WORD EMBEDDINGS FOR DOMAIN SPECIFIC SEMANTIC RELATEDNESS

by

Kyle Tilbury

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
October 2018

Table of Contents

List of Tables

List of Figures

Abstract

Word embeddings are becoming pervasive in natural language processing (NLP), with one of their main strengths being their ability to capture semantic relationships between words. Rather than training their own embeddings many NLP practitioners elect to use pre-trained word embeddings. These pre-trained embeddings are typically created and evaluated using general corpora. Thus, there is a deficiency in the understanding of their performance within a technical domain. In this thesis, we explore how the nature of the data used to train embeddings can affect their performance when computing semantic relatedness within different domains. The three main contributions are as follows. Firstly, we find that the performance of general pre-trained embeddings is lacking in the biomedical domain. Secondly, we provide key insights that should be considered when working with word embeddings for any semantic task. Finally, we develop new biomedical word embeddings and provide them as publicly available for use by others.

Acknowledgements

I would like to express sincere thanks to my supervisors Evangelos Milios and Meng He for their guidance and support. I would also like to thank Abidalrahman Moh'd and Aminul Islam for their valuable advice and feedback.

I acknowledge the financial support of the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Boeing Company.

Chapter 1

Introduction

The use of word embeddings (word vectors or word representations) has become prevalent in many natural language processing (NLP) tasks. These representations follow the distributional hypothesis idea that the meaning of a word can be derived from the company it keeps [20]. Word embeddings are typically learned from large unlabelled text copra. The learned representations are heavily dependant on the distributional statistics of the training corpus. A main strength is that embeddings capture the semantic relationships between words. Word representations are being being used in a wide variety of applications such as dependency parsers, deep learning approaches for generating image descriptions, sentiment classification, and semantic textual similarity [9, 23, 43, 42].

Many NLP practitioners rely on publicly available pre-trained word embeddings rather than training models themselves. Typically, these embeddings are pre-trained using massive general corpora, including Wikipedia and web crawl data. Many of these pre-trained representations perform well and competently transfer to new general domain problems [30]. However, they have not been thoroughly evaluated in specialised domains. Using these general pre-trained embeddings in an ad hoc manner in a technical domain could have implications, the extent of which are currently unknown.

To address this we evaluate popular state-of-the-art general pre-trained embeddings within the biomedical domain. We choose the biomedical domain due to the wealth of domain specific resources that are available such as publicly available data, semantic relatedness evaluation datasets, and pre-trained biomedical word embeddings. We propose and develop new biomedical embeddings using known strategies to improve word embeddings: the incorporation of phrase representations and the use of subword information from [31] and [2] respectively.

We measure the performance of the general pre-trained, biomedical pre-trained,

and the proposed word embeddings quantitatively using semantic relatedness datasets from both the general and biomedical domains and qualitatively using manual inspection approaches inspired by [24] and [11].

1.1 Contributions

This thesis focuses on word embeddings for domain specific semantic relatedness. We are interested in the semantic performance implications of the nature of the training data used to learn word embeddings. The main contributions of this thesis can be summarised here as:

- Evaluate state-of-the-art general pre-trained word embeddings in quantitative and qualitative semantic evaluations within the biomedical domain.
- Provide insights as to necessary precautions when training or working with word embeddings of any nature for semantic relatedness tasks. Firstly, regarding the training of word embeddings, we find the following: More training data does not always equate to better embeddings. A word embedding's semantic performance can be greatly impacted by how that embedding's training data was preprocessed. Secondly, we further the understanding into the problems that affect conventional embeddings. We find that the problem of multiple word senses being embedded into a single representation pervades both general and biomedical word embeddings trained with conventional methods.
- Develop biomedical word embeddings that outperform current publicly available pre-trained biomedical embeddings and make them available for future use.

Chapter 2

Background and Related Work

This chapter details the word embedding methods used throughput this thesis and provides instances where word embeddings are used in semantic tasks.

2.1 Word Embeddings

In this section, the three word embedding methods that pertain to this thesis are summarised following a brief overview of word embeddings.

A word embedding is a learned mapping of a word to a real valued vector. Embeddings are typically learned by optimising an auxiliary objective, such as predicting a word based on its context, in a large unlabelled corpus [54]. This follows from the distributional hypothesis that states words that are used and occur in the same contexts tend to purport similar meanings. As a result, word embeddings depend heavily on the distributional statistics of the corpus that they are learned from.

Although there are many approaches for learning word embeddings, three of the most popular methods, Word2vec [31, 29], GloVe [35], and fastText [2], are used in this thesis. Embeddings created using these methods all exhibit useful linear properties that capture meaningful semantic or syntactic concepts. For example, word embeddings capture the underlying concept that distinguishes terms. The distance between the vectors for "man" and "woman" is the same as the distance between the vectors for "King" and "Queen", showing that the intricate concept of gender is captured by the embeddings. Less intuitive morphological relationships are also captured. This is exhibited by the distance between "stronger" and "strongest" and the distance between "darker" and "darkest" being equivalent. Additionally, word embeddings enable the easy computation of the semantic relatedness between terms. This is done by simply computing the cosine similarity between the the vectors of two words.

2.1.1 Word2vec

Word2vec is one of the most well known word embeddings methods. It is a predictive model where word embeddings are learned by essentially predicting a word based on its context [29, 31]. There are two main "approaches" for Word2vec. The first is called continuous-bag-of-words (CBOW). CBOW computes the conditional probability of a target word given the context words surrounding it. The second approach, skip-gram, is the opposite of CBOW. Skip-gram predicts the surrounding context words given the central target word.

At a high level, either of these approaches can be conceptualised as a shallow neural network. With skip-gram, the architecture of the network can be considered as follows. The target word is the input layer. The word's surrounding context is the output layer. The middle, hidden layer, which has a dimension smaller than the input or output layers, is the embedded word representation. Essentially, both the word and its surrounding context are used to encode a representation of the word.

2.1.2 GloVe

GloVe is another popular approach for learning word embeddings [35]. Where as Word2vec and fastText are predictive models, GloVe is essentially a count-based model. GloVe embeddings are trained on the global word-word co-occurrence matrix, which tabulates how frequently words co-occur with one another in a given corpus. This co-occurrence count matrix is processed by normalising the counts and smoothing them, followed by factorisation to get lower dimensional representations. The underlying principle of GloVe is still derived from the distributional hypothesis. In a context, the co-occurrence ratios between two words are strongly connected to meaning. Despite being learned in an entirely different manner, GloVe and Word2vec embeddings are very similar.

2.1.3 fastText

The word embedding method used the most extensively in this thesis is fastText [2]. It can be viewed as an extension of Word2vec where word representations are replaced with the set of character n-grams appearing in that word. Like Word2vec, it has

both the CBOW and skip-gram models. The use of character n-grams, or subword information, allows the embeddings to perform better in morphological tasks as the n-grams approximate the morphemes of the words. This subword information also has the additional benefit of enabling the generation of representations for terms that were not in the training vocabulary. Since embeddings are learned for sets of characters n-grams, these n-gram embeddings can be used to build an embedding corresponding to the n-grams of an out of vocabulary (OOV) term.

2.2 Word Embeddings in Semantic Tasks

Word embeddings and word embedding based features have been used in many semantic based tasks. Semantic similarities and patterns of key phrases in scientific publications were explored using pre-trained word embedding models [25]. This work proposed Word Embedding Distance Pattern, which uses the head noun word embedding to generate distance patterns based on labelled keyphrases, as a feature to enhance conventional Named Entity Recognition. In another work, pre-trained word embeddings were used to construct sentence embeddings and the cosine similarity between sentence embeddings was used as a feature in a ridge regression model for sentence similarity [42].

Chapter 3

Methodology

This chapter begins with summaries of the pre-trained embeddings that we utilise throughout this thesis. This is followed by detailing the new proposed biomedical domain embeddings. Finally, we discuss using word embeddings for semantic relatedness and define the evaluation procedures.

3.1 Pre-trained Word Embeddings

This section outlines the five pre-trained word embeddings that are employed in this thesis. Three of these are popular, cutting edge pre-trained general embeddings trained using GloVe and fastText on massive general corpora. These corpora include Common Crawl, Wikipedia, the UMBC WebBase corpus, and the statmt.org news dataset. Common Crawl is a corpus containing petabytes of web crawl data collected. Wikipedia is an online open content encyclopedia. The UMBC WebBase corpus is a dataset of high quality English paragraphs containing over three billion words derived from a 2007 web crawl. The statmt.org news dataset consists of political and economic commentary crawled from news articles. The three embeddings trained on these corpora are the main focus of the evaluation of general word embeddings in a technical domain.

Since we are working in the biomedical domain we have access to available high quality pre-trained domain specific embeddings which are the two additional pre-trained embeddings compared in this work. Both of these biomedical embeddings are pre-trained on Pubmed which is a collection of citations for biomedical literature from life science journals and online books.

All of the pre-trained embeddings we use in this work are summarised in Table 3.1.

Embeddings	Vocabulary Size	Training Size (Tokens)	Reference
GloVe-Web	2.2 million	840 billion	[35]
fastText-Web	2 million	600 billion	[30]
fastText-WikiNews	1 million	16 billion	[30]
Word2vec-Pubmed-[10]	2.2 million	2.9 billion	[10]
Word2vec-Pubmed-[27]	2.7 million	3.6 billion	[27]

Table 3.1: Pre-trained Word Embeddings: Summary of the pre-trained word embeddings used in this work. The first three rows are general pre-trained embeddings and the last two rows are pre-trained biomedical embeddings.

3.1.1 Pre-trained GloVe Embeddings

The available general pre-trained GloVe embeddings are a popular resource for many machine learning and NLP practitioners. The specific pre-trained GloVe embeddings that we analyze in this work are:

- *glove.840B.300d.zip*: 2.2 million 300-dimensional word vectors trained on Common Crawl (840B tokens) which we refer to as **GloVe-Web**. Taken from [36].

3.1.2 Pre-trained fastText Embeddings

Recent work in pre-trained word embeddings combines a number of heuristics to train state-of-the-art, high quality, large pre-trained word emebddings which are made publicly available [30]. Pre-trained embeddings trained on a corpus of Wikipedia+News data and embeddings trained on Common Crawl web data are provided. We assess the quality of these embeddings when applied to relatedness in the biomedical domain. The embeddings we use are:

- *wiki-news-300d-1M-subword.vec.zip*: 1 million 300-dimensional word vectors trained with subword information on a Wikipedia dump from 2017, the UMBC webbase corpus and the statmt.org news dataset (16B tokens) which we refer to as **fastText-WikiNews**. Taken from [13].

- *crawl-300d-2M-subword.zip*: 2 million 300-dimensional word vectors trained with subword information on Common Crawl (600B tokens) which we refer

to as **fastText-Web**. Taken from [13].

3.1.3 Pre-trained Biomedical Embeddings

Within the biomedical domain there has been work to develop quality domain specific embeddings and provide high quality pre-trained biomedical word embeddings [10, 22, 27]. In this thesis, we use two different pre-trained trained biomedical embeddings. First, the embeddings developed in [10]:

- *PubMed-shuffle-win-2.bin*: 2.2 million 200-dimensional word vectors trained on Pubmed (2.89B tokens) which we refer to as **Word2vec-Pubmed-[10]**. Taken from [8].

The embeddings were trained using the Word2vec model with a substantial amount of work into parameter optimisation. For further details of the embedding training parameters we refer to [10].

Second, the more recent pre-trained biomedical embeddings from [27]:

- *pubmed2018_w2v_200D.bin*: 2.7 million 200-dimensional word vectors trained on Pubmed (3.58B tokens) which we refer to as **Word2vec-Pubmed-[27]**. Taken from [4].

The Word2vec-Pubmed-[27] embeddings are trained on the 2018 Pubmed baseline which is also used as an embedding training resource in this thesis. For further details on these embeddings refer to [5].

The main difference between these two pre-trained biomedical embeddings is that Word2vec-Pubmed-[10] embeddings are trained using Pubmed data from 2016, where as Word2vec-Pubmed-[27] embeddings are trained on the 2018 Pubmed baseline. Additionally, this Pubmed text was extracted and cleaned using different approaches for each of the embeddings.

