Author:
Sutton, Adam J

Title:
Concepts in Word Embeddings

Theory and Applications

Concepts in Word Embeddings: Theory and Applications

By

ADAM SUTTON

Department of Electrical and Electronic Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

OCTOBER 2020

Word count: 32094

Abstract

Word embeddings have become a integral part of machine learning solutions for natural language processing (NLP) challenges. They are learned by taking the statistical co-occurrence information from a corpus and representing it in a dense vector (often hundreds of dimensions). This has resulted in many NLP tasks employing word embedding solutions to achieve state of the art performance metrics. The embedding process produces a challenge of understanding and interpreting these vectors as humans.

In this thesis I aim to explore how to understand and explain what word embeddings are representing and why they improve the performance of many tasks. I achieve this by utilising concepts, human understandable lists of words that aim to define an abstract object or class.

Using concepts I define a method to show that they remain present in word embeddings, I then use this method to measure word embeddings understanding of these same concepts. I then use these measurements to provide reasoning for deciding on a "better" word embedding algorithms, or finding corpora that when embedded better represents a domain (such as medicine).

I then define and use a method to measure a word's association to a concept, and by extension association to a bias. I show that unwanted biases (such as gendered, and racial) exist in word embeddings. I further show that gendered biases are representative of real world statistics. I then show that while removing these biases may seem like an ideal solution, it decreases a word embeddings representation of the same real world statistics. I also show colour biases and compare them to real world psychology studies, showing that pink is has a feminine bias and pink and blue are positively biased.

I apply the methods I have defined for measuring biases to historical corpora (1800 - 1959) and look to see if occupational words change in gender or emotional biases over time. I then look at the semantic changes of words over those 150 years. Finally, I see if there is any correlation between bias changes and changes in the semantic meaning of a word over time.

Dedication and Acknowledgements

In memory of Lilian Watson. I also dedicate this work to Katie and my family; mom, dad, Lee.

I would like to thank my supervisor, Nello Cristianini for his guidance. I would also like to thank Thomas Lansdall-Welfare for his valuable support in the first year of my studies. I would also like to thank the rest of the team I worked alongside in the intelligent systems lab at the University of Bristol; Carina, Nouf, Sheng, Saatviga, Fabon, Chris, and Teresa.

I would like to thank EPSRC and the Communications CDT at the University of Bristol for their funding, support, teaching, and guidance over the past four years. I would like to thank Suzanne for helping me with all queries and questions that I should have already known the answer to. I would like to thank Oliver Johnson and Johnathan Lawry for their reviews for the CDT and the University respectively.

The University of Bristol enabled me to attempt this work however many people not directly involved in the process helped me get through just as much any funding or teaching could. I would like to thank Dave for the first drink, and hope we've not seen our last. I would like to thank Justin for teaching me everything he knew nothing about, but... and I would like to thank Roger for being a much better dancer than Jon. I would also like to thank Jon, Michael, Owen, Chrys, and everyone else who came for anything social related to drinks.

Outside of the University I'd like to thank all of the friends I've made through work and during my undergraduate who impacted me. I would like to thank the friends from the University of Birmingham; Kamlesh, Dale, Matthew, Tola, Jawad, Charlie and the rest of the group. Thank you "The Island" for making me enjoy work life for a while until I realised I mainly enjoyed them; Eros (look after my grave), Jon, Himesh, Bhavin, Rohit, Sham, Annie, Katie, Manisha, and Mandeep. Thanks to the after work pitcher heroes who always respected going downstairs; Sean, Lacky, Termesh, Jit, Johal, Salah, Jack, Ruben, Chris and everyone else who only ever came out for one. To the European travellers of Eros, Jit, and Arun hopefully we can travel again. Thanks Lee, Jimi, Carl, Scott, and Damien for the blues adventures although you're wrong about football and everything else. Last but not least thanks to former work colleagues for making those days better, Amandeep, Frank, Harji, and Jaspreet.

AUTHOR'S DECLARATION

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

TABLE OF CONTENTS

List of Tables

Table **Page**

List of Figures

Figure **Page**

CHAPTER 1

INTRODUCTION

Machines can fundamentally only operate according to a single thing - binary. Thanks to the effort of numerous people and projects, software has enabled machines to process and represent binary in a variety of ways. Data can be encoded, stored, and transmitted depending on the requirements of the task and defined by a programmer. Character encoding methods enable text to be represented as binary numbers for the benefit of humans to better represent the processes of machines. Machines have no intuition of language, so while they are able to represent text this way, they hold no understanding towards the meaning of the text it can represent.

Shannon's coding theorem can be seen as a description of how much error free data can be transmitted over a lossy channel [88]. For both machines in this process, the data being sent is inconsequential. For traditional computing processes, only the users of those machines are interested in the encoding and decoding method for their data of interest. If a machine doesn't need a human understanding of human language, why would this be a concern?

Artificial intelligence (AI) is the field of developing machines to be able to complete tasks that would normally require the intelligence of a human. Natural Language Processing (NLP) is a subdivision of artificial intelligence that focuses on enabling machines to understand, process, and generate human language. There are a wide range of practical implementations of NLP, from

predictive text on smart phones to translating from one language to another.

In the earlier days of AI (and by extension NLP) the most common approach was using logic to achieve results in various tasks. In 1954 the Georgetown-IBM experiment took place to demonstrate machine translation [36]. They demonstrated automatic translation of sixty sentences from Russian to English. These sentences were translated using six rules and logical implementations. This approach had explicit rules defined using empirical lexical knowledge of both English and Russian, no information about languages learned by the machine itself. Perceived as a success at the time, this work showed little progress after this and was ultimately defunded due to lack of results. A main factor in the struggle with this method was scalability; more rules would be required to translate more than the original sixty sentences shown. The number of rules to achieve a “full” translation of a language was deemed infeasible as a solution. The sentences originally chosen were also hand picked to remove errors in translation due to ambiguity and other issues with words chosen or sentence structure. Additional sentences would require more rules, which would require more experts to define these rules, time to ensure that rules didn’t contradict each other, and to manage ambiguity in words and sentence structure.

J.R. Firth is known for the famous quote "You shall know a word by the company it keeps" [22]. Over time this was proven to hold true, and words that appeared in similar situations were found to be similar both syntactically and semantically. NLP saw progress in various areas utilising corpus statistics to aid in making predictions. Machine Learning (ML) is another sub field of AI, which focuses on using data to estimate a mathematical model to accurately represent a given task, and make predictions. Machine learning takes advantage of data to train a mathematical model. Due to the large volume of natural language from various sources, NLP tasks can utilise large sets of training data.

As time has gone by ML has shown to be well suited to certain NLP tasks [89, 91, 110]. When deciding on inputs for ML algorithms, NLP naturally uses representations of words, characters, or sequences. These representations generally must be a numeric representation, which can be as simple as a numbering tokens to more complex representations. Word embeddings are a more recent advancement of representations for words in ML applications. Word embeddings can generally be seen as dense vector representations of word co-occurrence statistics. These

embeddings can also be seen as an encoding of the meaning of words for a natural language.

Word embeddings provide two key advantages for ML and NLP applications; task performance, and training time. Many NLP tasks have seen increases in performances when using word embeddings as part of their pipeline. When training an AI for an NLP task, if using unlearned or statistical representations for word inputs then that AI must spend time learning what the meaning on input words and sentences mean from a corpus. This can take a long time for a given AI to learn. However a language model can be learned once for a corpus and be used in multiple different scenarios to cut down on training time.

Word embeddings can be seen as a method that given a word converts it to a real numbered vector representation commonly called a "word vector". The dimensionality of the vector can generally be defined as a hyper parameter of the embedding algorithm of choice. Word embeddings are trained by a number of algorithms, such as GloVe, fastText, and Word2Vec. These algorithms use the co-occurrence statistics of a corpus, to generate a vector space for all words or tokens that have been learned. These vectors are then used as static inputs for their ML task.

Word embeddings have also been found to have uses outside of improving performance in machine learning tasks. The representations of words can also be used to learn about the representation words in the corpus or by extension the language itself. Nearest neighbours of words are commonly words that share the same meaning. It would be expected for breeds of dogs for example to be close to each other. Another feature of word embeddings is linear sub-structures that are present within them. A common example of this is the following vector calculation:

$$man - woman = king - queen \tag{1.1}$$

where the vector representations for man , $woman$, $king$ and $queen$ are such that they demonstrate the gender relationship between all words. These substructures can be seen in many examples, such as capital cities and countries. Syntactic relationships are also similarly demonstrable in this way; $strong$ and $stronger$ will have a similar relationship as $weak$ and $weaker$.

In this thesis, I use a number of embedding algorithms (and a number of corpora) for multiple different experiments. The majority of experiments are done using "context-free" word

embeddings. The three context-free embeddings used are word2vec, GloVe, and fastText [62], [75], [61]. There are also "deep-contextual word representations", that are used in parts of this work. BERT is an example of these type of embeddings [20].

Word vectors from "context-free" embeddings can be seen as static representations of words. "Bark" will have a static representation that will contain a weighted representation of bark from a tree and a dog's bark. "Deep-contextual" embeddings try to alleviate this solution by having the representation vectors be generated depending on words that are around it. "The dog barks" and "the bark peeled off the tree" should contain different vector representations for the word "bark".

1.1 Concepts

As machines are unable to naively understand natural language, they do not understand concepts that are intuitive or easier for humans to learn in the real world. Concepts can aid humans in understanding new language, or enable faster learning with transferable knowledge. *The Classical Theory of Concepts* defines a concept as a set of necessary conditions that if something satisfies all those conditions, it can be seen as that concept. A common example of this is the concept of a bachelor, which must meet the conditions of being unmarried, and a man [57].

Cook claims in the dictionary of philosophical logic that there are two methods in defining concepts, extensional and intensional [14]. Cook defines that intensional definition for a concept follows classical theory and is specifying the key features something requires to be counted as part of that concept (i.e. all prime numbers are positive integers only have two factors, the number itself and one). The extensional counterpart definition for a concept is to list every object that belongs to that concept (for example listing all prime numbers would extensionally define all primes).

In this work I use the extensional definition of a concept to demonstrate the ability of AI to represent a corpus (or a language). I use this method as a measure of quality different for word embedding methods and corpora in different experiments. The key part of this method is representing concepts, specifically concepts that I define extensionally. I use word lists to train neural networks to identify a subset of a concept, and then try distinguish between the remaining

members of that concept and random negative samples. If a neural network performs well at this task then I call the concept “learnable”.

I aim to identify the statistical significance of all findings in these experiments to see how a word embedding affects the learnability of a set of concepts. I first look at concepts that are defined by the Linguistic Inquiry Word Count (LIWC) [74], and see if they are learnable in context-free word embeddings. I then compare the statistical performance of the word embedding algorithms fastText, word2vec, and GloVe using linear classifiers and show that under identical training conditions fastText better represents LIWC concepts.

Further to this, I look to apply my method to deep contextual embeddings to see if LIWC concepts are indeed represented in them also. I do this by extracting BERT embeddings and using a linear classifier comparable to those used in previous experiments. I show that due to BERTs deeper representation of words that BERT embeddings cannot be learned from with a linear classifier. I also show that a more expressive and deep classifier network is required when using BERT embeddings attempting to learn LIWC concepts.

In the final experiment with this proposed method I examine the performance of two embeddings trained using BERT but using different corpora; one being trained from Books and Wikipedia [20], and another being trained with medical textbooks (BioBert [50]).

The purpose of this experiment is to see if a domain specialised corpus can translate to a specialised word embedding and perform better at our classification task. LIWC is used to test the representation of general concepts, and to test medical domain specific knowledge I use ICD-10 diagnoses codes and their respective categories as concepts [70]. I show that a domain specific corpus performs significantly better than q general purpose corpus at identifying concepts suitable to the specific domain. The inverse is also found to be true, a more general corpus (trained on news articles and Wikipedia) is better at learning concepts that could be considered more general knowledge.

With concepts being represented within context-free embeddings I turn to look at how these concepts can be used to measure how words are associated to them. LIWC has many word lists that encapsulate concepts of emotions, constructs, and aspects of daily life. If there is a method to quantify how similar (or associated) words of interest are to these concepts then that method

could be used to enable users to gain further understanding of an embedding, or by extension its source corpus.

The method I propose is to use a simple normalised vector mean of all word vectors from a concept word list to represent a concept. Words of interest can then be compared to this mean vector by taking the cosine similarity to retrieve a score between -1 and 1 to show how similar a word is to a concept. To validate this method for every concept of interest I calculate a vector mean from half of the vectors that represent concept words and get the similarity scores for the remaining words and an equal number of random negative samples. With these scores I use ROC curves to show that vector means are able to capture the general concepts. I compare these results with randomly sampled word lists (or random concepts) to show the significance of these results. The mean of a set of vectors is a good representation of the concept associated to that set of vectors.

1.2 Biases in Natural Language

Biases which are present in the real world are also known to be represented within text. Writers knowingly or unknowingly project their biases onto text, word embedding algorithms then learn these biases along with a representation of text [12]. These biases represented in word embedding algorithms are then carried over into end systems that use word embeddings as their foundation. An example of this is male and female biases being present in large-scale studies of gender bias [23, 40, 49].

For my contribution to this work I propose a new version of the Word Embedding Association Test (WEAT) presented in [12]. This method is designed to identify and quantify biases within word embeddings by repeating experiments in applied psychology, namely the Implicit Association Test(IAT). I iterate on this method by using LIWC concepts that from previous work are already proved to accurately encapsulate concepts that are applicable to test for biases.

I then use my proposed method to identify and measure biases within a popular pre-trained word embedding provided by GloVe. The pre-trained embeddings source corpus is a combination of Wikipedia pages and news articles [38]. I repeat the experiments done by Caliskan using our

improved method (with the same words of interest) and these embeddings. I find that European-American names are more similar to positive emotions when compared to African-American names, male names are more associated with work while female names are more associated with family, and that the academic disciplines of science and maths are more associated with male terms than the arts, which are more associated with female terms. As a further extension to this work I look at gender biases within occupations and visualise them. These occupational biases are found to be correlated with UK employment statistics, showing a consistency between real world statistics and biases learned from the corpus.

Another challenge of bias within embeddings is how should they be addressed. While biases found here may be problematic they do accurately represent the corpus it is learned from. I propose a simple orthogonal projection of word embeddings to remove such biases and I then repeat the previous experiments and then monitor the changes in biases. We find a reduction in biases when performing the same experiments as previous. A potentially negative side effect of removing this bias is a decrease in correlation with UK employment statistics, showing the embeddings reflect the fact that employment statistics are not unbiased.

While biases may be seen as a negative in the end goal of intelligent systems, the presence of these biases may be useful to exploit for interdisciplinary work. The field of psychology has a challenge of resources when performing experiments, they can take a lot of time, manpower, or resources to perform. Word embeddings can be generated in a somewhat timely fashion with the only material requirements being computation and an ideal corpus to learn.

Colours and their biases are a point of interest for psychologists, with colour biases such as pink being perceived as a more female colour [77]. Further studies have shown pink and blue to be positive, along with pink to be a female colour. These results require a large effort from psychologists with testing and getting human participants to take part in testing.

Word embeddings can be measured for these biases of colours using the WEAT method proposed earlier. I can extract gender and emotional biases for all colours of interest to see if colours are highly associated to our concepts that represent masculinity, femininity, positive emotions, and negative emotions. We find that the majority of colours were highly associated individually with all four of these concepts, showing that the majority of colours did not present

a bias. We found “pink” to be an exception, with multiple experiments showing pink as highly biased towards femininity.

1.3 Semantic Change Detection in Changing Corpora

Over time natural language changes, new words are formed, meanings (or their semantics) of words change, and some words are removed from common vocabulary. Detecting and explaining these changes of words is a field of interest for historians, psychologists, and linguists. Using qualitative methods can be time and resource consuming due to specialised knowledge being required, and the large amount of text available in corpus resources. Natural language also changes depending on source, with domain specific words becoming more prevalent or words having different meanings in different corpora.

Biases in words are also known to change over time, or in different corpora. Gender bias historically was found to be arbitrary, and as time progressed the gender biases we see today became prevalent through popular culture [17], [18]. Occupational biases historically have been a point of contention and an example of gender biases affecting career opportunities, an example of this is unequal numbers of male and females in highly specialised roles [34].

Bias is not the only type of change for our words of interest, we wish to look at changes in the meaning of individual words. This has historically been done using statistical methods looking at a corpora of interest to measure how co-occurrences of these words change over time [44, 102]. This work is shown to be generally well correlated to human evaluations [31].

I will look at both the semantic and bias change of words of interest over time. I will focus on the time period of 1800 - 1959, and I will learn embeddings from British newspapers of the time [49]. I will first look at the historical bias of occupations, using the LIWC-WEAT method as defined in the previous chapter. I will then used the method proposed and demonstrated by Kulkarni to show the semantic change of words from decade to decade [48]. Finally I will look at the significant changes to biases in occupations and if this indicates a semantic change in meaning for those occupations.

Firstly I will look at the historical bias of occupations by extracting occupations from modern

and historical occupations [66, 69] and using the LIWC-WEAT to measure words similarities to concepts. I find that a large number of occupations become significantly more masculine over time and very few become feminine. “Doctor” becomes a more negative occupation over time, and “teacher” becomes more female over time.

The next part of the work looks at using the method of aligning word embeddings to the same vector space to find the semantic change of words over time. I use the method proposed by Kulkarni to achieve this alignment [48]. I first show and visualise the alignment of a word well known, “fathom” showing how its meaning changed significantly over 150 years. I then apply this to the same occupations that we found to be change in bias significantly. We do not find a correlation between a change in bias and a change in semantic meaning of a word.

CHAPTER 2

LITERATURE REVIEW

Word embeddings and earlier iterations have been present in artificial intelligence literature for a very long time. Over the years earlier iterations used statistical methods based on the occurrence of words in documents and moved onto statistical words co-occurring to explain a corpus. Counting word appearances in documents is computationally cheaper (in terms of memory and processing time) than word co-occurrences, however co-occurrence matrices maybe provide generally superior performances. That issue was attempted to be resolved with solutions such as dimensionality reduction tasks, or extracting weights from a classification task to describe the meaning of a word. The most recent discoveries in word embeddings now look at the context of words in the learning and generation of word embeddings, along with taking context sentences into account.

In this section I assume knowledge of linear algebra, probability theory, functions of a euclidean space, and machine learning. I also assume some knowledge of the fields of natural language, statistics. If parts of this literature review are difficult to understand then the methods section (Chap. 2) may help to provide additional context.

2.1 Word Embedding Algorithms

2.1.1 Historical Semantic Spaces

2.1.1.1 TF-IDF

TF-IDF stands for term frequency-inverse document frequency. TF-IDF is a statistical measure of how important a word is to a document from a collection of documents. A word increases in importance to a document the number of times it appears in that document, but will decrease when that word is common to multiple documents in a corpus.

There are two components that contribute to TF-IDF; how often a term appears in a document, and how specific a term is to a document. These may have slightly different methods of calculation for different purposes. A common definition for the term frequency is:

$$TF(t) = \frac{freq(t,d)}{\sum_{t' \in d} freq(t',d)} \tag{2.1}$$

where t is the term of interest, and d is the document of interest. A common definition for inverse-document frequency is:

$$IDF(t) = \log\left(\frac{|D|}{|d \in D : t \in d|}\right) \tag{2.2}$$

where D is the collection of documents. The bottom of the fraction can be read as the number of documents where the term t appears. This assumes that term t does appear in the corpus, or the denominator can add 1 to avoid division by zero errors. Any base logarithm is fine for the IDF, however it must be consistent when working between queries. TF-IDF can then be calculated by multiplying the two results as such:

$$\text{TF-IDF(t)} = TF(t) * IDF(t) \tag{2.3}$$

The larger the result from a TF-IDF calculation, the more important that term is to a given

document. A result of 0 shows that the specific term does not appear in the document of interest. TF-IDF is very commonly used in text mining and information retrieval [82].

TF-IDF are also used as a baseline in many NLP tasks [108]. Terms - or words can be represented as N dimensional vectors where each document within that vector will represent the TF-IDF for that document. These terms can then be used as input for machine learning tasks, similar to word embeddings.

TF-IDF has some issues compared to future algorithms. If a word / term was to appear in all documents, then it would provide a TF-IDF of of 0 for all documents. However if that term was used significantly more in some document than in all others that information would not be captured by TF-IDF. TF-IDF does not deal with words relationships with each other in sentences, focusing on terms of interest and their occurrence in documents. Finally no model is learned from this method, it is a pure statistical which will perform worse than future work that learns how to represent a corpus.

2.1.1.2 Word Co-occurrence

Reasoning for word co-occurrence can be easily and intuitively summed up with the sentence "words with similar meanings will appear in similar contexts". This logic makes for an easy explanation for semantic representations of words, using co-occurrence statistics. Bullinaria and Levy have performed a computational study on methods of Extracting semantic representations from word co-occurrence statistics [11].

Generating word co-occurrence matrices is done by counting how often a word occurs and how often it occurs near other words of a given corpus of text. For this we have a target word t to be a word of interest, context words c the words that are nearby the target word, and a window size w which describes how many context words c are around the target t are considered as co-occurring. Due to corpora sizes differing these counts are normalised to give a "basic semantic vector" for t:

$$p(c|t) = \frac{p(c,t)}{p(t)} = \frac{n(c,t)}{\sum_c n(c,t)} \tag{2.4}$$

where $n(c,t)$ is the number of times context word c occurs in the window of target word t. This

results in co-occurrence counts that have the properties of probabilities and can be treated as such. The frequencies of target and context words can be seen respectively as:

$$f(t) = \frac{1}{W}\sum_{c} n(c,t) \quad , \qquad f(c) = \frac{1}{W}\sum_{t} n(c,t) \tag{2.5}$$

where W is the window size. These frequencies are the total co-occurrence counts divided by the number of times each word gets counted within the window size. Individual word probabilities are calculated as:

$$p(t) = \frac{1}{NW}\sum_{c} n(c,t) \quad , \qquad p(c) = \frac{1}{NW}\sum_{t} n(c,t) \tag{2.6}$$

which is also dividing the word frequency by N, which is the number of words in the corpus. A window can be defined in multiple ways, as words that precede the target word, secede the target word, or a mixture of both. There can also be further changes to this, such as individual vectors for separate right and left counts or having diminishing returns for words further away from the target in the context window (potentially following a triangle or Gaussian function).

Regardless of the details in the definition of the word vectors generated by co-occurrences, Bullinaria and Levy measured the performance of these semantic representations in a number of tests. These tests range from testing the semantic or syntactic representation of these vectors, generally by having a correct nearest neighbour vs a number of negative samples where the test aims to find the correct word.

They use a number of different distance metrics as part of the test to compare performances of both the distance measurements, and the different vector representations. Some distance measures include Euclidean distance, cosine distance, Manhattan distance, Kullback–Leibler divergence, and point-wise mutual information (PMI). Results showed that Positive PMI Cosine distance was the best performing distance measure for three of the four tests, and the final test which looks at syntactic distance shows the the Ratios-Cosine (which is PMI, without logarithmic functions) as the best performing.

A key disadvantage is the memory required to create the matrix can be very large. If there are $\mathbf{V}$ words of interest of interest then a matrix must be constructed that is of size $\mathbf{V} \otimes \mathbf{V}$. In the

case of **V** containing hundreds of thousands of words this matrix may become quite large. This may also be quite redundant, as this matrix can be sparse with most words no co-occurring with each other. Another disadvantage is the performance of word co-occurrences in comparison to modern methods that take advantage of machine learning.

This work shows that statistical word co-occurrence information when extracted from a corpus can be used to extract semantic and syntactic representations. It also shows that different measurement functions can differ vastly in performance.

2.1.1.3 Latent Semantic Indexing

Latent Semantic Analysis is an older technique used for natural language processing [16]. It is based on the assumption that similar documents will have more occurrences of words with similar meaning. Latent Semantic Indexing is a by product, by using Latent Semantic Analysis that instead aims to identify the patterns in words in a text rather than context of documents.

The original aim was to effectively index and retrieve documents based of search terms. The initial problem had 1000-2000 document abstracts and 5000-7000 indexed terms, along with a 100 dimensional representation that includes both terms and documents in the same space.

First an initial matrix (X) of words and and contexts are generated from titles, abstracts or contents and count the appearance of words of interest. Then singular value decomposition (SVD) is used to factorise that matrix into singular values and fields:

$$X = T_0 S_0 D_0' \tag{2.7}$$

where T_0 is a term by dimension matrix, D_0 is a document by dimension matrix and S_0 is a diagonal matrix of singular values. These values are then ordered by size from the first cell in descending order along the diagonal. The largest values of this matrix contribute to the evaluation of X. Using this knowledge, the dimensions of the decomposition can be reduced with a loss of accuracy, but a simpler model:

$$X \approx \hat{X} = TSD' \tag{2.8}$$

Latent Semantic Indexing was then used with three tasks, comparing the similarities of two terms, two documents, and the association between a term and a document. For example the similarity of two terms can be found by calculating the dot product between two row vectors of $\hat{X}$. This shows the rate of occurrence between two terms within the corpus studied.

While successful at the time for categorising relevant documents and terms along with lowering the dimensional space there was a downside to this process. The major flaw is polysemy - where a word may have multiple meanings (e.g. "crane"). These words only have a single representation in the space, and therefore their representation may not be a true representation of any single meaning within the matrix. Another disadvantage of the model is choosing an appropriate size for the approximation matrix, choosing a size too small will lose valuable information while a size too large will add more noise. This issue might also be different between different matrices of interest.

2.1.1.4 Vector Space Models

Turney and Pantel survey various vector space models for semantics [97]. They first look at term-document matrices and their relation to semantic representation. They then look at the word context matrix, and finally the pair-pattern matrix.

Counting the number of appearances of terms (commonly - but not always words) appearing in each document can generate a matrix $\mathbf{X}$ with n documents and m words of interest such that $\mathbf{X} \in \mathbb{R}^{m \times n}$. Where each of m rows in $\mathbf{X}$ represents a unique word or term, and each of n columns represents a document. The contents of the matrix can be normalised to represent the frequency of terms appearing in documents, generally speaking the matrix will be sparse. Document similarity (the similarity between two columns in $\mathbf{X}$) is shown to have some success in search engines, however there are draw backs to this method. The ordering of words is lost and only the appearance of a given word is counted (and in most cases most words won't appear in a given document).

Word context takes advantage of the same term-document matrix that is used previously and instead focuses on the rows to represent the semantic meaning of words. This work makes use of the distributional hypothesis that words that appear in similar contexts are likely to have

similar meanings [33]. The intuitive logic of this method is that the similarity of rows and by extension words should show words that share a similar meaning. However this may also be generalised from a term-document matrix, to a word-context matrix. The context could represent many things, such as words, sequences, or paragraphs.

For a pair pattern matrix, the focus is on the similarity of relations. Rows represent pairs of words ($w_1 : w_2$) while columns represents the pattern in which those words co-occur. An example could be $wood : tree$ and the pattern being "wood comes from a tree". This can be used as an example of an extension to the distributional hypothesis that patterns that co-occur with other patterns tend to have similar meanings [53]. An example of this is the patterns $mason : stone, carpenter : wood, potter : clay$ sharing the same general semantic relation of $artisan : material$. A potential extension of this is triple-pattern matrices, for example $mason : stone : masonry$ where the pattern could be "mason uses stone to build masonry". However as the number of terms increases these patterns become increasingly rare and result in a sparser matrix. It is possible however to turn these triple-patterns into pair-patterns, in the previous example this could be $mason : stone$, $stone : masonry$, and $mason : masonry$.

The rest of this work looks at applications of these matrices, and how they can be used in NLP tasks. Term-document matrices have uses in many classical NLP tasks, such as document retrieval and question answering. Word and term context matrices have multiple tasks that they can contribute to state of the art results at the time of publication, such as classification, information extraction, and clustering algorithms. Pair-pattern matrices have fewer uses however they are all focused on the relations and similarity of word patterns and pairs. Examples of this include analogical mapping, relational classification, and pattern similarity. As time has passed performance using vector space models has fallen behind more modern methods.

2.1.2 Context Free Embeddings

Word embedding algorithms can be generated taking advantage of the co-occurrence of words, assuming that words that appear together often have a semantic relationship. Three such algorithms that take advantage of this assumption are fastText [61], word2vec [59], and GloVe [75].

2.1.2.1 word2vec

word2vec aims to provide an efficient implementation of both the continuous bag of words and skip gram algorithms for computing vector representations of words [81] [62]. Continuous bag of words aims to predict a word given its context. Skip-gram will aim to predict the context given an input word. word2vec uses two layer neural networks, utilizing a single hidden layer which will then be used to represent the word vectors.

word2vec uses one-hot encoding where the initial, input vector size is the number of words in the vocabulary. This means that a input vector will be all zeros except for a one in the cell that represents a given word. word2vec has two distinct versions; skip-gram, and continuous bag of words.

