ESSAYS ON UNDERSTANDING ANALOGIES AND ASSOCIATIONS IN WORD EMBEDDING SPACES

by

Kawin Ethayarajh

A thesis submitted in conformity with the requirements
for the degree of Master of Science
Graduate Department of Computer Science
University of Toronto

Abstract

Essays on Understanding Analogies and Associations
in Word Embedding Spaces

Kawin Ethayarajh
Master of Science
Graduate Department of Computer Science
University of Toronto
2019

Despite word embeddings being a cornerstone of current methods in natural language processing, many famous properties of embeddings remain poorly understood. For example, analogies such as *'man is to woman as king is to ?'* can often be solved with vector algebra, despite word embeddings being trained in a completely unsupervised manner. This work aims to answer two open questions: (1) Why, and under what conditions, can vector algebra be used to solve word analogy tasks?; (2) Why do socially undesirable associations such as gender bias exist in word embedding spaces and can they be provably removed? To answer these questions, I build on prior theoretical work that frames neural embedding models as matrix factorization, showing how the inner product of two word vectors has an information theoretic interpretation for models such as skipgram and GloVe. Building on one of the conclusions of my theories – namely that adding two word vectors automatically down-weights the the more frequent word – I then design a simple sentence embedding method that achieves state-of-the-art performance on sentence similarity tasks. This highlights how answering theoretical questions on word embeddings, while important in its own right, can also yield marked improvements on empirical problems.

Contents

Chapter 1

Introduction

Distributed representations of words, better known as word embeddings or word vectors, are a cornerstone of current methods in natural language processing (NLP). They can be generated by a variety of models, all of which share Firth's philosophy [Firth, 1957] that the meaning of a word is defined by "the company it keeps". The simplest such models obtain word vectors by constructing a low-rank approximation of a matrix containing a co-occurrence statistic [Landauer and Dumais, 1997, Rohde et al., 2006]. In contrast, neural network models [Bengio et al., 2003, Mikolov et al., 2013b] learn word embeddings by trying to predict words using the contexts they appear in, or vice-versa. The use of embeddings has led to marked improvements on a wide array of NLP tasks, ranging from textual entailment [Gong et al., 2018] to dependency parsing [Chen and Manning, 2014].

Still, many famous properties of word embeddings remain poorly understood, even those that are touted as merits of distributed representations. One notable – and surprising – property is that word analogies can often be solved with vector algebra, despite word embeddings being trained in a completely unsupervised manner using non-linear embedding models [Mikolov et al., 2013b]. For example, *'king is to ? as man is to woman'* can be solved by finding the closest vector to $\vec{king} - \vec{man} + \vec{woman}$, which should be $\vec{queen}$. However, such phenomena are not exclusively serendipitous. Word embeddings can also capture undesirable associations such as gender bias, allowing stereotypical analogies such as *'doctor is to nurse as man is to woman'* to also hold in the vector space [Bolukbasi et al., 2016]. Moreover, the tests used to measure undesirable associations, such as the word embedding factual association test (WEFAT) [Caliskan et al., 2017], are not based on any rigorous theory.

In this work, I provide an answer to two open questions on word embedding properties: (1) Why, and under what conditions, can vector algebra be used to solve word analogy tasks?; (2) Why do socially undesirable associations such as gender bias exist in word embedding spaces and can they be provably removed? To answer these questions, I build on prior theoretical work that frames neural embedding models as matrix factorization

[Levy and Goldberg, 2014]. Specifically, I show how the inner product of two word vectors, not just a word and a context vector, has an information theoretic interpretation for common embedding models such as skipgram with negative sampling (SGNS). Since both word analogies and word associations involve linear operations on inner products, this paradigm makes both of them much more interpretable. The major findings of this work are as follows:

1. Where $\text{csPMI}(x,y) \triangleq \text{PMI}(x,y) + \log p(x,y)$, a linear word analogy (e.g., *'king is to queen as man is to woman'*) holds over a set of word pairs iff $\text{csPMI}(x,y)$ is the same for every word pair and $\text{csPMI}(x_1,x_2) = \text{csPMI}(y_1,y_2)$ for any two word pairs. This identity can, in turn, be used to prove the conjecture proposed in [Pennington et al., 2014] as to why word analogies hold.