3.2 Developing Domain Specific Biomedical Word Embeddings

This section delves into the development of domain specific word embeddings. First, we outline the corpora used to train embeddings in this thesis. Secondly, we describe a brief exploration of the effects data preprocessing can have on trained embeddings

and detail the data preprocessing used on the training corpora. Third, we enrich word embeddings by incorporating phrases into the learning process. Finally, we expound the training process with the fastText library [16, 15]. As stated above fastText word representations improve over more conventional word methods by incorporating subword information into the representations. We propose three different biomedical embeddings using two biomedical datasets and a combination of them. These are summarised in Table 3.2.

Embeddings	**Vocabulary Size**	**Training Size (Tokens)**
fastText-Pubmed	1.0 million	3.4 billion
fastText-OAS	3.9 million	10.9 billion
fastText-Pub+OAS	4.4 million	14.3 billion

Table 3.2: The Proposed Biomedical Word Embeddings: Summary of the biomedical word embeddings trained in this work. Three embedding models trained on:

(a) Pubmed citations containing the titles and abstracts from biomedical articles

(b) Pubmed Central Open Access Subset (OAS) plain text full biomedical articles

(c) Both Pubmed and OAS

3.2.1 Training Corpora

Two main corpora are used for training biomedical word embeddings in this thesis.

Pubmed is a database of citations that contain the titles and abstracts for more than 26 million articles [46]. Pubmed is distributed in XML format. We extract 27.84 million <ArticleTitle> and <AbstractText> from the articles in the 2018 baseline [49]. It should be noted that of these articles approximately 10 million articles had only titles and no abstracts. We then preprocess these extracted texts.

The Pubmed Central Open Access Subset (**OAS**) is, as of May 2018, a collection of around 2 milllion full-text open access biomedical and life science publications [48]. We use a version of OAS distributed in plain text format which contained approximately 1.64 million documents which we obtained from [47]. These texts have already gone through some unknown preprocessing steps to transform them into their plain text formats and we perform further preprocessing on them.

We additionally utilise a **Wikipedia** dataset to preform an experiment on preprocessing data for training word embeddings. An English Wikipedia dump from has 21-Mar-2018 with over 5 million articles was used [51].

3.2.2 Preprocessing

We first describe the preprocessing experiment on the Wikipedia dataset. Justified by the results of this experiment, we then detail the preprocessing steps used for the biomedical datasets.

Wikipedia Preprocessing Experiment

We experiment briefly with two preprocessing methods for fastText embedding training corpora.

The first method of preprocessing Wikipedia, **Preprocessing A**, is as follows. We utilise a modified version of the fastText Wikipedia preprocessing script [2]. The original version of the script, called *get-wikimedia.sh*, is available on the fastText code repository [14]. The modifications made to the script were superficial and do not change the preprocessing steps. The modified version of the script, called *preprocessing-wiki.sh* can be viewed in the code repository for this thesis on GitHub [44]. This script cleans the Wikipedia text, converts it to lower case, tokenizes, and splits and shuffles the sentences. A sample of the data after preprocessing can be seen in Figure 3.1.

```
little saigon
timezone cet
in extremely rare cases , severe reactions can happen including
carnivore a system developed by the us fbi to wiretap email .
living people
nifl premiership players
companies of azerbaijan
the body art . biennale de valencia , valencia , spain
establishments in scotland
marco antonio colonna ( march , )  cardinal - priest of ss . xii...
```

Figure 3.1: Sample of Preprocessed Wikipedia Text - **Preprocessing Method A**: A sample of 10 lines from the Wikipedia dataset after initial preprocessing with preprocessing method A. Note that one line has been truncated.

We now describe the second method of preprocessing Wikipedia, **Preprocessing B**. We make use of the WikiCorpus utility from *gensim* version 3.4.0 [39]. It is a tool expressly for the purpose of extracting a text corpus from a Wikipedia dump. We use the default parameters but without lemmatization. Text is tokenized and cleaned. A notable difference of this preprocessing method B from method A is the lack of sentence shuffling. A sample of the data after preprocessing method B can be seen in Figure 3.2.

anarchism is political philosophy that advocates self governed...
these are often described as stateless societies although...
anarchism holds the state to be undesirable unnecessary and...
anarchism is usually considered far left ideology and much of...
anarchism does not offer fixed body of doctrine from single...
collectivism strains of anarchism have often been divided into...
the word anarchism is composed from the word anarchy and the...
the first known use of this word was in various factions within...
the first political philosopher to call himself an anarchist...
on the other hand some use libertarianism to refer to...

Figure 3.2: Sample of Preprocessed Wikipedia Text - **Preprocessing Method B**: A sample of 10 lines from the Wikipedia dataset after initial preprocessing with preprocessing method B. Note that the lines of text have been truncated.

We show that preprocessing can be a very important step for some of the semantic relatedness methods that we use. During initial experimentation with generating fastText embeddings we discovered that the quality of the result can be very sensitive to preprocessing even when compared to what may seem to be a different but still reasonable means of preprocessing. We discuss this further and provide the results in Section 4.4.2.

Biomedical Corpora Preprocessing

We now summarise the preprocessing procedure for the Pubmed and OAS corpora. We use a different modified version of the script again from [2]. We treat this preprocessing script as a reasonable means of preprocessing text for training the proposed fastText embeddings for two reasons. First, due to the script being the better performing method in the preprocessing experiment. Second, the original script was used to preprocess data for many word embeddings across multiple languages. The

modified version for biomedical text we call *preprocessing-bio.sh*. It can be viewed on the GitHub repository for this thesis [44]. The modifications to the script for preprocessing the biomedical data were more substantial than the modifications for the Wikipedia preprocessing experiment. However, they were relatively simple in that it is solely modified to remove the Wikipedia specific steps. Samples of text from the preprocessed biomedical datasets are shown in Figure 3.3 and Figure 3.4. The corpora and their size after their respective preprocessing are shown in Table 3.3.

combination of topical methoxsalen and narrowband ultraviolet...
this paper aims to investigate the predictors of good outcome...
identify the types and prevalence of intestinal parasites among...
premature increases in amylase of postnatal rat parotid with...
clinical outcomes and radiologic results after cervical ...
pulmonary langerhans cell histiocytosis (plch) is usually...
a conventional patch - clamp technique was used to record the...
luteinizing hormone - releasing hormone receptor targeted ...
the pathognomonic triad of a rheumatoid pleural effusion (round...
evolutionary dynamics of human rotaviruses balancing reassortment...

Figure 3.3: Sample of Preprocessed Pubmed Text: A sample of 10 lines from the Pubmed dataset after initial preprocessing. Note that the lines of text have been truncated.

regional differences in prevalence of hiv - discordance in africa...
a minireview functions of the cumulus oophorus during oocyte...
the correlations of the pre derived distances with the ones ...
consisting of both pgs - pcl (with its collagenous
and traditional chinese medicine . phytother .
ann clin microbiol antimicrobann . clin . microbiol . antimicrobannals...
treatment results
harwood j james mt entomology in human and animal health new york...
upon review of our rhinoplasty osce , all residents and faculty...
as shown in the table , only bmi and severity of copd were...

Figure 3.4: Sample of Preprocessed OAS Text: A sample of 10 lines from the Pubmed dataset after initial preprocessing. Note that some of the lines of text have been truncated.

Corpus	Number of Tokens	File Size
PubMed	3.4 billion	19 GB
OAS	10.9 billion	58 GB

Table 3.3: Training Corpora: Training corpora and their respective sizes after initial preprocessing.

3.2.3 Learning Phrases

One approach that has been shown to enrich word embeddings is generating a token for each common phrase within the text [31]. The intuition is that words that appear frequently together and infrequently in other contexts can be combined into a phrase. This example helps illustrates the idea; 'New York Times' and 'Toronto Maple Leafs' would be replaced by unique tokens, such as "New_York_Times" and "Toronto_Maple_Leafs", in the training data, while a bigram like 'this is' would remain unchanged [31]. Even if the phrase representations are not used directly, they still improve the quality of the word embeddings overall as shown in [30].

We use the phrase (collocation) detection model from *gensim* version 3.4.0 [39]. This implementation has two methods of phrase detection inspired by [31] and [3]. We choose to use the method where phrases are detected using the simple statistical approach:

$$score(w_i,\ w_j) = \frac{(frequency(w_i w_j) - \delta) \cdot |V|}{frequency(w_i) \cdot frequency(w_j)}$$

where w_i and w_j are unigrams, $w_i w_j$ is the bigram of the two unigrams, $|V|$ is the size of the vocabulary, and δ is a discounting coefficient that stops phrases consisting of infrequent terms from being formed. A phrase is detected if $score(w_i,\ w_j)$ is above a defined threshold. Following phrase detection a phrase generation pass is performed over the data. Occurrences of the bigram $w_i w_j$ within the corpus are replaced with the unigram $w_i\_w_j$ where w_i and w_j are joined by an "_" character.

We perform three passes of phrase detection and generation over each of the preprocessed Pubmed and OAS datasets. The phrase generation script, called *GeneratePhrases.py*, can be seen in the code repository for this thesis on GitHub [44]. We use the default Phrases parameters except for *max_vocab_size*. We set

$$max\_vocab\_size = 200\ million$$

to avoid the chance of any words being pruned. Performing multiple passes over the data facilitates the creation of sensible phrases that can contain several words without greatly increasing the size of the vocabulary. For example, on the first phrase generation pass the phrase "new_york" could be formed out of the unigrams "new" and "york". During a subsequent phrase generation pass the phrase "new_york_city" could be formed out of the now previously generated unigram "new_york" and the unigram "city".

3.2.4 Training Embeddings

We train three word embeddings models:

- **fastText-Pubmed**
- **fastText-OAS**
- **fastText-Pub+OAS**

The fastText-Pubmed and fastText-OAS embeddings are trained on the Pubmed and OAS datasets as previously described. For fastText-Pub+OAS, we form the training dataset by first concatenating the Pubmed and OAS datasets. This Pub+OAS dataset is then shuffled and used as the training data for fastText-Pub+OAS. Recall that these embeddings are summarised in Table 3.2.

Parameter	**Argument**	**Value**
Minimum number of word occurrences	-minCount	5
Minimum length of char n-gram	-minn	3
Maximum length of char n-gram	-maxn	6
Loss function	-loss	ns (negative sampling)
Number of negatives sampled	-neg	5
Size of context window	-ws	5
Learning rate	-lr	0.05
Sampling threshold	-t	10^{-4}
Size of vectors	-dim	300

Table 3.4: Embedding Training Parameters: The parameters used to train the three proposed biomedical fastText skip-gram embeddings. For further detail on these parameters refer to [2, 15, 16].

We use the fastText library to train the embeddings [16]. As this thesis focuses on exploring how the nature of the training data affects the resultant embeddings, we preform no parameter optimisation for the proposed word embeddings. We simply employ the training parameters used to train a variety of fastText word embeddings in [2]. These parameters were used to train word embeddings on English, Czech, French, German and Spanish Wikipedia data, so we treat them as reasonable default parameters for our embeddings. Table 3.4 shows the embedding training parameters.

3.3 Word Embeddings for Semantic Relatedness

In this section, we describe the process of using embeddings to measure the semantic relatedness between words and phrases. In this thesis a phrase is considered to be a multi-word term like "chronic obstructive pulmonary disease" or "bee's knees". If a phrase was learned, as described in Section 3.2.3, then there is a single embedding corresponding to that phrase. If a phrase was not learned then we construct an embedding for it provided that its constituent words are in vocabulary.

Semantic relatedness between words or in vocabulary phrases, denoted as w_i and w_j, is computed by taking the cosine similarity between the vector representations of each term, defined as:

$$Rel(w_i,\ w_j) = CosSim(\mathbf{r}(w_i),\ \mathbf{r}(w_j))$$

where $\mathbf{r}(w_i) = \vec{w_i}$ is the vector representation corresponding to that word or multi-word term.

When a multi-word term that is out of vocabulary for a set of embeddings is involved, a single embedding for that phrase is generated using a simple additive approach. This additive approach for generating an embedding for multiple words was demonstrated in [31]. Word embeddings exhibit a type of linear property that makes it possible to combine them to form a meaningful phrase representation by an element-wise addition of their vector representations. An embedding for a multi-word term, denoted as p where $p = w_1, w_2, ..., w_n$, is generated by an element-wise addition of that phrase's constituent word embeddings, defined as:

$$\mathbf{r}(p) = \textstyle\sum_{w \in p}\ \mathbf{r}(w)$$

Then the semantic relatedness between a phrase and a word is computed by taking the cosine similarity between the composed phrase embedding and the word embedding, defined as:

$$Rel(p,\ w) = CosSim(\mathbf{r}(p),\ \mathbf{r}(w))$$

It should be noted that in some cases this approach may not be the best way to represent a phrase because many phrases have a meaning that is not a simple composition of the meanings of its individual words. This is demonstrated in [31]. For example, idioms like "hot potato" cannot be represented by the combination of the meanings of the constituent words. This is why it can be beneficial to directly learn embeddings for common phrases as described in Section 3.2.3.