Continuous Bag of Words

Continuous bag of words has multiple one hot input vectors of the words to be predicted (depending on the window size). It has multiple input vectors, with a single output vector with the probabilities for the output word. The training objective of the neural network is to maximise the probability of observing what is the correct output word given all of the input vectors.

Skip-gram

Skip-gram can be seen as an inverse of the continuous bag of words model. The input layer is a single one hot vector which represents the target word, and the outputs are vectors for the words around it (defined by window size). The training objective in this case is an inverse to the continuous bag of words object, it is aiming to minimise the summed prediction error across all of the words within the window in the output layer.

With both models of learning, the main aim of the neural network is not to accurately predict words based on context words (although the networks cost function is focused on this), but to use the hidden projection layer vectors as the word vectors when the neural network has finished optimising the solution.

There are various hyper-parameters that can be changed which can impact the representation of the word embedding. Two variables that are related to this are the dimensionality of a vector,

and the size of the training set. While this is a common approach for improving accuracy of a learning algorithm, it does show diminishing returns if the number of dimensions of a word vector are not also increased. The same behaviour is also observed if dimensionality is increased but the training set is not increased.

The performance of word2vec varies on using either skip-gram or a continuous bag of words approach. Using 640 dimensional word vectors accuracies for continuous bag of words was found to have a 24% semantic accuracy and 64% syntactic accuracy [59]. While Skip-Gram was found to have a semantic accuracy of 55% and a syntactic accuracy of 59%. Semantic accuracy is using linear algebra to predict a value. An example of this could be $france - paris = ? - rome$, which could be rearranged to $france - paris + rome = ?$. In this example we would aim for the nearest neighbour to that vector calculation to be "Italy". Syntactic accuracy would follow the same rules, but with words that have syntactic relationships, such as; "fast", "faster", "big", ?. Where the nearest neighbour would ideally be bigger.

Skip-gram proves to have a general better accuracy performance for semantic accuracy, but it also takes longer to train on equal sized vectors and training sets. However both models are an improvement in time, processing, and accuracy compared to earlier iterations of language representation models.

GloVe

GloVe is an unsupervised learning algorithm for obtaining vector representations of words [75]. It is trained on aggregated global word co-occurrence statistics from a corpus. These representations show linear substructures of the word vector space that has been generated.

The initial step is to create a word co-occurrence matrix where each cell counts the number of times one word occurs in context of another word. This can also be used to count the number of times any word appears in the context of a given word, and the probability that a word will appear in the context of that word.

Word co-occurrences can then be used for examination given words probabilities with various probe words. These probabilities are then used as ratios to create word vectors representation. In an example, $P(k|w)$ is the probability that the word k appears in the context of word w.

An example given for this from the GloVe paper itself is the probability of the word "solid" being closely related to "ice" or "steam". The probabilities of these word co-occurrences in the GloVe example corpus are $P(solid|ice) = 1.9 * 10^{-4}$ and $P(solid|steam) = 2.2 * 10^{-5}$. The ratio of these two probabilities will be calculated as $P(solid|ice)/P(solid|steam) = 8.9$ meaning that ice correlates better with the word ice than it does with the word steam.

This is considered a “large” ratio, meaning that ice co-occurs more often with solid, than with ice. In this situation the number is expected to be greater than one, while if it was smaller than one then it would mean that the bottom of the division is more likely to co-occur. While a ratio of one means an equal probability to co-occur, or equally as related.

The aim of GloVe is to maintain the linear structure of these ratios, so complex systems such as neural networks can learn from them at a faster rate. This is because when GloVe learning it is aiming to make it such that word vectors dot products equal the logarithm of a words probability of co-occurrence. This is due to the law of logarithms that the logarithm of a ratio is equal to the difference of the logarithms.

The unsupervised learning method is a weighted least squares regression model. A motivation for this is to account for co-occurrences that happen rarely, as these carry noise and less information overall than more frequent co-occurrences. The initial step will be to generate two sets of word vectors, and then use global bi-linear regression to make those vectors dot product equal to the logarithm of the words probability of co-occurrence.

Glove showed state of the art performance when compared to contemporaries. GloVe has a computationally expensive cost, but it is a one time up front cost. GloVe when compared to word2vec outperforms it when training on sets of similar or even larger size. It also proves to perform better with limited training time [75]. However, unlike word2vec it cannot be trained multiple times. It generates its best prediction by learning on the entire co-occurrence data set once. This is opposed to word2vec where additional sentences can be given to predict and learn from.

fastText

fastText is a semi-supervised learning algorithm that can be seen as a further refinement of word2vec. The initial training for fastText is to create a sentence vector from input words, or n-gram character sequences. This is done by averaging those inputs to generate a sentence vector, this sentence vector is then later used as part of a classification task. This will train both the sentence weights, and original n-gram / word input weights for the classification task. Later iterations using fastText worked on using these methods to generate word embeddings [43].

The key difference between fastText and word2vec is the use of sub-words; where word2vec will have a vocabulary for as many words as it wishes to learn, fastText will instead learn sequences of characters (interchangeable with sub-words) [9]. There are multiple benefits to using sub-words; having a vocabulary will mean words not in that vocabulary do not have representation. Sub-words make it so that all possible words can be represented, even if the representation may not be as accurate on rarely seen words. A potential example of this could be the following word. If we embed the word “playing” and choose to also embed 3 character n-grams then the word vector for “playing” will combine the vector representations for the following n-grams:

“playing”

[“playing”, “pla”, “lay”, “ayi”, “yin”, “ing”]

When the word “playing” needs to be adjusted due to training, all 3-grams that participate in the generation of that word vector will also be adjusted.

These can also be seen as word features, that can be combined together to make for sentence representation tasks that improve performance on downstream tasks such as sentiment analysis and tagging predictions [43]. This method starts with word representations that are then averaged into text representations as an input. A further improvement is using a hierarchical soft-maxing with a hashing function to allow for faster and more efficient memory mapping of representations [60].

2.1.3 Deep Contextual Word Embeddings

"Deep contextual word embeddings" are a more recent advancement in natural language that generate word embeddings as part of a pre-processing step. Popular examples of these algorithms are ELMo [76] and BERT [20]. Deep contextual word embeddings have shown to perform better in a lot of tasks compared to their context free counter parts (such as fastText), however there is a large trade off as there are more trainable parameters resulting in longer training times.

2.1.4 ELMo

The first prominent example of deep contextual word embeddings is ELMo. ELMo was defined by Peters et. al with the core function being that of a deep bidirectional language model [76]. The core idea of this work is to train two embeddings, one that is predicting preceding words and one predicting succeeding words. These two models can then have their respective vector representation for each word concatenated to provide an embedding. ELMo is split into two training tasks, the first task is training embeddings using two LSTMs each for the forwards and backwards direction of text. The second task is to then fine tune these representations for a specific machine learning task (such as classification or translation). The "word vectors" (also known as the training weights from the prediction task) from the LSTM are no longer trained, to cut down on training time and for transfer learning. The embeddings at each of the 3 layers are concatenated based on their bidirectional counterpart; the input layer (which are input embeddings such as GloVe), the first LSTM layer, and the second LSTM layer. For a single word or token this produces three vectors, they are then weighted based on the second tasks trainable parameter. After weighting all three vectors are summed together to provide a vector for a given word.

ELMo saw increases in performance compared to context free word embeddings in many tasks, however this came at the cost of vastly increased training times. Pre-training and fine tuning was able to help alleviate this issue by using pre-trained embeddings and fine tuning for specific tasks, but did not resolve it.

2.1.5 Self Attention

Attention was first presented in 2014 by Bahdanau et. al. used in the context of machine translation [5]. This attention mechanism was used in the decoding side of the translator, enabling the decoder to know what words from the encoded sentence are important for their decoded counterparts. This also relieved the challenge of the encoder trying to encode all information from the source sentence.

Vaswani et. al. introduced the idea of a transformer network, extracted from encoder decoder networks to improve performance and training time [99]. A key motivation for transformers was to solve the issue of long training times caused by recurrent neural networks, having to be sequential and impossible to train in parallel (i.e. predicting word t depends on word $t-1$, but also on the hidden state for $t-1$ which can only be known after observing in order). Transformers make use of self-attention, a key part of decoders in encoder / decoder algorithms, but in a way that removes this time limitation. Attention in encoding and decoding is the process of telling the network what tokens are important to the target token at the time. For a given word t this means that a weight will be calculated for all tokens around t (i.e. $t+1, t-1$) that indicates how important each is token to our target. These weights will then scale their respective tokens vector and be summed up to provide a vector representation for our target word. Intuitively words pay most attention to their own inputs, however some words will pay attention to other words in this context to define them. This work when applied to machine translation tasks has produced state of the art results, with a reduction in computation time when compared to convolutional or recurrent solutions. OpenAI also released GPT (Generative Pre-trained Transformer) using transformers in a forward reading fashion, to achieve improvements on state of the art for many tasks at the time [78].

2.1.6 BERT

Bidirectional Encoder Representations from Transformers (BERT) follows up on transformers to use them in a bidirectional manner [20]. BERT also follows ELMo in that training involves a split between two tasks, pre-training and fine-tuning.

Pre-training for BERT splits into a further two tasks, the first will be predicting masked

words from an input with a classifier and the second task is to predict the next sequence of words. Word masking is done to solve an issue of a bidirectional model, transformer layers enable words to be able to see “see themselves”. This is solved by masking 15% of input words during the pre-training process, with either a token indicating that token is masked, or replace with another token. This process is explained in Sec. 3.3.1. Transfer learning is the process of learning an AI system for a particular task, then applying what has been learned to another task.

The fine tuning process of BERT is an example of transfer learning. Depending on the task, the network will change the input and outputs accordingly (such as sentence classification, question answering, and named entity recognition).

BERT is primarily used on downstream tasks, however feature extraction can also be done to generate word embeddings to be put in existing models. BERT has 12 layers of encoders stacks on top of each other which can all contribute to embeddings, they find that the best performance for a named entity recognition task is achieved by concatenating the four last layers on a given embedded sequence.

2.2 Applications of Word Embeddings

Word embeddings are primarily designed to be applied to machine learning tasks to aid in faster training and improve the performance of numerous tasks. They have also found uses outside that core focus, and have been used in fields such as social studies, linguistics, and history.

2.2.1 Word Embeddings in Machine Learning

Word embeddings are used as a pre-training step for many state of the art machine learning tasks. They provide a neural network (or other potential machine learning algorithms) with a representation of a language and the corpus provided, before learning the main task at hand. This improves performance in two distinct ways; the first is that due to the networks knowledge of input words and tokens at the start of training, attention can be given to the final steps of the pipeline instead of just understanding the input. The second performance increase is that of time and data required. If a neural network only used input tokens such as unique identifiers for each

word instead of word embeddings more training would be required to learn word representations as well as the task. As more training is required more data may also be required.

Irsoy and Cardie, look at using deep recursive neural networks to understand compositions, and apply it to a sentiment analysis task [37]. They compare the performance of their deep model to standard recurrent neural networks that are shallow and find that they outperform their counterparts when doing sentiment analysis. In this task they use word2vec word embeddings across all experiments, positively contributing to the performance.

Nalisnick et. al. use dual word embeddings to improve document ranking utilizing word2vec vectors [65]. A key factor of their work is using embeddings generated by both the input word2vec embeddings, and the output word2vec embeddings. It is more common for only the input word embeddings to be extracted for most other tasks. They find that using both input and output embeddings in their task performs better than using input embeddings alone, or compared to other methods such as frequency statistics or latent semantic analysis.

Lei et. al. look to use word embeddings in part to help with redundant questions and answers in question answering forums on the internet [52]. They aim to achieve this by learning what questions are semantically related using a recurrent and convolutional model. They train their own word2vec word embeddings on a Wikipedia corpus along with a collection of posts from a question answering forum of their choice to better understand the language. Using word embeddings, and an extra pre-training step of an encoder decoder model shows their method to perform better .

2.2.2 Word Embeddings for Science

Word embeddings main function is to be a useful input or a pre-processing step for machine learning tasks as they provide downstream tasks with an understanding of a language before learning that specific task. They have also proven useful in work that isn't directly involved in machine learning or neural network tasks.

While biases being present within word embeddings may be problematic in general use, Garg. et. al found those biases useful to conduct a study about biases in the United States over the course of 100 years [27]. They study a range of GloVe and word2vec embeddings, along with pre

trained embeddings that use multiple different corpora such as BookCorpus, Corpus of Historical American English (COHA), Google Books, and the New York times. They compare common biases over time to real world statistics and how significant events affect those biases. Their method is capable of finding significant real world events from looking at changes in biases, such as the women's movement in the 1960s and the Asian immigration to the United States. They also find biases to be correlated with real world statistics, an example being employment statistics of men and women reflecting the biases of men and women within embeddings. This work shows an application of word embedding biases being use in social sciences to better understand commonly known biases.

Work here is not just limited to detecting biases within word embeddings, Kulkarni et. al. have utilised word embeddings for finding significant semantic changes in words [48]. They look at target words and try to see how those words have change. An example is "gay" changing over the course of a century; first it was seen to be similar to "dapper" and "cheerful" but changed to its modern meaning "homosexual". They show that their method performs better than frequency based methods. They also look at faster evolving corpora, such as twitter and amazon to show that linguistic shifts that happen very fast are also captured.

Further work has been done looking at the semantic relations using analogy based detection. Gladkova et. al. look at finding inflectional and derivational morphology, and lexicographic and encyclopedic semantics within word embeddings [28]. Inflectional for example tests an embeddings knowledge of plurals (student, becoming students). An example of derivation is adding suffixes to words, such as life becoming lifeless. Lexicography deals with understanding things such as synonyms, such as sofa and couch having the same meaning. Finally the encyclopedia section looks at the relationship between "things", such as countries and their main language such as "mexico" and "spanish". They compare results from GloVe to SVD counting based methods, and find that they have similar results among all categories.

2.3 Measuring the Performance of Word Embeddings

There has been a lot of work focused on providing evaluation and understanding for word embeddings. Schnabel et. al. have looked at two schools of evaluation; intrinsic and extrinsic [86]. Intrinsic evaluations involve using sub-tasks to evaluate the performance of a word embedding in isolation based on some criteria, an example of this is seeing if word embeddings have the linear relationship of countries and their capital cities. Extrinsic evaluations are judging the performance of a word embedding based only on how it performs on the intended end task. An example of this is a machine translation tool correctly translating sentences as intended.

Extrinsic evaluations alone are unable to define the general quality of a word embedding. The work also shows the impact of word frequency on results, particularly with the cosine similarity measure that is commonly used. Intrinsic methods have also had criticisms, with Faruqui et. al. calling word similarity and word analogy tasks unsustainable and showing issues with the method [21].

Intrinsic

Nematzadeh et. al. showed that GloVe and word2vec have similar constraints when compared to earlier work on geometric models [68]. For example, a human defined triangle inequality such as “asteroid" being similar to “belt" and “belt" being similar to “buckle" are not well represented within these geometric models.

Schwarzenberg et. al. have have defined “Neural Vector Conceptualisation" as a method to interpret what samples from a word vector space belong to a certain concept [87]. The method was able to better identify meaningful concepts related to words using non linear relations (when compared to cosine similarity). This method uses a multi class classifier with the Microsoft Concept Graph as a knowledge base providing the labels for training.

Extrinsic

Sommerauer and Fokkens have looked at understanding the semantic information that has been captured by word embedding vectors [92] using concepts provided by [19] and training

binary classifiers for these concepts. Their proposed method shows that when using a pretrained word2vec model some properties of words are represented within the embeddings, while others are not. For example, functions of a word and how they interact are represented (e.g. having wheels and being dangerous), however appearance (e.g. size and colour) is not as well represented.

General Language Understanding Evaluation (GLUE) is a collection of tools designed by Wang et. al. to evaluate the performance of word embedding models across a range of existing tasks [100]. The tasks included in GLUE are question answering, sentiment analysis, and textual entailment that have either sentences or sentence pairs as inputs. These tasks are focused on general knowledge embeddings, and are not suited to specialised corpora. When performing their work on GloVe and ELMo embeddings to get sentence representations they find that ELMo is the strongest performing embedding. They judge performance on tasks using a range of metrics, ranging from accuracy and F1 score to correlations.

2.3.1 Medical Performance Evaluation

Khattak et. al. have done extensive research looking at the performance of word embeddings for clinical text and tasks [46]. It ranges from examining statistical co-occurrence embeddings, context free embeddings, to variants of deep contextual embeddings. They compare intrinsic and extrinsic tasks measuring the performance of word embeddings at using clinical tasks. They find that intrinsic tasks while being understandable, do not always correlate to a high performance in end user tasks. They also note that extrinsic tasks are important as they will be the target task of an AI system, however they are difficult to interpret and understand decisions made.

Si et. al. look at enhancing clinical concept extraction when using deep contextualised word embeddings [90], partly as an extension of the work done by Zhu et. al.[109]. The focus of this work is to compare the performance of concept extraction in the clinical domain between context free embeddings such as GloVe and deep contextualised word embeddings such as BERT. Another part of the work is looking at pre-training on a clinical corpus compared to state of the art off the shelf corpus that are from a general domain. They find that BERT outperforms at concept extraction, and the domain specific corpus also improves performance on the same task.

Peng et. al. has taken inspiration from the work done with GLUE [100], to produce the

Biomedical Language Understanding Evaluation (BLUE) framework [73]. The aim of BLUE is the same as GLUE, to evaluate the performance of an embeddings understanding of language. However, the difference in this work is that it is focusing on biomedical language evaluation rather than performance in general language. BLUE has four different tasks using ten different data-sets specialised for the medical field. The tasks are sentence similarity, named entity recognition, relation extraction, document classification, and inference. As part of their work they compared the performance of ELMo and BERT trained on different corpora. They found that BERT trained using a specialised medical corpus to be the best performing embedding for the BLUE test.

2.4 Bias

2.4.1 Bias in AI

AI systems are growing in widespread recognition and publicity along with industries attempting to incorporate these systems into business processes that handle private and personal data. This makes them a very important aspect of modern society. This increased importance for AI systems has also increased demand for fairer AI, and a trust in intelligent systems to make decisions that represent human decisions fairly and transparently.

This has not proven to always be true, as there has been evidence of biases in intelligent systems with no ability to analyse or query the decision making process that these AIs have made [2, 24]. In response to these black box systems making unwanted decisions that seem to be biased, there has been work that attempts to look under the hood to understand how these black box systems make their decisions [45, 80, 83].

Fong and Vedaldi have been looking at peeling back black box systems that are becoming more popular in computer vision [25]. Deep neural networks in computer vision have shown to be state of the art in many tasks, however the AI provides little reasoning for the decisions made. This results in advancements being achieved by primarily by trial and error. Fong and Vedaldi attempt to look at what a convolutional neural network (CNN) has learned to do after training.

Jia et. al. looked to uncover deep networks by looking at the mistakes made in judgement

after training [39]. They use a CNN to train classifiers to take an input and correctly classify it as a family of mammals. This work finds that the image classifier was learning information more general than just a mammals family, as it was mistaking images for similar families within the mammal taxonomy.

Samek et. al. has attempted to develop a system that attempts to explain the decision making process of any classifier, for example text or images [83]. The purpose of this work is to give humans confidence in their decisions when using what would normally be a black box solution.

2.4.2 Bias in NLP / Word Embeddings

Bolukbasi et. al. look at the biases within word embeddings and corpora [10]. They use an example vector analogy to demonstrate bias such that:

$$\text{man} - \text{woman} \approx \text{computer programmer} - \text{homemaker}$$

This work demonstrates biases present within word embeddings that are commonly used in downstream systems. It also looks at occupations that can be described with a single word and finds the most male and female occupations. The most female occupations are words such as “nurse”, “receptionist”, and “librarian”. While male occupational words are “maestro”, “skipper”, “protege”. Finally they propose a method of debiasing that reduces the biases present within the embeddings while maintaining preferable traits of embeddings, such as nearest neighbours representing similar words and high performance on word analogy tasks.

Caliskan et. al. used word embeddings to look at racial, age, and other biases [12]. They look at repeating experiments from traditional psychology, and find similar results (such as insects being unpleasant, and flowers being pleasant). In terms of biases they find that males names are more associated to careers, while females are associated with family. European-American names are found to be more associated with positive words than African-American counterpart names, and names more common among younger people are more positive when compared to names more common among older people. They also related this work to the real world showing a correlation between the number of males or females in an occupation and that occupations

relative employment figures to those genders (for example, a job with a high percentage of males will be more likely to be more associated with male terms).

Gonen and Goldberg critique and show that common debiasing methods (such as removing bias via projections or generating embeddings that aim to be gender neutral) cover systematic gender biases up, but do not effectively remove them from word embeddings [29]. They argue that methods that take focus on the removal of the "gender direction", which is a vector calculation $he - she$ gender bias information can still contain gender biases within neutral words. Clustering the top 500 most gendered words for both genders before and after debiasing shows that clusters don't see significant change, with an 85% to 92% accuracy for post debiasing clusters. Another indication of biases still being present within debiased embeddings is showing that nearest neighbours still indicate such biases. For example "nurse" is no longer near explicitly feminine words but is still close to words biased in society as feminine, such as "receptionist", "caregiver", and "teacher". Plotting biases from embeddings versus percentage of words that generally show an unwanted gender bias shows there is a correlation, in both biased and debiased embeddings. A similar result is experienced when comparing occupational biases to the number of male socially marked words. This work shows that while explicit biases may have been mitigated with popular debiasing work, implicit biases are still found within the word embeddings and may propagate into downstream tasks.

2.4.3 Colour Bias in the Real World

In real world language, a common bias among western culture is that the colours pink and blue could be associated to females and males respectively [15]. Historically gender associations towards colour were initially arbitrary, and over time became more widespread through culture [17], [18]. In contemporary times, parents commonly choose pink as an aesthetic of choice for their daughters for every day life [77], [4]. This has an effect on the choice of colour for young girls, with them showing a particular affinity for pink [54], [105]. Later in life, young adult women are shown to choose other colours as their favourite, with blue and red being common choices [41] [1].

[1] This review of colour biases in the field of psychology was conducted by Domicele Jonauskaite and Christine Mohr for the forth coming manuscript "Colour and Affect in Natural Language - Domicele Jonauskaite, Adam Sutton, Christine Mohr, Nello Cristianini"

Traditional studies have shown that pink is considered to be a female colour, while blue is considered to be a male colour [13], [15], [56]. In addition to female representation, pink is also shown to represent other social groups, like children and homosexuals [7], [35], [47]. However later in life adult women have shown to disassociate with the colour of pink to avoid these representations [41].

In comparison to pink, red represents power, dominance, and higher status [1], [58], [93], [106]. This may provide an explanation of why the colour red is associated with both positive and negative associations [26] [67], [96], [101]. Pink and blue have both been highly associated with positive emotions, although blue has been associated with sadness as well (as can be seen in natural language with "feeling blue") [67], [85], [98], [6], [32], [84].

Studies have shown that colours expressed through language are associated to the same colour presented physically. Women dressed in red have been found to viewed as more attractive than women dressed in other colours [51]. However this association is also found to carry over into language, as was found when just mentioning the word of red in a description of a woman [72]. This language and physical relationship of colours is not only focused on gender; colours that were presented visually or in language were found to be associated with similar emotions for both emotional and gender associations of colour [42].

2.5 Semantic Change in Corpora

Language changing between corpora has been a long focus of linguistics. Juola uses quantitative methods and entropy estimation to calculate the Kullback-Leibler distance between corpora [44]. The main focus of this work is to show that they can demonstrate language changing and that this change can be perceived by algorithms. This work shows that, and also shows that the change of language is not uniform. The method used shows that language changed more in the period of 1949-1968 than when compared to 1939-1948, and can be quantified. This work however does not specify how language has changed, or what has changed about it. All it does is show linguistic separation of a corpus between defined time periods.

Gulordava and Baroni use vector space models and distributional similarity to detect semantic

change in Google's Book N-gram corpus [31]. The study aims to look at the change in meaning of words from 1960-1964 to 1995-1999, comparing the statistical and quantitative methods to intuitive assessments from a human's evaluation. The distributional method uses co-occurrence matrices with a window of 1, therefore only direct neighbours to a word can influence its meaning in the co-occurrence matrix. The cosine similarity between a word's score in the 60s and the 90s will give an indication of how much a word has changed, a higher score meaning less semantic change has taken place. This work finds that semantic change can be measured in this way, as human evaluations are shown to be correlated with the distributional method with a Pearson correlation of 0.51 and a p-value of < 0.01. This shows that distributional methods can contribute to learning the changing of words over time.

Wijaya and Yeniterzi attempt to understand the semantic change of words over centuries [102]. They also use co-occurrence statistics between words to show the semantic change of words over time, along with comparing these results to frequency based methods to detect change. They find that using frequency based methods (the number of times a term appeared in a corpus) is not always enough to identify a change in semantics. Frequency based methods also do not describe what change is being experienced, and some changes are not represented with a change in frequency. Using co-occurrence methods or k-means clustering on tf-idf scores show success in finding hidden topics or words that change in meaning over time. They are also able to show and reason with points of change in the meaning of the word over time (such as "Clinton" changing from "governor" to "president" in 1993).

Mitra et. al. look at words "developing", "losing", and "birthing" new meanings over time using a clustering method with the Google book project [64]. They used an unsupervised co-occurrence clustering method to find target words local neighbourhoods for each time period corpus. Change between words can be detected if the target words have different neighbourhoods in each corpora of interest. They demonstrate four ways a word can change in meaning over time; a new meaning ("birth"), a meaning becoming obsolete ("death"), a meaning changing to two or more clusters ("split"), a or multiple clusters unifying to a single meaning ("join"). An example of birth could be seen in the word "compiler", in a corpora from 2002-2005 one cluster of words with "compiler" includes words such as "hardware", "software", and "parser". Earlier corpora have no such cluster,

showing that a new meaning for that word has been found. These results were found to be correlated with both manual evaluation and automated evaluation using WordNet.

Kulikarni et. al's work can also be viewed here as it focused on linguistic change [48]. A key difference in this work compared to previous work in the same field is not only does it look at frequency and statistical based methods it also uses word embeddings to help in the process of learning and visualising word meanings changing over time. Their proposed method shows the performance of using word embedding in this task, and how it performs better than frequentist and syntactic based methods (an example of syntactic methods is part of speech tagging and seeing that "apple" becomes more of a proper noun from the late 70s than just a noun).

CHAPTER 3

METHODS AND RESOURCES

In this section I will be writing and describing all methods, statistical tests, machine learning, word embedding methods, and language resources used throughout my thesis.

3.1 Linear Algebra

3.1.1 Euclidean Norm

The Euclidean Norm for vector x with n dimensions can be defined as:

$$||x||_2 = \sqrt{\sum_{i=1}^{n} x_i^2} \tag{3.1}$$

3.1.2 Normalisation

Dividing a vector by its norm will provide a unit vector where the length of the vector will always result in a length of 1. The euclidean norm used in Sec. 3.1.1 can also be used to normalise the vectors to a unit length of 1 such as for vector x:

$$\hat{x} = \frac{x}{||x||_2} \tag{3.2}$$

3.1.3 Cosine Similarity

Cosine similarity is measure of similarity between two vectors, measuring the cosine of two angles between them. Two vectors with a similar direction will have a score closer to 1, a score of 0 means they are orthogonal, and a score of -1 means they are going the opposite direction (anti-correlated). The cosine similarity between two vectors x, y can be defined as:

$$\cos(x,y) = \frac{xy}{\|x\|\|y\|} = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n}(x_i)^2}\sqrt{\sum_{i=1}^{n}(y_i)^2}} \tag{3.3}$$

3.1.4 Euclidean Distance

The Euclidean distance is the straight line distance between two points that are in an Euclidean space. The Euclidean distance can also be called the l^2 distance, by looking at its relationship with the Euclidean norm 3.1.1

$$dis(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{3.4}$$

3.2 Context Free Word Embeddings

A corpus **C** is a collection of documents (e.g. from sources such as news articles, or Wikipedia). From **C**, we extract the set of all distinct words, and call it the vocabulary **V**. Each document in **C** is a string of words (i.e. their ordering is important for the embedding algorithm). An embedding is a function Φ defined such that $\Phi : \mathbf{V} \rightarrow \mathbf{R^d}$, mapping every word in the vocabulary to a d dimensional vector. Word vectors from a word embedding will be represented with w.

Using an embedding method Φ, which defines the method of going from a word in a vocabulary to an embedded space as: $\Phi\ (word_j \in \mathbf{V}) = w_j \in \mathbb{R}^d$. A word vector for a given word will be defined as w.