2. In an SGNS embedding space with no reconstruction error, the addition of two word vectors automatically down-weights the more frequent word. Since many weighting schemes are based on the idea that more frequent words should be down-weighted *ad hoc* [Arora et al., 2017a], the fact that this is done automatically provides novel justification for using addition to compose words.

3. In an SGNS or GloVe embedding space with no reconstruction error, the squared Euclidean distance between two words is a decreasing linear function of $\text{csPMI}(x,y)$. In other words, the more similar two words are (as measured by csPMI) the smaller the distance between their vectors. Although this is intuitive, it is also the first rigorous explanation of why the Euclidean distance in embedding space is a good proxy for word dissimilarity.

4. The subspace projection method for debiasing embeddings [Bolukbasi et al., 2016] is provably effective for any embedding model that implicitly does matrix factorization (e.g., GloVe, SGNS), when there is no reconstruction error. Debiasing vectors in this way is equivalent to training on an unbiased corpus. However, tests like WEAT and WEFAT [Caliskan et al., 2017] are not good measures of gender bias; they have theoretical flaws that easily exaggerate the extent of bias.

5. SGNS does not, *on average*, make the vast majority of words any more gendered in the vector space than they are in the training corpus; individual words may be slightly more or less gendered due to reconstruction error. However, for words that are gender-stereotyped (e.g., *nurse*) or gender-specific by definition (e.g., *queen*), SGNS amplifies the gender association in the training corpus.

I format these findings as two distinct and independent papers, one concerning the theoretical basis of word analogies and one concerning undesirable word embedding associations, each of which is given its own chapter. Both of these papers are co-authored with David Duvenaud and Graeme Hirst, with the former having been published as a preprint [Ethayarajh et al., 2018]. In the last chapter, which has been published as [Ethayarajh, 2018], I exploit one of the corollaries of my theory of linear word analogies (see item 3 above) and show just how effective weighted addition is for composing word vectors. I propose a sentence embedding approach called unsupervised smoothed inverse frequency (uSIF) that takes a weighted average of word vectors based on a random walk model and then denoises using singular value decomposition. Despite being simple and requiring no hyperparameter tuning – unlike similar methods, such as SIF [Arora et al., 2017b] – it achieves state-of-the-art results on several sentence similarity tasks. The success of this simple method highlights how answering theoretical questions about word embeddings, while an important research direction in its own right, can also yield marked improvements on empirical problems.

Chapter 2

Towards Understanding Linear Word Analogies

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst

2.1 Introduction

Distributed representations of words are a cornerstone of current methods in natural language processing. Word embeddings, also known as word vectors, can be generated by a variety of models, all of which share Firth's philosophy [Firth, 1957] that the meaning of a word is defined by "the company it keeps". The simplest such models obtain word vectors by constructing a low-rank approximation of a matrix containing a co-occurrence statistic [Landauer and Dumais, 1997, Rohde et al., 2006]. In contrast, neural network models [Bengio et al., 2003, Mikolov et al., 2013b] learn word embeddings by trying to predict words using the contexts they appear in, or vice-versa.

A surprising property of word vectors derived via neural networks is that word analogies can often be solved with vector algebra. For example, *'king is to ? as man is to woman'* can be solved by finding the closest vector to $\vec{king} - \vec{man} + \vec{woman}$, which should be $\vec{queen}$. It is unclear why linear operators can effectively compose embeddings generated by non-linear models such as skip-gram with negative sampling (SGNS). There have been two attempts to rigorously explain this phenomenon, but both have made strong assumptions about either the embedding space or the word distribution. The paraphrase model [Gittens et al., 2017] hinges on words having a uniform distribution rather than the typical Zipf distribution, which the authors themselves acknowledge is unrealistic. The latent variable model [Arora et al., 2016] makes many *a priori* assumptions about the word vectors, such as the assumption that word vectors are generated by randomly scaling vectors uniformly randomly sampled from the unit sphere.