Another way to generate a single embedding for a phrase is a simple unweighted averaging of the word embeddings of a phrase. For a multi-word term, where $p = w_1, w_2, ..., w_n$ and $|p|$ is the number of words in p , the averaging approach to generate is a single embedding for that phrase is defined as:

$$\mathbf{r}(p) = \frac{1}{|p|}\ \sum_{w \in p}\ \mathbf{r}(w)$$

This approach has been found to do well in representing short phrases [1]. However, when it comes to computing cosine similarity between vectors the averaging approach is equivalent to the additive approach. This is because cosine similarity is based on the the angles between vectors and these angles are not changed by the scaling of the vector's magnitude with the averaging approach.

Since both the additive and averaging approach have been shown to do well at representing phrases and are equivalent approaches when it comes to using the embeddings for evaluating relatedness of terms using cosine similarity, in this thesis we use the additive approach to generate embeddings for out of vocabulary phrases.

3.4 Evaluation

We evaluate the different word embeddings on semantic relatedness datasets in both the general and biomedical domain. We also perform qualitative experiments to further explore the differences between embeddings.

An evaluation of the pre-trained biomedical and the proposed biomedical embeddings within the general domain is important. Though the embeddings may be designed for use in technical domain specific applications, it may still be important that they capture the semantics of general English and be able to perform well in the general use case.

Additionally, evaluating the general pre-trained embeddings within the biomedical domain will give insights as to the ability of the general corpora to capture the specific language and nuances of the more technical biomedical domain.

It seems natural to have an implicit assumption that the embeddings trained with technical resources will contain information that enhances their performance in technical tasks. We examine the correctness of this assumption by evaluating the embeddings in the following ways:

- Semantic Relatedness Evaluation
    1. Gauging the performance of the general pre-trained, biomedical pre-trained and the proposed biomedical embeddings within the general domain.
    2. Thoroughly assessing the general pre-trained, biomedical pre-trained and the proposed biomedical embeddings in the biomedical domain.
- Qualitative Evaluation
    1. Examine similar terms to biomedical candidate terms for general pre-trained embeddings and the proposed embeddings.
    2. Explore how polysemous biomedical terms are embedded by general pre-trained and the proposed embeddings.

3.4.1 Semantic Relatedness Evaluation

In this thesis, semantic relatedness is defined as any semantic relationship between terms, for example “car” and “tire’ are semantically related. Semantic relatedness includes semantic similarity which is considered special case of relatedness, for example “car” and “automobile” are semantically similar.

The standard method for evaluation on semantic relatedness datasets is followed. Each dataset consists of many instances of the form:

$$(w_1,\ w_2,\ relatedness\ judgement)$$

where w_1 and w_2 are words and/or multi-word terms and *relatedness judgement* is a judgement of the relatedness, usually assigned by humans, between w_1 and w_2 that is treated as the gold standard of relatedness for that pair.

To evaluate a semantic relatedness measure on these datasets, the measure being evaluated is used to assign a relatedness score to each word pair. In other words, we get $Rel(w_1,\ w_2)$ for every instance in the dataset. This facilitates the direct testing of the correlation of the semantic relatedness measures with the human relatedness judgements. The correlation coefficient used in past works is somewhat inconsistent between the use of Spearman's ρ and Pearson's r. Both measures are reported in thesis.

In past works involving the evaluation of word embeddings on relatedness or similarity datasets, words that appear in the evaluation dataset but not in an embedding's vocabulary are typically omitted from the correlation calculation. Since different sets of embeddings may have varying vocabulary, this may not be the most fair way to compare the different embeddings. If certain words exist in one embedding but not in another, this may lead to computing the correlation between different sets of words for different embeddings, and the correlations should not be directly compared. This is especially important in the biomedical domain where the vocabulary differences between embeddings are drastic. To account for this, the intersection of vocabularies for all embeddings is computed. Evaluation is done on the pairs in the evaluation datasets that exist within the vocabulary intersection. This reduces the size of the evaluation datasets, so the number of pairs being evaluated, after removing the pairs not in the intersection of the vocabulary, are included where applicable.

Relatedness in the General Domain

First, ten datasets are used to assess each embedding's performance in the general domain. Using a wide array of general datasets gives insight into important factors that are not strictly domain related, but that should still be considered when using a method for semantic relatedness. These factors include the rareness of words in the

English language, the part of speech of the words, and the abstractness of the terms. The datasets are summarised in Table 3.5.

Dataset	Word Pairs	Reference
MC	30	[32]
MEN	3000	[6]
MTURK	287	[38]
MTURK	771	[19]
RG	65	[41]
RW	2034	[26]
SE	518	[7]
SL	999	[21]
WS	353	[17]
YP	130	[52]

Table 3.5: General Relatedness Evaluation Datasets. We provide these datasets in the code repository for this thesis on GitHub [44].

Miller and Charles (**MC-30**) is 30 noun word pairs with a human assigned similarity score [32]. These word pairs are for semantic similarity evaluation.

MEN-3000 consists of 3,000 word pairs together with human assigned relatedness judgement [6]. Pairs represent a balanced range of relatedness levels. The dataset contains examples involving both similarity and relatedness.

MTURK-771 contains 771 word pairs along with human assigned relatedness judgements [19]. Similarly, **MTURK-287** contains 287 word pairs with associated human relatedness judgements [38].

Rubenstein and Goodenough (**RG-65**) is the most classic dataset for word similarity tasks consisting of 65 word pairs with annotation of human similarity judgement [41].

The Rare Word dataset (**RW-2034**) is a similarity dataset with 2034 word pairs focusing on rare or morphologically complex words [26].

For a new semantic similarity evaluation dataset, the English word pairs from SemEval 2017 Task 2: Multilingual and Cross-lingual Semantic Word Similarity [7] were used. The pairs from both test and trial data were utilised. We refer to this set as **SE-518**. It consists of 518 pairs of words and multi-word terms where as all other general evaluation datasets typically only include pairs of single words.

SimLex-999 (**SL-999**) provides 999 pairs and their associated similarity scores [21]. The dataset is structured to focus evaluation of how well measures capture similarity based around conceptual distinctions including concreteness and part of speech. It contains a balanced selection of concrete, for example "dog" and "cup", and abstract concepts, such as "envy" and "deny". SimLex-999 is comprised of 666 noun to noun pairs, 222 verb to verb pairs, and 111 adjective to adjective pairs.

The WordSimilarity-353 Test Collection (**WS-353**) contains 353 pairs that contain instances of both relatedness and similarity type relationships [17]. The human scores estimate the relatedness of the word pairs.

Yang and Powers (**YP-130**) is a dataset of 130 verb pairs specifically designed for evaluating relatedness between verbs [52].

Relatedness in the Biomedical Domain

Second, the performance of each method on four biomedical domain semantic relatedness datasets is evaluated. These datasets are summarised in Table 3.6.

Dataset	Word Pairs	Reference
Mayo	101	[45]
MiniMayo	29	[34, 28]
UMNRel	587	[33]
UMNSim	566	[33]

Table 3.6: Biomedical Relatedness Evaluation Datasets. We provide these datasets in the code repository for this thesis on GitHub [44].

Medical Coders Set (**Mayo-101**) is a set of 101 medical concept pairs manually rated by medical coders for semantic relatedness. The concepts consist of single words and many multi-word terms [45].

Medical Coders High Reliability Subset (**MiniMayo-29**) is a subset of 29 medical concept pairs, comprised of both single and multi-word terms, manually rated by medical coders for semantic relatedness with high inter-rater agreement [34, 28].

Medical Residents Relatedness Set (**UMNRel-588**) is a set of 588 UMLS concept pairs manually rated for semantic relatedness [33]. Similarly, Medical Residents Similarity Set (**UMNSim-566**) is set of 566 Unified Medical Language System (UMLS)

concept pairs manually rated for semantic similarity [33]. In both of these datasets the concepts consist mainly of single words but multi-word terms do appear.

3.4.2 Qualitative Evaluation

Two qualitative experiments are also performed to further explore how the nature of the training data can affect the resulting embeddings. These evaluations highlight some overarching problems with conventional word embeddings. Furthermore, they emphasise the need for caution when using embeddings for semantic tasks without appropriate considerations.

Nearest Neighbours of Candidate Terms

The first qualitative evaluation in the biomedical domain is inspired by the approach in [24]. In their approach, the five most similar words, by cosine similarity, to some selected candidate terms are retrieved and manually inspected. The perceived quality of these similar words can provide insights into the quality of the embeddings with how well similar words are mapped within the embedding space.

The approach used in this thesis is as follows. We arbitrarily select one candidate biomedical term from each of the biomedical relatedness datasets in Table 3.6. The selected words, chosen such that they exist within the unmodified vocabulary of the embeddings being compared, can be seen in Table 3.7. For each candidate term we retrieve the 10 nearest neighbours, i.e. the 10 words with the highest cosine similarity across the entire vocabulary, for each set of embeddings being compared. Inspecting these nearest neighbour terms can help understand how the nature of the embedding's training data affects their semantics.

Word Embedding Sense Ambiguity

The second qualitative evaluation explores how the embedding of a term with multiple word senses is affected by the nature of the embedding training data.

Words can have multiple word senses, but conventional embedding training methods fail to distinguish between them. Consider the term "bat". The two most obvious senses of the word, taken from the Oxford English Dictionary, are:

Term	Origin Dataset	Qualitative Task
polyp	Mayo-101	Nearest Neighbours
antibiotic	MiniMayo-29	Nearest Neighbours
prozac	UMNRel-587	Nearest Neighbours
cardiomyopathy	UMNSim-566	Nearest Neighbours
culture	-	Ambiguous Word Sense
acid	-	Ambiguous Word Sense

Table 3.7: Candidate Terms for Qualitative Evaluations: Four arbitrarily selected biomedical candidate terms for use in the nearest neighbour qualitative evaluation. Two polysemous terms that relate to the biomedical domain for use in the ambiguous word sense qualitative analysis.

1. "An implement with a handle and a solid surface used for hitting a ball in sports."

2. "A mainly nocturnal mammal capable of sustained flight."

Due to conventional word embeddings learning only one representation per word, the resulting single representation for "bat" will be an amalgam which attempts to capture all meanings of the word within the training corpus.

The word sense that is captured by an embedding can be induced by looking at the nearest neighbours to the word. For example, if the nearest neighbours to the term "bat" are: batting, baseball, wiffle, corked, ball, etc. then it is clearly the sport implement word sense that is embedded. If the nearest neighbours to "bat" are: bats, leaf-nosed, free-tailed, roundleaf, noctule, etc. then the flying mammal sense is captured by the representation. However, the nearest neighbours will typically include all these terms as nearest neighbours to "bat" because there is only a single representation for the word.

There has been development of word embedding approaches that attempt to learn multiple representations per word [12, 40]. However, these approaches are more computationally intensive and are not so easily accessible when compared to the conventional word embedding approaches.

This qualitative evaluations examines whether the problem of polysemy and conventional word embeddings can be abated in a technical domain by using embeddings

trained in that domain. Consider the following, an NLP researcher is working on some problem dealing with the taxonomy of bats and wants to use word embeddings. Using general pre-trained embeddings could negatively affect their results as the embedding for "bat" represents a mixture of semantic meanings. However, if they used domain specific embeddings, trained without any text from the sports domain, then the embedding for "bat" would better represent the desired semantic meaning of the term. In other words, this evaluation assess whether domain specific embeddings may be more representative, than general trained embeddings, of the intended meaning of a polysemous word in that domain.

Like with the previous qualitative task, we perform this analysis by looking at the nearest neighbours to candidate terms. We arbitrarily selected two terms that have multiple word senses some of which may be related to the biomedical domain. These terms, "acid" and "culture", can be seen in Table 3.7. We then retrieve the 20 most similar terms, by cosine similarity, for each candidate term. With these most similar terms, terms that are simple alternative word forms of the candidate term are discarded. For example, with the candidate term "culture" we discard the most similar term "cultures". These terms do not help illustrate the word sense being embedded and thus are treated as extraneous.

Chapter 4

Results

This chapter begins with the presentation of general domain relatedness evaluation results. This is followed by the biomedical domain relatedness evaluation results. Then the findings for the qualitative evaluations are provided. Finally, the results regarding the training of embeddings are reported and discussed.