The machine learning problem associated with learning word embeddings is posed as follows: given a corpus **C** find a mapping Φ that assigns vectors to each element of the vocabulary **V**. Many ways are possible to define an embedding, and this thesis will cover a few key algorithms that are used to generate word embeddings. For clarity, a word embedding is the function of

taking an input word and returning a d dimensional word vector. However the phrase "word embedding" and "word vector" may be used interchangeably throughout this work.

When performing similarity measurements, word vectors are commonly normalised to unit length:

$$\hat{w} = \frac{w}{||w||} \tag{3.5}$$

Two words vectors w_1 and w_2 within a vector space can be compared by taking the normalised dot product of their words:

$$\langle \hat{w_1}, \hat{w_2} \rangle = \sum_{i=1}^{n} \hat{w}_{1,i} \cdot \hat{w}_{2,i} \tag{3.6}$$

As both word vectors are normalised, this is equivalent to the cosine similarity between the two word vectors as described in Sec.3.1.3. A cosine similarity closer to 1 means that the vectors are similar to each other, while a cosine similarity of 0 means that the vectors are orthogonal to each other.

3.2.1 word2vec

The first method we describe to generate word embeddings is to generate them using shallow neural networks, with the idea that a good embedding of a word should be very informative for the prediction of other words that co-occur with it. Word2vec [62] aims at finding representations that can be used for this purpose. It uses skip-gram or continuous bag of words and uses the weights from the network to create vector representations for each word. Continuous bag of words uses the surrounding words of a given window size to predict the current word, while skip-gram inversely uses the current word to predict the surrounding words. Overall the basic signal being exploited is still that of co-occurrence.

For the duration of this thesis all word embeddings generated by the word2vec algorithm will use the skip-gram variant, as it generally has proven to have the better performance when compared to continuous bag of words. The (naive) loss function for the skip gram model makes

use of the softmax function and is as follows:

$$J = -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})} \tag{3.7}$$

where j_c^* is the index of the correct output from the output layer of the neural network, u_{c,j_c^*} would then represent the score of the correct word j appearing in the context c. $u_{j'}$ is the score for all incorrect words appearing in the correct context. C is the size of the context window, while V is all words in the vocabulary. This loss function is loss for the skip-gram algorithm, without any computational improvements that are common within word2vec such as negative sampling or hierarchical softmax [62].

Negative sampling is the process of sampling a small subset of training samples (in the range of 2-20) to have their weights updated when performing back-propagation. If 5 negative samples were chosen then it would only require $6 \times d$ weights to be updated out of $\mathbf{V} \times d$ possible weights. Negative samples are selected based on the frequency of words appearing in the corpus, with more frequent words more likely to be sampled [62].

The conventional softmax (as shown as part of the word2vec loss function in Eqn.3.7) has a computational complexity of $O(\mathbf{V})$. Every word in $\mathbf{V}$ must have the probability of it occurring computed, which can be a long task as $\mathbf{V}$ can reach hundreds of thousands of words. Hierarchical softmax has a complexity of $O(\log \mathbf{V})$, this is achieved by having each word in $\mathbf{V}$ represent a leaf in a tree. The tree is then traversed to compute the probability of a word. The probability of each word will be the product of probabilities on the path from the root to the word. The construction of these trees is crucial to the performance of the model, as these trees are human defined. In the case of [62], binary huffman trees are utilised.

3.2.2 GloVe

GloVe is a popular algorithm to infer a semantic word-embedding from a given corpus using a co-occurrence matrix [75].

Let X represent the word-word co-occurrence matrix, whose entries X_{ij} tabulate the number of times word j occurs in the context of word i. Its dimension will be $|V| \times |V|$. GloVe uses this

co-occurrence matrix and the following cost function to learn a set of feature embeddings:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - logX_{ij})^2 \tag{3.8}$$

To simplify, the inner product between two word-vectors should provide information about the conditional probability of co-occurrence. Other information used in the definition includes the use of f as a weighting function to reduce the emphasis of commonly co-occurring words such that:

$$f(x) = \begin{cases} (x/x_{max})^{\alpha}, & \text{if} x < x_{max} \\ 1, & \text{otherwise} \end{cases} \tag{3.9}$$

Where x_{max} is the number of times X_{ij} must co-occur to achieve maximum weighting and α is the scaling factor for values of x between 0 and x_max. α is commonly set to 0.75 due to empirical evidence that it results in better performance.

3.2.3 fastText

fastText can be see as an improvement to word2vec seen in Sec.3.2.1. Like word2vec, fastText can use the continuous bag of words model or the skip-gram model. fastText differs by making use of different optimisations to further improve performance [43]. A key feature of fastText is sub-word embeddings. With both word2vec and GloVe if a word doesn't exist in the embeddings vocabulary then there will be no vector representation for that word. fastText answers this by using sub-word information, sequences of characters that can be combined to generate any word possible. This helps performance in two distinct ways; words that don't have a representation in previous embeddings will now always have a representation, and sub-word information will improve information on rarer occurring words.

Generally sub-words are a bag of n-gram characters where n is between 3 and 6. Special characters '<' and '>' are added to denote the beginning and ending of a word. The entire word is also embedded, regardless of its n-gram representation. A hashing function is used to hash approximately 2×10^6 n-grams. A word is then represented by its own word representation and

the n-grams that represent it. A word can be scored by:

$$s(w,c) = \sum_{g \in G_w} z_g^\top v_c \tag{3.10}$$

where z are the vector representations for for each n-gram g in G_w and v_c is the context vector. Using this and denoting the logistic loss function ℓ such that $\ell : x \rightarrow \log 1 + e^{-x}$ we can describe the loss function as:

$$\sum_{t=1}^{T} \left[\sum_{c \in C_t} \ell(s(w_t, w_c)) + \sum_{n \in N_{t,c}} \ell(-s(w_t, n)) \right] \tag{3.11}$$

where $N_{t,c}$ is the set of negative examples sampled from the vocabulary (another optimisation available to word2vec and fastText). In the case of a word not being present in the vocabulary, word representations can be generated using the n-gram representations and the hashing function. While the representation may not necessarily be accurate, it is still a preferable solution compared to being unable to compute words not seen during training.

3.3 Deep Contextual Word Embeddings

The previous word embeddings explained in this chapter can be defined as “context-free word embeddings”. The word vectors generated by these embeddings are static and do not take their context into account when representing a word. A key benefit of this is representing words that are polysemic, having multiple meanings. An example word of this would be “bark”; this word could mean the bark on a tree, or the bark of a dog. The same word has two different meanings, however the vector representation in context-free embeddings would be the same. This results in a loss of performance for multiple NLP tasks, as the nuance of some words are lost. This section will briefly explain how BERT generates embeddings, the key differences, and some notation for BERT embeddings [20].

As opposed to context-free word embeddings BERT uses transformers, an attention mechanism that learns the contextual relationships between input tokens (which can be word or character sequences) [99]. The transformer model tries to learn using the entire window of input

tokens, with no directional input (i.e. reading the left or right most words first). This results in what is known as a “bi-directional model”. BERT works in two major steps, the first step is commonly known as the “pre-training” step which is a semi-supervised task where the objective is to predict masked words from input sentences and then predict the next sentence. The second major step can be any of multiple different supervised NLP challenges, an example of this could be a classifier with training data of input sentences, and labels. The second step is widely known as “fine tuning”.

The “pre-training” step has two tasks that it tries to improve on. One task is the “masked language model”, word sequences will have a percentage of input tokens (commonly 15%) masked and those tokens will be predicted by a classification layer. The second task is “next sentence predictions”, pairs of sentences will be input and a classifier will try to predict if given sentence B is the correct sentence that follows sentence A. Sentence B will 50% of the time be the correct sentence from the original document, while the rest of the time it will be a random negative sample. These input sentences will have masking happening at the same time as sentence prediction, and BERT will calculate a loss function based on the performance of both of these tasks.

Fine tuning is the process of taking embeddings generated from the pre-training tasks and now applying them to different NLP tasks. These tasks include but are not limited to sentence prediction, question and answering tasks, and part of speech tagging.

3.3.1 BERT Representation

BERT represents words as N number of character sequences (also known as sub words), depending on the vocabulary. The character sequences that are present in the vocabulary are fixed and can represent any possible sequence that has appeared in the corpus. This is due to single characters being embedded. Each character sequence has its own representation depending on the context of the word around it. All character sequences that are in the BERT vocabulary start with “##”, except for those sequences that are the start of the word. A potential example of this could be the following word:

“playing”

may be broken down in a BERT vocabulary as:

“play”, “##ing”

If the embedding dimension is set to 768 (which is commonly used in BERT) this would result in the word “playing” being represented as a 2x768 matrix. Mathematically, embeddings for single words (and by extension all sentences, and character sequences) in BERT can be viewed as a $\mathbb{R}^{N\times d}$ matrix:

$$w = \begin{bmatrix} w'_{11} & \dots & w'_{i1} & \dots & w'_{N1} \\ \vdots & \ddots & \vdots & & \vdots \\ w'_{1j} & \dots & w'_{ij} & \dots & w'_{Nj} \\ \vdots & & \vdots & \ddots & \vdots \\ w'_{1d} & \dots & w'_{id} & \dots & w'_{Nd} \end{bmatrix}$$

where N is the number of word tokens, d is the dimensionality for the embedding, and w' is the embedded sub-word such that $\Phi(subword_i \in V) = w_i \in \mathbb{R}^d$.

3.4 Concepts in Word Embeddings

In much of the work presented here we make use of the notion of a ‘concept’, which is defined as any subset of the vocabulary, that is a set of words. Sometimes we will use the expression “list of words", for consistency with the literature in social psychology, but we will never make use of the order in that list, so that we effectively use “list" as another expression for “set", in this thesis. We define this as a set of words $\mathbf{L} \subseteq \mathbf{V}$ (or for an embedding a set of points in $\mathbb{R}^d$ such that $\Phi(\mathbf{L}) \subseteq \Phi(\mathbf{V})$).

We use the word vectors from a word list to define this concept in an embedding. In general, a concept is defined as any subset of a set (or a “universe"). We would normally define a concept as an unordered list of words that have been created, validated, and understood by humans that should be learnable by machines. However for the purpose of this work a concept can be defined as any subset of words from $\mathbf{V}$). This use is consistent with the Extensional Definition of a concept

used in logic, and the same definition of concept as used in the probably approximately correct model of machine learning [3].

3.5 Comparing Embedded Words

Two word vectors $\mathbf{w}_1$ and $\mathbf{w}_2$ within an embedding space can be compared by taking the dot product of their words:

$$\langle \hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2 \rangle = \sum_{i=1}^{n} \hat{\mathbf{w}}_{1,i} \cdot \hat{\mathbf{w}}_{2,i}. \tag{3.12}$$

As both word vectors are normalised, this is equivalent to the cosine similarity between the two word vectors, as seen in 3.1.3. A cosine similarity closer to 1 means that the vectors are similar to each other, while a cosine similarity of 0 means that the vectors are orthogonal to each other.

In addition to comparisons between individual word vectors, we can compare an individual word vector to a set of word vectors which we will call a words association to a concept. This is done by finding the mean of the set, normalising the resulting vector and calculating the dot product with the individual word vectors as follows:

$$\langle \hat{\mathbf{w}}, \hat{\mu} \rangle = \sum_{i=1}^{n} \hat{\mathbf{w}}_i \cdot \frac{\mu_i}{||\mu||}. \tag{3.13}$$

The resulting calculation gives us how closely an individual word is associated with a larger set of words. This association can be used to assess how closely related a given word is to different topics or concepts within the embedding space.

Using the notions that we have defined above; it is now possible to interpret the information represented within the embedding. In this work I consider two concepts antithetical if they can be considered the opposite of each other within their context. The antithetical concepts used here are he / she, and positive / negative emotions. These are arbitrarily chosen, and may not be

perfectly antithetical. Male and female terms do not take gender neutrality into account. Positive and negative emotions may not take into account emotions that might not be considered either, such as surprise or confusion.

Given a word vector w we wish to understand the bias of, and two antithetical concepts μ_1 and μ_2 as we define the function F:

$$F(w, \mu_1, \mu_2) = \langle \hat{w}, \hat{\mu_1} \rangle - \langle \hat{w}, \hat{\mu_2} \rangle \tag{3.14}$$

Using this scoring method, we can quantify the bias for a given word. These scores can be visualised by use of scatter plots, but they can also be condensed into a single number for each word, as defined in Eq.3.14.

3.6 Removing Bias

To remove bias, first two vectors have to be identified that contain contrasting directions of the bias. These two vectors ($\mathbf{v}_1$ and $\mathbf{v}_2$) must be considered "opposite" of each other semantically, in terms of the bias that is required to be removed. The following method of debiasing is equivalent to [10]:

$$\mathbf{v}_b = \hat{\mathbf{v}}_1 - \hat{\mathbf{v}}_2 \tag{3.15}$$

where the vector $\mathbf{v}_b$ will have the direction of bias in the embedding. For example, he and she are different genders and could potentially be used to capture a gender direction. They would be composed as sets such as he, him, his for male terms, and she, her, hers for female terms.

Using this bias direction, all word vectors can now have that component removed by projecting them into a space that is orthogonal to the bias vector:

$$\mathbf{v}_\perp = \hat{\mathbf{v}} - (\hat{\mathbf{v}} \cdot \hat{\mathbf{v}}_b^T) \cdot \hat{\mathbf{v}}_b \tag{3.16}$$

where $\mathbf{v}_\perp$ is the original word vector with the biased component removed. The rank of the matrix

of orthogonal projected vectors will be reduced by one in a non-trivial embedding set. These orthogonal word vectors are required to again be normalised for further analysis.

For clarity; in this work individual word vectors are not used in debiasing, instead I use the mean of concept word lists. This would lead to a more appropriate notation of:

$$\mu_b = \hat{\mu}_1 - \hat{\mu}_2 \tag{3.17}$$

to obtain the vector that represents bias. The orthogonal projection can then be represented as:

$$\mathbf{w}_\perp = \hat{\mathbf{w}} - (\hat{\mathbf{w}} \cdot \hat{\mu}_b^T) \cdot \hat{\mu}_b \tag{3.18}$$

where $\hat{\mathbf{w}}$ is a word that we would wish to debias for the two antithetical concepts.

3.7 Statistical Methods

3.7.1 Wilcoxon Signed-Rank Test

The Wilcoxon signed-rank test is a statistical hypothesis test that is used throughout the work [104]. It is non parametric, as in situations where the test is used, assumptions cannot be made about the probability distribution of the data. In particular we use this test as some of the data used in this experiment cannot be assumed to be Gaussian due to lack of samples.
Three assumptions are made for the Wilcoxon signed-rank test:

1. Data is paired and comes from the same population
2. Each pair is chosen randomly and independently
3. The data is measured on an at least interval scale

If these assumptions hold, there can be a null hypothesis of “difference between the pairs of samples following a symmetric distribution around zero”. If this hypothesis is found to not be true, then these two distributions are viewed as significantly different. We can formulate the test as:

Table 3.1: A confusion matrix for a binary classifier

		Actual Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN
	Totals	P	N

Table 3.2: Methods of calculating performance statistics of a binary classifier

Metrics	Formulae
TPR / Recall	$\frac{\mathbf{TP}}{\mathbf{P}}$
FPR	$\frac{\mathbf{FP}}{\mathbf{N}}$
Accuracy	$\frac{\mathbf{TP}+\mathbf{TN}}{\mathbf{P}+\mathbf{N}}$
Precision	$\frac{\mathbf{TP}}{\mathbf{TP}+\mathbf{FP}}$

$$W = \sum_{i=1}^{N_r} [\text{sgn}(x_{2i} - x_{1i}) \cdot R_i] \tag{3.19}$$

Where:

1. x represents the i'th pair of observations that when subtracted don't equal zero
2. sgn is the sign function (assigning 1 or -1 for a positive or negative number respectively
3. N_r is the number of pairs that don't equal zero when calculating the difference
4. and R_i is the ordered rank (starting from the smallest absolute value of $x_{2i} - x_{1i}$ to be 1)

Using W, it is possible to reject the null hypothesis by comparing it to a critical value from a reference table [55]. If $|W| > W_{crit}$ then then null hypothesis is found to be rejected.

3.7.2 Confusion Matrices and Receiver Operating Characteristic

A receiver operating characteristic (or a ROC curve) makes use of a binary classification confusion matrix to produce graphical plots displaying its True Positive and False Positive ratios. This can be considered a measure of performance for a binary classifier.

A basic confusion matrix is shown at 3.1. From a confusion matrix there are more measures of performance of a binary classifier that can be seen. Throughout this work we make use of the True Positive Ratio (TPR, or recall), False Positive Ratio (FPR), Precision, Accuracy, and the Area

Under Curve (AUC). The TPR, FPR, precision, and accuracy are derived by formulae shown in 3.2. The AUC is the probability that the binary classifier will score a randomly chosen positive sample high than a randomly chosen negative sample.

Throughout this work there will be many uses of the all of the metrics listed when using binary classifiers for machine learning tasks.

3.8 Other Machine Learning Methods

3.8.1 Linear Classifier

A classifier is a function that maps elements of an input space (a universe, or in our case a vocabulary) to a class. A binary classifier maps inputs to one of two classes. A linear classifier is a function that classifies vectors of a vector space R^d into two classes, as follows:

$$f : \mathbb{R}^d \rightarrow \{0,1\}, f(x) = \sigma(\langle x, w \rangle + b) \tag{3.20}$$

We will learn linear classifiers from data, using the Perceptron Algorithm on a set of labeled data, which is a set of vectors labeled as belonging to class 1 or class 0. As we will learn concepts formed by words, and linear classifiers only operate on vectors, we will apply them to the vector space generated by the word embedding, as follows.

A linear classifier is a simple supervised machine learning model used to classify membership of an input. We will use a single layer perceptron with embeddings as input to see if it is possible for a perceptron to predict half of a word list, while being trained on its other half.

Given a word list $\mathbf{L}$ such that $\Phi(\mathbf{L}) \subseteq \Phi(\mathbf{V}) \subseteq \mathbb{R}^d$ we will define the words not in this list as $\mathbf{L}^c = \mathbf{V} \setminus \mathbf{L}$. We will use $\mathbf{L}$ and $\mathbf{L}^c$ to define a train set and test set for our perceptron. We will first uniformly sample half of the words of $\mathbf{L}$, we will then sample in equal amount from $\mathbf{L^c}$. We will then append these two word lists to make $\mathbf{L_{train}}$. To produce a test set $\mathbf{L_{test}}$ we will take the remaining words that haven't been sampled from $\mathbf{L}$, and sample the same number of words again from from $\mathbf{L^c}$.

A member of the training set can be defined as $l_i \in \Phi(\mathbf{L_{train}})$. We define our prediction function

$\hat{y}$ as:

$$\hat{y} = \sigma((\sum_{i}^{d} \theta_i l_i) + b) \tag{3.21}$$

where θ and b are the training parameters of the classifier and σ is the sigmoid function. We will then train the perceptron using the cross entropy:

$$J = -\frac{1}{|\mathbf{L}|} \sum_{i}^{|\mathbf{L}|} y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i) \tag{3.22}$$

where y_i is the correct class of the training sample.

3.8.2 K-Nearest Neighbour Embedding Alignment

We turn to the problem of comparing two different word embeddings, for example obtained by the same algorithm on two different training corpora. This comparison can highlight differences in the process of generating those corpora (e.g. the culture that produced them). The main problem is that the above algorithms are not designed in a way to provide comparable coordinates for different embeddings, indeed they only focus on optimising some properties of the relative distances between words. Any comparison between word embeddings should be performed at the level of pairwise word distances.

Two assumptions are made for the method in [48]: that spaces are equivalent under a linear transformation, and that the meaning of most words (in a neighbourhood) did not shift over time (so that local structure is preserved). We aim to learn a linear transformation $\boldsymbol{W}(w)_{t' \Rightarrow t} \in \mathbb{R}^{d \times d}$ such that applying $\boldsymbol{W}(w)$ to word w at time t' will map the word to what its position would be at time t.

This is achieved by taking the k nearest neighbours to the word w that is being checked for linguistic change. The aim is to learn a matrix $\boldsymbol{W}$ such that those nearest neighbours from time period t' are accurately transformed to their counterparts in time t. To achieve this, we use the

following linear regression model:

$$\underset{t' \Rightarrow t}{\boldsymbol{W}}(\mathrm{w}) = \underset{\boldsymbol{W}}{\operatorname{argmin}} \sum_{w_i \in k-NN(\phi'_t(w))} ||\phi'_t(w_i)\boldsymbol{W} - \phi_t(w_i)||_2^2 \tag{3.23}$$

When we transform the words from their source to the target embedding at time t we will calculate the k nearest neighbours again at the transformed position of the word for each decade and calculate the pairwise distances for all words and the temporal instances of the word we are examining. This will be used to visualise the movement of words throughout time.

Time Series Construction

To monitor the changing of words over time, all words can be aligned to the initial embedding space (Φ_0) using the linear mapping $\boldsymbol{W}$ learned from Eq.3.23. A distributional times series can be constructed by calculating the distance in the embedding space between the transformed word at time t ($\Phi_t(w)\boldsymbol{W}_{t\to 0}(w)$) and the original word at time 0 ($\Phi_0(w)$) as follows [48]:

$$\tau_t(w) = 1 - \frac{(\Phi_t(w)\boldsymbol{W}_{t\to 0}(w))^T \Phi_0(w)}{||(\Phi_t(w)\boldsymbol{W}_{t\to 0}(w)||_2 ||\Phi_0(w)||_2} \tag{3.24}$$

This would be done for every further word embedding that takes place after the initial time period to construct a time series that provides a metric for the movement of the word w over time periods t.

3.9 Language Resources

3.9.1 Linguistic Inquiry Word Count

Some studies in this work uses lists of words generated by the LIWC project [74], a long-running effort in social psychology to handcraft, vet and validate lists of words of clinical value to psychologists. They typically aim at capturing concerns, interests, emotions, and topics of psychological significance. LIWC lists are well suited to an experiment of this kind as the words within them are common and relevant to any cross-domain corpus.

Table 3.3: Sample words from the LIWC word lists used in experiments

Full Name	Sample Words	List Name
Positive Emotions	happy, pretty, good	posemo
Negative Emotions	hate, worthless, enemy	negemo
Anger Processes	hate, kill, pissed	anger
Biological Processes	eat, blood, pain	bio
Relativity	area, bend, exit	relative
Affective Processes	happy, ugly, bitter	affect
Social Processes	talk, us, friend	social
Work Concerns	work, class, boss	work
Family Concerns	mom, brother, cousin	family
Health Concerns	weak, heal, blind	health

Tab.3.3 shows samples of the ten word lists used in this study as well their full names, and their name when used in the context of this study. Most word lists used have hundreds of words in them. Family is the smallest word list with a total of 54 words being used. These word samples will be what are used to extensionally define the word list as a concept within the embedding. This is consistent with the Extensional Definition of a concept used in logic, and the same definition of concept as used in the PAC model of machine learning [3].

3.9.2 Office of National Statistics

For some experiments in this work we take data provided from the Office of National Statistics. We first generated a list of 62 occupations from data published by the Office of National Statistics [69], filtering the list to only include those occupations for which there is reliable employment statistics and can be summarised by a single word in the embedding, *e.g.* doctor, engineer, secretary. We have also sourced an older list of occupations that will be used in different experiments later on, these lists were also provided by the Office of National Statistics using census data from the 1800's [66]. This older list of provides 107 unique occupations to use in experiments that were not in the list of 62 modern occupations, bringing us a total of 169 occupations.

3.9.3 ICD-10 Diagnosis Chapters

ICD-10 is the tenth iteration of the International Statistical Classification of Diseases and Related Health Problems, a medical classification list by the World Health Organisation (WHO) [70]. It

Table 3.4: Sample diagnosis codes from ICD-10 used in experiments. There are twenty two chapters of diagnoses, this is a small sample of those lists along with two samples of diagnoses that exist in each list.

Full Name	Sample Words	List Name
Blood Diseases And Immune Disorders	Hypersplenism, Eosinophilia	BDID
Skin And Subcutaneous Tissue Disease	Pyoderma, Other sunburn	SSTD
Infectious And Parasitic	Typhoid fever, Balantidiasis	IP
Respiratory Diseases	Mixed asthma, Stannosis	RD
Perinatal Period Conditions	Neonatal coma, Extreme immaturity	PPC
Neoplasms	Multiple myeloma, Myeloid sarcoma	NP
Nervous System Diseases	Chronic meningitis, Myoclonus	NSD

contains codes and formal names for diseases, aliments symptoms, and causes of injury or death. These codes and names are split into twenty two different chapters (categories of diagnoses). In total there are approximately 12,000 diagnoses split among these chapters. I will be using these chapters as sets (or unordered lists) of training samples in classification work.

Tab.3.4 is a sample of the diagnoses chapters from ICD-10, along with some samples words that appear and an abbreviation for each chapter. A full table appears in in the appendix with samples for each word (App.A). The smallest number of diagnoses in a chapter is " Blood Diseases and Immune Disorders" with 161 diagnoses. ICD-10 is used in this work along with BERT and BioBERT. Some diagnoses are sequences of words, rather than just words so they cannot be represented by a single vector, in comparison to 3.9.1. However both embeddings I use with ICD-10 uses character sequences, so there are representations for character sequences within each embedding.

3.9.4 British Historical Newspapers

Find my past have provided the University of Bristol access to their 150 year collection of periodicals for the purpose of research. This corpus contains news articles over 150 years from 1800 to 1959. This data (which cannot be released for Copyright reasons) has used in other related studies [49].

3.9.5 GloVe Pre-trained Embeddings

GloVe provides pre-trained embeddings trained from a variety of corpora using Wikipedia, newspaper articles, and twitter [75]. In this work I use the "6 billion token" GloVe pre-trained word embeddings in various experiments throughout this work. [1]

3.9.6 BERT Pre-Trained Word Embeddings

BERT provides pre-trained word embeddings with various versions trained on different hyper-parameters (namely the number of transformer blocks L, and the hidden layer size H) [20]. In this work the model known as “bert-base-uncased” is used in experiments. For these experiments L is set to 12 and H is set to 768 . [2]

3.9.7 BioBERT

BioBERT provides pre-trained word embeddings specifically for the biomedical field [50]. Using the same embedding pre-training step as BERT the source corpus is different, focusing on the biomedical field to better represent language used in the specialised field. BioBERT provides pre-trained word embeddings and are publicly available for download. [3]

[1] The GloVe pre-trained models are available at the following url: https://nlp.stanford.edu/projects/glove/
[2] The BERT pre-trained models are available at the following url: https://github.com/google-research/bert
[3] The BioBERT pre-trained models are available at the following url: https://github.com/dmis-lab/biobert

CHAPTER 4

STATISTICAL ANALYSIS OF EMBEDDINGS THROUGH CONCEPTS

An important problem in designing word embeddings is how to evaluate their quality, a measure of quality can be used to compare the merits of different algorithms, different training sets, and different parameter settings. Importantly, it can also be used as an objective function to design new and more effective procedures to learn embeddings from data. Currently, most word embedding methods are trained based on statistical co-occurrence information and are then assessed based on criteria that differ to the original training task.

I propose a criterion of quality for word embeddings, and then present a statistical methodology to compare different embeddings. The criterion would fall under the intrinsic class of methods in the classification of Schnabel et. al. [86], and has similarities with both their coherence criterion and with their categorisation and relatedness criteria. However it makes use of the notion of “concept learnability" based on statistical learning ideas. I make use of extensional definitions of concepts, as they have been defined by [3].

The key part in this study is that of a “concept". If the set of all words in a corpus is called a vocabulary, I define any subset of the vocabulary as a concept. I categorise a concept as learnable if it is possible for a learning algorithm to be trained on a random subset of its words, and then recognise the remaining words. I believe that concept learnability captures the essence of semantic similarity, and if the list of words has been carefully selected, vetted and validated by

rigorous studies, it can provide an objective way to measure the quality of the embedding.

In the first experiment I will measure the learnability of LIWC lists. I compare LIWC lists to randomly generated word lists for pretrained word embeddings of four different algorithms (Randomly Generated Embeddings, GloVe [75], word2vec [62], and fastText [61]). I show that LIWC concepts are represented in all embeddings except those randomly generated, through statistical testing.

In the second experiment I compare the performance of the three embedding algorithms using the same method as previous, however for this experiment I train with the same hyper parameters and corpus ([103]) across all three word embeddings [1]. This experiment shows that fastText performs the best, performing significantly better than both word2vec and GloVe.

The third experiment I perform focuses on deep contextual embeddings from BERT. I first look to see if a classifier similar to that used in previous experiments will perform similarly when using BERTs deeper embeddings [20]. I find that a linear classifier is not as accurate when learning the same concepts as before using BERT embeddings. These concepts however prove to still be learnable, when using a deeper classifier that is defined by BERT.