In this paper, we explain why – and under what conditions – word analogies in GloVe and SGNS embedding spaces can be solved with vector algebra, without making the strong assumptions past work has. We begin by formalizing word analogies as functions that transform one word vector into another. When this transformation is simply the addition of a displacement vector – as is the case when using vector algebra – we call the analogy a *linear analogy*. Using the expression $\text{PMI}(x,y) + \log p(x,y)$, which we call the *co-occurrence shifted pointwise mutual information* (csPMI) of a word pair (x,y), we prove that in both SGNS and GloVe spaces without reconstruction error, a linear analogy holds over a set of ordered word pairs iff $\text{csPMI}(x,y)$ is the same for every word pair and $\text{csPMI}(x_1,x_2) = \text{csPMI}(y_1,y_2)$ for any two word pairs. By then framing vector addition as a kind of word analogy, we offer several new insights into the compositionality of words:

1. Past work has often cited the Pennington et al. conjecture [Pennington et al., 2014] as an intuitive explanation of why vector algebra works for analogy solving. The conjecture is that an analogy of the form *a is to b as x is to y* holds iff $p(w|a)/p(w|b) \approx p(w|x)/p(w|y)$ for every word w in the vocabulary. While this is sensible, it is not based on any theoretical derivation or empirical support. We provide a rigorous proof that this is indeed true.

2. Consider two words x,y and their sum $\vec{z} = \vec{x} + \vec{y}$ in an SGNS embedding space with no reconstruction error. If z were in the vocabulary, the similarity between z and x (as measured by the csPMI) would be the log probability of y shifted by a model-specific constant. This implies that the addition of two words automatically down-weights the more frequent word. Since many weighting schemes are based on the idea that more frequent words should be down-weighted *ad hoc* [Arora et al., 2017a], the fact that this is done automatically provides novel justification for using addition to compose words.

3. Consider any two words x,y in an SGNS or GloVe embedding space with no reconstruction error. The squared Euclidean distance between $\vec{x}$ and $\vec{y}$ is a decreasing linear function of $\text{csPMI}(x,y)$. In other words, the more similar two words are (as measured by csPMI) the smaller the distance between their vectors. Although this is intuitive, it is also the first rigorous explanation of why the Euclidean distance in embedding space is a good proxy for word dissimilarity.

Although our main theorem only concerns embedding spaces with no reconstruction error, we also explain why, in practice, linear word analogies hold in embedding spaces with some noise. We conduct experiments that support the few assumptions we make and show that the transformations represented by various word analogies correspond to different csPMI values. Without making the strong assumptions of past theories, we thus offer a rigorous explanation of why, and when, word analogies can be solved with vector algebra.

2.2 Related Work

PMI Pointwise mutual information (PMI) captures how much more frequently x, y co-occur than by chance: $\text{PMI} = \log[p(x,y)/(p(x)p(y))]$ [Church and Hanks, 1990].

Word Embeddings Word embeddings are distributed representations of words in a low-dimensional continuous space. Also called word vectors, they capture semantic and syntactic properties of words, even allowing relationships to be expressed algebraically [Mikolov et al., 2013b]. Word vectors are generally obtained in two ways: (a) from neural networks that learn representations by predicting co-occurrence patterns in the training corpus [Bengio et al., 2003, Mikolov et al., 2013b, Collobert and Weston, 2008]; (b) from low-rank approximations of word-context matrices containing a co-occurrence statistic [Landauer and Dumais, 1997, Levy and Goldberg, 2014].

