clinical Documents for Research.” *American Journal of Clinical Pathology* 121 (2): 176–86.

https://doi.org/10.1309/e6k33gbpe5c27fyu.

Gupta, Manish, and Puneet Agrawal. 2021. “Compression of Deep Learning Models for Text: A Survey.” https://arxiv.org/abs/2008.05221.

Hansell, David M., Alexander A. Bankier, Heber MacMahon, Theresa C. McLoud, Nestor L. Müller, and Jacques Remy. 2008. “Fleischner Society: Glossary of Terms for Thoracic Imaging.” *Radiology* 246 (3): 697–722. https://doi.org/10.1148/radiol.2462070712.

Harris, Paul A., Robert Taylor, Robert Thielke, Jonathon Payne, Nathaniel Gonzalez, and Jose G. Conde. 2009. “Research Electronic Data Capture (REDCap)—A Metadata-Driven Methodology and Workflow Process for Providing Translational Research Informatics Support.” *Journal of Biomedical Informatics* 42 (2): 377–81. https://doi.org/10.1016/j.jbi.2008.08.010.

Henry, Sam, Clint Cuffy, and Bridget McInnes. 2017. “Evaluating Feature Extraction Methods for Knowledge-Based Biomedical Word Sense Disambiguation.” In, 272281. Vancouver, Canada,: Association for Computational Linguistics. https://doi.org/10.18653/v1/W17-2334.

Herdağdelen, Amaç, Katrin Erk, and Marco Baroni. 2009. “Measuring Semantic Relatedness with Vector Space Models and Random Walks.” In, 5053. TextGraphs-4. USA: Association for Computational Linguistics.

Hill, Felix, Roi Reichart, and Anna Korhonen. 2015. “SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation.” *Computational Linguistics* 41 (4): 665–95. https://doi.org/10.1162/coli_a_00237.

Hinton, G. E., J. L. McClelland, and D. E. Rumelhart. 1986. “Distributed Representations.” In, 77109. Cambridge, MA, USA: MIT Press.

Hliaoutakis, Angelos. 2005. “Semantic Similarity Measures in MeSH Ontology and Their Application to Information Retrieval on Medline.” PhD thesis. http://www.intelligence.tuc.gr/publications/Hliautakis.pdf.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9 (8): 1735–80. https://doi.org/10.1162/neco.1997.9.8.1735.

Honnibal, Matthew, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python.* Zenodo. https://doi.org/10.5281/zenodo.1212303.

Howard, Jeremy, and Sebastian Ruder. 2018. “Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).” In. Association for Computational Linguistics. https://doi.org/10.18653/v1/p18-1031.

“Hugging Face Models and Datasets.” n.d. https://huggingface.co/models.

Jiang, Steven, Weiyi Wu, Naofumi Tomita, Craig Ganoe, and Saeed Hassanpour. 2020. “Multi-Ontology Refined Embeddings (MORE): A Hybrid Multi-Ontology and Corpus-Based Semantic Representation Model for Biomedical Concepts.” *Journal of Biomedical Informatics* 111 (November): 103581. https://doi.org/10.1016/j.jbi.2020.103581.

Jiang, Zhenchao, Lishuang Li, Degen Huang, and Liuke Jin. 2015. “Training Word Embeddings for Deep Learning in Biomedical Text Mining Tasks.” In *2015 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 625–28. https://doi.org/10.1109/BIBM.2015.7359756.

Johnson, Alistair E. W., Tom J. Pollard, Lu Shen, Li-wei H. Lehman, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G. Mark. 2016. “MIMIC-III, a Freely Accessible Critical Care Database.” *Scientific Data* 3 (1): 1–9. https://doi.org/10.1038/sdata.2016.35.

Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. “Bag of Tricks for Efficient Text Classification.” *arXiv:1607.01759 [Cs]*, August. http://arxiv.org/abs/1607.01759.

Jurafsky, Daniel, and James H. Martin. 2020. *Speech and Language Processing : An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.* 3 (draft). https://web.stanford.edu/~jurafsky/slp3/.