I then look at the impact of BERT embeddings trained on different corpora using this classification task. I compare the performance of a BERT embedding trained from a more general corpus [20] to an embedding trained from a medical domain corpus, BioBERT [50]. I find that general concepts are better represented within the general domain embeddings, while the specialised domain is better at identifying specialised concepts.

Finally, for context free embeddings I look to see if LIWC concepts have their meaning captured within a single vector to be used for further experiments. I calculate the vector mean for samples of the positive and negative emotional word lists. These sample means are then used to distinguish between the remaining samples of the word list, and negative samples. This work shows that vector means accurately capture the concepts described by their word lists.

[1] The work presented in the first and second experiment has been published and presented in AIAI 2020 "On the Learnability of Concepts" where I worked with Nello Cristianini [94]

4.1 Measuring the Learnability of LIWC Lists

I will measure the performance of a linear classifier by using the receiver operating characteristic (ROC) curve, a quantity defined as the performance of a binary classifier as its prediction threshold is changed between the lowest probable prediction and its highest probable prediction. This curve plots the TPR and FPR at each classification threshold possible. For posterity, I also show the accuracy and precision of the classifier.

My first experiment will look at the three word embedding algorithms of GloVe, word2vec, and fastText with regards to how they perform using pre-trained word embeddings readily available online. For the training set $\mathbf{L_{train}}$, I uniformly random sample half of the words from the list $\mathbf{L}$ I am experimenting on. I then sample an equal number of words from $\mathbf{L^c}$. For the test set $\mathbf{L_{test}}$ I take the remaining words from $\mathbf{L}$, and again sample another equal set of negative test samples from $\mathbf{L^c}$.

This method is repeated 1,000 times, and for each iteration of this test I sample new word lists for $\mathbf{L_{train}}$ and $\mathbf{L_{test}}$ each time. This method of a linear classifier has been defined in Eqn. 3.21 and Eqn. 3.22. This experiment is performed for the 10 LIWC word lists listed in Tab. 3.3. I take their average performance metrics across all 1,000 iterations of the experiment performed (for display purposes).

4.1.1 Learning Concepts from Embeddings

4.1.1.1 Random Embeddings

In this section I look using at concepts that are defined by Linguistic Enquiry Word Count (LIWC) [74] word lists to see if they can be represented using randomly generated word embeddings. In this experiment I hypothesise that randomly generated word embeddings will be unable to correctly predict members of a LIWC word list that has a semantic consistency in the real world.

The embedding algorithm in this experiment that I use is sampled from a Gaussian distribution with a μ of 0 and a variance of 1 ($\sim \mathcal{N}(0,1)$). This Gaussian distribution is sampled for each dimension for each word vector within the embedding. The vocabulary $\mathbf{V}$ will be the same vocabulary as that used by the GloVe pretrained embeddings [75]. However no corpus $\mathbf{C}$ is

Table 4.1: Average Performance of Linear Classifiers using LIWC word lists on randomly generated word embeddings to identify members of its own set.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.500	0.484	0.488	0.498	0.495
$\mathbf{L_{negemo}}$	492	0.502	0.492	0.486	0.503	0.505
$\mathbf{L_{anger}}$	184	0.494	0.487	0.499	0.494	0.492
$\mathbf{L_{bio}}$	558	0.506	0.492	0.483	0.505	0.504
$\mathbf{L_{relative}}$	632	0.500	0.423	0.423	0.279	0.503
$\mathbf{L_{affect}}$	908	0.499	0.490	0.490	0.500	0.499
$\mathbf{L_{social}}$	396	0.495	0.485	0.493	0.496	0.493
$\mathbf{L_{work}}$	322	0.503	0.496	0.489	0.503	0.500
$\mathbf{L_{family}}$	54	0.495	0.509	0.518	0.495	0.505
$\mathbf{L_{health}}$	232	0.504	0.499	0.499	0.499	0.499
$\mathbf{L_{random(max)}}$	400	0.57	0.572	0.4	0.571	0.566
$\mathbf{L_{random(avg)}}$	400	0.496	0.482	0.490	0.496	0.493

required for these embeddings as statistical co-occurrence from a corpus is not used.

To achieve this I use a linear classifier training on half of a LIWC word list along with the same number of negative samples (sampled uniformly from the vocabulary **V**). I then test on the remaining words from the LIWC list, along with another equal number of samples from **V** and look at the performance of the binary classifier. This method is as described in Sec. 3.21.

As shown in Tab. 4.1, randomly generated word embeddings fail to reflect word lists that have real world semantic meanings such as LIWC. All word lists perform equally as random in predicting members of the concept that it is representing. This confirms that random embeddings are unable to capture semantic information in its embedding space, confirming the hypothesis.

4.1.1.2 GloVe

I set GloVe to be an embedding algorithm to test (Φ), with the corpus **C** being a collection of Wikipedia and Gigaword 5 news articles. These embeddings are pretrained and available online on the GloVe web-page [38]. These word embeddings are open for anyone to use, and can be used to repeat these experiments.

Tab. 4.2 shows the performance and statistics of ten different word lists from LIWC. $\mathbf{L_{random(avg)}}$ shows the average performance of concepts defined from random word lists. $\mathbf{L_{random(max)}}$ shows the best performance of that statistic from all random iterations.

Table 4.2: Average Performance of a Linear Classifiers using LIWC word lists on GloVe word embeddings to identify members of its own set. Random word lists are also tested to obtain a p-value and compare performances. These embeddings perform better than random word lists resulting in a p-value of < 0.001

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.915	0.902	0.079	0.919	0.964
$\mathbf{L_{negemo}}$	492	0.913	0.913	0.085	0.915	0.965
$\mathbf{L_{anger}}$	184	0.888	0.880	0.103	0.896	0.950
$\mathbf{L_{bio}}$	558	0.895	0.871	0.087	0.909	0.954
$\mathbf{L_{relative}}$	632	0.937	0.935	0.059	0.940	0.979
$\mathbf{L_{affect}}$	908	0.910	0.906	0.085	0.914	0.962
$\mathbf{L_{social}}$	396	0.906	0.887	0.075	0.922	0.962
$\mathbf{L_{work}}$	322	0.899	0.880	0.081	0.916	0.959
$\mathbf{L_{family}}$	54	0.884	0.893	0.125	0.881	0.956
$\mathbf{L_{health}}$	232	0.895	0.880	0.105	0.893	0.953
$\mathbf{L_{random(max)}}$	400	0.547	0.32	0.115	0.617	0.574
$\mathbf{L_{random(avg)}}$	400	0.500	0.198	0.198	0.502	0.501

An accuracy of approximately 0.9 shows a high general performance. The precision and recall show that these word lists are able to accurately discern remaining members of its list and words that are not a part of the concept. After a thousand iterations of random word lists the best performing random lists (shown in $\mathbf{L_{random(max)}}$) were performing worse than each LIWC word list, giving a p-val of < 0.001 for each word list.

4.1.1.3 word2vec

The next embedding algorithm I use is word2vec (Φ), with the corpus $\mathbf{C}$ being a dump of Wikipedia from April 2018 [107] using the conventional skip-gram model. These embeddings are available online from the Wikipedia2Vec web-page [107]. These word embeddings are available for anyone to use, and can be used to repeat these experiments.

Tab. 4.3 shows the performance and statistics of ten different word lists from LIWC while using the word2vec embedding algorithm. $\mathbf{L_{random(avg)}}$ and $\mathbf{L_{random(max)}}$ again show the average and best performances of random word lists.

An accuracy of approximately 0.9 shows a high general performance, although it performs generally worse than GloVe's pre-trained embeddings. The precision and recall are again performing much better than random as shown in Sec. 4.1.1.1. This shows that the word2vec embedding

Table 4.3: Average Performance of Linear Classifiers using LIWC word lists on word2vec embeddings to identify members of its own set. Random lists word are also tested to obtain a p-value and compare performances. These embeddings perform better than all random word lists resulting in a p-value of < 0.001

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.904	0.914	0.115	0.888	0.959
$\mathbf{L_{negemo}}$	492	0.923	0.920	0.081	0.919	0.970
$\mathbf{L_{anger}}$	184	0.890	0.906	0.126	0.879	0.953
$\mathbf{L_{bio}}$	558	0.890	0.901	0.120	0.882	0.954
$\mathbf{L_{relative}}$	632	0.911	0.952	0.135	0.876	0.963
$\mathbf{L_{affect}}$	908	0.886	0.947	0.177	0.842	0.950
$\mathbf{L_{social}}$	396	0.893	0.911	0.123	0.881	0.957
$\mathbf{L_{work}}$	322	0.877	0.910	0.154	0.855	0.947
$\mathbf{L_{family}}$	54	0.874	0.912	0.164	0.853	0.953
$\mathbf{L_{health}}$	232	0.893	0.899	0.113	0.889	0.959
$\mathbf{L_{random(max)}}$	400	0.545	0.27	0.055	0.68	0.576
$\mathbf{L_{random(avg)}}$	400	0.498	0.128	0.130	0.494	0.500

Table 4.4: Average Performance of Linear Classifiers using LIWC word lists on fastText embeddings to identify members of its own set. Random lists word are also tested to obtain a p-value and compare performances. These embeddings perform better than random word lists resulting in a p-value of < 0.001

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.928	0.925	0.068	0.931	0.977
$\mathbf{L_{negemo}}$	492	0.937	0.934	0.067	0.932	0.978
$\mathbf{L_{anger}}$	184	0.940	0.965	0.084	0.919	0.981
$\mathbf{L_{bio}}$	558	0.917	0.933	0.098	0.905	0.970
$\mathbf{L_{relative}}$	632	0.933	0.966	0.099	0.907	0.977
$\mathbf{L_{affect}}$	908	0.886	0.947	0.177	0.842	0.950
$\mathbf{L_{social}}$	396	0.927	0.920	0.074	0.925	0.973
$\mathbf{L_{work}}$	322	0.918	0.914	0.077	0.922	0.970
$\mathbf{L_{family}}$	54	0.966	0.975	0.041	0.960	0.995
$\mathbf{L_{health}}$	232	0.931	0.940	0.078	0.924	0.980
$\mathbf{L_{random(max)}}$	400	0.51	0.04	0.0	1.0	0.562
$\mathbf{L_{random(avg)}}$	400	0.500	0.007	0.006	0.427	0.505

algorithm Φ applied to the corpus $\mathbf{C}$ yields word vectors that represent the real world meaning of words. The AUC is extracted from the scores of the sigmoid within the classifier. Overall word2vec performs slightly worse than GloVe embeddings in most metrics. However while the source corpora are very similar, GloVe has additional sources of information. The p-values for these word lists in comparison to random word lists is again < 0.001 showing that these word lists that have a real world representation are represented accurately within the embedding.

Table 4.5: AUC performance of word lists for each embedding algorithm used in these experiments, along with the average AUC for an embedding across all lists. Bold denotes the embedding algorithm that performs best for a given word list. Italic denotes the best performing list for each embedding algorithm.

L	GloVe	word2vec	fastText
$\mathbf{L_{posemo}}$	0.961	0.929	**0.965**
$\mathbf{L_{negemo}}$	0.965	0.945	**0.973**
$\mathbf{L_{anger}}$	0.957	0.928	**0.970**
$\mathbf{L_{bio}}$	0.960	0.935	**0.974**
$\mathbf{L_{relative}}$	***0.971***	*0.927*	*0.961*
$\mathbf{L_{affect}}$	**0.960**	0.944	0.958
$\mathbf{L_{social}}$	0.960	0.925	**0.973**
$\mathbf{L_{work}}$	0.947	0.909	**0.970**
$\mathbf{L_{family}}$	0.948	0.864	**0.963**
$\mathbf{L_{health}}$	0.952	0.923	**0.975**
Mean	0.958	0.922	**0.968**
Median	0.960	0.927	**0.970**

4.1.1.4 fastText

The third and final word embedding algorithm (Φ) I test is fastText [43]. The corpus **C** is a collection of Wikipedia, “UMBC WebBase corpus" and statmt.org news [61]. These embeddings are also pretrained word embeddings that are available from the fastText website.

Tab. 4.4 shows the performance statistics of the fastText word embeddings using my proposed method to evaluate word embeddings. $\mathbf{L_{random(avg)}}$ and $\mathbf{L_{random(max)}}$ show the random performance, while the other lists are LIWC word lists and their respective performances.

A precision of 1 in the best performing random word lists are insignificant as the recall is shown to be poor, due to predicting most samples to be negative. The p-val of all of the word lists defined by LIWC is < 0.001 as after one thousand iterations no random list outperformed any of LIWC lists. This again means that these word lists represent a real world concept, and that the embeddings are able to capture this information of this concept by using members of the set within the embedding to define it.

4.1.2 Comparing Embeddings

In this section I will be comparing the performance of the three word embedding algorithms used in the previous experiment. However, for this experiment the hyper parameters and the corpus

trained will be fixed for the purpose of direct comparison. All embeddings have been generated by myself using the three word embedding algorithms word2vec (skip-gram), GloVe, and fastText.

The AUC metric previously shown can be viewed as a measure of the learnability of an embedded concept. This compares the TPR and the FPR and shows the performance at each threshold that is possible within the classifier on for a given word lists test set.

This AUC shows the performance of that binary classifier, and also as a measure of the quality of each embedding and a measure of the quality of each word list. The better the performance of an embedding, the higher perceived quality of that embedding. The better a list performs on all embeddings (except for random which I have demonstrated not to encode any semantic or syntactic information), the higher the quality of that list.

To accurately compare the performance of the embedding algorithms, the same test as shown in Sec. 4.1.1 is performed. However I ensure that a number of parameters are kept the same for each embedding, to maintain fairness. For this test, the corpus used to train will be identical between all embeddings. The corpus ($\mathbf{C}$) used for all three embedding algorithms will be a dump from the English Wikipedia taken from the first of July, 2019 [103]. The embedding dimension d will be set to 300. A word must appear a minimum of five times to be embedded, and the context window of all words is five.

In Tab. 4.5 the AUC performance of all three embedding algorithms is shown. The fastText embedding algorithm is shown to be the highest performing embedding for 8 of the 10 lists that have been tested. Glove performs best on two lists, and generally performs better than word2vec overall. These performances are consistent with previous comparisons of these word embeddings [75] [61]. The word list $\mathbf{L}_{\mathbf{relative}}$ is shown to have the best overall performance across all three non-random embeddings, demonstrating the quality of that list.

I tested the statistical significance of the performance differences observed between GloVe and fastText. To this purpose I performed a Wilcoxon signed-rank test, using the median of the AUCs from each embedding as the test statistic [104]. I use the Wilcoxon signed-rank test as the distribution of AUC scores is unknown.

I propose a null hypothesis that the median difference of fastText and GloVe AUCs (as shown in Tab. 4.5) are 0. I used a sample size of 10 as the difference of no pairs are equal to zero. Setting

the alpha to 0.01 for a one sided (right) tail test, where the test statistic W_{crit} is 5. It is shown the resulting W_{test} to be 3, which leads me to reject the null hypothesis and show that fastText outperforming GloVe is statistically significant for the word lists that I am testing with a p-value of 0.0088.

4.2 Concepts in Deep Contextual Embeddings

Word embeddings in the previous section are what are known as "context free embeddings", a key part of these embeddings is that they only take the window of words or tokens into account once when learning word embeddings. "Deep contextual word embeddings" are a more recent advancement in natural language that generate word embeddings depending on the context around it. Popular examples of these algorithms are ELMo [76] and Bert [20].

In this section I investigate if concepts within these new embeddings are learned, and compare their performance to context free embeddings in the previous work (Sec. 4.1). I will again be using LIWC word lists for comparison. For this work it is imperative to note that deep contextual word embeddings will rarely be represented by a single d dimensional vector. Words are now more likely to be represented by N character sequences that when combined will represent the word. The resulting representation for a given word will be a $N \times d$ matrix, where N is the number of character sequences (or tokens) needed to construct the word representation and d is the dimension of representation for those sequences.

These changes to generating embeddings for words pose a new challenge for the method of learning concepts with a classifier to check the performance of word embeddings. A list of words, and the potential negative samples could have different numbers of character sequences to represent different members of the set. A common solution to this is to find the largest number of sequences in your training and testing sets possible as an input and to then pad every other input to be this size. "Padding" in this instance is a design choice specific to the embedding which indicates a token that can be ignored as part of the input.

In all future work in this section, I will be working with BERT embeddings, that will be using padding where applicable to my work. As input embeddings for single words are $N \times d$, another

Table 4.6: Average Performance of linear classifiers using LIWC word lists on flattened BERT embeddings to identify members of its own set. Random lists are also tested to obtain a p-value and compare performances. These embeddings perform significantly worse than previous iterations of work

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.676	0.412	0.065	0.433	0.678
$\mathbf{L_{negemo}}$	492	0.544	0.171	0.079	0.145	0.055
$\mathbf{L_{anger}}$	184	0.563	0.153	0.048	0.160	0.558
$\mathbf{L_{bio}}$	558	0.499	0.100	0.102	0.233	0.500
$\mathbf{L_{relative}}$	632	0.656	0.443	0.127	0.415	0.658
$\mathbf{L_{affect}}$	908	0.529	0.181	0.124	0.131	0.526
$\mathbf{L_{social}}$	396	0.665	0.433	0.096	0.421	0.664
$\mathbf{L_{work}}$	322	0.672	0.413	0.062	0.444	0.667
$\mathbf{L_{family}}$	54	0.746	0.605	0.119	0.595	0.721
$\mathbf{L_{health}}$	232	0.584	0.230	0.070	0.632	0.605

issue is how to use a linear classifier identify words belonging to a concept. For this work I am using BERT pre-trained embeddings [20]. I am using the "bert-base-uncased" embeddings where d is set to 768 and 12 layers (commonly denoted as **L**) are used in the pre-training method. **L** is generally seen as the number of layers used in the pre-training task of predicting surrounding tokens and correct sentences as explained in Sec. 3.3.

4.2.1 Learning Classifiers in Deep Contextual Embedding

The first part of this work I will try to adapt the embeddings generated from BERT to work within a linear classifier. To do this I will flatten the input embedding to be $1 \times d * N$. Where N is the max number of tokens needed to represent a word from both the training and testing set, and d is the number of dimensions representing one token. This work is being done as a “fair” comparison to the work done with context free word embeddings from Sec. 4.1.1, and to see how BERT embeddings work when using a linear algorithm.

Tab. 4.6 shows the performance of linear classifiers trained using flattened BERT embeddings to identify members of their respective LIWC word lists. In comparison to any of the context free embeddings in Sec. 4.1.1, it is clear that BERT is not as able to recognise and represent concepts suitably for a linear classifier. It is more important to note that flattening BERT embeddings is not a standard method, and was only done to compare embedding algorithms in a “fair” environment

Table 4.7: Performance of deep sequence classifiers using LIWC word lists on BERT embeddings to identify members of its corresponding LIWC set. Random lists are also tested to obtain a p-value and compare performances. These embeddings perform better than random word lists resulting in a p-value of < 0.001

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	392	0.952	0.938	0.037	0.955	0.966
$\mathbf{L_{negemo}}$	492	0.926	0.976	0.125	0.893	0.982
$\mathbf{L_{anger}}$	184	0.917	0.933	0.089	0.901	0.981
$\mathbf{L_{bio}}$	558	0.896	0.941	0.141	0.854	0.959
$\mathbf{L_{relative}}$	632	0.959	0.961	0.047	0.950	0.977
$\mathbf{L_{affect}}$	908	0.928	0.941	0.078	0.922	0.976
$\mathbf{L_{social}}$	396	0.943	0.957	0.079	0.933	0.984
$\mathbf{L_{work}}$	322	0.943	0.957	0.079	0.933	0.984
$\mathbf{L_{family}}$	54	0.922	0.868	0.031	0.964	0.984
$\mathbf{L_{health}}$	232	0.905	0.936	0.133	0.867	0.944
$\mathbf{L_{random(max)}}$	400	0.51	0.04	0.0	1.0	0.562
$\mathbf{L_{random(avg)}}$	400	0.500	0.007	0.006	0.427	0.505

where the constraints are as similar as possible.

BERT has its own network heads defined for a multitude of different tasks. An appropriate network for this classification challenge posed would be sequence classification where the pre-trained VERT embeddings are used as input to transformer networks connected end to end to produce a prediction. This doesn't require flattening the embeddings for the sequences, however it will still require padding. Tab. 4.7 shows the performance of a deep sequence classifier using transformers work with BERT embeddings to identify members of its corresponding LIWC set. This performance is much more inline with context free embeddings and a linear classifier to identify concept words. It is unfair to compare those algorithms to BERT however, as the classifiers use different architectures and are vastly different in terms of what can be modelled between them. Linear classifiers are generally considered the weakest and computationally cheapest type of classifiers, while a BERT deep classifier can be considered computationally expensive.

In this section I have shown how to use classification to measure the performance of deep contextual word embeddings by adapting the linear classifier method demonstrated earlier. I have showed that LIWC list elements are identifiable by BERT embeddings when trained with a deep classifier. BERT when using a deep classifier showed comparable results to fastText,

GloVe, and word2vec. However these results are not fair comparisons as BERT utilises sequences of characters to form these words resulting in a matrix for an input word, where context free embeddings will have a single vector representation for a word giving two different computational challenges.

4.2.2 BERT and different source corpora

So far in this chapter I have shown word embeddings are able to capture concepts and learn them when given examples of that concept. I have also shown a use of this to mark the quality of context free word embedding algorithms. I consider there to be three inputs for this method; the embedding algorithm, the corpus, and the word list that encapsulates a concept. Linear classifiers are limited to samples being single words only, however with BERT and other deeper representations single words can still not be learned with a linear classifier.

In this experiment I will be looking at measuring the performance of different corpora for different tasks, while using BERT as the embedding algorithm for the task. I am going to compare the performance of two different variants of BERT; "bert-base-uncased" [20] and BioBERT [50]. I will be using deep sequential classifiers to gauge the performance of these embeddings on LIWC [74] and ICD-10 [70].

4.2.2.1 Comparing LIWC performances

The first experiment I will perform is looking at the performance of LIWC on BERT and BioBERT pre trained embeddings. Intuitively, we would believe that LIWC should perform better as a classifier using BERT, rather than BioBERT. For a fair comparison between LIWC and ICD-10 codes in this experiment, the negative samples making up the training and test sets for LIWC will be the words from LIWC lists that are not the focus of the training. An example would be that for $\mathbf{L_{posemo}}$, positive samples remain the words from the “posemo” word list but negative samples for testing or training could be from any of the other nine word lists. This is done as ICD-10 samples have varying lengths of words, and even words will have varying numbers of tokens (due to sub-word sequences being embedded instead of words). This allows for a more

Table 4.8: Performance of deep sequence classifiers using LIWC word lists on BERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are samples from the LIWC word lists not being trained in the binary classifier.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	404	0.901	0.916	0.114	0.890	0.956
$\mathbf{L_{negemo}}$	500	0.904	0.956	0.149	0.867	0.959
$\mathbf{L_{anger}}$	184	0.897	0.894	0.100	0.903	0.955
$\mathbf{L_{bio}}$	558	0.887	0.863	0.087	0.911	0.947
$\mathbf{L_{relative}}$	632	0.938	0.958	0.082	0.920	0.976
$\mathbf{L_{affect}}$	908	0.901	0.904	0.102	0.896	0.951
$\mathbf{L_{social}}$	396	0.864	0.880	0.153	0.854	0.928
$\mathbf{L_{work}}$	322	0.860	0.865	0.145	0.849	0.924
$\mathbf{L_{family}}$	54	0.907	0.852	0.037	0.958	0.986
$\mathbf{L_{health}}$	232	0.841	0.790	0.105	0.887	0.911

Table 4.9: Performance of deep sequence classifiers using LIWC word lists on BioBERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are the word lists not being trained in the binary classifier.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{posemo}}$	404	0.768	0.821	0.169	0.821	0.877
$\mathbf{L_{negemo}}$	500	0.834	0.869	0.202	0.814	0.904
$\mathbf{L_{anger}}$	184	0.734	0.660	0.189	0.785	0.766
$\mathbf{L_{bio}}$	558	0.853	0.873	0.167	0.844	0.910
$\mathbf{L_{relative}}$	632	0.878	0.911	0.154	0.853	0.942
$\mathbf{L_{affect}}$	908	0.848	0.871	0.174	0.830	0.914
$\mathbf{L_{social}}$	396	0.795	0.885	0.296	0.753	0.873
$\mathbf{L_{work}}$	322	0.699	0.763	0.361	0.665	0.726
$\mathbf{L_{family}}$	54	0.852	0.741	0.037	0.952	0.925
$\mathbf{L_{health}}$	232	0.785	0.832	0.263	0.767	0.838

even comparison between ICD-10 and LIWC, although the focus is on the algorithm and source corpora.

Tab. 4.8 shows the performance of deep sequence classifiers using LIWC word lists on BERT embeddings. Note that this table is different to the Tab. 4.7, as the negative samples are those words LIWC words that are not a part of the positive class. These results are as expected, with a slight decrease in performance compared to using a large vocabulary.

Tab. 4.9 shows the performance of deep sequence classifiers on BioBERT embeddings when the classifiers are trained to identify members of LIWC word lists. These results show a noticeable downgrade in performance of the classifier with a large difference in all performance metrics. To

Table 4.10: Performance of deep sequence classifiers using ICD-10 diagnoses chapters on BERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are chapters not being trained in the binary classifier. The full lists of performance and results can be found in the appendix.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{BDID}}$	161	0.832	0.871	0.198	0.772	0.924
$\mathbf{L_{SSTD}}$	332	0.855	0.863	0.153	0.863	0.918
$\mathbf{L_{IP}}$	697	0.901	0.930	0.125	0.869	0.963
$\mathbf{L_{RD}}$	224	0.817	0.871	0.241	0.795	0.896
$\mathbf{L_{PPC}}$	333	0.865	0.865	0.135	0.880	0.956
$\mathbf{L_{NP}}$	702	0.972	0.961	0.017	0.983	0.993
$\mathbf{L_{NSD}}$	276	0.841	0.879	0.199	0.820	0.939

compare the significance of these experiments I am again going to perform a Wilcoxon signed rank test to measure the statistical significance seen between BERT and BioBERT. As previously done in Sec. 4.1.2, I will compare the distribution of AUCs for each embedding. With an alpha of 0.01, the null hypothesis of the median difference is rejected. The p-value for this experiment is 0.002. This experiment shows that LIWC performs significantly better on BERT embeddings trained on Wikipedia and more a general corpus compared to BioBERT embeddings trained on medical a medical corpus.

4.2.2.2 Comparing ICD-10 performances

The second experiment will look at how the same embeddings are able to correctly distinguish correct ICD-10 diagnosis titles when trained on half of the chapter, and tested on the remaining half. The negative samples for this experiment in training and testing are randomly sampled from the chapters that are not from the positive set. There are no diagnoses that are repeated in any chapters. In this work I hypothesise that BioBERT will out-perform BERT in these classification tasks.

Tab. 4.10 shows a sample of the performance of ICD-10 trained classifiers on BERT embeddings. The full table showing the performance of all ICD-10 chapters can be found in App. B. BERT shows that it is generally capable of correctly identifying members of each chapter when trained appropriately.

Tab. 4.11 shows a sample of the performance when training ICD-10 classifiers on BERT

Table 4.11: Performance of deep sequence classifiers using ICD-10 diagnoses chapters on BioBERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are chapters not being trained in the binary classifier. The full lists of performance and results can be found in the appendix.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{BDID}}$	161	0.925	0.974	0.119	0.882	0.993
$\mathbf{L_{SSTD}}$	332	0.967	0.960	0.025	0.977	0.988
$\mathbf{L_{IP}}$	697	0.973	0.979	0.033	0.964	0.990
$\mathbf{L_{RD}}$	224	0.920	0.940	0.102	0.908	0.989
$\mathbf{L_{PPC}}$	333	0.967	0.961	0.026	0.977	0.990
$\mathbf{L_{NP}}$	702	0.997	0.997	0.003	0.997	0.999
$\mathbf{L_{NSD}}$	276	0.928	0.993	0.140	0.880	0.991

emebeddings. The full table showing all ICD-10 chapters performances with BioBERT are available at App. C. There is a visible improvement in the performance of BioBERT embeddings in this task when compared to BERT embeddings. However to find the statistical significance of this experiment I again perform a Wilcoxon signed rank test, comparing the median of the difference of both vectors (and a null hypothesis that the median of that difference is 0). I find that again that with a alpha of 0.01 the null hypothesis is rejected, and that BioBERT performs significantly better in this task. The p-value of this test was shown to be < 0.0009 showing a strong indication that BioBERT is a much better choice than BERT in tasks within the field of medicine.