4.1 Evaluation in the General Domain

In this section, we report the results of the evaluation within the general domain. We begin with the examination of how well each set of embeddings captures general vocabulary. This is done by reporting the out of vocabulary (OOV) percentages of each set of embeddings for the general relatedness datasets. Following this is the semantic relatedness evaluation for all of the word embeddings in the general domain.

4.1.1 General Vocabulary Coverage

The general pre-trained embeddings are markedly ahead of most of the biomedical embeddings in terms of learned vocabulary when it comes to the general domain. The biomedical embeddings, both pre-trained and the proposed, using only Pubmed have the highest instances of OOV paris in the general evaluation datasets. The Word2vec-Pubmed-[10] embeddings have the the highest amount of OOV pairs across all embeddings. There is a clear link between the size of the training corpus and the number of OOV terms. For embedding training sizes see Table 3.1 and Table 3.2.

It is not necessary to have large amounts of general training data to achieve good vocabulary coverage in the general domain though. This is shown by the OOV results for fastText-Pub+OAS which covers very similar amount of vocabulary to fastText-WikiNews despite fastText-Pub+OAS being trained on biomedical data with 1.7 billion fewer tokens. The percentages of pairs that are OOV in the general evaluation datasets for each word embedding model are reported in Table 4.1.

Dataset	**GloVe-Web**	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed-[10]**	**Word2vec-Pubmed-[27]**	**fastText-Pubmed**	**fastText-OAS**	**fasttext-Pub+OAS**
MEN-3000	-	-	-	0.10%	0.03%	-	-	-
MTURK-287	-	-	0.70%	9.41%	0.70%	0.70%	0.35%	0.35%
MTURK-771	-	-	-	0.13%	-	-	-	-
RG-65	-	-	-	1.54%	-	-	-	-
RW-2034	1.72%	1.77%	3.29%	19.62%	14.70%	15.44%	7.03%	5.65%
SE-518	1.16%	0.97%	1.54%	8.69%	1.74%	2.32%	1.35%	1.16%
WS-353	-	-	-	2.27%	-	-	-	-

Table 4.1: General Out of Vocabulary Results: The percentage of pairs from each general evaluation dataset that are OOV corresponding to each set of embeddings. A "-" indicates that no pairs were OOV for that dataset and embedding. MC-30, SL-999, and YP-130 do not appear as none of their pairs were OOV for any set of embeddings.

4.1.2 General Domain Relatedness Results

The embeddings trained on general data perform the best when it comes to semantic relatedness evaluation between general terms. The state of the art pre-trained embeddings fastText-WikiNews and fastText-Web performed best. The biomedical embeddings all performed similarly in the general domain, with the proposed fastText-Pubmed performing best among biomedical embeddings by a small margin.

Though general training data is not necessary to obtain good vocabulary coverage of the general domain, it is the case that the lack of general training data has a measurable negative impact on an embedding's semantic performance in the general domain.

We provide the average correlation for all embeddings in Figure 4.1. This average is across all ten general relatedness datasets from Table 3.5. Both Spearman and Pearson correlation are illustrated. Recall that these results are for the pairs of terms that are not OOV across all sets of embeddings.

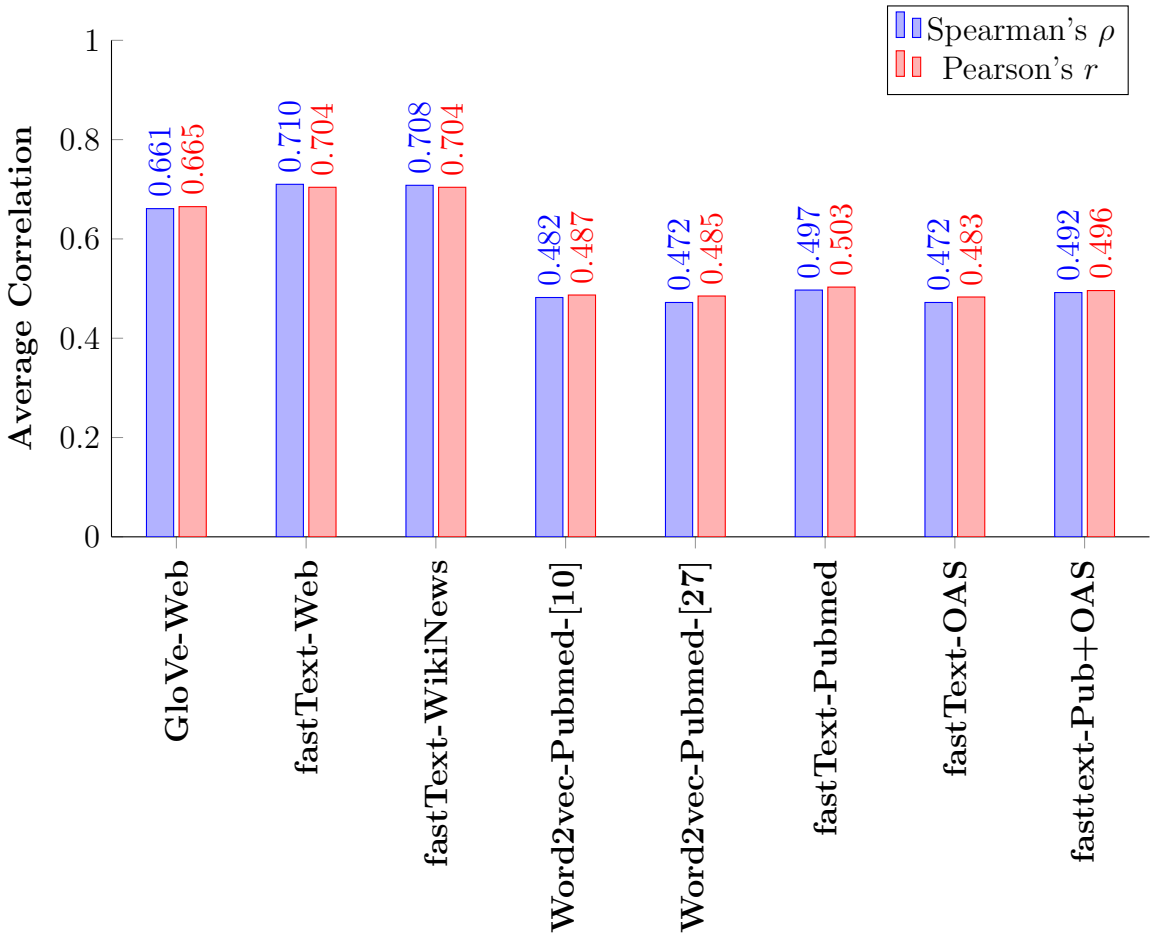

Figure 4.1: General Relatedness Results: The average correlation between the human relatedness judgements and the embedding based relatedness scores across all the general relatedness evaluation datasets.

4.2 Evaluation in the Biomedical Domain

This section begins with reporting the out of vocabulary percentages on the biomedical relatedness datasets for all methods. This allows us to asses how well each embeddings training data covers biomedical domain specific terms. We then present three relatedness evaluations between various sets of embeddings. Firstly, the relatedness correlation results of all embeddings, those trained in both the general and biomedical domains, in the biomedical domain. Secondly, a comparison between the relatedness performance of the pre-trained biomedical embeddings compared to the proposed biomedical embeddings. Finally, we report the results when using each fastText embedding to generate representations for any OOV terms, evaluating the

embeddings on the entirety of each relatedness dataset.

4.2.1 Biomedical Vocabulary Coverage

It is necessary to have biomedical training data to achieve high learned vocabulary coverage in the biomedical domain. The general domain GloVe-Web embeddings, trained on 840 billion tokens, and the general domain fastText-Web embeddings, trained on 600 billion tokens, have the best biomedical domain coverage of the general pre-trained embeddings. However, they only approximate the vocabulary coverage of the biomedical embedding with the most OOV terms, Word2vec-Pubmed. This is despite these general embeddings being trained with at least 300 times as many tokens as Word2vec-Pubmed which was trained with 2.9 billion tokens.

Preprocessing can also affect the learned vocabulary of a set of embeddings as shown by the differences in OOV between the three different embeddings trained on Pubmed data of similar size. This could also be explained partly by the different model and training parameters. We report the percentages of pairs that are OOV in the biomedical evaluation datasets for each word embedding model in Table 4.2.

Dataset	**GloVe-Web**	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed-[10]**	**Word2vec-Pubmed-[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
Mayo-101	13.9%	12.9%	18.8%	10.9%	6.9%	9.9%	7.9%	7.9%
UMNRel-587	15.3%	18.1%	39.0%	17.2%	5.1%	6.5%	5.6%	2.9%
UMNSim-566	14.1%	16.3%	35.7%	15.0%	3.7%	4.4%	4.4%	2.1%

Table 4.2: Biomedical Out of Vocabulary Results: The percentage of pairs from each biomedical evaluation dataset that are OOV corresponding to each set of embeddings. MiniMayo-29 does not appear as none of its pairs were OOV for any set of embeddings.

4.2.2 Biomedical Domain Relatedness Results

On the biomedical relatedness evaluation datasets, all biomedical embeddings outperform general trained embeddings. These results are shown in Figure 4.2. We provide the average correlation, both Spearman and Pearson, across all four biomedical relatedness datasets from Table 3.6. Recall that these results are for the pairs of terms that are not OOV across all embeddings.

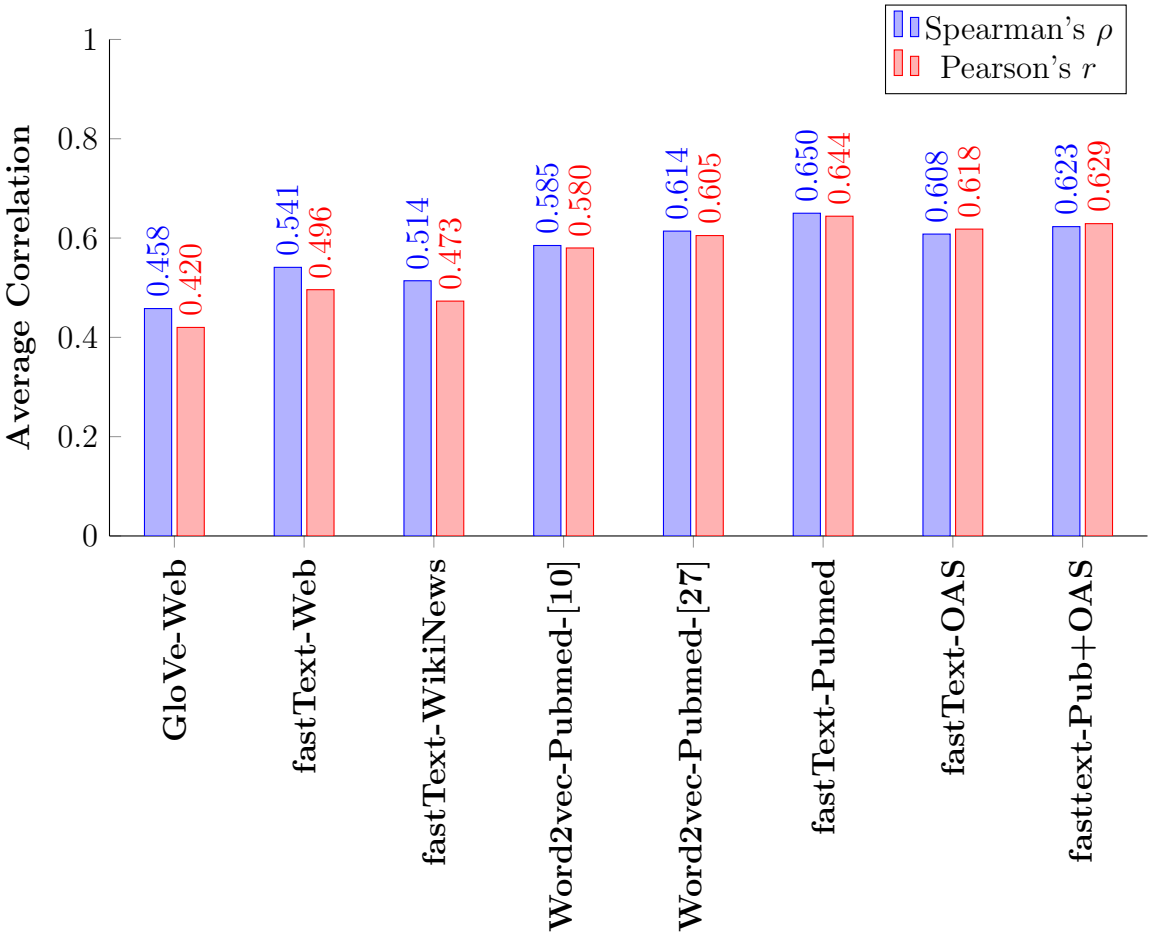

Figure 4.2: Biomedical Relatedness Results: The average of Spearman's ρ between the embedding based relatedness and the human assigned relatedness across all the biomedical relatedness evaluation datasets

4.2.3 Pre-trained Biomedical Embeddings vs. Proposed Biomedical Embeddings

The proposed fastText-Pubmed embeddings achieve the highest performance among all the biomedical embeddings with fastText-Pub+OAS achieving the second best performance. We report the full correlation results across the four biomedical datasets as well as the average of the general datasets, comparing biomedical embeddings in Table 4.3. Since the proposed embeddings outperform the currently available pre-trained biomedical embeddings, we will provide them as publicly available for use.