SGNS The objective of skip-gram with negative sampling (SGNS) is to maximize the probability of observed word-context pairs and to minimize the probability of k randomly sampled negative examples. For an observed word-context pair (w, c), the objective would be $\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c' \sim P_n}[\log(-\vec{w} \cdot \vec{c}')]$, where c' is the negative context, randomly sampled from a scaled distribution P_n. Words that appear in similar contexts will therefore have similar embeddings. Though no co-occurrence statistics are explicitly calculated, [Levy and Goldberg, 2014] proved that SGNS is in fact implicitly factorizing a word-context PMI matrix shifted by $-\log k$.

Latent Variable Model The latent variable model [Arora et al., 2016] was the first attempt to rigorously explain why word analogies can be solved algebraically. It is a generative model that assumes that word vectors are generated by the random walk of a "discourse" vector on the unit sphere. Gittens et al.'s criticism of this proof is that it assumes that word vectors are known *a priori* and generated by randomly scaling vectors uniformly sampled from the unit sphere (or having properties consistent with this sampling procedure) [Gittens et al., 2017]. The theory also relies on word vectors being uniformly distributed (isotropic) in embedding space; however, experiments by [Mimno and Thompson, 2017] have found that this generally does not hold in practice, at least for SGNS.

Paraphrase Model The paraphrase model [Gittens et al., 2017] was the only other attempt to rigorously explain why word analogies can be solved algebraically. It proposes that any set of context words $C = \{c_1, ..., c_m\}$ is semantically equivalent to a single word c if $p(w|c_1, ..., c_m) = p(w|c)$. One problem with this is that the number of possible context sets far exceeds the vocabulary size, precluding a one-to-one mapping; the authors circumvent this problem by replacing exact equality with the minimization of KL divergence. Assuming that the words have a uniform distribution, the paraphrase of C can then be written as an unweighted sum of its context vectors.

However, this uniformity assumption is unrealistic – word frequencies obey a Zipf distribution, which is Pareto [Piantadosi, 2014].

2.3 The Structure of Word Analogies

2.3.1 Formalizing Analogies

A word analogy is a statement of the form *"a is to b as x is to y"*, which we will write as *(a,b)::(x,y)*. It asserts that *a* and *x* can be transformed in the same way to get *b* and *y* respectively, and that *b* and *y* can be inversely transformed to get *a* and *x*. A word analogy can hold over an arbitrary number of ordered pairs: e.g., *"Berlin is to Germany as Paris is to France as Ottawa is to Canada ..."*. The elements in each pair are not necessarily in the same space – for example, the transformation for *(king,roi)::(queen,reine)* is English-to-French translation. For *(king,queen)::(man,woman)*, the canonical analogy in the literature, the transformation corresponds to changing the gender. Therefore, to formalize the definition of an analogy, we will refer to it as a transformation.

Definition 1 *An analogy f is an invertible transformation that holds over a set of ordered pairs S iff* $\forall\ (x,y) \in S, f(x) = y \wedge f^{-1}(y) = x$.

The word embedding literature [Mikolov et al., 2013b, Pennington et al., 2014] has focused on a very specific type of transformation, the addition of a displacement vector. For example, for *(king,queen)::(man,woman)*, the transformation would be $\vec{king} + (\vec{woman} - \vec{man}) = \vec{queen}$, where the displacement vector is expressed as the difference $(\vec{woman} - \vec{man})$. To make a distinction with our general class of analogies in Definition 1, we will refer to these as *linear analogies*.

Definition 2 *A linear analogy f is an invertible transformation of the form* $\vec{x} \mapsto \vec{x} + \vec{r}$. *f holds over a set of ordered pairs S iff* $\forall\ (x,y) \in S, \vec{x} + \vec{r} = \vec{y}$.

Co-occurrence Shifted PMI Theorem *Let W be an SGNS or GloVe word embedding space with no reconstruction error and S be a set of ordered word pairs such that* $\forall\ (x,y) \in S, \vec{x}, \vec{y} \in W$ *and* $|S| > 1$. *A linear analogy f holds over S iff* $\exists\ \gamma \in \mathbb{R}, \forall (x,y) \in S, PMI(x,y) + \log p(x,y) = \gamma$ *and for any* $(x_1,y_1),(x_2,y_2) \in S$, $PMI(x_1,x_2) + \log p(x_1,x_2) = PMI(y_1,y_2) + \log p(y_1,y_2)$.