4.3 Using Cosine Similarity to Score Words

The cosine similarity of individual words in a context-free embedding gives a simple view on how close those words are to each other within an embedding and therefore it can be utilised as a proxy for semantic similarity. This measures a simple similarity of one word to another within the embedding which may have inaccuracies due to the nature of natural language.

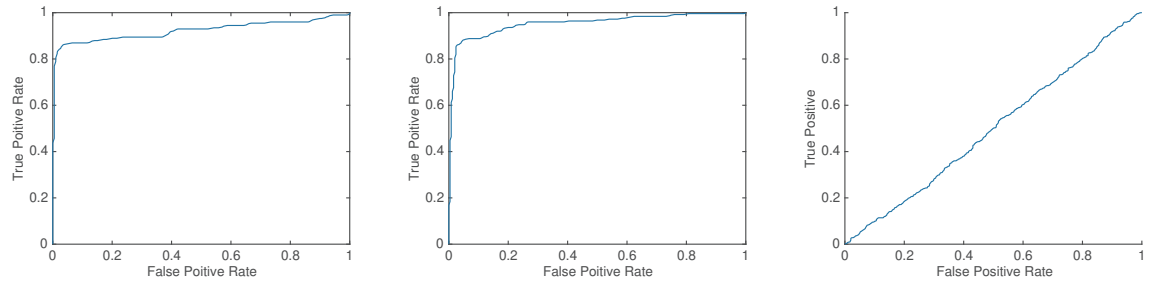
In future work I plan to measure how similar words are to a single concept. The work here shows that it is possible to accurately encapsulate concepts using a linear classifier on context free embeddings. The purpose of this work would be to show that measuring words to concepts (lists of words) would show non trivial results.

To achieve this I create a single d dimensional vector that summarises the concept underlying

(a) ROC of $\Phi(\mathbf{L}_{posemo})$ vectors with a AUC of 0.91

(b) ROC of $\Phi(\mathbf{L}_{negemo})$ vectors with an AUC of 0.92

(c) ROC of $\Phi(\mathbf{L}_{random})$ vectors with an AUC of 0.49



(d) Precision Recall of $\Phi(\mathbf{L}_{posemo})$

(e) Precision Recall of $\Phi(\mathbf{L}_{negemo})$

(f) Precision Recall of $\Phi(\mathbf{L}_{random})$

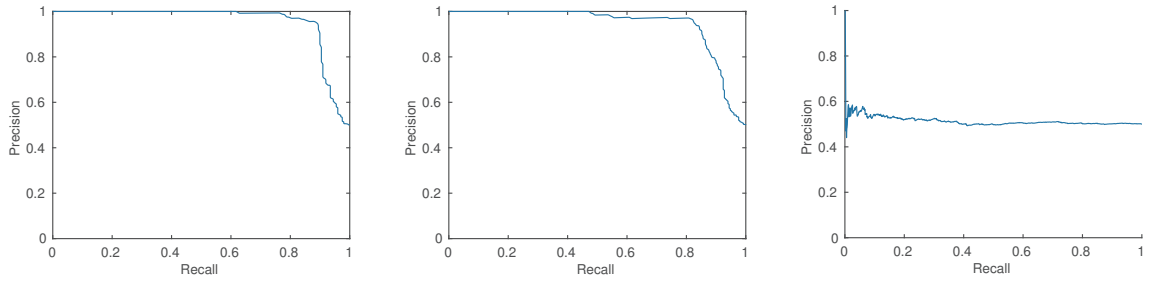


FIGURE 4.1. Precision Recall and ROC Curves for a modern corpus $\mathbf{C}$

a group of words. I used a list of words ($\mathbf{L}$) that form a concept and using the vector representations of those words find a general vector that summarises that concept within the embedding space. I therefore used the vector mean as described in Sec. 3.5 (Eqn. 3.13) to summarise half of the word list (producing $\mathbf{L}_{train}$), while using the remaining words as the real true samples of the test set ($\mathbf{L}_{test}$). I uniformly randomly sample from $\mathbf{V}$ the same number of test samples as $\mathbf{L}_{train}$ and append them to $\mathbf{L}_{test}$ to complete the test set.

I then produced a ROC curve, along with a measure of precision and recall seeing if it is possible to classify the words from the set better than random words. To validate this, I also followed the same process but with a random sampling of test and training words to verify that LIWC lists perform better than random lists in this task (which I performed on a random set). For my test I will be testing $\mathbf{L}_{posemo}$ and $\mathbf{L}_{negemo}$.

To show the human defined lists as capture significant information, I then sampled 200 random words to train on, and 400 random words to test on (denoted as $\mathbf{L}_{rand}$), where 200 were positive samples and 200 were negative samples. The difference in performance between $\mathbf{L}_{rand}$

and $\mathbf{L}_{posemo}$ and $\mathbf{L}_{negemo}$ shows the embeddings accurate representation of these concepts.

Fig. 4.1(a) and Fig. 4.1(b) show the ROC performance of both $\mathbf{L}_{posemo}$ and $\mathbf{L}_{negemo}$ word sets. With an AUC of 0.91 and 0.92 respectively this shows that it is possible to predict (better than random) the missing words within this set. In comparison to Fig. 4.1(c) using $\mathbf{L}_{rand}$ having an AUC of 0.49 shows a random classification for words within the set.

The precision and recall testing have similar results when looking at both the provided word lists and the random sampled lists also. Fig. 4.1(d) and Fig. 4.1(e) show good precision and recall for the word lists with an accurate performance while Fig. 4.1(f) shows an almost random performance as expected from a random list of words.

4.4 Discussion

In this chapter, I have shown that word embeddings are able to capture the meaning of human defined word lists. I have shown the ability of embedding algorithms in learning concepts from word lists. In particular I have shown this quality in word2vec, GloVe and fastText. This section has shown that learning embeddings from real data can represent real world concepts defined extensionally, utilising word lists provided by LIWC.

I have also shown the relative performance of GloVe, fastText, and word2vec when using LIWC word lists to form concepts using similar corpora that derive most of their corpus from Wikipedia. fastText performs better in the majority of situations for all word lists tested from LIWC, while GloVe outperforms word2vec generally. However as all algorithms use slightly different corpora, this result may change depending on the corpus used.

This measure of performance of word embeddings can be used in the future as a measure of “quality" of word embeddings. While there are other methods that look at the performance of word embeddings by evaluating their performance in a specific task [79], this method differs in that it looks at an embeddings general ability to accurately represent human defined concepts. There has also been criticism of evaluating word embeddings using only word similarity tasks [21]. This method can also be used in another way as a measure of the quality of word lists and their ability to accurately describe a concept, providing an assumption or proof that an embedding is

performing suitably to the users needs.

This method can be used to measure the performance of the three main inputs into the task. This methods three key inputs are the source corpus, the embedding algorithm, and the word lists that are being tested. If you keep two inputs static then you are able to compare the performance of the third input. In the previous experiment I did this with embedding algorithms, while testing them on the same word lists and source corpus.

Future work with this method would involve extensive testing of the method varying differing hyper parameters to see the optimal performance of these embedding algorithms. An example of this is the impact of embedding dimension on performance.

Further work could be focused on the performance of different word lists and concepts within word embeddings. The benefit of this could be to validate word lists that are not as carefully curated as LIWC word lists. These word lists may come from different fields, as LIWC is focused on clinical psychology other word lists may perform differently. Different source corpora may also change the performance of these word lists due to the meaning of some words changing from domain to domain. The LIWC project is an expensive and time consuming task to generate and maintain these hand crafted word lists, and this method may help in generating word lists at a fraction of the cost. Following this work on context free embeddings I have look at using this method with deep contextual embeddings. I compared the performance of two BERT embeddings that have been pre-trained on two different corpora. The first embedding for comparison was the "bert-base-uncased" embeddings that are provided by the original BERT paper [20]. The corpus used to train this embedding are a combination of Wikipedia and BookCorpus. BookCorpus is a collection of books used in NLP tasks [111]. The second embedding used is "BioBERT", which is a "biomedical language representation model designed for biomedical text mining tasks such as biomedical named entity recognition, relation extraction, and question answering" [50]. BioBERT has been found to perform the best on tasks within the biomedical field.

Using the method I aimed to test the performance of both embeddings by seeing if these embeddings could accurately distinguish between members of a set when provided with positive and negative samples. To show and compare the performance of an embedding's "general knowledge" I used LIWC word lists as the input for deep sequence classifiers [74]. To look at domain specific

biomedical knowledge I used "ICD-10" diagnoses names. With this experiment I wanted to see if a more general purpose corpus would perform better with general concepts (such as emotions and general life), and a domain specific corpus would perform better with word lists that are more much associated to that domain.

I showed that both embeddings were able to generally learn the concepts that are described by both sets of word lists. More importantly when comparing the embeddings performance in each task, I found that domain specific embeddings perform better than general embeddings when our classifier is identifying domain specific concepts. LIWC classifiers can be seen as a very general performance measure of word embeddings, as they are clinically proven and look at psychology, aspects of life, and widely agreed common concepts. While BioBERT was somewhat able to represent these LIWC concepts, it was shown to be significantly worse than BERT.

Using the same words lists I have shown it is possible to measure similarity to a concept within an embedding, if those word lists encapsulate a single concept. Overall this shows that using cosine similarity in a classification context I can find words from a LIWC list given a subset of that list, but this does not work with random word lists. It also shows that these concept word lists are able to measure the similarity to these concepts.

This work shows the method of using a classifier to validate the performance of embeddings. While embeddings are used upstream for real world tasks, these tasks may take a long time to train or fine tune even with pre-trained word embeddings. Using this as a method of evaluation for embedding algorithms, corpora, or word lists can provide two key benefits for NLP and machine learning. The training and testing time taken in these tasks is trivial; training for a single word list generally takes less than a minute in real time. The time taken generating embeddings and then fine tuning for many tasks can take a long time and slow down the general development process of AIs. If these tests are used as a cheap and approximate estimation of embedding quality then more focus can be spent on other challenges in the pipeline of developing AIs. The second benefit is that this is a quantifiable and human understandable metric of performance for these embeddings. Multiple embedding algorithms and NLP pipelines are claimed to have desirable performance at specific tasks without much reason to give humans confidence in those decisions. When choosing “the best” word embedding algorithm or training corpus for a NLP task,

this work and method may be able to give confidence in the decisions taken outside of single high metric of performance on that end task which cannot easily be explained.

Finally I looked at using cosine similarity to score words and find their similarity to concepts. I show that it is possible that mean vectors composed from a human defined word list represent a concept. The LIWC concepts of positive and negative emotions are used and are shown to perform significantly better than random concepts, which are made from randomly generated word lists. Mean vectors being able represent LIWC concepts enables them to be used in future work which focuses on the similarity between these concepts and words of interest.

CHAPTER 5

BIASES

Embeddings are primarily generated from data that has been gathered "in the wild". Data that has been gathered in this manner are prone to contain unwanted biases, that can carry over into embedding algorithms. In this chapter I make three contributions towards measuring, understanding, and mitigating this problem. I present a rigorous method to measure biases based on the use of LIWC word lists such as those in Sec.3.9.1; I observe how gender bias in occupations reflects actual gender bias in the same occupations in the real world. Finally, I demonstrate how a simple projection can significantly reduce the effects of embedding bias.

With this improved experimental setting (provided by LIWC word lists), I show that European-American names are viewed more positively than African-American names, male names are more associated with work while female names are more associated with family, and that the academic disciplines of science and maths are more associated with male terms than the arts, which are more associated with female terms. Using this new methodology, I then show the gender bias in the way different occupations are represented by the embedding. Furthermore, I use the latest official employment statistics in the UK, and find that there is a correlation between the ratio of men and women working in different occupation roles and how those roles are associated with gender in the word embeddings. This suggests that at least for gender and occupations that these biases in the embeddings reflect biases in the world.

Finally, I present a method of reducing gender bias from the word embeddings. Having established that there is a direction in the embedding space that correlates with gender, I use a simple orthogonal projection to remove that dimension from the embedding and reduce the bias. After projecting the embeddings, I analyse the effect on bias in the embeddings by considering the changes in biases between the words, demonstrating that the biases in the modified embeddings now correlate less to UK employment statistics among other things.

In all of these experiments, the first step is to obtain semantic vectors from a word embedding that I am to analyse. I use GloVe embeddings [75], pre-trained using a window size of 10 words on a combination of Wikipedia from 2014, and the English Gigaword corpus [71], where each of the 400,000 words in the vocabulary for this embedding are represented by a 300-dimensional vector. These vectors capture, in a quantitative way, the nuanced semantics between words necessary to perform meaningful analysis of words, reflecting the semantics found in the underlying corpus used to build them.

The Wikipedia data includes the page content from all English Wikipedia pages as they appeared in 2014 when a snapshot was taken. The English Gigaword corpus is an archive of newswire text data from seven distinct international sources of English newswire covering several years up until the end of 2010 [71].

5.1 Bias in Word Embeddings

5.1.1 LIWC Word Embedding Association Test (LIWC-WEAT)

In this experiment, I introduce the LIWC Word Embedding Association Test (LIWC-WEAT), where I have measured the association (defined in Eqn.3.13) between sets of target words with larger sets of attribute words known to relate to sentiment and gender coming from the LIWC lexica [74] [1]. I first use the target words from [12] which were originally used in [30], allowing a direct comparison of these findings with the original WEAT.

This approach differs from that of [12] in that while I use the same set of target words in each test, I use an expanded set of attribute words, allowing a more rigorous, systematic study of the

[1]This work has been published and presented at IDA 2018 "Biased embeddings from wild data: Measuring, understanding and removing" where I worked with Thomas Lansdall-Welfare, and Nello Cristianini [95]

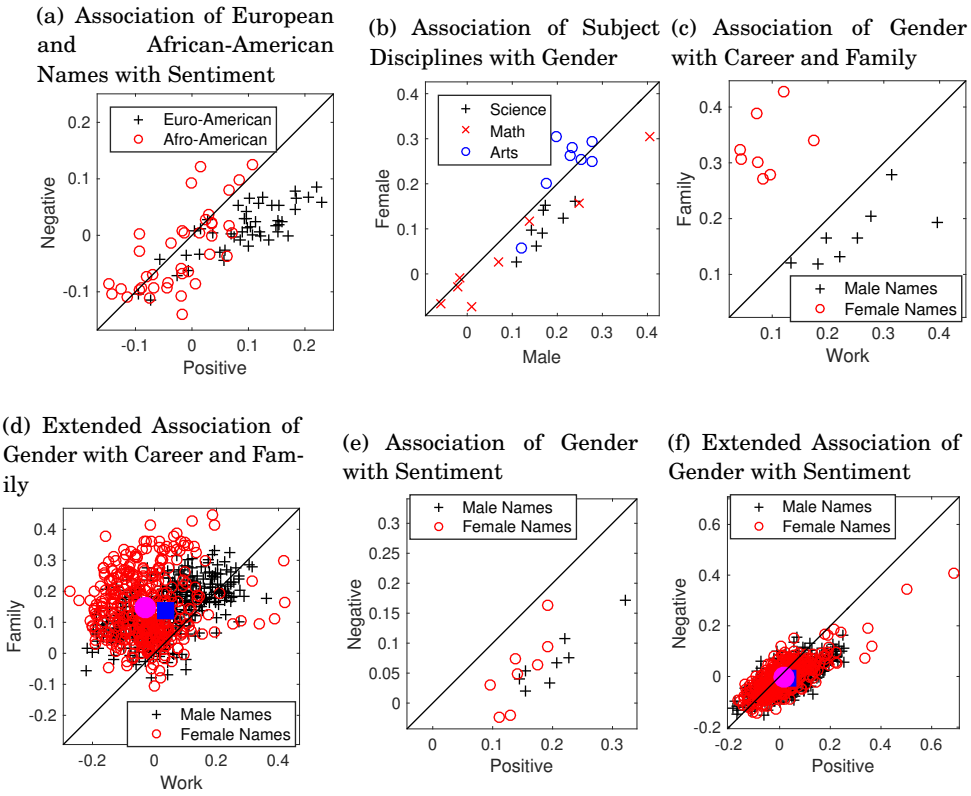


FIGURE 5.1. Association between different words and concepts in experiment 1, produced by the proposed LIWC Word Embedding Association Test (LIWC-WEAT).

associations found within the word embeddings. For this, attribute words are sourced from the LIWC, which has been explained in Sec.3.9.1. The categories specified in the LIWC lexica are based on many factors, including emotions, thinking styles, social concerns, and are well suited for non specific word embeddings such as these. For each of the original word categories used in [12], I matched them with their closest equivalent within the LIWC categories, for example matching the word lists for 'career' and 'family' with the 'work' and 'family' LIWC categories as seen in Tab.3.3.

I tested the association between each target word and the set of attribute words using the method described in Sec. 3.5, focusing on the differences in association between sentimental terms and European- and African-American names (the same names used in Caliskan's work [12]), subject disciplines to each of the genders, career and family terms with gendered names, as well as looking at the association between gender and sentiment.

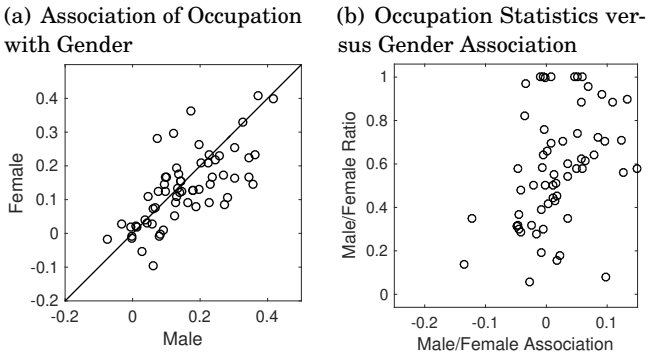


FIGURE 5.2. Gender biases and its relation to the number of men and women working in those roles

5.1.1.1 Association of European and African-American Names with Sentiment

Taking the list of target European-American and African-American names used in [12], I test each of them for their association with the positive and negative emotion concepts found in [74] by using the methodology described by Eq. 3.13 in Sec. 3.5, replacing the short list of words used to originally represent pleasant and unpleasant attribute sets.

This test found that while both European-American names and African-American names are more associated with positive emotions than negative emotions, the test showed that European-American names are more associated with positive emotions than their African-American counterparts, as shown in Fig. 5.1(a). This finding supports the association test in [12], where they also found that European-American names were more associated with pleasant words than African-American names in the corpus.

5.1.1.2 Association of Subject Disciplines with Gender

A further test was conducted to find the association between words related to different subject disciplines (*e.g.* arts, maths, science) with each of the genders using the 'he' and 'she' categories from LIWC [74].

The results of the test again support the findings of [12], with mathematics and science terms being more closely associated with males, while art terms are more closely associated with females, as shown in Fig. 5.1(b).

Table 5.1: List of the top 10 gender biased occupations from LIWC-WEAT.

Gender	Occupations most associated with a gender
Male	Manager, Engineer, Coach, Executive, Surveyor, Secretary, Architect, Driver, Police, Caretaker, Director
Female	Housekeeper, Nurse, Therapist, Bartender, Psychologist, Designer, Pharmacist, Supervisor, Radiographer, Underwriter

5.1.1.3 Association of Gender with Career and Family

Taking the list of target gendered names used in [12], I now test each of them for their association with the career and family concepts using the categories of 'work' and 'family' found in LIWC [74].

As shown in Fig. 5.1(c), the set of male names were more associated with the concept of work, while the female names were more associated with family, mirroring the results found in [12].

Extending this test, I generate a much larger set of male and female target names from an online list of baby names[2]. Repeating the same test on this larger set of names, male and female names were much less separated than suggested by previous results, with only minor differences between the two, as shown in Fig. 5.1(d).

5.1.1.4 Association of Gender with Sentiment

Extending the number of tests performed in the original WEAT study, I additionally decided to test the set of target male and female names and computed their association with the positive and negative emotions. This found that both sets of names are considered to be positive, similarly to the European-American and African-American names used in the previous test, but with male names appearing to be slightly more positive, as shown in Fig. 5.1(e).

I further tested these associations using the extended list of gendered baby names, as in Sec. 5.1.1.3, finding that there is no clear difference between the positive and negative sentiment attached to names of different gender in the word embedding.

[2]Baby names were taken from `http://bit.ly/2Dmqjco`, separated into two gendered lists.

5.1.2 Gender Biases in Occupations

In this experiment, I test the association between different occupations and gender categories coming from LIWC [74]. The association between each of the occupations is further contrasted against official employment statistics for the United Kingdom detailing the actual number of people working in each job role.

5.1.2.1 Association of Occupation with Gender

I first generated a list of 62 occupations from data published by the Office of National Statistics [69], filtering the list to only include those occupations for which there is reliable employment statistics and can be summarised by a single word in the embedding, *e.g.* doctor, engineer, secretary. For each of these occupations, I tested their association with each of the genders, as shown in Fig. 5.2(a), with the top ten occupations associated with each gender shown in Table 5.1. I found there was a 70% (p-value $< 10^{-10}$) correlation in the closeness of association between occupations and each of the gender attribute sets.

5.1.2.2 Occupation Statistics versus Occupation Association

Using the list of occupations from the previous section, I compared their association with each of the genders with the ratio of the actual number of men and women working in those roles, as recorded in the official statistics [69], where 1 indicates only men work in this role, and 0 only women. I find that there is a strong, significant correlation ($\rho = 0.57, p$-value $< 10^{-6}$) between the word embedding association between gender and occupation and the number of people of each gender in the United Kingdom working in those roles. This supports a similar finding for U.S. employment statistics using an independent set of occupations found in [12].

5.1.3 Minimising Associations via Orthogonal Projection

In this experiment, I deploy a method for reducing bias from word embeddings, first published in [10], and repeat all previous association tests related to gender reported in this paper, empirically showing the effect of bias removal on word associations.

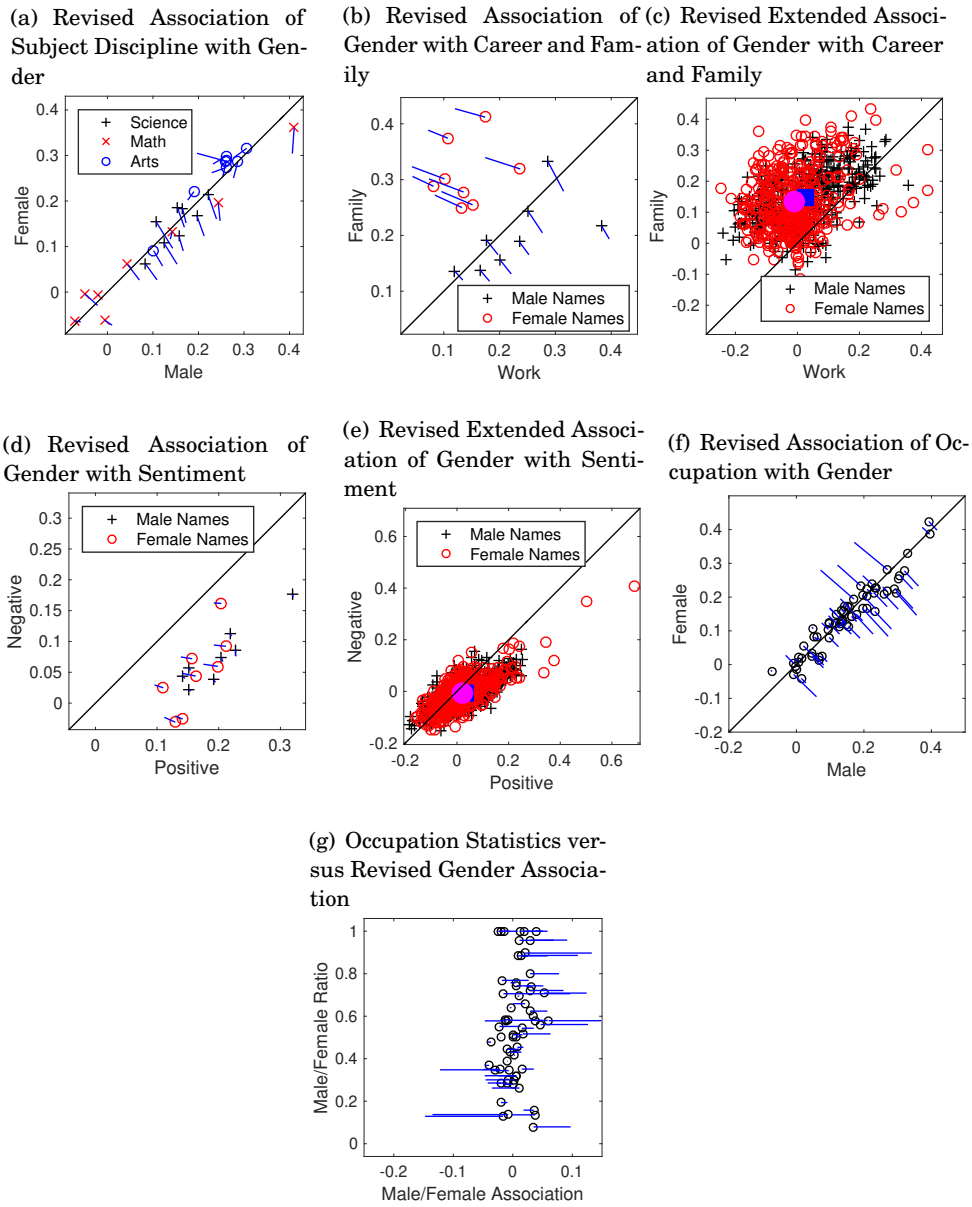


FIGURE 5.3. Association between different words and concepts in Experiment 3 after word vectors have been debiased via orthogonal projection in the gender direction. Line-traces shown in blue indicate where points have moved from after debiasing

5.1.3.1 Finding an Orthogonal Projection for Gender

To remove gender from the embedding, first find a projection within the space that best encapsulates the gender differences between words. To find the best projection, I began from a list of 5 gendered pronoun pairs in LIWC [74]. For each of the pairs, they are paired with their

gender-opposite, for example pairing “he” and “she”, “himself” and “herself” and so on. Taking the word vector from the embedding for each pronoun, I then computed their difference, as described in Sec. 3.6, giving a set of 5 potential gender projections.

Each gender projection was tested against an independent set of paired gender words sourced from WordNet [63] (containing implicit gendered words such as king and queen). After applying the gender projection to the test word-pairs, following the procedure of [10], I measured the average cosine similarity between the word-pairs. The gender projection that led to the WordNet word-pairs that are most similar (highest cosine similarity) was then selected as the ideal gender projection, corresponding to the difference between the vectors for “himself” and “herself”.

5.1.3.2 Revised Association Tests

Using the orthogonal gender projection found in the previous section, I repeated the tests from the LIWC-WEAT in Sec. 5.1.1 that were related to gender. This included the association of science, mathematics and the arts with gender, the association of male and females names with sentiment, work and family, and the ranking of occupations by their gender association.

In Experiment 1, I previously found that the disciplines of science and maths were slightly more associated with male terms in the embedding, while the arts were closer to female terms. The association of each of these subject disciplines with gender after orthogonal projection was found to be more balanced, with closer to equal association for both male and female terms, shown in Fig. 5.3(a).

Male and Females names tested in [12] showed a clear distinction in their association with work and family respectively, with my replication of the test in Sec. 5.1.1.3 finding the same results. Performing the same tests again after applying the gender projection to both name lists, I aimed to quantify the change in associations. I calculated the change in the distance between the centroids of each set of names before and after applying the orthogonal gender projection, finding that the association with work for males and family for females reduced, closing the gap between male and female names by 37.5% for the target names found in the original WEAT and 66% for the extended list of names respectively.

In the experiment looking at the association of positive and negative emotions with male and

female names, male and female names were both positive, with male names being slightly more associated with positive emotions than female names. The same findings were also true when using a larger set of names and making the same comparison. Applying the orthogonal gender projection to the word vectors, I again looked at how much the difference between the two sets was reduced. I found that for the target names found in the original WEAT, the distance between the two sets of names was reduced by 27%, while for the extended list the difference was reduced by 40%.

In Experiment 2, I find that there was a significant correlation of 70% between the male and female association of each occupation, while comparing the associations with official statistics of the number of men and women in each role showed a correlation of 53%. Again, applying the orthogonal gender projection and repeating these tests, I find that, on average, occupations moved closer to having an equal association with each of the genders (Fig. 5.3(f)) and that their association with gender was not significantly correlated ($\rho = 0.178, p\text{-value} = 0.167$) with the number of men and women working in each role.

5.2 Discussion

In this chapter, I introduced the LIWC-WEAT, a set of objective tests extending the association tests in [12] by using the LIWC lexica to measure bias within word embeddings. I found bias in both the associations of gender and race, as first described in [12], while additionally finding that male names have a slightly higher positive association than female names.