	Evaluation Dataset	*Pairs Evaluated*	**Word2vec-Pubmed[10]**	**Word2vec-Pubmed[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
Spearman	**Mayo-101**	88	0.484	0.512	0.556	0.522	0.536
	MiniMayo-29	29	0.749	0.773	0.804	0.752	0.771
	UMNRel-587	472	0.516	0.560	0.608	0.574	0.582
	UMNSim-566	471	0.574	0.622	0.659	0.627	0.640
	General Datasets	-	0.482	0.472	0.497	0.472	0.492
Pearson	**Mayo-101**	88	0.484	0.526	0.560	0.533	0.536
	MiniMayo-29	29	0.716	0.717	0.785	0.782	0.796
	UMNRel-587	472	0.520	0.567	0.606	0.576	0.584
	UMNSim-566	471	0.592	0.641	0.671	0.639	0.654
	General Datasets	-	0.487	0.485	0.503	0.483	0.496

Table 4.3: Biomedical Word Embeddings Compared: The pre-trained biomedical embeddings compared with the proposed biomedical embeddings. The best performance is highlighted in blue and the second best performance is highlighted in green.

4.2.4 fastText Generated Representations for OOV

Provided a fastText model was trained with subword information, the model can be used to generate representations for OOV words and phrases. This allows the computation of relatedness between some pairs that are not possible with other models. For example, some instances of OOV terms in the biomedical relatedness evaluation datasets are due to misspellings. Instances of these in Mayo-101 are in the terms "rheumatoid_arthriits", "buterfly_rash" and "varicsoe_vein".

Our proposed biomedical fastText embeddings are the least negatively impacted by generating representations for OOV pairs. This implies that, in addition to having best vocabulary coverage, the biomedical embeddings generate more accurate, in terms of semantics, representations for those OOV terms than the general embeddings

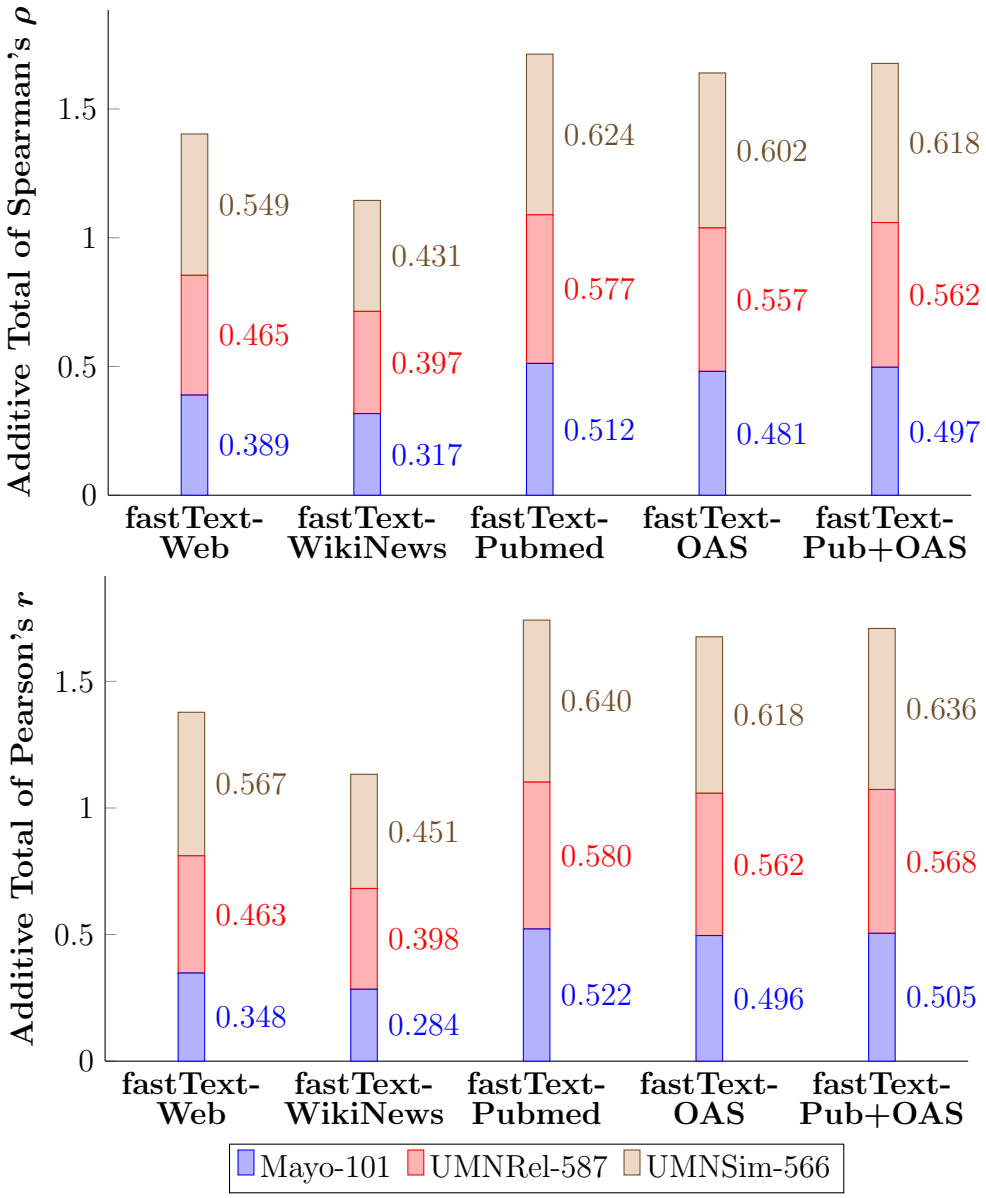

Figure 4.3: Biomedical Relatedness with OOV Generated Embeddings: The performance of the fastText embeddings when using each set of embeddings to generate representations (vectors, embeddings) for all terms. In other words, no OOV pairs in the evaluation datasets.

do. We present the correlation results after using each fastText embedding model to generate representations for all OOV terms or phrases in each biomedical relatedness dataset, except for MiniMayo-29 as all pairs were in vocabulary for every fastText embedding, in Figure 4.3. In other words we first used the fastText models to generate representations for OOV terms to obtain 100% vocabulary coverage for Mayo-101,

UMNRel-587, and UMNSim-566. Then the correlation between the human score and embedding based relatedness score is computed as normal.

Even though full vocabulary coverage can be achieved with general pre-trained fastText models, specialised domain trained embeddings still perform better. The net change in correlation results between the evaluation of embeddings on only in vocabulary pairs and between the fastText embeddings after generating representations for all OOV terms is shown in Table 4.4.

Correlation Coefficient	**fastText-Web**	**fastText-WikiNews**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
Spearman	-0.059	-0.105	-0.027	-0.013	-0.014
Pearson	-0.054	-0.094	-0.016	-0.006	-0.003

Table 4.4: Net Correlation Change with OOV Generated Embeddings: The net change in correlation results between: (a) Embedding relatedness for all pairs after generating representations for OOV terms and (b) Embedding relatedness for only in vocabulary pairs. In other words, this table reports correlation results of (a) subtract the correlation results of (b).

4.3 Qualitative Evaluation

For the qualitative evaluation, the proposed fastText-Pubmed embeddings are compared with the general pre-trained fastText-WikiNews. Though fastText-WikiNews performs slightly worse on the relatedness evaluations than fastText-Web, fastText-WikiNews is used for the following reason. The qualitative evaluation relies on retrieving nearest neighbours to a term and the nearest neighbours for most terms in fastText-Web are just various word forms (i.e. misspellings, containing addition punctuation, etc) of the query word. So, an infeasible amount of nearest neighbour terms would have to be retrieved to see any terms that are not the same as the candidate term. Thus, it would be difficult to infer anything meaningful from the results if was used. Firstly, the evaluation nearest neighbours to biomedical candidate terms is presented. Then, secondly, the evaluation of word embedding sense ambiguity is given.

4.3.1 Quality of Nearest Neighbours

As we are not biomedical domain experts, we cannot speak as to the quality of the most similar terms given by each embedding. However, we see very clear differences in the levels of specificity of the nearest neighbour terms between the embeddings. The nearest neighbours to the biomedical candidate terms for fastText-WikiNews and fastText-Pubmed are provided in Table 4.7. The fastText-WikiNews nearest neighbour terms are quite general when compared to fastText-Pubmed which gives more technical terms. This is best exemplified by the candidate term "prozac". With fastText-WikiNews, we can see notable generic terms like "meds" and non-closely related terms like "oxycontin".

4.3.2 Capturing Appropriate Word Senses

By looking at a sample of the nearest neighbours to "culture", we can infer what word sense is captured by the embeddings. The embeddings pre-trained on WikiNews embed the "the arts and other manifestations of human intellectual achievement regarded collectively" sense of "culture". Our proposed embeddings, trained on Pubmed, embed the "the cultivation of bacteria, tissue cells, etc., in an artificial medium containing nutrients" sense of "culture".

Through the examination of the nearest neighbours to "acid", we can infer what word sense is being embedded by each model. The problem of multiple word senses being tangled into the single embeddings of "acid" is evident for both the general and biomedical embeddings. We can see the "mineral (or inorganic)" sense of acid. This is illustrated by "hydrochloric" and "sulfuric" in the fastText-WikiNews nearest neighbours and by "hcl" in the fastText-Pubmed nearest neighbours. Additionally, we can see the "organic compound" sense of acid in both embeddings as well. Shown by "oxalic" and "butyric" in the fastText-WikiNews nearest neighbours and by "keto" and "amino" in the fastText-Pubmed nearest neighbours.

The nearest neighbours to "culture" and "acid" that provide an idea of the word senses embedded by fastText-WikiNews and fastText-Pubmed are presented in Table 4.5.

These results highlight two important things. First, we can see that if a word has multiple word senses, the sense that gets embedded can be affected by the nature

Embedding	Sample of nearest neighbours to "culture"
fastText-WikiNews	sub-culture, mass-culture, cultural, multi-culture, anti-culture, cross-culture, high-culture, multiculture, culturism, mono-culture, self-culture, non-culture, politics, sub-cultural, culture-oriented, culturology, sub-cultures, society
fastText-Pubmed	culturing, explant_cultures, culture_set, cultureable, culturepal, print_cultures, cultivation, cell_culture, subcultures, ecoculture, primoculture, cultured, batchculture, subculturing, incultured, primocultures, precultures
	Sample of nearest neighbours to "acid"
fastText-WikiNews	oxalic, oxoacid, butyric, acidic, hydrochloric, benzoic, uric, acetic, phosphoric, carbonic, sulfuric, ascorbic
fastText-Pubmed	acetic, keto, salicylic, amide, citric, glutamic, ester, fatty, amino, hyaluronic, hcl, caffeic, palmitic

Table 4.5: Word Embedding Sense Ambiguity: An example of how the nature of the training data can affect which sense of a word is embedded illustrated using the polysemous terms "culture" and "acid".

of the training data. This could have a positive or negative effect when using the embedding for semantic tasks depending on the intended sense of an embedding versus the actual embedded sense. Second, it is clear that the issue of multiple word senses being into embedded into a single vector pervades conventional embeddings in both the biomedical and general domain. This again could have consequences when it comes to using word embeddings for semantic tasks. If many senses are tangled into one vector then that vector will not do a perfect job in semantically representing the word in any context. These two insights highlight the need for consideration as to the word senses actually being embedded and caution when using word embeddings for particularly sensitive semantic relatedness tasks.