Throughout the rest of this paper, we will refer to $\text{PMI}(x,y) + \log p(x,y)$ as the *co-occurrence shifted PMI* (csPMI) of *x* and *y*. In sections 1.3.2 to 1.3.4, we prove the csPMI Theorem. In section 1.3.5, we explain why, in practice, perfect reconstruction is not needed to solve word analogies using vector algebra. In section 1.4, we explore what the csPMI Theorem implies about vector addition and Euclidean distance in embedding spaces.

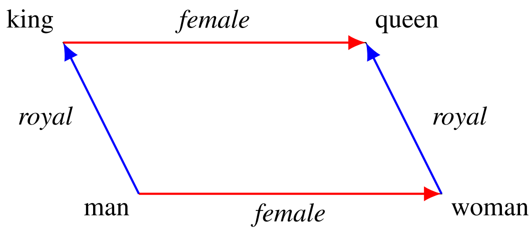

Figure 2.1: The parallelogram structure of the linear analogy *(king,queen)::(man,woman)*. A linear analogy transforms the first element in an ordered word pair by adding a displacement vector to it. Arrows indicate the directions of the semantic relations.

2.3.2 Analogies as Parallelograms

Lemma 1 *Where $\langle \cdot, \cdot \rangle$ denotes the inner product, a linear analogy f holds over a set of ordered word pairs S iff $\exists\ \gamma' \in \mathbb{R}, \forall\ (x,y) \in S, 2\langle \vec{x}, \vec{y} \rangle - \|\vec{x}\|_2^2 - \|\vec{y}\|_2^2 = \gamma'$ and $2\langle \vec{x}_1, \vec{x}_2 \rangle - \|\vec{x}_1\|_2^2 - \|\vec{x}_2\|_2^2 = 2\langle \vec{y}_1, \vec{y}_2 \rangle - \|\vec{y}_1\|_2^2 - \|\vec{y}_2\|_2^2$ for any $(x_1,y_1),(x_2,y_2) \in S$.*

When $S = \{(x_1,y_1)\}$, Lemma 1 holds for $\gamma' = 2\langle \vec{x}_1, \vec{y}_1 \rangle - \|\vec{x}_1\|_2^2 - \|\vec{y}_1\|_2^2$. When $|S| > 1$, consider the $|S|-1$ subsets of the form $\{(x_1,y_1),(x_2,y_2)\} \subset S$. f holds over every subset $\{(x_1,y_1),(x_2,y_2)\}$ iff it holds over S. We start by noting that by Definition 2, f holds over $\{(x_1,y_1),(x_2,y_2)\}$ iff:

$$\vec{x}_1 + \vec{r} = \vec{y}_1 \wedge \vec{x}_2 + \vec{r} = \vec{y}_2 \tag{2.1}$$

By rearranging (2.1), we know that $\vec{x}_2 - \vec{y}_2 = \vec{x}_1 - \vec{y}_1$ and $\vec{x}_2 - \vec{x}_1 = \vec{y}_2 - \vec{y}_1$. Put another way, x_1, y_1, x_2, y_2 form a quadrilateral in vector space whose opposite sides are parallel and equal in length. By definition, this quadrilateral is then a parallelogram. In fact, this is often how word analogies are visualized in the literature (see Figure 2.1).