Biases found in the embedding were also shown to reflect biases in the real world and the media, where I found a correlation between the number of men and women in an occupation and its association with each set of male and female names.

Finally, using a projection algorithm [10], I showed the ability to reduce the gender bias shown in the embeddings, resulting in a decrease in the difference between associations for all tests based upon gender.

These experiments have demonstrated the effect of one debiasing procedure for reducing the association a given word has in a word embedding generated from natural language corpus

with concepts related to gender. Being able to do so relies on a set of gendered terms which are obtained from pairings with opposite meaning, allowing me to find an orthogonal projection within the space. This will not always be possible for every type of bias that may be desirable to remove (or at least reduce) in an embedding because there will not always be suitable word vector pairs that can be used to represent a given bias.

Other biases which are present may also be impossible to detect with this LIWC-WEAT method, as a pre-defined and validated list of words from LIWC were required to perform the tests. Other potentially undesired biases such as race or age are not currently able to be captured using the LIWC lexica, and thus different, carefully considered sets of words would need to be curated.

Another issue is objectively defining bias. In this work I used word lists that we trust accurately represent concepts, and then measure words to these concepts. In reality biases may not be a one dimensional value, and the word lists that we trust to help us understand biases may not be perfect. This is a philosophical issue where faith must be placed in some definition of bias, or concepts to be able to measure them. In these experiments, LIWC was used for this and with reason.

General solutions to this problem are probably impossible, for philosophical reasons, but I believe that biases can at least be mitigated or compensated for, by removing specific undesirable sub-types of bias, given there are ways to measure and detect them in the first place. However, in this process, care should also be taken as it may introduce or compound other existing biases in the embeddings.

If we want AI to take a central position in society, we need to be able to detect and remove any source of possible discrimination, to ensure fairness and transparency, and ultimately trust in these learning systems. Principled methods to measure biases will certainly need to play a central role in this, as will an understanding of the origins of biases, and new developments in methods that can be used to remove biases once detected.

There must also be care in removing all biases within certain intelligent systems. Systems that understand differences in gender, race, and other biases may be beneficial to certain AI systems. The objective of “fairness” and “transparency” may also be difficult to agree upon.

It may considered “fair” for an AI to understand that occupations are more associated to the better represented gender. Others may consider it harmful to perpetuate biases even if they are presented within statistics. There may also be other subjective opinions on fairness that find both opinions presented as unacceptable. Finding a consensus on “fairness” could prove to be a very difficult task, due to its subjectivity. This may provide an increased emphasis on “transparency”, if people using systems are aware of the biases present within them this may enable them to make informed decisions that are aware of issues with an intelligent system.

Further work in this direction will include removing bias in n-gram embeddings, embeddings that include multiple languages and new procedures for both generating better projections to remove a given bias, using debiased embeddings as an input to an downstream system and testing performance, and learning word embeddings which can be generated without chosen directions by construction.

5.3 Applying the LIWC-WEAT to Colours

In this section I will be using at the LIWC-WEAT method used previously, to show the multi discipline potential of this work. I have been able to identify and measure biases within word embeddings (also showing that these biases are also present in real world statistics). We now look towards applying these methods in the field of psychology to see if the empirical studies performed in that domain are consistent with observations found in word embeddings [3].

In this section we look at the biases of colour words. An extremely common example of colours being biased is that blue is seen as positive, while pink is seen as feminine. It is possible to use the method described to measure gender, and emotional biases within word embeddings (as seen in Sec.5). We will look at both the individual associations to a single concept (as shown in Eqn.3.13) and the difference between antithetical concepts (Eqn.3.14).

With this method we look at eleven colours that have been embedded in GloVe pre-trained word embeddings [38].These embeddings are generated from Wikipedia articles from 2014 and

[3]The present work is part of the forthcoming paper "Colour and Affect in Natural Language - Domicele Jonauskaite, Adam Sutton, Christine Mohr, Nello Cristianini" - and my contribution has been defining and using the method of analysis and providing results based on initial theory, research, and discussion by Domicele Jonauskaite and Christine Mohr

news articles from years previous. These embeddings are useful for being repeatable, and having a good range of high quality text for the algorithm to learn from. We will be look at both the emotional (positive, negative) biases of words and the gender (masculine, feminine) biases of words.

We will be studying 11 colour words which are the basic colour terms for British English [8]. These colour words are; “red”, “orange”, “yellow”, “green”, “blue”, “purple”, “pink”, “brown”, “grey”, “white”, and“ black”. Along with these colour words we chose eight words that we would consider to be anchor words. We chose four particular words as they can be defined by four of the word lists that are being used. These words are happy, sad, priest, and nun. We believe that happy and sad anchor the positive and negative scores for emotional bias, We believe the same to be true for priest and nun with masculine and feminine bias. We chose day, night, life, and death as extra sample antithetical words to be displayed.

5.3.1 Descriptive Analysis

To test if the words of interest are biased emotionally or by gender we must calculate the associations of a large sample of the words within the entire vocabulary. To this end we select the 100,000 most common words and calculate the four associations (positive, negative, he, she) using Eqn.3.13, and calculate their biases (emotion, and gender) using Eqn.3.14.

For each association, and bias we wish to test we will first compute their respective scores. We then look at colour words that are to be considered outliers. No assumptions can be made about the distributions of associations and biases. We will consider the associations being in the 95^{th} percentile or greater as highly associated to a concept. For gender and emotional biases being below the 5^{th} percentile or above the 95^{th} percentile are outlying observations of interest. Being below the 5^{th} percentile for gender would indicate a feminine bias, and above the 95^{th} percentile a masculine bias for gender scores. For emotional biases; being below the 5^{th} percentile for gender would indicate a negative bias, and above the 95^{th} percentile a positive bias.

We will first look at the colour words for the associations to single concepts. The concepts here are generated from the word lists; he, she, positive and negative. we will then look at the biases using (what I consider) two antithetical concepts; gender (he vs she) and emotion (positive vs

Table 5.2: Percentiles for associations to the concepts he, she, positive emotions, and negative emotions for all eleven colours of interest. A percentile of 97.2 for "red" in its association to the "he" concept means it is more similar to that concept than 97.2 of all other words in the vocabulary.

Colour	He	She	Positive	Negative
Red	97.2	94.7	96.7	91.6
Blue	96.4	96.7	96.8	78.5
Green	98.1	97.5	96.7	79.8
Yellow	95.3	94.6	93.2	87.4
Orange	87.2	93	89.1	60.4
Pink	89.2	98.7	95.5	79.1
Brown	99.1	98.7	96	89
White	98.7	98.5	97.8	94
Black	98.3	98.7	96.7	95.3
Purple	91	95.8	93.2	82.1
Grey	84.5	92.9	82.4	48.1

negative).

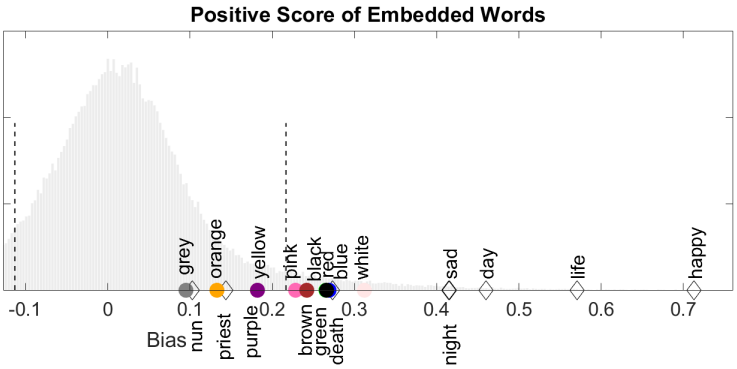
Figures.5.4(a) and 5.4(b) show the positive and negative associations of the colour words being tested, the anchor terms, along with a histogram showing the 100,000 most occurring words scores. Predictably, "happy" and "sad" are the most associated words in positive and negative respectively of the words chosen. The scores show a bell-like shape, but it is not a Gaussian distribution. Tab.5.2 shows the percentiles for all colours associations to concepts, including positive and negative associations.

In Fig.5.4(a) all colours are shown as more positive than the majority of the 100,000 words scored. "Grey", "orange", "purple", and "yellow" are all below the 95^{th} percentile. The remaining colours are shown to be in the 95th percentile or higher. All anchor words were also shown to be more positively associated than most words.

Fig.5.4(b) that the majority of colours are shown to be more negative associated than the common word, with "grey" being the only exception. "Black" is shown to be the only word past the 95^{th} percentile. The majority of anchor words are shown to be far past the 95^{th} percentile also, with "sad" predictably being the most negative associated word.

Figures.5.5(a) and 5.5(b) show the single gender associations of the colour words being tested, the anchor terms, and real scores from all 100,000 words. These figures show that all words of interest seem to be more gender biased than most words. Again both scores show a bell-like

(a) Positive associations of words with comparison distribution



(b) Negative associations of words with comparison distribution

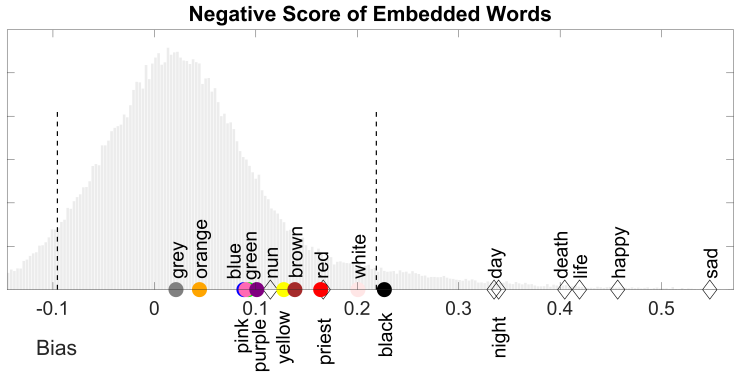


FIGURE 5.4. Histograms showing the positive and negative associations of embedded words, particularly looking at colour.

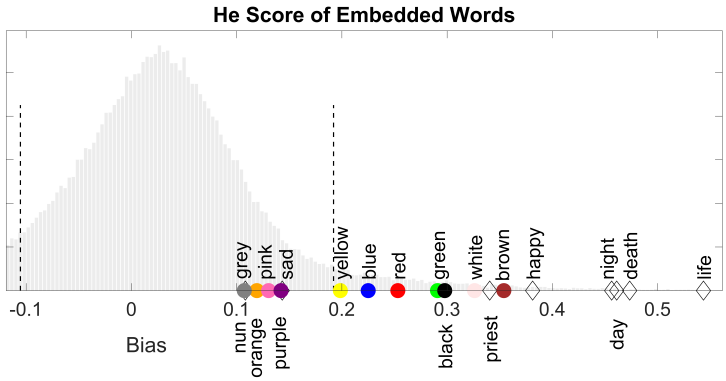
shape, but it is not a Gaussian distribution. “Life” is shown to be the most outstanding word in both scores. Tab.5.2 shows the percentiles for all colours associations to concepts, including feminine and masculine associations.

Fig.5.5(a) shows that the colours “yellow”, “blue”, “red”, “green”, “black”, “white”, and “brown” are shown to be in the 95^{th} percentile. “Grey”, “orange”, “pink”, and “purple” are below that percentile. The anchor term “priest” shows that it is a masculine word with its high percentile.

Fig.5.5(b) has the colours “grey”, “orange”, “yellow”, “red” below the 95^{th} percentile. With “purple”, “blue”, “green”, “white”, “black”, “brown”, “pink” above the 95^{th} percentile. “Pink” is shown to be the most feminine colour by this scoring. Many of the anchor terms are also showed to be highly associated to femininity.

The final comparisons in Figures.5.6(a) and 5.6(b) show the gender and emotional/valence

(a) Male association of words with comparison distribution



(b) Female association of words with comparison distribution

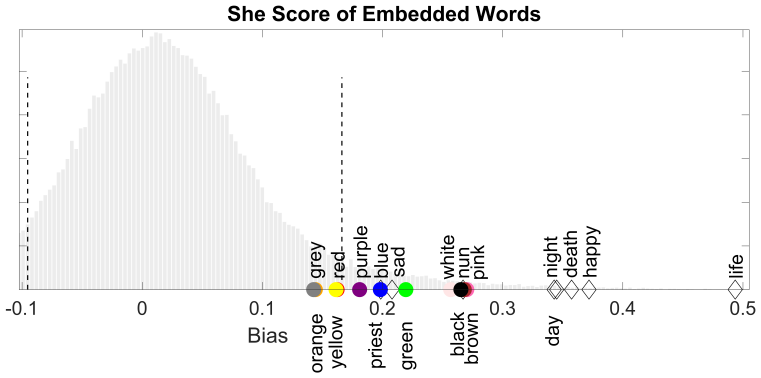


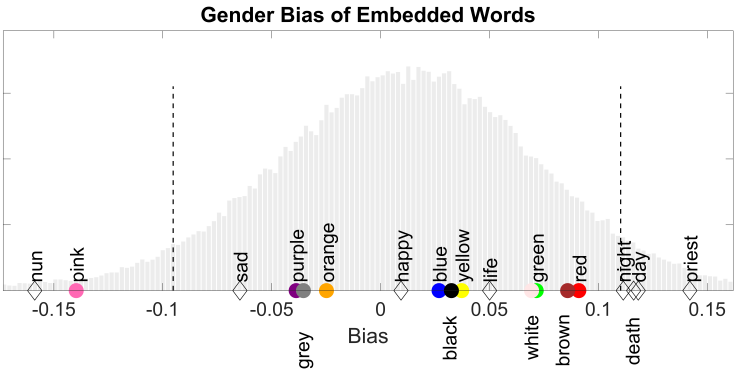
FIGURE 5.5. Histograms showing the male and female associations of embedded words, particularly looking at colour

biases to our target words of choice and our anchor words. These results show that the majority of colours fall between the 5^{th} and the 95^{th} percentiles for gender bias. However, more colours are shown to in the 95^{th} percentile for emotional biases. The percentiles for gender and valence colour biases are also shown in 5.3.

In Fig.5.6(a) words that are further to the right are to be considered more male biased, while further to the left of the distribution means they are female biased. Only one colour, pink is shown to be highly gender biased being below the 5^{th} percentile. “Nun” and “priest” are shown to be gender biased also in the correct positions for their anchor words. “Death”, “night’, and “day’ are also shown to be in the 95^{th} percentile.

Words to the left in Fig.5.6(b) are shown to be more negative, and words to the right are more positive. All colours are shown to be more positively biased than most words. Five colours are

(a) Gender bias of words with comparison distribution



(b) Emotional bias of words with comparison distribution

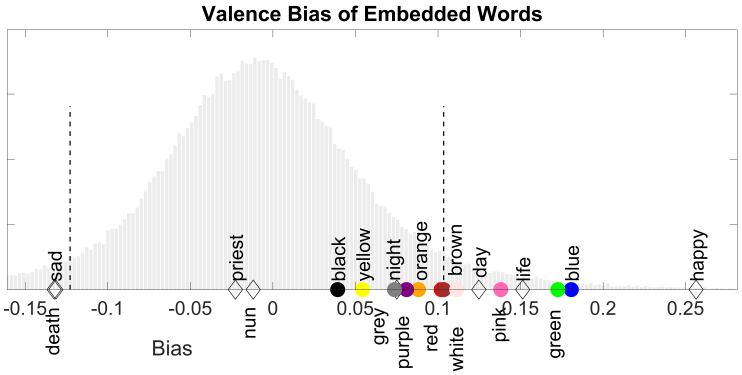


FIGURE 5.6. Histograms showing the gender and emotional biases of embedded words, particularly looking at colour

shown to be in the 95^{th} percentile; “pink”, “green”, and “blue”, “white” and “brown”. Of the anchor terms “day”, “life”, and “happy” are shown to be in the 95^{th} percentile showing positive emotional bias, while “death” and “sad” are below the 5^{th} percentile showing a negative emotional bias.

From this work I conclude that a large collections of colours are associated to emotional and gender concepts, but may be highly associated to the antithetical concepts. An example of this would be that while “green” is both seen as a positive word and a negative word. This results in “green” not having a clear distinction in bias for either positive or negative association. However this does indicate that the majority colours are heavily associated to emotion and gender in general. There are a few exceptions to this; grey was not seen as associated to either gender, “pink” was seen as biased towards females, and “pink”, “green” and “blue” are viewed more positively than negative.

Table 5.3: Percentiles for Gender and Emotional biases for all eleven colours of interest. A 90.8 percentile for the word "red" for gender bias shows that the word "red" is more male biased than 90.8 of all words in the vocabulary.

Colour	Gender Bias Percentile	Emotional Bias Percentile
Red	90.8	94.8
Blue	60	99
Green	84.2	99.8
Yellow	66.7	85
Orange	27.4	93
Pink	1.8	97.6
Brown	89.3	95
White	83.4	95.8
Black	63.4	79.2
Purple	20.6	91.8
Grey	22.2	90.3

5.3.2 Discussion

In this section I have applied the LIWC-WEAT method defined and used in Sec.5.1.1. We have applied this method into the discipline of psychology, looking particularly at the biases of colours in an natural English language corpus (sourced from Wikipedia and notable news sources). I again used LIWC word lists this time jointly in collaboration to define vectors that can then be used to compare words associations to that concept.

When originally looking just at associations between LIWC word lists and colour words we originally found that the majority of colours were associated with positive and negative emotions, as well as male and female terms. A low similarity between colours and comparison vectors in Figures.5.5(a), 5.5(b), 5.4(a), and 5.4(b) would indicate that colours are simply not associated to these concepts. However this was not found to be true for any colours tested.

Moving onto comparing how words score when subtracting antithetical concepts (he and she, positive and negative) we found that fewer colours were biased in these findings. "Pink", "green", "blue", "brown", and "white" were found to be much more positive than negative, with all of those colours being in the 95^{th} for positive emotional bias. "Pink" was also found to be more female biased than 95% of words.

This shows that most colours can be strongly associated with either gender, and positive or negative emotions. Colours appear and are used in a multitude of ways within natural language

and this work seems to indicate as such. When we use colours in sentences when not directly describing a colour itself, they can appear somewhat ubiquitously. While we would consider a phrase such as “black death” to be a negative phrase for the Bubonic plague, we would also consider “in the black” to be positive financial news. I feel that this information is encoded within the embeddings.

Despite this work, it is clear that some colours are biased more towards a single association. “Pink”, “green”, “blue”, “white” and “brown” are all much more associated with positive emotions than negative. While they were shown to be highly associated with positive emotions like many other colour words, they had a relatively lower association with negative emotions. For gender, the majority of colours also contain a high association to both male and female terms. This again may indicate that these colours are used commonly in the context of both genders. However, this results in only one colour being a remarkable standout; “pink” was the only colour to show a strong gender bias. “Pink” may be be the only word that can be considered as exceptionally more feminine biased than any other colours that we tested.

CHAPTER 6

HISTORICAL CORPORA

The main purpose of this chapter is to look at more general applications of the knowledge gained from the previous two chapters for the benefit of outside fields and disciplines. The content within this chapter will have inter-disciplinary ties with social and historical sciences, along with linguistics.

In this chapter I step back from just analysis of biases, and consider semantic change in more general terms, focusing on words that significantly changed their meaning over long periods of time [48]. An example is “fathom”, changing in meaning over several decades. Originally understood to be a unit of measurement, by the 1950s assumed a meaning closer to “understanding” or “conceiving”. I demonstrate this example in the work.

In this section first confirm that positive and negative emotional concepts are represented in historical word embeddings, using the same method in Sec.4.3 with LIWC lists. I then use the method I have defined in Chapter.5 to measure historical biases in occupations and in colours over time. I look at what occupations have shown to become significantly more negative, positive, male, or female from decade to decade. I do this using embeddings generated in decade long windows and measuring biases at each decade from 1800 to 1959. The data used in this experiment is based on a corpus of historical newspapers and has been used previously [49]. I then use the method proposed by Kulkarni to look at occupations that have changed in bias over the decades,

showing that a change in bias may not be accompanied with a more general change in the meaning of a word [1].

Finally I look into colours representation within historical word embeddings, continuing the work from the previous chapter. I look at gender, and emotional biases and then compare if changes in those biases correlated with changes in the semantic meaning of a word.

6.1 Concepts in Historical News

I have previously shown that LIWC word lists accurately define their intended concepts for word embeddings trained in a modern corpus (Chap. 4). I also showed that when using a mean vector of a given concept we can capture these concepts and measure them using a cosine similarity (Sec.4.3).

In this section I validate that these same LIWC word lists still accurately describe positive and negative emotional concepts when using word embeddings trained on historical corpora comprised of newspapers [49]. In this experiment I will test embeddings trained from corpora from two different decades; the 1800s and the 1950s. These samples are the earliest and latest decade intervals from the historical news corpus used in the rest of this chapter.

App.D shows the precision / recall, and the ROC curves for positive, negative, and random word lists. These results show that these concepts are learnable in historical corpora showing similar performances when compared to modern embedding performances. Random word lists again show insignificant performance in comparison to the well defined word lists. These results show that these word lists represent their respective concepts and can be used to measure biases in historical word embeddings.

6.2 Bias in Historical News

In this section I look to see if the biases that were found within modern corpora are present within historical corpora, and how these biases may have changed over time. To do this, I created

[1] The data used in this experiment is not available to the public and was supplied to the University of Bristol from Find my past

Table 6.1: A sample of occupational words that are shown to have the highest gender or emotional biases. I also show the words with the largest coefficients. Bold represents word scores that are the highest or lowest for that category. A negative score indicates that a word is more negative or female, and a positive indicates the inverse respectively. A coefficient is the linear component of a simple linear regression. A positive coefficient shows words becoming more positive or male biased, and a negative shows the inverse biases.

Word	1800 Emo	1950 Emo	Emo Coeff	1800 Gen	1950 Gen	Gen Coeff
Coachman	-0.1937	-0.0114	**0.0181**	0.0408	0.0311	0.0031
Doctor	0.0560	-0.1546	**-0.0173**	0.1833	0.0326	-0.0071
Shipbuilder	-0.0381	-0.0508	0.0020	-0.2282	0.0416	**0.0210**
Teacher	0.1900	0.0167	-0.0110	0.0211	-0.1685	**-0.0145**
Artist	**0.2204**	**0.1705**	-0.0033	0.0733	-0.0541	-0.0076
Driver	**-0.2366**	-0.2376	0.0014	-0.0287	0.1407	0.0114
Police	-0.1485	**-0.2991**	-0.0084	0.1050	**0.1682**	0.0071
Nurse	-0.0447	-0.0436	0.0026	-0.2943	**-0.2396**	0.0036
Minister	-0.0049	-0.0087	0.0011	**0.2186**	0.0833	0.0087
Milliner	-0.0073	0.1026	0.0065	**-0.3350**	-0.2003	-0.0041

16 distinct corpora using the "Find my past" corpus formed by British newspaper articles from 1800 to 1959 [49]. Each corpus contained articles written in different decades, between 1800 and 1959. I also used word2vec as the embedding algorithm (Φ), due to time constraints in learning the embeddings.

I tested occupations that can be described in a single word. This can be used to look at the similarity to positive or negative directions from $\Phi(L_{posemo})$ and $\Phi(L_{negemo})$, or the gender directions from $\Phi(L_{he})$ and $\Phi(L_{she})$. The word occupations have been sourced from the office of national statistics for 2017 [69]. However, these occupations may not be as relevant in the 1800s and early 1900s, so I also sourced an older list of occupations to also look at the change of those words over time [66].

To visualise the change in associations to these concepts over time I subtract the score of one association from its opposing association to visualise it as a time series as explained in Eqn.3.14. For example, the gender association for a given word from a given decade will be the association to $\Phi(L_{he})$, minus the $\Phi(L_{she})$ association. If the score is above 0 then a word will be "more male" and below will represent a "more female" word.

Due to the focus of looking at the change in association of words over time I will look at the most extreme examples of words changing in association over time. This has been shown in the

Table 6.2: Words that regression coefficients that are found to be statistically significant such that its coefficient is at least 3 standard deviations away from the mean.

Word Trend	Words
Male	bricklayer, plumber, fruiterer, moulder, shipwright, builder, joiner, butcher, saddler, wheelwright, confectioner, carpenter, merchant, innkeeper, shipbuilder, plasterer, upholsterer, fishmonger, miner, blacksmith, grocer, schoolmaster, baker, fisher, ironmonger, glover, farmer, porter, driver, dealer, solicitor, druggist, shopkeeper, manufacturer, painter, draper, surgeon, greengrocer, printer, tailor, seaman, plumber, plasterer, conservationist, carpenter, auctioneer, coach, driver, architect, solicitor, manager, cook, accountant, engineer, waiter, mason, surveyor
Female	secretary, artist, teacher
Positive	supervisor, buyer, coachman, tailor, dressmaker, milliner, pedlar
Negative	director, analyst, surveyor, solicitor, instructor, barrister, clerk, engineer, manager, teacher, doctor, conservationist, shopkeeper, labourer, porter, solicitor, clerk, teacher, clergyman, bailiff, painter, mechanic, miner, proprietor

Table 6.1. Coefficients are from a simple linear regression and looking at the regression coefficient to show the impact time has on the association of the word. Table 6.1 also shows the extremes of the emotional and gender associations of words in the earliest and latest decade embedding.

"Artist" and "driver" was shown to be the most positive and negative associated occupations respectively in the 1800s. Artist remains the most positive association at the latest embedding, while "police" is shown to be the most negative. The earliest embedding also shows "minister" to be the most male occupation, and "milliner" to be the most female. At the final chronological embedding however it is shown that "police" is the most male and "nurse" the most female.

The word with the highest positive change is the word "coachman" which was a word from the list of older occupations, while doctor has the largest negative bias change over time. "Shipbuilder" was shown to become more of a male associated word over time, with Teacher becoming a more female word over time.

Of 169 occupations I found that that 79 of them are becoming more positive and 90 more negative over time. Of the same 169 occupations 139 are becoming more male associated, while 30 more female. However, these regression coefficients alone do not show bias changes that are statistically significant. With Table 6.2, I tried to find what words change in bias is statistically significant. I do this by permuting positive / negative or male / female scores per decade and calculating their regression coefficients $1,000$ times. Sampling the mean and standard deviation of this, I found what words change in association is significant with a p-value of < 0.001.

(a) Fathom's nearest semantic neighbours in 1800

(b) Fathom's nearest semantic neighbours in 1910

(c) Fathom's nearest semantic neighbours in 1950

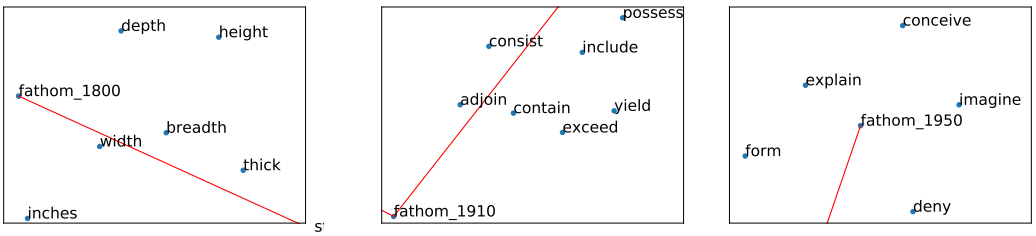


FIGURE 6.1. Nearest Neighbours of the word "fathom" over a 150 year period, showing the change in semantic meaning over time. This figure is created using pairwise distances of the nearest neighbouring vectors to the word of interest.

6.3 Semantic Drifts in Historical News

In word embeddings, nearest neighbours can be seen as words that carry the same or similar meaning to a given word. For example, the nearest neighbours to "boy" could be words such as "man", "child", or "infant". These words are all nouns and semantically have a intuitive relationship with "boy", with "man" being of the same gender while "child" and "infant" refer to a younger age.

Using this logic a word can reflect its meaning by the "neighbours that it keeps". By looking at the nearest neighbours of words from multiple word embeddings from different decades I show that is possible to demonstrate the semantic changes in words from the 1800's to the 1950's.

Given two embeddings at time t (1800-1809) and t' (decade intervals from 1810-1959) of the same set of words, and a word w, I want to compare the position of this word in the two embeddings. I aim to compute its expected position in t based on its position in t', and then compare this with its actual position in t.

I find the k nearest neighbour words to the target word and use them to estimate a linear transformation from $\Phi_{t'}(w)$ to $\Phi_t(w)$. The same transformation can then be used to compute the expected position of word w from time t' to time t.

Using the equation as specified in Eq.3.23 it is possible to align a word from one embedding to another depending on its k nearest neighbours. Using this newly aligned word it is now possible to look at the rest of the words within its potential new neighbourhood to see the linguistic change

in meaning of a word.

For this work I use the same decade by decade embeddings used previously to look at the change of associations between occupations and concepts. I will first confirm the method defined in [48] by looking at words known to have changed over the time period 1800 to 1950, and then apply it to the words that I have demonstrated to have significant changes in biases.