Kalyan, Katikapalli Subramanyam, and S. Sangeetha. 2020. “SECNLP: A Survey of Embeddings in Clinical Natural Language Processing.” *Journal of Biomedical Informatics* 101 (January): 103323. https://doi.org/10.1016/j.jbi.2019.103323.

Kitaev, Nikita, Lukasz Kaiser, and Anselm Levskaya. 2020. “Reformer: The Efficient Transformer.” In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020.* OpenReview.net. https://openreview.net/forum?id=rkgNKkHtvB.

Lee, Jinhyuk, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. “BioBERT: A Pre-Trained Biomedical Language Representation Model for Biomedical Text Mining.” *Bioinformatics* 36 (4): 1234–40. https://doi.org/10.1093/bioinformatics/btz682.

Lehman, Eric, Sarthak Jain, Karl Pichotta, Yoav Goldberg, and Byron Wallace. 2021. “Does BERT Pretrained on Clinical Notes Reveal Sensitive Data?” In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 946–59. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2021.naacl-main.73.

Levy, Omer, and Yoav Goldberg. 2014. “Neural Word Embedding as Implicit Matrix

Factorization." In, 21772185. NIPS'14. Cambridge, MA, USA: MIT Press.

Levy, Omer, Yoav Goldberg, and Ido Dagan. 2015. "Improving Distributional Similarity with Lessons Learned from Word Embeddings." *Transactions of the Association for Computational Linguistics* 3 (December): 211–25. https://doi.org/10.1162/tacl_a_00134.

Li, Xiaoya, Xiaofei Sun, Yuxian Meng, Junjun Liang, Fei Wu, and Jiwei Li. 2020. "Dice Loss for Data-Imbalanced NLP Tasks." In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 465–76. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.45.

Lin, Chen-Tan, Marlene McKenzie, Jonathan Pell, and Liron Caplan. 2013. "Health Care Provider Satisfaction With a New Electronic Progress Note Format: SOAP Vs APSO Format." *JAMA Internal Medicine* 173 (2): 160. https://doi.org/10.1001/2013.jamainternmed.474.

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. "RoBERTa: A Robustly Optimized BERT Pretraining Approach." *arXiv:1907.11692 [Cs]*, July. http://arxiv.org/abs/1907.11692.

Manzini, Thomas, Lim Yao Chong, Alan W Black, and Yulia Tsvetkov. 2019. "NAACL-HLT 2019." In, 615621. Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1062.

Mao, Yuqing, and Kin Wah Fung. 2020. "Use of word and graph embedding to measure semantic relatedness between Unified Medical Language System concepts." *Journal of the American Medical Informatics Association: JAMIA* 27 (10): 1538–46. https://doi.org/10.1093/jamia/ocaa136.

McInnes, Bridget T., Ted Pedersen, and Serguei V.S. Pakhomov. 2009. "UMLS-Interface and UMLS-Similarity : Open Source Software for Measuring Paths and Semantic Similarity." *AMIA Annual Symposium Proceedings* 2009: 431–35. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2815481/.

Meng, Yu, Jiaxin Huang, Guangyuan Wang, Chao Zhang, Honglei Zhuang, Lance Kaplan, and Jiawei Han. 2019. "Spherical Text Embedding." In, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, 82088217. Curran Associates, Inc. http://papers.nips.cc/paper/9031-spherical-text-embedding.pdf.

Micikevicius, Paulius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, et al. 2018. "Mixed Precision Training." *arXiv:1710.03740 [Cs, Stat]*, February. http://arxiv.org/abs/1710.03740.

Microsoft Presidio. 2020. https://microsoft.github.io/presidio/.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Efficient Estimation of Word Representations in Vector Space." *arXiv:1301.3781 [Cs]*, September. http://arxiv.org/abs/1301.3781.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. "Distributed Representations of Words and Phrases and Their Compositionality." In, 31113119. NIPS'13. Red Hook, NY, USA: Curran Associates Inc.