To prove the first part of Lemma 1, we let $\gamma' = -\|\vec{r}\|_2^2$. A quadrilateral is a parallelogram iff each pair of opposite sides is equal in length. For every possible subset, $\vec{r} = (\vec{y}_1 - \vec{x}_1) = (\vec{y}_2 - \vec{x}_2)$. This implies that $\forall\ (x,y) \in S$,

$$\gamma' = -\|\vec{y} - \vec{x}\|_2^2 = 2\langle \vec{x}, \vec{y} \rangle - \|\vec{x}\|_2^2 - \|\vec{y}\|_2^2 \tag{2.2}$$

However, this condition is only necessary and not sufficient for the parallelogram to hold. The other pair of opposite sides, which do not correspond to $\vec{r}$, are equal in length iff $-\|\vec{x}_1 - \vec{x}_2\|_2^2 = -\|\vec{y}_1 - \vec{y}_2\|_2^2 \iff 2\langle \vec{x}_1, \vec{x}_2 \rangle - \|\vec{x}_1\|_2^2 - \|\vec{x}_2\|_2^2 = 2\langle \vec{y}_1, \vec{y}_2 \rangle - \|\vec{y}_1\|_2^2 - \|\vec{y}_2\|_2^2$, as stated in Lemma 1. Note that the sides that do not equal $\vec{r}$ do not necessarily have a fixed length across different subsets of S.

2.3.3 Analogies in the Context Space

Definition 3 *Let W be an SGNS or GloVe word embedding space and C its corresponding context space. Let k denote the number of negative samples, $X_{x,y}$ the frequency, and b_x, b_y the learned biases for GloVe. If there is no reconstruction error, for any words x, y with $\vec{x}, \vec{y} \in W$ and $\vec{x}_c, \vec{y}_c \in C$:*

$$\begin{aligned} \text{SGNS}: \quad & \langle \vec{x}, \vec{y}_c \rangle = \text{PMI}(x,y) - \log k \\ \text{GloVe}: \quad & \langle \vec{x}, \vec{y}_c \rangle = \log X_{x,y} - b_x - b_y \end{aligned} \tag{2.3}$$

SGNS and GloVe generate two vectors for each word in the vocabulary: a context vector, for when it is a context word, and a word vector, for when it is a target word. Context vectors are generally discarded after training. The SGNS identity in (2.3) is from [Levy and Goldberg, 2014], who proved that SGNS is implicitly factorizing the shifted word-context PMI matrix. The GloVe identity is simply the local objective for a word pair [Pennington et al., 2014]. Since the matrix being factorized in both models is symmetric, $\langle \vec{x}, \vec{y}_c \rangle = \langle \vec{x}_c, \vec{y} \rangle$.

Lemma 2 *A linear analogy $f : \vec{x} \mapsto \vec{x} + \vec{r}$ holds over a set of ordered pairs S in an SGNS or GloVe word embedding space W with no reconstruction error iff $\exists\ \lambda \in \mathbb{R}, g : \vec{x}_c \mapsto \vec{x}_c + \lambda \vec{r}$ holds over S in the corresponding context space C.*

In other words, an analogy f that holds over S in the word space has a corresponding analogy g that holds over S in the context space. The displacement vector of g is simply the displacement vector of f scaled by some $\lambda \in \mathbb{R}$. To prove this, we begin with (2.1) and any word w in the vocabulary:

$$\begin{aligned} & \vec{x}_2 - \vec{y}_2 = \vec{x}_1 - \vec{y}_1 \\ \iff & \langle \vec{w}_c, (\vec{x}_2 - \vec{y}_2) - (\vec{x}_1 - \vec{y}_1) \rangle = 0 \\ \iff & \langle \vec{w}, (\vec{x}_{2c} - \vec{y}_{2c}) - (\vec{x}_{1c} - \vec{y}_{1c}) \rangle = 0 \\ \iff & \vec{x}_{2c} - \vec{y}_{2c} = \vec{x}_{1c} - \vec{y}_{1c} \end{aligned} \tag{2.4}$$

Note that we can rewrite the second equation as the third because the matrices being factorized in (2.3) are symmetric and there is no reconstruction error. We can simplify from the second-last step because not all word vectors lie in the same hyperplane, implying that $(\vec{x}_{2c} - \vec{y}_{2c}) - (\vec{x}_{1c} - \vec{y}_{1c}) = \vec{0}$.