After aligning the target word from each decade to the embedding at time t (which for the purpose of this experiment I have decided to be $\Phi(C_{1800})$, I will then compute the pairwise distance between the word through each decade, and its k nearest neighbours (in this case 10) in the embedding at time t to visualise what the most similar words the aligned word in the embedding from 1800-1809.

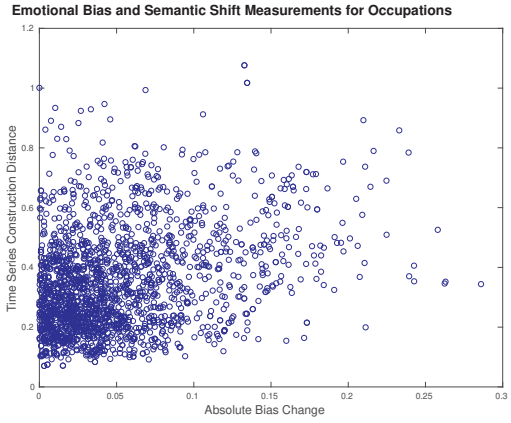
To visualise the change in meaning of a word over time the words “fathom” will be tested to see its meaning changes decade by decade. “Fathom” is known to have changed in meaning over time. Originally being used as a unit of measurement, but eventually meaning understanding or conceiving a concept.

Fig.6.1 shows fathom and its change in meaning from decade to decade. From 1800-1909 it shows that the word doesn’t change too much in meaning with the nearest words being “depth”, “width”, and “inches”. While the nearest words at the end of from the later embeddings are “conceive”, “explain”, and “find”. This shows that the linguistic change of words is visible within word embeddings.

6.3.1 Semantic Drifts in Biased Occupations

In this experiment I visualised words that I have demonstrated to change the most over time in their sentiment and gender biases. Coachman was found to be the word that becomes the most positive over the time period of interest. In Fig.6.3(c) “coachman” does not show a significant shift linguistically. Over 159 years of embeddings the nearest neighbours with k set to 10 shows that only 31 different words were the nearest neighbours year on year. “Doctor” as shown in Fig.6.3(a) does not have a significant linguistic change, although it is viewed more negatively as time passed. “Teacher” was shown to become more female over time in previous experiments however Fig.6.3(d) does not show a significant linguistic change. Finally, “shipbuilder” (as seen

(a) Correlation of Relative Emotional Bias Changes and Times Series Constructions



(b) Correlation of Relative Gender Bias Changes and Times Series Constructions

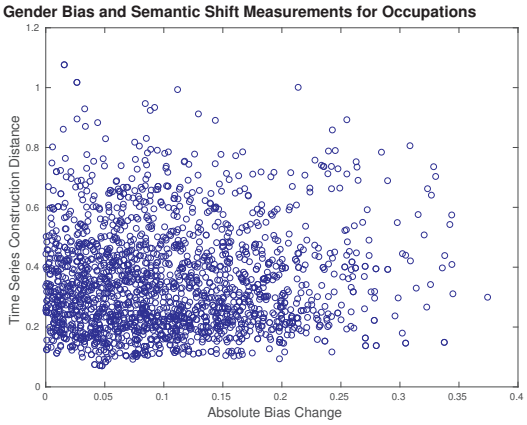


FIGURE 6.2. Correlations between occupational words biases and their change in semantic meaning. A single point represents a words relative bias difference from time t' to time t and its times series construction for the same time periods.

in Fig.6.3(b)) shows to become more male over time, however there is little change when using k-NN's to visualise this.

I then decided to look at all occupational biases and look at the correlation between them and the values from the time series construction described in 3.8.2. I first took all 169 occupations, and calculated their emotional and gender biases for each decade. This would result in 16 decades of bias scores for all 169 words. Because I am looking for any changes in bias and not for a particular change, I will subtract the bias at time t' from the bias at time t and then take the absolute value from that. This will provide me with 15 absolute differences in bias starting from the embedding at time t.

I then calculate all of the possible distances from the time series construction method as explain in Eqn.3.24. Of 2535 possible data points (169 words, with 15 decades of distances from the original embedded word per word), 2008 provided times series calculations. This is due to some words not existing in the vocabulary at time t and thus the word couldn't be aligned to for any decade, or a word not existing in time t' and thus that word could not be aligned for that specific decade. With these pairs of results I then graph and compute the Pearson correlation to see if there is any significant correlation.

Table 6.3: For the words “doctor” and “coachman”, these are the counts of the k-nearest neighbours that show changes in emotional bias. k in this test is set to 10, and we look at any neighbouring words that appear in at least one decade. Neutral words for these biases are those with a coefficient of <0.001.

Nearest to Word	Trending Positive	Trending Negative	Neutral
Coachman	20	7	5
Doctor	7	15	9

Table 6.4: For the words “ship-builder” and “teacher”, these are the counts of the k-nearest neighbours that show changes in gender bias. k in this test is set to 10, and we look at any neighbouring words that appear in at least one decade. Neutral words for these biases are those with a coefficient <0.001

Nearest to Word	Trending Male	Trending Female	Neutral
Shipbuilder	22	14	14
Teacher	12	44	2

In this experiment I take the absolute value of biases, and ignore the sign as any change in bias is of interest. Fig.6.2(a) and Fig.6.2(a) show the scatter plots for both biases of interest and their time series constructions. For gender biases and semantic change from time t of occupations, I find a Pearson correlation of 0.0393, showing no correlation between the change in gender bias and the change in the meaning of a word over time. When comparing emotional biases to the same time series distances of occupations, the Pearson correlation is 0.3253.

In conclusion, occupations that were shown to change the most in sentiment and gender over time showed little change when looking at their semantic change over time. There is also no significant correlation between a change in bias and a significant semantic change in the meaning of a word.

Due to this, I experimented on the nearest neighbours of these words to see how their associations to these concepts change over time. I will use the previous method as before of using a time series of bias with the scoring associations and computing a gradient to see if these words change in associations over time. Words that have a regression coefficient between -0.001 and 0.001 will be considered as neutral words regarding if they are becoming more gender or emotionally biased.

Table 6.3 shows the counts of positive and negative trending words. Coachman as a word was found to become more positive over time, while doctor was found to become more negative

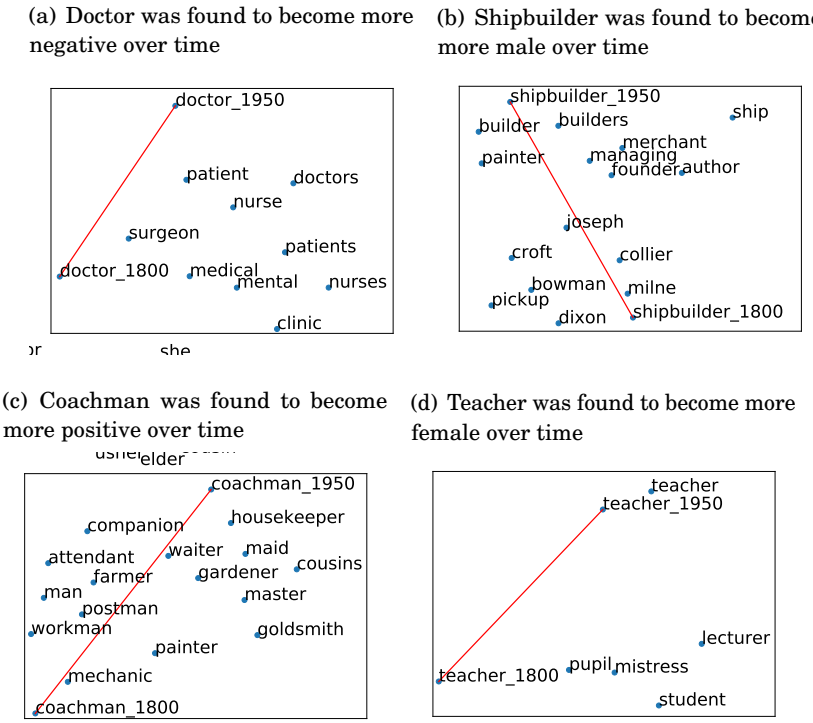


FIGURE 6.3. Detecting semantic change in words that have shown large changes in bias over time

over time. The nearest neighbours to these words also showed similar results generally following the same association to their respective word. Table 6.4 again shows similar results with the closest words to shipbuilder and the closest words to teacher becoming more male and female respectively. Which is the same change in bias as the target words being tested.

This shows a limitation of the k-nearest neighbours method to show the change in words from decade to decade. While the algorithm is intended to work on statistically significant changes it is limited by the words nearest to it, where most of those words are also experiencing a similar change in its association to these concepts.

6.4 Historical Representation of Colours

In this section I am going to look at the historical representation of colours. I will look at both their semantic change over time to see if there is a significant change in the meaning of any colours, and I will look at changes in bias and if any bias changes in emotion or gender are

Table 6.5: Historical gender and emotional biases of colours along with their coefficient's. I also show the words with the largest coefficients. Bold represents word scores that are the highest or lowest for that category. A negative score indicates that a word is more negative or female, and a positive indicates the inverse respectively.

Word	1800 Emo	1950 Emo	Emo Coeff	1800 Gen	1950 Gen	Gen Coeff
Red	0.0369	**0.0887**	0.0051	-0.0482	-0.0682	-0.0046
Blue	-0.0133	0.0334	0.0057	-0.0819	**0.0119**	0.0041
Green	**0.1066**	0.0595	**-0.0025**	-0.0361	-0.1385	-0.0035
Yellow	0.0017	0.0506	0.0041	**0.0605**	-0.0756	-0.0135
Orange	**-0.1112**	0.0151	**0.0065**	-0.0844	-0.1367	-0.0003
Pink	0.0663	0.0774	0.0038	-0.1233	-0.1392	-0.0033
Brown	-0.0059	-0.0035	0.0044	**-0.1942**	-0.0099	**0.0102**
White	0.0363	**-0.0082**	0.0038	-0.0311	**-0.1655**	**-0.0134**
Black	-0.0362	-0.0026	0.0020	-0.1080	-0.0635	0.0004
Purple	-0.0372	0.0341	0.0053	-0.1456	-0.1451	-0.0012
Grey	-0.0290	0.0209	0.0048	-0.0449	-0.0633	-0.0018

significant over the time period listed. Modern day biases for colours have been discussed in Sec.5.3, this work focuses on historical biases and also looks if those changes in biases may also lead to a change in meaning.

I will repeat the experiment looking at historical gender and emotional biases using the same method as Sec.6.2. I will use the same colours used in Sec.5.3 for this experiment.

Tab. 6.5 shows the gender and emotional biases for all colours of interest at the earliest and the latest decade I have embedded. I also display the regression coefficients for both gender and emotional biases to indicate what colours are appearing to change in their gender and emotional bias.

Of 11 colours tested, nine were found to be insignificant when looking at their trend for gender bias. The two colours found to be changing significantly in bias are "green" and "grey". "Green" was found to become significantly more male over time, while "grey" was found to become significantly more female over time.

For the emotional biases of colours, three were found to become significantly more positive as time passed. "Brown", "Yellow", and "White" were found to become more significantly to become more positive with no colours becoming significantly more negative.

I finally check to see see if there is any correlation between changes in bias and significant changes in meaning of words. I will again compare the relative bias changes to the times series

construction for all 11 colours across 15 time periods. This method has been done previously with occupations as shown in Sec.6.3.1.

Both emotional and gender bias changes show no significant correlation when compared to times series calculations of colour words. Taking the absolute value of emotional biases of colours and looking at its correlation to their semantic change over time gives a correlation of 0.1989. The correlation between gender biases and the semantic change of a word is shown to be 0.1451. This shows to be consistent with previous work looking for a correlation between a change in bias indicating a change in meaning.

6.5 Discussion

In this section I have validated and demonstrated the method of measuring word bias using word lists provided by LIWC [74]. I used historical corpora to look at biases over a 160-year period creating embeddings at ten-year intervals. I verified that LIWC word lists are again encapsulating the concepts I aimed to measure, and then measured the change of various biases over time. Looking at those associations from decade to decade I find that most occupations in the study have become more male over time. Half of the occupations I have tested are found to become more positive, while half more negative over the decades.

Using the method proposed in [48] I found that given a words change in emotional or gender biases does not indicate a statistically significant shift in that words meaning. I also find that neighbours of words that change in bias are more likely to also experience the same change in that bias. I find there is no statistical correlation between words changing over time and the change in their biases over the same time periods.

I finally look at the colour biases over the decades of interest. I show that “green” was becoming more male, and “grey” is becoming more female over time. I also found “yellow”, “white”, and “brown” to become significantly more positive over time. I again look for a statistical correlation in these changes in biases and changes in the meaning of these colours, only to find no such evidence.

Biases have been found to change within embeddings over time, even though current methods

to look at change in the meaning of words over time do not manage to capture what could be a significant issue for these embeddings. This method has shown an ability to look at important biases within an embedding at a greater granularity. Other biases not related to gender or race as presented here may also be found using this methodology.

Biases changing are not easily identifiable with broad stroke methods. They may be subtle changes that may not present with a change in the meaning of a word. Over 150 years a teacher has shown to always be a profession of someone who teaches others. However with further inspection of these embeddings using my method, we can see that the mental image of a teacher has changed from a man to a woman. These biases present in the embeddings are also indicative of them being present in the historical corpora, and by extension those who had contributed this corpora.

Measuring any concept with an accurate word list should provide reliable associations between words and concepts and can be used to look at additional biases. Word lists however must represent the concepts accurately in the word embedding, historical corpora must be accounted for when using word lists and the same must be said for corpora generated in a different context (ranging from forums, twitter, and other casual social media). My method I proposed in Chapter. 4 would help resolve this issue. Furthermore, this method may also be able to generate or extend lists for other concepts, even in historical corpora.

CHAPTER 7

CONCLUSION

As word embeddings become increasingly powerful and necessary for state of the art performance, these systems have become more difficult for humans to understand the reasoning for decisions that they make. This has lead to a large number of black box solutions that humans may fail to trust entirely. In this work I have shown multiple methods of using concepts to help humans understand multiple different word embedding algorithms, and give transparency when using these black box intelligent systems.

In this thesis I have proposed methods to aid in better understanding what constitutes a "better" word embedding. This method makes use of three key inputs; a word embedding algorithm, a corpus for the algorithm to learn from, and a word list such that all members of that list represent a concept. If we choose two of those three inputs to be "ground truths" that we know to perform as intended, then we can validate the third input to see if it performs as desired. We can also use this method to compare the relative performance of one of the inputs (i.e. we can compare word embedding algorithms performances provided we train on the same corpus and test with the same word lists).

When applying this method, we first confirm that it works. We show that word lists that represent a concept are understood by three different context free word embeddings, provided by off the shelf pre-trained embeddings. These word embeddings confirmed that LIWC word lists

were significantly capturing the concept that they aim to define. This experiment also showed that word embeddings are capturing information, when compared to random word embeddings.

I then applied my method to compare the performance of three popular "context-free" word embedding algorithms. I used the same source corpus to train all three embedding algorithms, and tested the resulting embeddings on the same word lists. I found that "fastText" was the best performing word embedding in this task, with "GloVe" and "word2vec" coming in second and third respectively. This work also showed the LIWC list of "relative" to generally be the highest performing concept.

As an extension to this work, I looked to apply the same experiment to "deep contextual embeddings", namely BERT to see if linear classifiers could learn concepts from these embedding representations. This proved to be much less effective, than when using context free embeddings. However using a deep sequence classifier, which is defined as part of BERT showed to have strong performance. This showed that deep contextual embeddings could be tested in this way and how well BERT represented LIWC concepts.

Using this method I looked at the performance of two different corpora and word lists using the BERT algorithm. I compared how two different sets of word lists defined as concepts (LIWC and ICD-10) perform on two corpora from different domains. I show that LIWC concepts are better represented when the embedding algorithm learns from a general corpus, such as news, books, and Wikipedia. The inverse is true with ICD-10 diagnoses codes, which perform significantly better using a corpus that focuses on the medical domain.

All of this work shows that concepts known in the real world, are represented in corpora and by extension can be learned by language representation models. We show the significance of a collection of words representing a concept, and can be used to test any of three key inputs. We applied the method to compare different embedding algorithms and corpora used for training, but it is also possible to test how well word lists capture a concept. The linear classifiers performance cannot be compared to the performance of the deep sequential classifiers, due to the difference in computational and expressive power.

This method should give some confidence to those who wish to use word embeddings as part of an intelligent pipeline for a given task. If you have concepts that are key to a field, that can be

encapsulated by words (or sequences) then you can use this method to choose embeddings that provide a strong human understanding of why they are the best to all other options available. The cheaper computation time can also be seen as a benefit, as generally speaking tasks that use embeddings in a pipeline can take up to days or weeks to train.

The final contribution of the first chapter also demonstrates the same concepts being understood when using linear algebra. A benefit of this work is linear algebra is used in many methods that assess performance. Many tests are designed to examine linear substructures are present with embeddings (such as a linear relationship between cities and capitals). This work also only compromises of linear algebra and can indicate weather a concept word list correctly represents its intended concept and be used to measure words association to it.

Future work utilizing concepts can go in multiple different directions. As the method has three inputs, all three could be of interest in future work. Word Embedding algorithms can be tested over multiple hyper-parameters, and different numbers of dimensions. New embedding algorithms can also be tested and adapted using the method proposed here. Different corpora have been tested in this work and showed domain specific performances. Additional domains can be tested, as long as there are suitable word lists to test. Word lists can also be the focus of the method. All word lists used in this work were sourced from expert knowledge in the domains of psychology and medicine. Some corpora may not be well represented by such word lists in certain tasks. This method can be used with validation techniques to show word lists where some members may not be suited to the word list. This method could also be used to provide confidence in creating or augmenting word lists for concepts that currently are unavailable. Confidence in the method alone may not be enough, but experts could validate these lists.

Chapter two uses the knowledge of concepts being present and understood within word embeddings. We use concepts to generate a mean vector, a single vector that represents the general meaning of the concept. Using LIWC lists as the provider of these concepts means that we can in good faith believe that these word lists represent the concepts they claim to. LIWC lists have been established and maintained for a very long time and has been used in clinical psychology, and trust can be placed in them to encapsulate a concept.

I first proposed the LIWC-WEAT (LIWC-Word Embedding Association Test), to identify,

measure and compare biases within word embeddings. I applied this to common pre-trained GloVe embeddings and found racial and gender biases present within the word embeddings. I also look at the gender biases for occupations and compare to real world statistics. There proved to be a strong correlation between the biases of an occupation and the ratio of men to women within those roles.

Using a projection algorithm I then removed these biases from the embeddings and repeated these experiments to see the results. I found that biases were reduced, however this came at the cost of the word embeddings losing some representation of real world statistics. The occupational statistics were no longer correlated with the occupational gender biases. This may be a benefit for the fairness of the AI system, although it comes with a reduction in representing current occupational statistics.

In another application of my method I look to another discipline; psychology. Colour biases have long be seen in natural language and society. Trying to identify and interpret them using traditional methods can be quite a challenge of manpower and resources for psychologists. Word embeddings can help alleviate some of these issues as a suitable corpus and word embedding algorithm is required to perform experiments on certain biases.

In this work I look at gender and emotional biases for colours of interest. We look at the individual associations of words to the concept of male and female terms, and positive and negative terms. We show that a lot of colours are highly associated to all these concepts. When looking at the difference between the opposing concepts we see that pink, blue, and green are shown to be strongly positively biased and pink is the only colour that demonstrates a strong female gender bias.

This application of my proposed method shows interdisciplinary usage and provides a generally easier method compared to traditional methods of interviewing people to try to discover or learn about such biases. I don't believe that this method removes the need for traditional psychology methods of testing when looking for such biases, but it may provide evidence for researchers to commit resources towards research which may otherwise be seen as a risk.

In the future all AI's may benefit from understanding biases present, and this includes systems using the word embedding algorithm in their pipeline. The method of measurement can

be used to measure biases of interest present in corpora. Future embedding algorithms (including BERT) currently do not have a suitable method of measuring embeddings similarity to words and other concepts. A method for these algorithms would be useful to help with understanding biases in future systems. Other areas of academia may also benefit from understanding biases present within systems.

In my final chapter I look at another application of using concepts, this time focusing on semantic and bias change in words over long periods of time. I first look at biases in historical news, focusing on occupations sourced from modern and traditional occupations for the time period. I look at the occupations that change in emotional bias, and gender bias.

I find that from the 169 occupations tested, a large number of occupations are becoming significantly more male and only three are being increasingly biased towards female terms. Seven of the 169 occupations are becoming more significantly more positive, and 24 negative. I calculate this by generating temporal corpora for every 10 years from 1800 to 1959 generating 16 points of a data.

I then look to compare how changes in biases of a word leads to a change in the semantic meaning of a word. I first validate the embedding align method first proposed by Kulkarni by applying it to the word “fathom”, a word known for its semantic change over our period of time. We show that fathom clearly changes in meaning over time, and visualise it with its nearest neighbours.

I then look at the occupations we have discovered to significantly change in biases over time to see if the meaning of those words has also experienced a change. I find that generally speaking a change in bias doesn’t not necessarily make for a significant change in semantic meaning for a word. However the majority of the nearest neighbours to occupational words also appear to experience a similar change in bias over time, perhaps explaining why the change in meaning of a word may not be visible by the embedding alignment or time series method.

There is a open field of future work possible when applying this work to different disciplines and industries. Historical corpora of other languages can also be looked at to see bias and semantic changes in words. It may also be of interest to see words other biases change over time. Colour biases have been observed here from the perspective of psychology and history, however

this kind of work may have commercial applications beyond colour. Advertisers may benefit to know which colours are viewed more favourably in a society, and this could be extended past colours to other things.

Concepts defined extensionally are human understandable. LIWC word lists that were used in this work as concepts are understandable for humans with no explicit teaching of what they should be representing. ICD-10 diagnoses chapters are domain specific word lists that we have also used for concepts. While non-experts may struggle in accurately understanding what concepts members of those chapters are, a domain expert would have little issue. However these same humans cannot possibly understand the innermost workings of word embeddings. Using concepts to help peel away the black box from word embeddings help society with a tool for transparency and trust in AI systems which are becoming increasingly important in their daily life.

They provide a practical use for engineers hoping to make intelligent systems that take advantage of language models. These models generally do not attempt to or poorly explain the reasoning behind their decisions. However using concepts to show the performance of a key input to that algorithm can help make informed decisions, and better improve those systems.

For society at large word embeddings have multiple applications to psychologists, historians, and other fields within the humanities and social sciences. The work here on modern biases and historical studies is only a small sample of the potential applications of this work. Traditional studies and experiments within this fields may be time consuming and a large commitment of resources. Using word embeddings and concepts, along with linear algebra and statistical testing may alleviate some of these issues.

Transparency is a key issue facing word embeddings, and by extension all AI systems that employ machine learning algorithms. Small improvements for AIs on performance metrics are celebrated as significant results, with little focus on explaining the significance of these differences. I have defined a statistically rigorous method that can give confidence and transparency that certain word embedding algorithms performances are reasonable and understandable to humans.

Another negative from the black box of word embeddings is unwanted biases being learned from corpora that are also biased. Transparency is again required here for society to have trust

in these systems, and to enable actions to be taken to mitigate these biases. There may be some applications to word embeddings containing biases, such as providing the humanities or other disciplines an efficient and large scale tool to understand biases. It may be necessary to trade these biases for a reduction on performance, but it may be required to achieve fairness in these systems. Regardless of removing such biases, awareness of their existence should be considered to truly be right for the right reasons.

APPENDIX A - FULL ICD-10 TABLE

Table A.1: Sample diagnosis codes from ICD-10 used in experiments.

Full Name	Sample Words	List Abbreviation
Blood Diseases And Immune Disorders	Hypersplenism, Eosinophilia	BDID
Circulatory Diseases	Pelvic varices, Cardiomegaly	CD
Congenital Malformations Deformations and Chromosomal Abnormalities	Microcephaly, Anencephaly	CMDCA
Digestive Diseases	Anodontia, Mottled teeth	DD
Ear and Mastoid Process Disease	Petrositis, Presbycusis	EMPD
Endocrine Eutritional Metabolic Diseases	Myxoedema coma, Beriberi	EEMD
External Causes Morbidity Morality	By parent, Macrolides	ECMM

Continued on next page

Table A.1 – continued from previous page

Full Name	Sample Words	List Abbreviation
Eye and Adnexa Diseases	Chalazion, Scleritis	EAD
Genitourinary System Disease	Hydroureter, Pyonephrosis	GSD
Health Status and Contact	Isolation, Sterilization	HSCI
Infectious and Parasitic	Typhoid fever, Botulism	IP
Injury Poisoning and External	Concussion, Flail chest	IPE
Mental and Behavioral Disorders	Delirium, Hypomania	MBDD
Musculoskeletal System And Connective Tissue Disease	Reiter disease, Flail joint	MSCTD
Pregnancy Childbirth Puerperium	Missed abortion, Twin pregnancy	PCP
Respiratory Diseases	Stannosis, Farmer Lung	RD
Skin And Subcutaneous Tissue Disease	Pyoderma, Other sunburn	SSTD
Symptoms Signs And Abnormal Clinical Labratory Findings	Cough, Heartburn	SSACLF
Perinatal Period Conditions	Neonatal coma, Extreme immaturity	PPC
Neoplasms	Multiple myeloma, Myeloid sarcoma	NP
Nervous System Diseases	Chronic meningitis, Myoclonus	NSD

APPENDIX B

APPENDIX B - FULL ICD-10 BERT PERFORMANCE

Table B.1: Performance of deep sequence classifiers using ICD-10 dianogses chapters on BERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are the word lists not being trained in the binary classifier.

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{BDID}}$	161	0.832	0.871	0.198	0.772	0.924
$\mathbf{L_{SSTD}}$	332	0.855	0.863	0.153	0.863	0.918
$\mathbf{L_{CD}}$	344	0.878	0.859	0.103	0.890	0.953
$\mathbf{L_{CMDCA}}$	614	0.884	0.906	0.138	0.870	0.943
$\mathbf{L_{DD}}$	393	0.855	0.932	0.158	0.848	0.959
$\mathbf{L_{EMPD}}$	100	0.780	0.709	0.133	0.867	0.875
$\mathbf{L_{EEMD}}$	317	0.886	0.888	0.116	0.899	0.946
$\mathbf{L_{ECMM}}$	3319	0.987	0.981	0.008	0.992	0.999
$\mathbf{L_{EAD}}$	223	0.865	0.870	0.139	0.870	0.930

Continued on next page

Table B.1 – continued from previous page

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{GSD}}$	392	0.855	0.896	0.186	0.824	0.934
$\mathbf{L_{HSCI}}$	626	0.952	0.939	0.035	0.964	0.985
$\mathbf{L_{IPE}}$	1270	0.937	0.931	0.057	0.948	0.984
$\mathbf{L_{MBDD}}$	395	0.957	0.985	0.070	0.932	0.992
$\mathbf{L_{MSCTD}}$	472	0.843	0.864	0.178	0.829	0.914
$\mathbf{L_{PCP}}$	399	0.927	0.951	0.097	0.911	0.971
$\mathbf{L_{SSACLF}}$	339	0.897	0.885	0.089	0.920	0.951
$\mathbf{L_{IP}}$	697	0.901	0.930	0.125	0.869	0.963
$\mathbf{L_{RD}}$	224	0.817	0.871	0.241	0.795	0.896
$\mathbf{L_{PPC}}$	333	0.865	0.865	0.135	0.880	0.956
$\mathbf{L_{NP}}$	702	0.972	0.961	0.017	0.983	0.993
$\mathbf{L_{NSD}}$	276	0.841	0.879	0.199	0.820	0.939

APPENDIX C

APPENDIX C - FULL ICD-10 BIOBERT PERFORMANCE

Table C.1: Performance of deep sequence classifiers using ICD-10 dianogses chapters on BioBERT embeddings to identify members of its corresponding set. In this work the negative samples used in training and testing are the word lists not being trained in the binary classifier.