Mnih, Andriy, and Geoffrey Hinton. 2008. "A Scalable Hierarchical Distributed Language Model." In, 10811088. NIPS'08. Red Hook, NY, USA: Curran Associates Inc.

Montani, Ines, and Matthew Honnibal. 2018. *Prodigy: A New Annotation Tool for Radically Efficient Machine Teaching. Artificial Intelligence*. Vol. to appear.

Neumann, Mark, Daniel King, Iz Beltagy, and Waleed Ammar. 2019. "ScispaCy: Fast and Robust Models for Biomedical Natural Language Processing." In. https://doi.org/10.18653/v1/W19-5034.

Pakhomov, Serguei, Greg Finley, Reed McEwan, Yan Wang, and Genevieve B. Melton. 2016. "Corpus Domain Effects on Distributional Semantic Modeling of Medical Terms." *Bioinformatics* 32 (23): 3635–44. https://doi.org/10.1093/bioinformatics/btw529.

Pakhomov, Serguei, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, and Genevieve B. Melton. 2010. "Semantic Similarity and Relatedness Between Clinical Terms: An Experimental Study." *AMIA Annual Symposium Proceedings* 2010: 572–76. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3041430/.

Paszke, Adam, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, et al. 2019. "PyTorch: An Imperative Style, High-Performance Deep Learning Library." In *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, 8024–35. Curran Associates, Inc. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Pedersen, Ted, Serguei V.S. Pakhomov, Siddharth Patwardhan, and Christopher G. Chute. 2007. "Measures of Semantic Similarity and Relatedness in the Biomedical Domain." *Journal of Biomedical Informatics* 40 (3): 288–99. https://doi.org/10.1016/j.jbi.2006.06.004.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. "Scikit-Learn: Machine Learning in Python." *Journal of*

Machine Learning Research 12: 2825–30.

Peng, Yifan, Shankai Yan, and Zhiyong Lu. 2019. "Transfer Learning in Biomedical Natural Language Processing: An Evaluation of BERT and ELMo on Ten Benchmarking Datasets." In, 5865. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/W19-5006.

Percha, Bethany, Yuhao Zhang, Selen Bozkurt, Daniel Rubin, Russ B. Altman, and Curtis P. Langlotz. 2018. "Expanding a Radiology Lexicon Using Contextual Patterns in Radiology Reports." *Journal of the American Medical Informatics Association* 25 (6): 679–85. https://doi.org/10.1093/jamia/ocx152.

Peters, Matthew, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. "Deep Contextualized Word Representations." In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics. https://doi.org/10.18653/v1/n18-1202.

Quint, Leslie E., Douglas J. Quint, and James D. Myles. 2008. "Frequency and Spectrum of Errors in Final Radiology Reports Generated With Automatic Speech Recognition Technology." *Journal of the American College of Radiology* 5 (12): 1196–99. https://doi.org/10.1016/j.jacr.2008.07.005.

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. "Language Models Are Unsupervised Multitask Learners." https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf.

Rao, Delip, and Brian McMahan. 2019. *Natural Language Processing with PyTorch: Build Intelligent Language Applications Using Deep Learning*. 1st edition. Sebastopol, CA: O'Reilly Media.

Řehůřek, Radim, and Petr Sojka. 2010. "Software Framework for Topic Modelling with Large Corpora." In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, 45–50. Valletta, Malta: ELRA.

Resnik, Philip. 1995. "Using Information Content to Evaluate Semantic Similarity in a Taxonomy." *arXiv:cmp-Lg/9511007*, November. http://arxiv.org/abs/cmp-lg/9511007.

Roberts, Kirk, Dina Demner-Fushman, Ellen M Voorhees, William R Hersh, Steven Bedrick, and Alexander J Lazar. n.d. "Overview of the TREC 2018 Precision

Medicine Track." In. https://dmice.ohsu.edu/hersh/trec-18-pm.pdf.

Roberts, Kirk, Ellen M Voorhees, and William R Hersh. n.d. "Overview of the TREC 2015 Clinical Decision Support Track." In. https://trec.nist.gov/pubs/trec25/papers/Overview-CL.pdf.