4.4 Training Embeddings

This section provides results and discusses insights pertaining to the process of training embeddings. First, the findings that training with the smaller Pubmed dataset

results in better embeddings than those trained on the larger OAS corpus. This contradicts the generally accepted notion that more training data always means better word embeddings. Second, the sensitivity of fastText to preprocessing is shown by the relatedness results for word embeddings which differ by only the training corpora preprocessing. Third and finally, the time cost of training embeddings is presented and discussed.

4.4.1 Training Data for Biomedical Embeddings

While in theory larger training corpora are thought to improve learned word embeddings, this is not what we found with our biomedical corpora and embeddings. Despite the OAS and Pubmed+OAS embeddings being trained with much larger corpora than the embeddings trained only on Pubmed, they do not perform as well for semantic relatedness in the general and biomedical domains. This is shown in Figure 4.1, Figure 4.2, Table 4.3, and Figure 4.3.

An explanation for this is due to the format of the OAS data. The data is full text of research articles, but they are provided without a thorough explanation of the format or text extraction process. Manual inspection of text within OAS reveals that a proportion of the text is not biomedical research content, i.e. references, non-relevant forewords, etc, and could be adding noise to the data thus negatively affecting learning.

This finding agrees with previous embedding results in the biomedical domain that found Word2vec embeddings trained on Pubmed performed better than embeddings trained on OAS in [10].

4.4.2 Caution on Preprocessing for Training Embeddings

Though both of the preprocessing methods may seem to be a reasonable approaches, one method performs consistently worse across all relatedness datasets. Recall both **Preprocessing A** and **Preprocessing B** from Section 3.2.2. Preprocessing A cleans the Wikipedia text by removing markup, converts to lowercase, tokenizes, and splits and shuffles sentences. Preprocessing B cleans the Wikipedia text by removing markup, removes punctuation, and tokenizing. Embeddings trained on the data preprocessed with Preprocessing B achieve lower performance than those trained

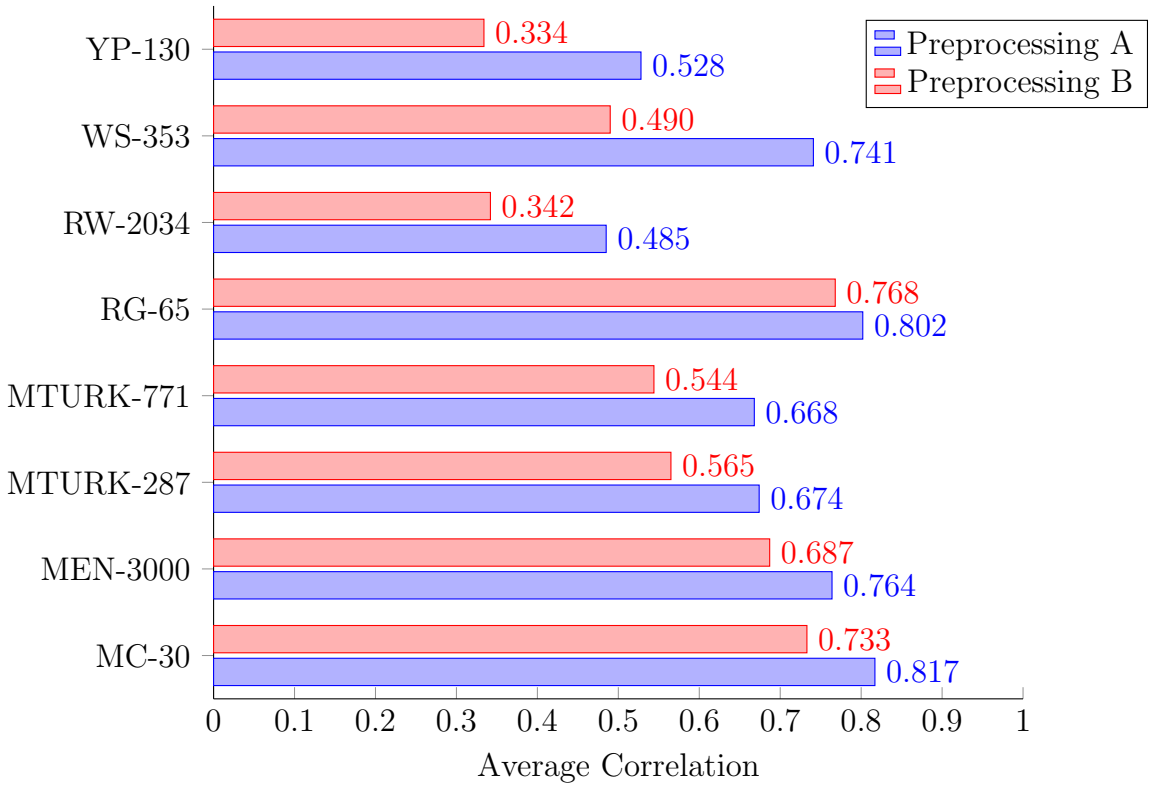

Figure 4.4: Preprocessing Effects on Embedding Relatedness Performance: The difference in Spearman's ρ between two sets of embeddings trained using the same parameters and training data, but with different preprocessing applied to the training data. **Preprocessing A** and **Preprocessing B** are detailed in Section 3.2.2.

on data using Preprocessing A. This is illustrated in Figure 4.4. Both embeddings used the Wikipedia corpus and the fastText model and parameters from Table 3.4 with the difference between the embeddings being the data preprocessing. The most significant difference between the two preprocessing methods is that Preprocessing B does not split and shuffle the sentences. Our intuition as to why the lack of shuffling impacts performance so negatively is as follows. If sentences are not split and shuffled, the word embedding model will be trained on chunks sentences from the same document. These sentences would be related to the semantics of that document. So training a word embedding model on a large amount of sentences in one semantic area could steer the model, too far, into that area. Then, when the sentences of the next document are reached, the embedding model would be dramatically steered in a different direction and so on. When training with shuffled sentences the model would not be biased in a similar way. Sentence order, and thus the semantics of

the sentences, would be presented in a random order after they are shuffled. This highlights the sensitivity of word embedding methods, fastText in particular, to data preprocessing.

4.4.3 Cost of Generating Domain Specific Embeddings

We show that with the availability of the high quality domain specific data from Pubmed and a modest time investment of 26 hours, it is possible to train embeddings that achieve state of the art in-domain performance on semantic relatedness tasks. We trained all models using fastText Version 0.1.0 run on a computer with an Intel Core i5-3550 CPU, 24 GB of RAM, running Ubuntu 18.04.1. The time cost of training our proposed embeddings is presented in Table 4.6. Whether training domain specific embeddings is worth it would depend highly on the application of the embeddings and data availability. As accessible tools like fastText make it possible to learn high quality embeddings quickly, careful consideration should be exercised before choosing to use off-the-shelf general pre-trained embeddings.

	fastText-Pubmed	**fastText-OAS**	**fastText-Pub+OAS**
Training Size (Tokens)	3.37 billion	10.92 billion	14.30 billion
Training File Size	19 GB	58 GB	74 GB
Embeddings Learned	1.01 million	3.99 million	4.39 million
User Time	372,923.47	1,138,744.02	1,516,415.28
System Time	1,546.02	5,850.14	7,244.92
Elapsed Time	26:06:42	79:52:10	106:17:12

Table 4.6: Embedding Training Times: The time taken to train each set of embeddings on each respective training corpus

Candidate Term	*Nearest Neighbours*	
	fastText-WikiNews	**fastText-Pubmed**
polyp	polyps	polyps
	cyst	polypoid
	polypoid	polypi
	polypus	adenoma_polyp
	adenoma	pseudopolyps
	tumor	polypectomy
	Polyps	sphenochoanal
	nodule	ethmochoanal
	fibroid	antrochoanal
	tumour	pedunculated
antibiotic	antibiotics	antibiotics
	non-antibiotic	antibiotical
	antibiotic-resistance	antimicrobial
	antimicrobial	antibioties
	antibiotic-resistant	polyantibiotic
	antibacterial	lactam_antibiotics
	Antibiotic	carbapenem
	pre-antibiotic	vancomycin
	post-antibiotic	beta_lactam
	antibacterials	fluoroquinolone
prozac	Prozac	fluoxetine
	valium	paroxetine
	anti-depressants	sertraline
	anti-depressant	citalopram
	meds	wellbutrin
	ritalin	norfluoxetine
	benzos	ssri
	anti-psychotics	sertralin
	antidepressants	anti_depressants
	oxycontin	zoloft
cardiomyopathy	cardiomyopathies	dilated_cardiomyopathy
	Cardiomyopathy	cardiomyopathie
	myopathy	myocarditis
	cardiomegaly	tachycardiomyopathy
	myopathies	dysplasia/cardiomyopathy
	angiopathy	endomyocardiopathy
	arrhythmia	hcm
	arrhythmogenic	cardiomyositis
	cardiomyocyte	leiomyopathy
	amyloidosis	endomyocardiopathies

Table 4.7: Biomedical Candidate Terms Nearest Neighbours: A sample of nearest neighbours to the biomedical candidate terms comparing the pre-trained fastText-WikiNews embeddings and the proposed fastText-Pubmed embeddings.

Chapter 5

Conclusion

This thesis focused on word embeddings for domain specific semantic relatedness. We evaluated state-of-the-art general pre-trained word embeddings in quantitative and qualitative semantic evaluations within the biomedical domain. We compared these general embeddings performance to domain specific biomedical embeddings and gained understanding as to the performance implications of the nature of the training data used to learn word embeddings. The results demonstrated that, for semantic relatedness in the biomedical domain, general pre-trained embeddings are lacking in performance when compared to their domain specific counterparts. In more wide-reaching terms, this supports the idea that these massive general pre-trained embeddings should not be hastily used on an ad hoc basis as embeddings that will transfer to any problem in any domain.

We gained and provided insights as to necessary precautions when training or utilising word embeddings for semantic relatedness. For training, an important outcome was the finding that the commonly touted "more training data = better embeddings" does not always hold true. This could be very closely linked to the importance of preprocessing training data appropriately. We highlighted that data preprocessing before training can have a large impact on an embedding's semantic performance. Our findings also provide insights regarding the general broad use of word embeddings. We showed why consideration should be given to what word sense a representation actually embeds.

Finally we developed our own biomedical word embeddings. Comparing our proposed biomedical word embeddings to cutting edge pre-trained biomedical embeddings demonstrated ours to be superior embeddings in semantic relatedness evaluation. Due to the fact that our biomedical embeddings outperform current publicly available pre-trained word biomedical embeddings, we make them available for future use by the public. A possible way to further improve these biomedical embeddings is

through optimisation of the training model. Additionally, we would like to evaluate the embeddings, both biomedical and general, in downstream NLP tasks, like question answering, to further assess the performance and differences among the embeddings.

Almost every NLP system based on deep learning uses word embeddings as input on some level. These deep learning based systems cover many areas including question answering [50], natural language inference [18], and image captioning [53]. Despite there being many methods for generating word embeddings that suit very specific problems, like multi-sense embeddings [37], pre-trained embeddings are a valuable resource. Pre-trained embeddings provide an quick and easy to use resource for all of these NLP systems. The insights provided in this thesis are applicable to the general process of pre-training embeddings. Since pre-trained embeddings are being widely used, it is necessary to understand them. To help further understanding of pre-training embeddings, future research into how corpus statistics directly affect embeddings or into the impact of processes like stemming or lemmatization can be done. Additionally, this thesis resulted in new state-of-the-art domain specific pre-trained embeddings. Providing more pre-trained embeddings for varying domains further increases the applicability and ease of use of pre-trained embeddings as a whole.

Bibliography

[1] Sanjeev Arora, Yingyu Liang, and Tengyu Ma. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*, 2017. `https://openreview.net/pdf?id=SyK00v5xx`.

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

[3] Gerlof Bouma. Normalized (pointwise) mutual information in collocation extraction. In *Proceedings of the German Society for Computational Linguistics and Language Technology*, pages 31–40, 2009.

[4] George Brokos and Ion Androutsopoulos. Athens University of Economics and Business Natural Language Processing Group: Biomedical word embeddings. `http://nlp.cs.aueb.gr/software.html`. Accessed: 2018-09-05.

[5] George Brokos and Ion Androutsopoulos. Athens University of Economics and Business Natural Language Processing Group: Biomedical word embeddings README. `https://ia802807.us.archive.org/21/items/pubmed2018_w2v_200D.tar/README.txt`. Accessed: 2018-09-20.