L	Size	Accuracy	Recall	FPR	Prec	AUC
L_{BDID}	161	0.925	0.974	0.119	0.882	0.993
L_{SSTD}	332	0.967	0.960	0.025	0.977	0.988
L_{CD}	344	0.968	0.976	0.960	0.040	0.990
L_{CMDCA}	614	0.963	0.968	0.043	0.958	0.993
L_{DD}	393	0.944	0.974	0.084	0.916	0.992
L_{EMPD}	100	0.950	0.982	0.089	0.931	0.987
L_{EEMD}	317	0.959	0.988	0.075	0.939	0.990
L_{ECMM}	3319	0.995	0.996	0.007	0.993	0.999
L_{EAD}	223	0.951	0.957	0.056	0.948	0.993

Continued on next page

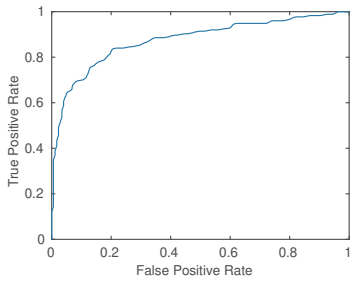
Table C.1 – continued from previous page

L	Size	Accuracy	Recall	FPR	Prec	AUC
$\mathbf{L_{GSD}}$	392	0.969	0.964	0.025	0.974	0.993
$\mathbf{L_{HSCI}}$	626	0.978	0.981	0.025	0.975	0.997
$\mathbf{L_{IPE}}$	1270	0.978	0.985	0.030	0.973	0.998
$\mathbf{L_{MBDD}}$	395	0.967	0.990	0.055	0.946	0.993
$\mathbf{L_{MSCTD}}$	472	0.932	0.979	0.114	0.895	0.985
$\mathbf{L_{PCP}}$	399	0.987	1.000	0.026	0.976	0.995
$\mathbf{L_{SSACLF}}$	339	0.935	0.923	0.051	0.955	0.986
$\mathbf{L_{IP}}$	697	0.973	0.979	0.033	0.964	0.990
$\mathbf{L_{RD}}$	224	0.920	0.940	0.102	0.908	0.989
$\mathbf{L_{PPC}}$	333	0.967	0.961	0.026	0.977	0.990
$\mathbf{L_{NP}}$	702	0.997	0.997	0.003	0.997	0.999
$\mathbf{L_{NSD}}$	276	0.928	0.993	0.140	0.880	0.991

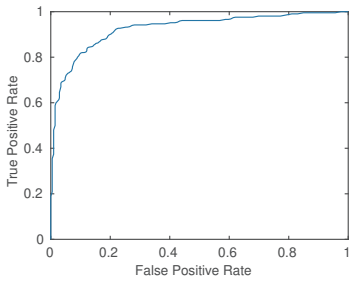
APPENDIX D

APPENDIX D - PRECISION RECALL AND ROC CURVES FOR HISTORICAL CORPORA TRAINED IN 1800 AND 1950

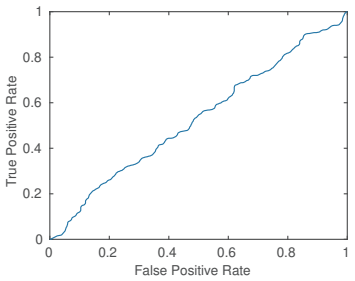
(a) ROC of $\Phi(\mathbf{L}_{posemo})$ vectors with an AUC of 0.9



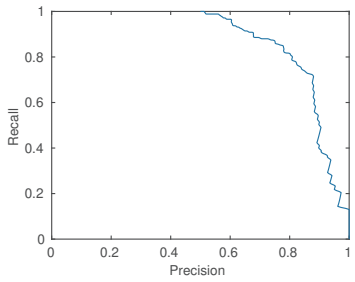
(b) ROC of $\Phi(\mathbf{L}_{negemo})$ vectors with an AUC of 0.92



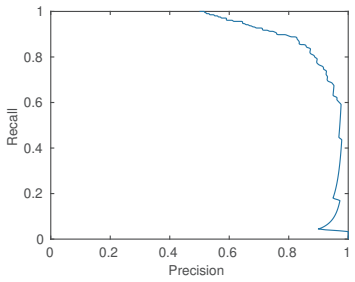
(c) ROC of $\Phi(\mathbf{L}_{random})$ vectors with an AUC of 0.5



(d) Precision Recall of $\Phi(\mathbf{L}_{posemo})$



(e) Precision Recall of $\Phi(\mathbf{L}_{negemo})$



(f) Precision Recall of $\Phi(\mathbf{L}_{random})$

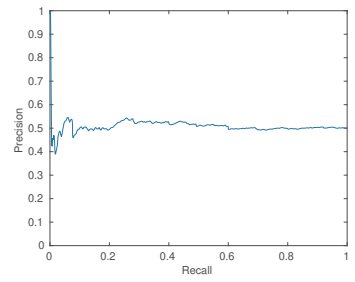
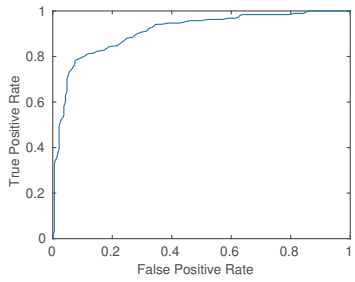
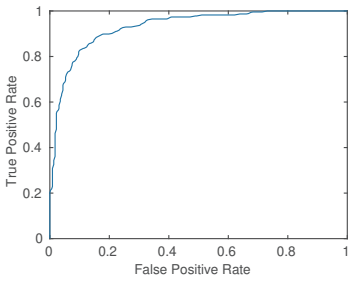


FIGURE D.1. Precision Recall and ROC Curves for historical corpora $\mathbf{C}_{1800}$

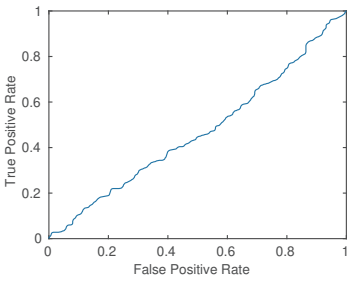
(a) ROC of $\Phi(\mathbf{L}_{posemo})$ vectors with an AUC of 0.91



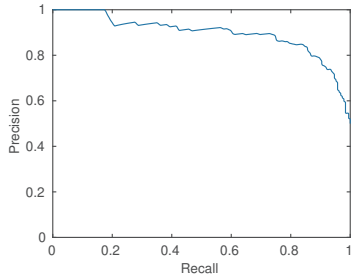
(b) ROC of $\Phi(\mathbf{L}_{negemo})$ vectors with an AUC of 0.94



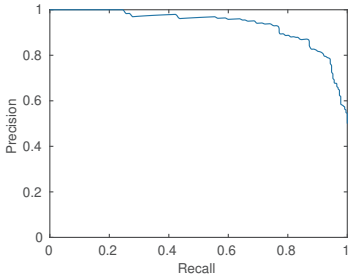
(c) ROC of $\Phi(\mathbf{L}_{random})$ vectors with an AUC of 0.49



(d) Precision Recall of $\Phi(\mathbf{L}_{posemo})$



(e) Precision Recall of $\Phi(\mathbf{L}_{negemo})$



(f) Precision Recall of $\Phi(\mathbf{L}_{random})$

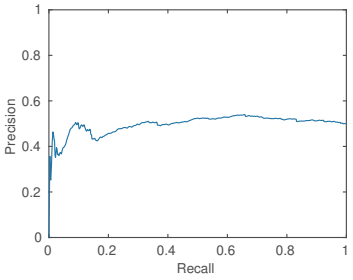


FIGURE D.2. Precision Recall and ROC Curves for historical corpora $\mathbf{C}_{1950}$

BIBLIOGRAPHY

[1] F. M. ADAMS AND C. E. OSGOOD, *A cross-cultural study of the affective meanings of color*, Journal of cross-cultural psychology, 4 (1973), pp. 135–156.

[2] J. ANGWIN, J. LARSON, S. MATTU, AND L. KIRCHNER, *Machine bias: There's software used across the country to predict future criminals. and it's biased against blacks*, ProPublica, May, 23 (2016).

[3] M. ANTHONY AND N. BIGGS, *Computational learning theory*, vol. 30, Cambridge University Press, 1997.

[4] C. J. AUSTER AND C. S. MANSBACH, *The gender marketing of toys: An analysis of color and type of toy on the disney store website*, Sex roles, 67 (2012), pp. 375–388.

[5] D. BAHDANAU, K. CHO, AND Y. BENGIO, *Neural machine translation by jointly learning to align and translate*, arXiv preprint arXiv:1409.0473, (2014).

[6] K. A. BARCHARD, K. E. GROB, AND M. J. ROE, *Is sadness blue? the problem of using figurative language for emotions on psychological tests*, Behavior research methods, 49 (2017), pp. 443–456.

[7] A. BEN-ZEEV AND T. C. DENNEHY, *When boys wear pink: A gendered color cue violation evokes risk taking.*, Psychology of Men & Masculinity, 15 (2014), p. 486.

[8] B. BERLIN AND P. KAY, *Basic color terms: Their universality and evolution*, Univ of California Press, 1991.

[9] P. BOJANOWSKI, E. GRAVE, A. JOULIN, AND T. MIKOLOV, *Enriching word vectors with subword information*, Transactions of the Association for Computational Linguistics, 5 (2017), pp. 135–146.

[10] T. BOLUKBASI, K.-W. CHANG, J. Y. ZOU, V. SALIGRAMA, AND A. T. KALAI, *Man is to computer programmer as woman is to homemaker? debiasing word embeddings*, in Advances in Neural Information Processing Systems, 2016, pp. 4349–4357.

[11] J. A. BULLINARIA AND J. P. LEVY, *Extracting semantic representations from word co-occurrence statistics: A computational study*, Behavior research methods, 39 (2007), pp. 510–526.

[12] A. CALISKAN, J. J. BRYSON, AND A. NARAYANAN, *Semantics derived automatically from language corpora contain human-like biases*, Science, 356 (2017), pp. 183–186.

[13] Y. CHEN, J. YANG, Q. PAN, M. VAZIRIAN, AND S. WESTLAND, *A method for exploring word-colour associations*, Color Research & Application, 45 (2020), pp. 85–94.

[14] R. T. COOK, *Dictionary of Philosophical Logic*, Edinburgh University Press, 2009.

[15] S. J. CUNNINGHAM AND C. N. MACRAE, *The colour of gender stereotyping*, British Journal of Psychology, 102 (2011), pp. 598–614.

[16] S. DEERWESTER, S. T. DUMAIS, G. W. FURNAS, T. K. LANDAUER, AND R. HARSHMAN, *Indexing by latent semantic analysis*, Journal of the American society for information science, 41 (1990), p. 391.

[17] M. DEL GIUDICE, *The twentieth century reversal of pink-blue gender coding: A scientific urban legend?*, Archives of sexual behavior, 41 (2012), pp. 1321–1323.

[18] M. DEL-GIUDICE, *Pink, blue, and gender: An update*, Archives of Sexual Behavior, 46 (2017), pp. 1555–1563.

[19] B. J. DEVEREUX, L. K. TYLER, J. GEERTZEN, AND B. RANDALL, *The centre for speech, language and the brain (cslb) concept property norms*, Behavior research methods, 46 (2014), pp. 1119–1127.

[20] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv preprint arXiv:1810.04805, (2018).

[21] M. FARUQUI, Y. TSVETKOV, P. RASTOGI, AND C. DYER, *Problems with evaluation of word embeddings using word similarity tasks*, arXiv preprint arXiv:1605.02276, (2016).

[22] J. R. FIRTH, *Papers in Linguistics 1934-1951: Repr*, Oxford University Press, 1961.

[23] I. FLAOUNAS, O. ALI, T. LANSDALL-WELFARE, T. DE BIE, N. MOSDELL, J. LEWIS, AND N. CRISTIANINI, *Research methods in the age of digital journalism: Massive-scale automated analysis of news-content—topics, style and gender*, Digital Journalism, 1 (2013), pp. 102–116.

[24] A. W. FLORES, K. BECHTEL, AND C. T. LOWENKAMP, *False positives, false negatives, and false analyses: A rejoinder to machine bias: There's software used across the country to predict future criminals. and it's biased against blacks*, Fed. Probation, 80 (2016), p. 38.

[25] R. FONG AND A. VEDALDI, *Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks*, arXiv preprint arXiv:1801.03454, (2018).

[26] J. M. B. FUGATE AND C. L. FRANCO, *What color is your anger? assessing color-emotion pairings in english speakers*, Frontiers in psychology, 10 (2019), p. 206.

[27] N. GARG, L. SCHIEBINGER, D. JURAFSKY, AND J. ZOU, *Word embeddings quantify 100 years of gender and ethnic stereotypes*, Proceedings of the National Academy of Sciences, (2018), p. 201720347.

[28] A. GLADKOVA, A. DROZD, AND S. MATSUOKA, *Analogy-based detection of morphological and semantic relations with word embeddings: what works and what doesn't.*, in Proceedings of the NAACL Student Research Workshop, 2016, pp. 8–15.

[29] H. GONEN AND Y. GOLDBERG, *Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them*, arXiv preprint arXiv:1903.03862, (2019).

[30] A. G. GREENWALD, D. E. MCGHEE, AND J. L. SCHWARTZ, *Measuring individual differences in implicit cognition: the implicit association test.*, Journal of personality and social psychology, 74 (1998), p. 1464.

[31] K. GULORDAVA AND M. BARONI, *A distributional similarity approach to the detection of semantic change in the google books ngram corpus.*, in Proceedings of the GEMS 2011 workshop on geometrical models of natural language semantics, 2011, pp. 67–71.

[32] M. HANADA, *Correspondence analysis of color–emotion associations*, Color Research & Application, 43 (2018), pp. 224–237.

[33] Z. S. HARRIS, *Distributional structure*, Word, 10 (1954), pp. 146–162.

[34] J. HUANG, A. J. GATES, R. SINATRA, AND A.-L. BARABÁSI, *Historical comparison of gender inequality in scientific careers across countries and disciplines*, Proceedings of the National Academy of Sciences, 117 (2020), pp. 4609–4616.

[35] H. L. HUGHES, *Pink tourism: Holidays of gay men and lesbians*, CABI, 2006.

[36] J. HUTCHINS, *The first public demonstration of machine translation: the georgetown-ibm system, 7th january 1954*, noviembre de, (2005).

[37] O. IRSOY AND C. CARDIE, *Deep recursive neural networks for compositionality in language*, in Advances in neural information processing systems, 2014, pp. 2096–2104.

[38] C. D. M. JEFFREY PENNINGTON, RICHARD SOCHER, *Wikipedia 2014 + gigaword 5 pretrained word embeddings*.
`http://nlp.stanford.edu/data/glove.6B.zip`.
Accessed: 2019-07-10.

[39] S. JIA, T. LANSDALL-WELFARE, AND N. CRISTIANINI, *Freudian slips: Analysing the internal representations of a neural network from its mistakes*, in Advances in Intelligent Data Analysis XVI, 2017, pp. 138–148.

[40] S. JIA, T. LANSDALL-WELFARE, S. SUDHAHAR, C. CARTER, AND N. CRISTIANINI, *Women are seen more than heard in online newspapers*, PLOS ONE, 11 (2016), pp. 1–11.

[41] D. JONAUSKAITE, N. DAEL, L. CHÈVRE, B. ALTHAUS, A. TREMEA, L. CHARALAMBIDES, AND C. MOHR, *Pink for girls, red for boys, and blue for both genders: Colour preferences in children and adults*, Sex Roles, 80 (2019), pp. 630–642.

[42] D. JONAUSKAITE, C. A. PARRAGA, M. QUIBLIER, AND C. MOHR, *Feeling blue or seeing red? similar patterns of emotion associations with colour patches and colour terms*, i-Perception, 11 (2020), p. 2041669520902484.

[43] A. JOULIN, E. GRAVE, P. BOJANOWSKI, AND T. MIKOLOV, *Bag of tricks for efficient text classification*, arXiv preprint arXiv:1607.01759, (2016).

[44] P. JUOLA, *The time course of language change*, Computers and the Humanities, 37 (2003), pp. 77–96.

[45] M. KAHNG, P. Y. ANDREWS, A. KALRO, AND D. H. P. CHAU, *Activis: Visual exploration of industry-scale deep neural network models*, IEEE transactions on visualization and computer graphics, 24 (2018), pp. 88–97.

[46] F. K. KHATTAK, S. JEBLEE, C. POU-PROM, M. ABDALLA, C. MEANEY, AND F. RUDZICZ, *A survey of word embeddings for clinical text*, Journal of Biomedical Informatics: X, 4 (2019), p. 100057.

[47] V. KOLLER, *Not just a colour': pink as a gender and sexuality marker in visual communication*, Visual communication, 7 (2008), pp. 395–423.

[48] V. KULKARNI, R. AL-RFOU, B. PEROZZI, AND S. SKIENA, *Statistically significant detection of linguistic change*, in Proceedings of the 24th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2015, pp. 625–635.

[49] T. LANSDALL-WELFARE, S. SUDHAHAR, J. THOMPSON, J. LEWIS, F. N. TEAM, N. CRISTIANINI, A. GREGOR, B. LOW, T. ATKIN-WRIGHT, M. DOBSON, ET AL., *Content analysis of 150 years of british periodicals*, Proceedings of the National Academy of Sciences, 114 (2017), pp. E457–E465.

[50] J. LEE, W. YOON, S. KIM, D. KIM, S. KIM, C. H. SO, AND J. KANG, *BioBERT: a pre-trained biomedical language representation model for biomedical text mining*, Bioinformatics, (2019).

[51] G. K. LEHMANN, A. J. ELLIOT, AND R. J. CALIN-JAGEMAN, *Meta-analysis of the effect of red on perceived attractiveness*, Evolutionary Psychology, 16 (2018), p. 1474704918802412.

[52] T. LEI, H. JOSHI, R. BARZILAY, T. JAAKKOLA, K. TYMOSHENKO, A. MOSCHITTI, AND L. MARQUEZ, *Semi-supervised question retrieval with gated convolutions*, arXiv preprint arXiv:1512.05726, (2015).

[53] D. LIN AND P. PANTEL, *Dirt@ sbt@ discovery of inference rules from text*, in Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 323–328.

[54] V. LOBUE AND J. S. DELOACHE, *Pretty in pink: The early development of gender-stereotyped colour preferences*, British Journal of Developmental Psychology, 29 (2011), pp. 656–667.

[55] R. LOWRY, *Concepts and applications of inferential statistics*, Directory of Open Educational Resources (DOER), (2014).

[56] M. P. LUCASSEN, T. GEVERS, AND A. GIJSENIJ, *Texture affects color emotion*, Color Research & Application, 36 (2011), pp. 426–436.

[57] J. MALPAS AND D. DAVIDSON, *The Stanford Encyclopedia of Philosophy (Winter 2012 Edition)*, Stanford University, 2012.

[58] S. V. MENTZEL, L. SCHÜCKER, N. HAGEMANN, AND B. STRAUSS, *Emotionality of colors: An implicit link between red and dominance*, Frontiers in psychology, 8 (2017), p. 317.

[59] T. MIKOLOV, K. CHEN, G. CORRADO, AND J. DEAN, *Efficient estimation of word representations in vector space*, arXiv preprint arXiv:1301.3781, (2013).

[60] T. MIKOLOV, A. DEORAS, D. POVEY, L. BURGET, AND J. ČERNOCKỲ, *Strategies for training large scale neural network language models*, in 2011 IEEE Workshop on Automatic Speech Recognition & Understanding, IEEE, 2011, pp. 196–201.

[61] T. MIKOLOV, E. GRAVE, P. BOJANOWSKI, C. PUHRSCH, AND A. JOULIN, *Advances in pre-training distributed word representations*, in Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.

[62] T. MIKOLOV, I. SUTSKEVER, K. CHEN, G. S. CORRADO, AND J. DEAN, *Distributed representations of words and phrases and their compositionality*, in Advances in neural information processing systems, 2013, pp. 3111–3119.

[63] G. A. MILLER, *Wordnet: a lexical database for english*, Communications of the ACM, 38 (1995), pp. 39–41.

[64] S. MITRA, R. MITRA, M. RIEDL, C. BIEMANN, A. MUKHERJEE, AND P. GOYAL, *That's sick dude!: Automatic identification of word sense change across different timescales*, arXiv preprint arXiv:1405.4392, (2014).

[65] E. NALISNICK, B. MITRA, N. CRASWELL, AND R. CARUANA, *Improving document ranking with dual word embeddings*, in Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 83–84.

[66] NATIONAL ARCHIVES, *Census records: England, wales, channel islands, isle of man*, 2017.

[67] K. NAZ AND H. EPPS, *Relationship between color and emotion: A study of college students*, College Student J, 38 (2004), p. 396.

[68] A. NEMATZADEH, S. C. MEYLAN, AND T. L. GRIFFITHS, *Evaluating vector-space models of word representation, or, the unreasonable effectiveness of counting words near other words.*, in CogSci, 2017.

[69] OFFICE FOR NATIONAL STATISTICS, *Statistical bulletin: Annual survey of hours and earnings: 2017 provisional and 2016 revised results*, 2017.

[70] W. H. ORGANISATION, *Icd-10 version:2019*.
https://icd.who.int/browse10/2019/en.
Accessed: 2020-07-02.

[71] R. PARKER, D. GRAFF, J. KONG, K. CHEN, AND K. MAEDA, *English gigaword fifth edition ldc2011t07. dvd*, Philadelphia: Linguistic Data Consortium, (2011).

[72] A. D. PAZDA AND A. J. ELLIOT, *Processing the word red can enhance women's perceptions of men's attractiveness*, Current Psychology, 36 (2017), pp. 316–323.

[73] Y. PENG, S. YAN, AND Z. LU, *Transfer learning in biomedical natural language processing: An evaluation of bert and elmo on ten benchmarking datasets*, arXiv preprint arXiv:1906.05474, (2019).

[74] J. W. PENNEBAKER, M. E. FRANCIS, AND R. J. BOOTH, *Linguistic inquiry and word count: Liwc 2007*, Mahway: Lawrence Erlbaum Associates, 71 (2001).

[75] J. PENNINGTON, R. SOCHER, AND C. MANNING, *Glove: Global vectors for word representation*, in Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[76] M. E. PETERS, M. NEUMANN, M. IYYER, M. GARDNER, C. CLARK, K. LEE, AND L. ZETTLEMOYER, *Deep contextualized word representations*, arXiv preprint arXiv:1802.05365, (2018).

[77] A. POMERLEAU, D. BOLDUC, G. MALCUIT, AND L. COSSETTE, *Pink or blue: Environmental gender stereotypes in the first two years of life*, Sex roles, 22 (1990), pp. 359–367.

[78] A. RADFORD, K. NARASIMHAN, T. SALIMANS, AND I. SUTSKEVER, *Improving language understanding by generative pre-training*, 2018.

[79] P. RAJPURKAR, J. ZHANG, K. LOPYREV, AND P. LIANG, *Squad: 100,000+ questions for machine comprehension of text*, arXiv preprint arXiv:1606.05250, (2016).

[80] M. T. RIBEIRO, S. SINGH, AND C. GUESTRIN, *Why should i trust you?: Explaining the predictions of any classifier*, in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 1135–1144.

[81] X. RONG, *word2vec parameter learning explained*, arXiv preprint arXiv:1411.2738, (2014).

[82] G. SALTON AND C. BUCKLEY, *Term-weighting approaches in automatic text retrieval*, Information processing & management, 24 (1988), pp. 513–523.

[83] W. SAMEK, A. BINDER, G. MONTAVON, S. LAPUSCHKIN, AND K.-R. MÜLLER, *Evaluating the visualization of what a deep neural network has learned*, IEEE transactions on neural networks and learning systems, (2017).

[84] J. L. SANDFORD, *Turn a colour with emotion: a linguistic construction of colour in english*, JAIC-Journal of the International Colour Association, 13 (2014).

[85] K. B. SCHLOSS, C. WITZEL, AND L. Y. LAI, *Blue hues don't bring the blues: questioning conventional notions of color–emotion associations*, JOSA A, 37 (2020), pp. 813–824.

[86] T. SCHNABEL, I. LABUTOV, D. MIMNO, AND T. JOACHIMS, *Evaluation methods for unsupervised word embeddings*, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 298–307.

[87] R. SCHWARZENBERG, L. RAITHEL, AND D. HARBECKE, *Neural vector conceptualization for word vector space interpretation*, arXiv preprint arXiv:1904.01500, (2019).

[88] C. E. SHANNON, *A mathematical theory of communication*, The Bell system technical journal, 27 (1948), pp. 379–423.

[89] A. SHRIVASTAVA, T. PFISTER, O. TUZEL, J. SUSSKIND, W. WANG, AND R. WEBB, *Learning from simulated and unsupervised images through adversarial training*, in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 3, 2017, p. 6.

[90] Y. SI, J. WANG, H. XU, AND K. ROBERTS, *Enhancing clinical concept extraction with contextual embeddings*, Journal of the American Medical Informatics Association, 26 (2019), pp. 1297–1304.

[91] D. SILVER, A. HUANG, C. J. MADDISON, A. GUEZ, L. SIFRE, G. VAN DEN DRIESSCHE, J. SCHRITTWIESER, I. ANTONOGLOU, V. PANNEERSHELVAM, M. LANCTOT, ET AL., *Mastering the game of go with deep neural networks and tree search*, Nature, 529 (2016), pp. 484–489.

[92] P. SOMMERAUER AND A. FOKKENS, *Firearms and tigers are dangerous, kitchen knives and zebras are not: Testing whether word embeddings can tell*, arXiv preprint arXiv:1809.01375, (2018).

[93] C. SORIANO AND J. VALENZUELA, *Emotion and colour across languages: implicit associations in spanish colour terms*, Social Science Information, 48 (2009), pp. 421–445.

[94] A. SUTTON AND N. CRISTIANINI, *On the learnability of concepts*, in IFIP International Conference on Artificial Intelligence Applications and Innovations, Springer, 2020, pp. 420–432.

[95] A. SUTTON, T. LANSDALL-WELFARE, AND N. CRISTIANINI, *Biased embeddings from wild data: Measuring, understanding and removing*, arXiv preprint arXiv:1806.06301, (2018).

[96] T. M. SUTTON AND J. ALTARRIBA, *Color associations to emotion and emotion-laden words: A collection of norms for stimulus construction and selection*, Behavior research methods, 48 (2016), pp. 686–728.

[97] P. D. TURNEY AND P. PANTEL, *From frequency to meaning: Vector space models of semantics*, Journal of artificial intelligence research, 37 (2010), pp. 141–188.

[98] P. VALDEZ AND A. MEHRABIAN, *Effects of color on emotions.*, Journal of experimental psychology: General, 123 (1994), p. 394.

[99] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. N. GOMEZ, Ł. KAISER, AND I. POLOSUKHIN, *Attention is all you need*, in Advances in neural information processing systems, 2017, pp. 5998–6008.

[100] A. WANG, A. SINGH, J. MICHAEL, F. HILL, O. LEVY, AND S. R. BOWMAN, *Glue: A multi-task benchmark and analysis platform for natural language understanding*, arXiv preprint arXiv:1804.07461, (2018).

[101] L. B. WEXNER, *The degree to which colors (hues) are associated with mood-tones.*, Journal of applied psychology, 38 (1954), p. 432.

[102] D. T. WIJAYA AND R. YENITERZI, *Understanding semantic change of words over centuries*, in Proceedings of the 2011 international workshop on DETecting and Exploiting Cultural diversiTy on the social web, 2011, pp. 35–40.

[103] WIKIMEDIA, *enwiki dump on 20190701*.
`https://dumps.wikimedia.org/enwiki/\20190701/`.
Accessed: 2019-07-07.

[104] F. WILCOXON, *Individual comparisons by ranking methods*, in Breakthroughs in statistics, Springer, 1992, pp. 196–202.

[105] W. I. WONG AND M. HINES, *Preferences for pink and blue: The development of color preferences as a distinct gender-typed behavior in toddlers*, Archives of sexual behavior, 44 (2015), pp. 1243–1254.

[106] Y. WU, J. LU, E. VAN DIJK, H. LI, AND S. SCHNALL, *The color red is implicitly associated with social status in the united kingdom and china*, Frontiers in psychology, 9 (2018), p. 1902.

[107] I. YAMADA, A. ASAI, H. SHINDO, H. TAKEDA, AND Y. TAKEFUJI, *Wikipedia2vec: An optimized tool for learning embeddings of words and entities from wikipedia*, arXiv preprint 1812.06280, (2018).

[108] X. YANG, C. MACDONALD, AND I. OUNIS, *Using word embeddings in twitter election classification*, Information Retrieval Journal, 21 (2018), pp. 183–207.

[109] H. ZHU, I. C. PASCHALIDIS, AND A. TAHMASEBI, *Clinical concept extraction with contextual word embedding*, arXiv preprint arXiv:1810.10566, (2018).

[110] J.-Y. ZHU, T. PARK, P. ISOLA, AND A. A. EFROS, *Unpaired image-to-image translation using cycle-consistent adversarial networks*, arXiv preprint arXiv:1703.10593, (2017).

[111] Y. ZHU, R. KIROS, R. ZEMEL, R. SALAKHUTDINOV, R. URTASUN, A. TORRALBA, AND S. FIDLER, *Aligning books and movies: Towards story-like visual explanations by watching movies and reading books*, in arXiv preprint arXiv:1506.06724, 2015.