Rosario, Barbara, and Marti Hearst. 2001. "Classifying the Semantic Relations in Noun Compounds via a Domain-Specific Lexical Hierarchy." In. https://www.aclweb.org/anthology/W01-0511.

Rosario, Barbara, and Marti A. Hearst. 2004. "Classifying Semantic Relations in Bioscience Texts." In, 430es. ACL '04. USA: Association for Computational Linguistics. https://doi.org/10.3115/1218955.1219010.

Rose, Stuart, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. "Automatic Keyword Extraction from Individual Documents." In, 1–20. John Wiley & Sons, Ltd. https://doi.org/10.1002/9780470689646.ch1.

Rumelhart, David E., Geoffrey E. Hinton, and Ronald J. Williams. 1986. "Learning Representations by Back-Propagating Errors." *Nature* 323 (6088): 533–36. https://doi.org/10.1038/323533a0.

Sajadi, Armin. 2014. "Graph-Based Domain-Specific Semantic Relatedness from Wikipedia." In, edited by Marina Sokolova and Peter van Beek, 381–86. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-06483-3_42.

Sanh, Victor, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. "DistilBERT, a Distilled Version of BERT: Smaller, Faster, Cheaper and Lighter." https://arxiv.org/abs/1910.01108.

Savova, Guergana K, James J Masanz, Philip V Ogren, Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, and Christopher G Chute. 2010. "Mayo Clinical Text Analysis and Knowledge Extraction System (cTAKES): Architecture, Component Evaluation and Applications." *Journal of the American Medical Informatics Association* 17 (5): 507–13. https://doi.org/10.1136/jamia.2009.001560.

Schnabel, Tobias, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. "EMNLP 2015." In, 298307. Lisbon, Portugal: Association for Computational Linguistics. https://doi.org/10.18653/v1/D15-1036.

Schwartz, Ariel S., and Marti A. Hearst. 2003. "A simple algorithm for identifying abbreviation definitions in biomedical text." *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 451–62.

Sechidis, Konstantinos, Grigorios Tsoumakas, and Ioannis Vlahavas. 2011. "On the Stratification of Multi-Label Data." In, 145–58. Springer Berlin Heidelberg.

https://doi.org/10.1007/978-3-642-23808-6_10.

Shivade, Chaitanya, Preethi Raghavan, Eric Fosler-Lussier, Peter J. Embi, Noemie Elhadad, Stephen B. Johnson, and Albert M. Lai. 2014. "A review of approaches to identifying patient phenotype cohorts using electronic health records." *Journal of the American Medical Informatics Association: JAMIA* 21 (2): 221–30. https://doi.org/10.1136/amiajnl-2013-001935.

Si, Yuqi, Jingqi Wang, Hua Xu, and Kirk Roberts. 2019. "Enhancing Clinical Concept Extraction with Contextual Embeddings." *Journal of the American Medical Informatics Association* 26 (11): 1297–1304. https://doi.org/10.1093/jamia/ocz096.

Smith, Leslie N. 2018. "A Disciplined Approach to Neural Network Hyper-Parameters: Part 1 – Learning Rate, Batch Size, Momentum, and Weight Decay," March. http://arxiv.org/abs/1803.09820v2.

Socher, Richard, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. 2012. "CoNLL-EMNLP 2012." In, 12011211. Jeju Island, Korea: Association for Computational Linguistics. https://www.aclweb.org/anthology/D12-1110.

Soysal, Ergin, Jingqi Wang, Min Jiang, Yonghui Wu, Serguei Pakhomov, Hongfang Liu, and Hua Xu. 2017. "CLAMP – a Toolkit for Efficiently Building Customized Clinical Natural Language Processing Pipelines." *Journal of the American Medical Informatics Association* 25 (3): 331–36. https://doi.org/10.1093/jamia/ocx132.

Strubell, Emma, Ananya Ganesh, and Andrew McCallum. 2019. "Energy and Policy Considerations for Deep Learning in NLP." In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3645–50. Florence, Italy: Association for Computational Linguistics. https://doi.org/10.18653/v1/P19-1355.