[6] Elia Bruni, Nam-Khanh Tran, and Marco Baroni. Multimodal distributional semantics. *Journal of Artificial Intelligence Research*, 49:1–47, 2014.

[7] Jose Camacho-Collados, Mohammad Taher Pilehvar, Nigel Collier, and Roberto Navigli. Semeval-2017 task 2: Multilingual and cross-lingual semantic word similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 15–26, 2017.

[8] Cambridge Language Technology Lab: BioNLP-2016. `https://github.com/cambridgeltl/BioNLP-2016`. Accessed: 2018-09-05.

[9] Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, 2014.

[10] Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. How to train good word embeddings for biomedical NLP. In *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, pages 166–174, 2016.

[11] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Learning topic-sensitive word represenations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, volume 2017, pages 441–447, 2017.

[12] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Learning topic-sensitive word representations. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers)*, pages 441–447. Association for Computational Linguistics, 2017.

[13] fastText: English word vectors. `https://fasttext.cc/docs/en/english-vectors.html`. Accessed: 2018-09-05.

[14] fastText: get-wikimedia.sh. `https://github.com/facebookresearch/fastText/blob/master/get-wikimedia.sh`. Accessed: 2018-09-10.

[15] fastText: Library for efficient text classification and representation learning. `https://fasttext.cc/`. Accessed: 2018-09-01.

[16] fastText Version 0.1.0. `https://github.com/facebookresearch/fastText/`. Accessed: 2018-03-15.

[17] Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. Placing search in context: The concept revisited. In *Proceedings of the 10th International Conference on World Wide Web*, pages 406–414. ACM, 2001.

[18] Yichen Gong, Heng Luo, and Jian Zhang. Natural language inference over interaction space. In *International Conference on Learning Representations*, 2018. `https://openreview.net/pdf?id=r1dHXnH6-`.

[19] Guy Halawi, Gideon Dror, Evgeniy Gabrilovich, and Yehuda Koren. Large-scale learning of word relatedness with constraints. In *Proceedings of the 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pages 1406–1414. ACM, 2012.

[20] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[21] Felix Hill, Roi Reichart, and Anna Korhonen. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695, 2015.

[22] Zhenchao Jiang, Lishuang Li, Degen Huang, and Liuke Jin. Training word embeddings for deep learning in biomedical text mining tasks. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on*, pages 625–628. IEEE, 2015.

[23] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[24] Omer Levy and Yoav Goldberg. Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 302–308, 2014.

[25] Sijia Liu, Feichen Shen, Vipin Chaudhary, and Hongfang Liu. MayoNLP at SemEval 2017 Task 10: Word Embedding Distance Pattern for keyphrase classification in scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 956–960, 2017.

[26] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013.

[27] Ryan McDonald, George Brokos, and Ion Androutsopoulos. Deep relevance ranking using enhanced document-query interactions. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.

[28] Bridget T McInnes, Ted Pedersen, and Serguei VS Pakhomov. UMLS-Interface and UMLS-Similarity: Open source software for measuring paths and semantic similarity. In *AMIA Annual Symposium Proceedings*, volume 2009, page 431. American Medical Informatics Association, 2009.

[29] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[30] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119, 2013.

[32] George A Miller and Walter G Charles. Contextual correlates of semantic similarity. *Language and cognitive processes*, 6(1):1–28, 1991.

[33] Serguei Pakhomov, Bridget McInnes, Terrence Adam, Ying Liu, Ted Pedersen, and Genevieve B Melton. Semantic similarity and relatedness between clinical terms: an experimental study. In *AMIA Annual Symposium Proceedings*, volume 2010, page 572. American Medical Informatics Association, 2010.

[34] Ted Pedersen, Serguei VS Pakhomov, Siddharth Patwardhan, and Christopher G Chute. Measures of semantic similarity and relatedness in the biomedical domain. *Journal of Biomedical Informatics*, 40(3):288–299, 2007.

[35] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[36] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global vectors for word representation. https://nlp.stanford.edu/projects/glove/. Accessed: 2018-09-01.

[37] Mohammad Taher Pilehvar, Jose Camacho-Collados, Roberto Navigli, and Nigel Collier. Towards a seamless integration of word senses into downstream nlp applications. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, volume 2017, page 18571869, 2017.

[38] Kira Radinsky, Eugene Agichtein, Evgeniy Gabrilovich, and Shaul Markovitch. A word at a time: computing word relatedness using temporal semantic analysis. In *Proceedings of the 20th International Conference on World Wide Web*, pages 337–346. ACM, 2011.

[39] Radim Řehůřek and Petr Sojka. Software framework for topic modelling with large corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. European Language Resource Association. http://is.muni.cz/publication/884893/en.

[40] Joseph Reisinger and Raymond J Mooney. Multi-prototype vector-space models of word meaning. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 109–117. Association for Computational Linguistics, 2010.

[41] Herbert Rubenstein and John B Goodenough. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633, 1965.

[42] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. DLS@CU at Semeval-2016 Task 1: Supervised models of sentence similarity. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 650–655, 2016.

[43] Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565, 2014.

[44] Thesis code repository on Github: Word embeddings for domain specific semantic relatedness. https://github.com/ktilbury/WORD-EMBEDDINGS-FOR-DOMAIN-SPECIFIC-SEMANTIC-RELATEDNESS. Accessed: 2018-09-16.

[45] University of Minnesota Pharmacy Informatics Laboratory: Medical coders set. http://rxinformatics.umn.edu/SemanticRelatednessResources.html. Accessed: 2018-09-01.

[46] U.S. National Library of Medicine: Pubmed. https://www.ncbi.nlm.nih.gov/pubmed. Accessed: 2018-09-06.

[47] U.S. National Library of Medicine: PMC FTP service. `ftp://ftp.ncbi.nlm.nih.gov/pub/pmc/oa_bulk/`. Accessed: 2018-05-06.

[48] U.S. National Library of Medicine: PMC Open Access Subset. `https://www.ncbi.nlm.nih.gov/pmc/tools/openftlist/`. Accessed: 2018-09-05.

[49] U.S. National Library of Medicine: Pubmed annual baseline. `ftp://ftp.ncbi.nlm.nih.gov/pubmed/baseline`. Accessed: 2018-05-06.

[50] Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 189–198, 2017.

[51] Wikimedia downloads. `http://dumps.wikimedia.your.org/`. Accessed: 2018-03-21.

[52] Dongqiang Yang and David Martin Powers. Verb similarity on the taxonomy of WordNet. In *Proceedings of the Third International WordNet Conference*, pages 121–128. Citeseer, 2006.

[53] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016.

[54] Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3):55–75, 2018.

Appendix A

Embedding Relatedness Full Results

A.1 General Domain

The full correlation results for all the pre-trained general embeddings, pre-trained biomedical embeddings, and the proposed biomedical embeddings on each of the ten general relatedness datasets, from Table 3.5, are reported in this section. As these correlation results are computed on the words that are in the intersection of all the embedding's vocabularies, i.e. for the pairs of terms that are not OOV across all sets of embeddings, the number of pairs evaluated is reported for each dataset. For the full general domain relatedness results using Spearman's ρ see Table A.1. For results using Pearson's r see Table A.2.

Evaluation Dataset	*Pairs Evaluated*	*General Pre-trained*			*Biomedical Pre-trained*		*Proposed Embeddings*		
		GloVe-Web	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed[10]**	**Word2vec-Pubmed[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
MC-30	30	0.786	0.850	0.886	0.670	0.474	0.562	0.387	0.494
MEN-3000	2997	0.805	0.815	0.803	0.591	0.625	0.669	0.646	0.654
MTURK-287	260	0.707	0.744	0.738	0.389	0.491	0.466	0.524	0.521
MTURK-771	770	0.716	0.752	0.729	0.507	0.524	0.534	0.567	0.562
RG-65	64	0.765	0.867	0.882	0.518	0.413	0.499	0.395	0.494
RW-2034	1617	0.462	0.577	0.545	0.377	0.392	0.411	0.430	0.426
SE-518	469	0.658	0.697	0.696	0.483	0.523	0.532	0.530	0.535
SL-999	999	0.408	0.471	0.441	0.322	0.329	0.318	0.311	0.313
WS-353	345	0.736	0.734	0.713	0.545	0.541	0.553	0.542	0.536
YP-130	130	0.572	0.592	0.647	0.423	0.413	0.422	0.390	0.387
	Average	0.661	0.710	0.708	0.482	0.472	0.497	0.472	0.492

Table A.1: Full General Relatedness Results - Spearman

Evaluation Dataset	*Pairs Evaluated*	*General Pre-trained*			*Biomedical Pre-trained*		*Proposed Embeddings*		
		GloVe-Web	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed[10]**	**Word2vec-Pubmed[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
MC-30	30	0.790	0.839	0.854	0.664	0.524	0.597	0.473	0.539
MEN-3000	2997	0.806	0.794	0.783	0.584	0.614	0.654	0.628	0.635
MTURK-287	260	0.752	0.754	0.749	0.444	0.546	0.528	0.571	0.567
MTURK-771	770	0.705	0.733	0.723	0.508	0.513	0.521	0.543	0.538
RG-65	64	0.772	0.841	0.866	0.502	0.439	0.514	0.423	0.490
RW-2034	1617	0.452	0.558	0.548	0.371	0.390	0.398	0.417	0.414
SE-518	469	0.653	0.682	0.672	0.492	0.525	0.536	0.532	0.535
SL-999	999	0.437	0.496	0.473	0.321	0.322	0.310	0.306	0.309
WS-353	345	0.732	0.718	0.713	0.539	0.534	0.539	0.531	0.528
YP-130	130	0.553	0.621	0.661	0.442	0.440	0.428	0.404	0.401
	Average	0.665	0.704	0.704	0.487	0.485	0.503	0.483	0.496

Table A.2: Full General Relatedness Results - Pearson

A.2 Biomedical Domain

The full correlation results on each of the four biomedical relatedness datasets, from Table 3.6, for all the pre-trained general embeddings, pre-trained biomedical embeddings, and the proposed biomedical embeddings are reported in this section. The number of pairs evaluated is reported per dataset as these correlation results are computed for the pairs of terms that are not OOV across all sets of embeddings. For the full biomedical domain relatedness results using Spearman's ρ see Table A.3. For results using Pearson's r see Table A.4.

Evaluation Dataset	*Pairs Evaluated*	*General Pre-trained*			*Biomedical Pre-trained*		*Proposed Embeddings*		
		GloVe-Web	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed[10]**	**Word2vec-Pubmed[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
Mayo-101	80	0.378	0.441	0.409	0.473	0.516	0.552	0.522	0.534
MiniMayo-29	29	0.470	0.583	0.593	0.749	0.773	0.804	0.752	0.771
UMNRel-587	347	0.462	0.544	0.500	0.513	0.539	0.577	0.534	0.546
UMNSim-566	353	0.522	0.594	0.553	0.603	0.628	0.666	0.624	0.640
	Average	0.458	0.541	0.514	0.585	0.614	0.650	0.608	0.623

Table A.3: Full Biomedical Relatedness Results - Spearman

Evaluation Dataset	*Pairs Evaluated*	*General Pre-trained*			*Biomedical Pre-trained*		*Proposed Embeddings*		
		GloVe-Web	**fastText-Web**	**fastText-WikiNews**	**Word2vec-Pubmed[10]**	**Word2vec-Pubmed[27]**	**fastText-Pubmed**	**fastText-OAS**	**fastText-Pub+OAS**
Mayo-101	80	0.373	0.402	0.370	0.477	0.526	0.553	0.531	0.532
MiniMayo-29	29	0.313	0.442	0.479	0.716	0.717	0.785	0.782	0.796
UMNRel-587	347	0.470	0.546	0.489	0.519	0.544	0.576	0.537	0.547
UMNSim-566	353	0.524	0.595	0.554	0.610	0.635	0.662	0.624	0.640
	Average	0.420	0.496	0.473	0.580	0.605	0.644	0.618	0.629

Table A.4: Full Biomedical Relatedness Results - Pearson

Appendix B

Training Better Embeddings

This appendix dives deeper into how various factors influence the semantic relatedness performance of the proposed embeddings. The following factors are examined:

1. Using subword information when training
2. Additional preprocessing of the training data
3. The number of phrase generation passes done over the training data

The starting point for all embedding training data in this appendix is the Pubmed dataset as described up to Section 3.2.2. In other words, it is the extracted preprocessed Pubmed corpus but with no phrase generation yet applied.