Stubbs, Amber, Christopher Kotfila, and Özlem Uzuner. 2015. "Automated Systems for the de-Identification of Longitudinal Clinical Narratives: Overview of 2014 I2b2/UTHealth Shared Task Track 1." *Journal of Biomedical Informatics* 58 (December): S11–19. https://doi.org/10.1016/j.jbi.2015.06.007.

Stubbs, Amber, Christopher Kotfila, Hua Xu, and Özlem Uzuner. 2015. "Identifying Risk Factors for Heart Disease over Time: Overview of 2014 I2b2/UTHealth Shared Task Track 2." *Journal of Biomedical Informatics* 58 (December): S67–77. https://doi.org/10.1016/j.jbi.2015.07.001.

Su, Jianlin, Yu Lu, Shengfeng Pan, Bo Wen, and Yunfeng Liu. 2021. "RoFormer: Enhanced Transformer with Rotary Position Embedding." *arXiv Preprint arXiv:2104.09864*.

Tay, Yi, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. “Long Range Arena : A Benchmark for Efficient Transformers.” In *International Conference on Learning Representations*. https://openreview.net/forum?id=qVyeW-grC2k.

Tay, Yi, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. “Efficient Transformers: A Survey.” https://arxiv.org/abs/2009.06732.

Tayyar Madabushi, Harish, Elena Kochkina, and Michael Castelle. 2019. “Cost-Sensitive BERT for Generalisable Sentence Classification on Imbalanced Data.” In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, 125–34. Hong Kong, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/D19-5018.

Tenney, Ian, Dipanjan Das, and Ellie Pavlick. 2019. “BERT Rediscovers the Classical NLP Pipeline.” *arXiv:1905.05950 [Cs]*, August. http://arxiv.org/abs/1905.05950.

TH, Muneeb, Sunil Sahu, and Ashish Anand. 2015. “Evaluating Distributed Word Representations for Capturing Semantics of Biomedical Concepts.” In, 158163. Beijing, China: Association for Computational Linguistics. https://doi.org/10.18653/v1/W15-3820.

Turian, Joseph, Lev Ratinov, and Yoshua Bengio. 2010. “Word Representations: A Simple and General Method for Semi-Supervised Learning.” In, 384394. ACL ’10. USA: Association for Computational Linguistics.

Turney, Peter D. 2006. “Similarity of Semantic Relations.” *Computational Linguistics* 32 (3): 379416. https://doi.org/10.1162/coli.2006.32.3.379.

Uzuner, O., I. Goldstein, Y. Luo, and I. Kohane. 2008. “Identifying Patient Smoking Status from Medical Discharge Records.” *Journal of the American Medical Informatics Association* 15 (1): 14–24. https://doi.org/10.1197/jamia.m2408.

Uzuner, O., Y. Luo, and P. Szolovits. 2007. “Evaluating the State-of-the-Art in Automatic De-Identification.” *Journal of the American Medical Informatics Association* 14 (5): 550–63. https://doi.org/10.1197/jamia.m2444.

Uzuner, Ozlem. 2008. “Second i2b2 workshop on natural language processing challenges for clinical records.” *AMIA … Annual Symposium proceedings. AMIA Symposium*, November, 1252–53.

Uzuner, Özlem, Brett R South, Shuying Shen, and Scott L DuVall. 2011. “2010 I2b2/VA Challenge on Concepts, Assertions, and Relations in Clinical Text.” *Journal of the American Medical Informatics Association* 18 (5): 552–56. https://doi.org/10.1136/amiajnl-2011-000203.

Van Rossum, Guido, and Fred L. Drake. 2009. *Python 3 Reference Manual*. Scotts

Valley, CA: CreateSpace.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000–6010. NIPS'17. Red Hook, NY, USA: Curran Associates Inc.

Wang, Alex, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. "SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems." In *Advances in Neural Information Processing Systems*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/file/4496bf24afe7fab6f046bf4923da8de6-Paper.pdf.

Wang, Alex, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. "GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding." In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 353–55. Brussels, Belgium: Association for Computational Linguistics. https://doi.org/10.18653/v1/W18-5446.

Wang, Yanshan, Sijia Liu, Naveed Afzal, Majid Rastegar-Mojarad, Liwei Wang, Feichen Shen, Paul Kingsbury, and Hongfang Liu. 2018. "A comparison of word embeddings for the biomedical natural language processing." *Journal of Biomedical Informatics* 87 (November): 12–20. https://doi.org/10.1016/j.jbi.2018.09.008.

Warmerdam, Vincent, Thomas Kober, and Rachael Tatman. 2020. "Going Beyond T-SNE: Exposing Whatlies in Text Embeddings." In *Proceedings of Second Workshop for NLP Open Source Software (NLP-OSS)*, 52–60. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.nlposs-1.8.

Wiley, Laura K, Anushi Shah, Hua Xu, and William S Bush. 2013. "ICD-9 Tobacco Use Codes Are Effective Identifiers of Smoking Status." *Journal of the American Medical Informatics Association* 20 (4): 652–58. https://doi.org/10.1136/amiajnl-2012-001557.

Wolf, Thomas, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, et al. 2020. "Transformers: State-of-the-Art Natural Language Processing." In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 38–45. Online: Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlpdemos.6.

Xu, Kelvin, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention." In *Proceedings of the 32nd International Conference on Machine Learning*, edited by Francis Bach and David Blei, 37:2048–57. Proceedings of Machine Learning Research. Lille, France: PMLR.http://proceedings.mlr.press/v37/xuc15.html .

Ye, Cheng, and Daniel Fabbri. 2018. "Extracting Similar Terms from Multiple EMR-Based Semantic Embeddings to Support Chart Reviews." *Journal of Biomedical Informatics* 83 (July): 63–72. https://doi.org/10.1016/j.jbi.2018.05.014.

Yin, Zi, and Yuanyuan Shen. 2018. "On the Dimensionality of Word Embedding." In, 895906. NIPS'18. Red Hook, NY, USA: Curran Associates Inc.

Yu, Zhiguo, Byron C. Wallace, Todd Johnson, and Trevor Cohen. 2017. "Retrofitting Concept Vector Representations of Medical Concepts to Improve Estimates of Semantic Similarity and Relatedness." *Studies in Health Technology and Informatics* 245: 657–61. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6464117/.

Zaheer, Manzil, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, et al. 2020. "Big Bird: Transformers for Longer Sequences." In *Advances in Neural Information Processing Systems*, edited by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, 33:17283–97. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/c8512d14 2a2d849725f31a9a7a361ab9-Paper.pdf.

Zech, John, Margaret Pain, Joseph Titano, Marcus Badgeley, Javin Schefflein, Andres Su, Anthony Costa, Joshua Bederson, Joseph Lehar, and Eric Karl Oermann. 2018. "Natural Language-based Machine Learning Models for the Annotation of Clinical Radiology Reports." *Radiology* 287 (2): 570–80. https://doi.org/10.1148/radiol.2 018171093.

Zeng, Zexian, Yu Deng, Xiaoyu Li, Tristan Naumann, and Yuan Luo. 2019. "Natural Language Processing for EHR-Based Computational Phenotyping." *IEEE/ACM transactions on computational biology and bioinformatics* 16 (1): 139–53. https: //doi.org/10.1109/TCBB.2018.2849968.

Zhang, Jinghe, Kamran Kowsari, James H. Harrison, Jennifer M. Lobo, and Laura E. Barnes. 2018. "Patient2Vec: A Personalized Interpretable Deep Representation of the Longitudinal Electronic Health Record." *IEEE Access* 6: 65333–46. https: //doi.org/10.1109/ACCESS.2018.2875677.

Zhang, Yijia, Qingyu Chen, Zhihao Yang, Hongfei Lin, and Zhiyong Lu. 2019.

"BioWordVec, Improving Biomedical Word Embeddings with Subword Information and MeSH." *Scientific Data* 6 (1). https://doi.org/10.1038/s41597-019-0055-0.