All word embedding models in this appendix are trained with the parameters from Table 3.4 unless otherwise specified.

B.1 Using Subword Information

In this section, the performance impact, on using an embedding for semantic relatedness, of training a word embedding model with subword information is evaluated.

Two word embedding models are trained:

- **fastText-pubmed-Subword**: Trained on Pubmed with fastText model parameters from Table 3.4.
- **fastText-pubmed-NonSubword**: Trained on Pubmed with fastText model parameters from Table 3.4 with the exception of not training with subword information.

The resulting embeddings are evaluated, as described in Section 3.4.1, on both the general relatedness datasets, Table 3.5, and the biomedical relatedness datasets, Table 3.6.

The following general pre-trained embeddings, from [2], are also compared:

- **crawl-300d-2M**: trained on Common Crawl (600B tokens). Retrieved from [13].
- **crawl-300d-2M-subword**: trained with subword information on Common Crawl (600B tokens). These are the embeddings used previously in this work that were referred to as fastText-Web. Retrieved from [13].

Dataset	**Pairs Evaluated**	**Net Correlation Change When Trained w/Subword** *Spearman*	*Pearson*
MC-30	30	-0.0234	-0.0141
MEN-3000	3000	-0.0003	-0.0005
MTURK-287	285	-0.0007	-0.0029
MTURK-771	771	0.0003	-0.0001
RG-65	65	-0.0250	-0.0135
RW-2034	1720	0.0023	0.0009
SE-518	506	0.0035	0.0031
SL-999	999	0.0006	0.0008
WS-353	353	-0.0023	-0.0015
YP-130	130	-0.0019	-0.0013
Mayo-101	91	-0.0029	-0.0027
MiniMayo-29	29	-0.0055	-0.0025
UMNRel-587	549	-0.0031	-0.0026
UMNSim-566	541	-0.0037	-0.0011
Average	-	**-0.0044**	**-0.0027**

Table B.1: Relatedness Impact - Subword Information (**Pubmed**): The net difference in correlation on semantic relatedness datasets between word embeddings trained with subword information versus those trained without. In other words, correlation results of **fastText-pubmed-Subword** subtract correlation results of **fastText-pubmed-NonSubword**.

Results

In this instance, the proposed biomedical embeddings trained using subword information perform slightly worse over all, in terms of correlation with the human semantic relatedness score, than the embeddings trained without the use of subword

Dataset	Pairs Evaluated	Net Correlation Change When Trained w/Subword	
		Spearman	*Pearson*
MC-30	30	-0.0024	-0.0101
MEN-3000	3000	-0.0308	-0.0307
MTURK-287	287	0.0079	0.0151
MTURK-771	771	-0.0106	-0.0010
RG-65	65	0.0078	-0.0070
RW-2034	1994	-0.0238	-0.0143
SE-518	512	-0.0227	-0.0197
SL-999	999	-0.0321	-0.0225
WS-353	353	-0.0624	-0.0295
YP-130	130	-0.0401	-0.0576
Mayo-101	88	0.0917	0.0521
MiniMayo-29	29	-0.0105	-0.0157
UMNRel-587	475	-0.0152	0.0011
UMNSim-566	469	-0.0216	-0.0151
Average	-	**-0.0118**	**-0.0111**

Table B.2: Relatedness Impact - Subword Information (**Web**): The net difference in correlation on semantic relatedness datasets between word embeddings trained with subword information versus those trained without. In other words, correlation results of **crawl-300d-2M-subword** subtract correlation results of **crawl-300d-2M**.

information. The net differences in correlation per dataset between fastText-pubmed-Subword and fastText-pubmed-NonSubword is reported in Table B.1. This net difference is calculated as correlation of fastText-pubmed-Subword subtract correlation of fastText-pubmed-NonSubword. Similarly, the net correlation change between crawl-300d-2M-subword and crawl-300d-2M is reported in Table B.2. Embeddings trained using subword information performing slightly worse over all is also the case with the embeddings pre-trained on web crawl data from Bowjanowski et al. [2].

The semantic relatedness performance decrease of training using subword information is not significant. What cannot be easily quantified is the added utility of being able to generate representations for OOV terms with a subword trained embedding model. We would argue that this additional functionality far outweighs the slight performance impact on the semantic relatedness evaluation datasets. Another area where the subword information trained embeddings could benefit is in morphology

dependant tasks as shown in [2].

B.2 Additional Training Data Preprocessing

In this section, the effects of additional training data preprocessing are investigated.

The biomedical preprocessing used, as described in Section 3.2.2, leaves various forms of punctuation in the training data. We explore the implications that this possible extraneous training data has on the semantic relatedness performance of a set of word embeddings. To do this, additional preprocessing is performed in order to remove the most obvious punctuation characters from the training data. The following characters are removed:

```
/$*.^|@#{}~&()_:;%+"='\'',‘><?!-
```

Then the embeddings trained on the data with punctuation are compared with embeddings trained on the data without punctuation. The two word embedding models trained are:

- **fastText-pubmed_Full**: Trained with fastText model parameters from Table 3.4 on Pubmed dataset preprocessed as described in Section 3.2.2.
- **fastText-pubmed_NoPunc**: Trained with fastText model parameters from Table 3.4 on Pubmed dataset preprocessed as described in Section 3.2.2 followed by additional processing to remove obvious punctuation.

To remove the punctuation, a simple shell script is utilised. This script, called *preprocessing-extra.sh*, can be seen in the code repository for this thesis on GitHub [44].

Results

In terms of semantic relatedness there is no significant difference between embeddings, but the embeddings trained on the data with no punctuation do perform better. The net correlation change between the embeddings trained with the additionally preprocessed, to remove punctuation, data and the embeddings trained on the full data is reported in Table B.3.

An interesting result is that fastText-pubmed_NoPunc has equal or better vocabulary coverage than fastText-pubmed_Full. This is despite fastText-pubmed_NoPunc

Dataset	Pairs Evaluated	Net Correlation Change w/Extra Preprocessing	
		Spearman	*Pearson*
MC-30	30	0.0162	0.0155
MEN-3000	3000	0.0034	0.0039
MTURK-287	285	-0.0001	0.0028
MTURK-771	771	0.0088	0.0024
RG-65	65	0.0078	0.0038
RW-2034	1720	-0.0063	0.0070
SE-518	506	0.0009	0.0019
SL-999	999	0.0021	-0.0041
WS-353	353	-0.0101	-0.0003
YP-130	130	0.0091	0.0003
Mayo-101	91	0.0113	-0.0013
MiniMayo-29	29	0.0122	-0.0022
UMNRel-587	549	-0.0014	-0.0089
UMNSim-566	541	-0.0002	0.0057
Average	-	**0.0038**	**0.0019**

Table B.3: Relatedness Impact - Additional Preprocessing: The net difference in correlation on semantic relatedness datasets between word embeddings trained with the additional preprocessing versus those trained without. In other words, correlation results of **fastText-pubmed_NoPunc** subtract correlation results of **fastText-pubmed_Full**.

		fastText-pubmed_Full	**fastText-pubmed_NoPunc**
	Training Tokens	3.38 Billion	2.88 Billion
	Embeddings Learned	999,653	879,757
Out of Vocabulary	MTURK-287	1%	1%
	RW-2034	16%	15%
	SE-518	2%	2%
	Mayo-101	10%	10%
	UMNRel-587	7%	6%
	UMNSim-566	4%	4%

Table B.4: Embedding Differences - Additional Preprocessing: The differences between the embeddings trained on the full data versus the data with no punctuation.

being trained on approximately 500 million fewer tokens and having 120,000 fewer words in its vocabulary. The differences between the embeddings in terms of number

of training tokens, number of embeddings learned, and out of vocabulary statistics are reported in Table B.4. Note that only the evaluation datasets with OOV pairs, i.e. MTURK-287, RW-2034, SE-518, Mayo-101, UMNRel-587, UMNSim-566, are reported.

B.3 The Number of Phrase Generation Passes

This section details the experimentation with how the number of phrase detection and generation passes on the training data can affect resulting word embedding performance.

All of the following embeddings are trained with the fastText model parameters from Table 3.4 on the Pubmed dataset with no punctuation as preprocessed in the previous section. The number of phrase detection and generation passes applied to the training data, as described in Section 3.2.3, is varied. Then the four following word embedding models are trained:

- **fastText-pubmed_Phrased0**: Zero passes of phrase generation on the training data.
- **fastText-pubmed_Phrased1**: One pass of phrase generation on the training data.
- **fastText-pubmed_Phrased2**: Two passes of phrase generation on the training data.
- **fastText-pubmed_Phrased3**: Three passes of phrase generation on the training data.

During each subsequent phrase detection pass the number of phrases detected increases. We show the number of phrases detected during each pass in Table B.5. Additionally we can see how many tokens were combined into phrases during each generation pass from the decreasing corpus size. Desipte the number of detected phrases increasing each pass, the number of tokens combined into phrases decreases slightly.

Phrasing Pass	Corpus Size (Tokens)	Phrases Detected
1	2.84 Billion	997,551
2	2.67 Billion	6,390,696
3	2.56 Billion	10,520,936

Table B.5: Phrase Generation Passes: The change in corpus size and the number of phrases detected during each pass.

Results

Phrase detection and generation has a positive impact on embedding semantic relatedness performance on average. The embeddings trained on the data with three phrase generation passes applied achieved a 0.11 stronger correlation, both Spearman and Pearson, than the embeddings trained on the data with no phrase generation applied. On some evaluation datasets, like MiniMayo-29 or MTURK-287, embedding relatedness performance got worse or stayed the same as the number of phrase generations applied to the training data increased. There is no obvious relationship between the nature of the relatedness evaluation dataset and whether or not phrase generation improves the performance on that dataset. This phenomena could simply be attributed to the properties of the Pubmed training dataset. The correlation results calculated using Spearman's ρ are reported in Table B.6. Similarly, the results with Pearson's r are reported in Table B.7.

Dataset	*Pairs Evaluated*	*Spearman's ρ*			
		Phrased0	**Phrased1**	**Phrased2**	**Phrased3**
MC-30	30	0.555	0.530	0.525	0.553
MEN-3000	3000	0.672	0.668	0.662	0.660
MTURK-287	285	0.460	0.455	0.471	0.461
MTURK-771	771	0.543	0.546	0.550	0.559
RG-65	65	0.470	0.484	0.492	0.471
RW-2034	1714	0.394	0.410	0.415	0.413
SE-518	506	0.522	0.527	0.523	0.539
SL-999	999	0.321	0.323	0.325	0.324
WS-353	353	0.519	0.528	0.526	0.516
YP-130	130	0.430	0.452	0.471	0.487
Mayo-101	91	0.571	0.612	0.660	0.671
MiniMayo-29	29	0.810	0.801	0.770	0.767
UMNRel-587	547	0.584	0.588	0.594	0.594
UMNSim-566	538	0.637	0.634	0.631	0.626
Average	-	**0.535**	**0.540**	**0.544**	**0.546**

Table B.6: Relatedness Impact - Phrase Generation Passes (**Spearman**): The semantic relatedness correlation results, using Spearman's ρ, for the embeddings with varying numbers of phrase generation passes applied to the training data.

Dataset	*Pairs Evaluated*	*Pearson's r*			
		Phrased0	**Phrased1**	**Phrased2**	**Phrased3**
MC-30	30	0.587	0.593	0.579	0.572
MEN-3000	3000	0.656	0.655	0.651	0.650
MTURK-287	285	0.525	0.520	0.531	0.526
MTURK-771	771	0.528	0.537	0.541	0.550
RG-65	65	0.486	0.490	0.499	0.469
RW-2034	1714	0.386	0.400	0.405	0.404
SE-518	506	0.530	0.540	0.539	0.551
SL-999	999	0.311	0.314	0.319	0.319
WS-353	353	0.509	0.514	0.517	0.509
YP-130	130	0.433	0.445	0.470	0.478
Mayo-101	91	0.580	0.607	0.659	0.673
MiniMayo-29	29	0.785	0.787	0.765	0.770
UMNRel-587	547	0.587	0.586	0.591	0.592
UMNSim-566	538	0.649	0.642	0.638	0.634
Average	-	**0.539**	**0.545**	**0.550**	**0.550**

Table B.7: Relatedness Impact - Phrase Generation Passes (**Pearson**): The semantic relatedness correlation results, using Pearson's r, for the embeddings with varying numbers of phrase generation passes applied to the training data.