Thus a linear analogy with displacement vector $(\vec{y}_1 - \vec{x}_1)$ holds over S in the word embedding space iff an analogy with displacement vector $(\vec{y}_{1c} - \vec{x}_{1c})$ holds over S in the context space. This is supported by empirical findings that word and context spaces perform equally well on word analogy tasks [Pennington et al., 2014]. Since there is an analogous parallelogram structure formed by x_1, y_1, x_2, y_2 in the context space, there is some linear map from $\vec{w} \mapsto \vec{w}_c$ for each word $w \in S$. The real matrix A describing this linear map is symmetric: $\langle \vec{x}, \vec{y}_c \rangle = \vec{x}^T A \vec{y} = (A^T \vec{x})^T \vec{y} = \langle \vec{x}_c, \vec{y} \rangle$ for any $(x,y) \in S$. This implies that $C = AW$, since $\langle \vec{w}, \vec{x}_c \rangle = \langle \vec{w}_c, \vec{x} \rangle$ for any word w.

Since A is a real symmetric matrix, by the finite-dimensional spectral theorem, there is an orthonormal basis of W consisting of eigenvectors of A. If A had distinct eigenvalues, opposite sides of the parallelogram formed by x_1, y_1, x_2, y_2 in the word space could be stretched by different factors. This would imply that the quadrilateral formed by x_1, y_1, x_2, y_2 in the context space is not a parallelogram, which is a contradiction. Therefore A can only have non-distinct eigenvalues. Because A's eigenvectors are a basis for W and all have the same eigenvalue λ, all word vectors lie in the same eigenspace (i.e., $C = \lambda W$). Experiments done by [Mimno and Thompson, 2017] provide some empirical support of this result.

2.3.4 Proof of the csPMI Theorem

From Lemma 1, we know that if a linear analogy f holds over a set of ordered pairs S, then $\exists\ \gamma' \in \mathbb{R}, \forall\ (x,y) \in S, 2\langle \vec{x}, \vec{y} \rangle - \|\vec{x}\|_2^2 - \|\vec{y}\|_2^2 = \gamma'$. Because there is no reconstruction error, by Lemma 2, we can rewrite the inner product of two word vectors in terms of the inner product of a word and context vector. Using the SGNS identity in (2.3), we can rewrite (2.2):

$$\begin{aligned} \gamma' &= 2\langle \vec{x}, \vec{y} \rangle - \|\vec{x}\|_2^2 - \|\vec{y}\|_2^2 \\ &= (1/\lambda)\langle \vec{y} - \vec{x}, \vec{x}_c - \vec{y}_c \rangle \\ \lambda\gamma' &= 2\ \text{PMI}(x,y) - \text{PMI}(x,x) - \text{PMI}(y,y) \\ &= \text{csPMI}(x,y) - \log p(x|x)p(y|y) \end{aligned} \tag{2.5}$$

We get the same equation using the GloVe identity in (2.3), since the learned bias terms cancel out. For $\log p(x|x)p(y|y)$ to not be undefined, every word in S must appear in its own context at least once in the training corpus. However, depending on the size of the corpus and the context window, this may not necessarily occur. For this reason, we assume that $p(w,w)$, the probability that a word co-occurs with itself, follows the Zipf distribution of $p(w)$ scaled by some constant $\rho \in (0,1)$. We find this assumption to be justified, since the Pearson correlation between $p(w)$ and non-zero $p(w,w)$ is 0.825 for uniformly randomly sampled words in Wikipedia. We can therefore treat $\log p(x|x)p(y|y)\ \forall\ (x,y) \in S$ as a constant $\alpha \in \mathbb{R}^-$. Rewriting (2.5), we get

$$\lambda\gamma' + \alpha = \text{csPMI}(x,y) \tag{2.6}$$

The second identity in Lemma 1 can be expanded analogously, implying that f holds over a set of ordered pairs S iff (2.6) holds for every pair $(x,y) \in S$ and $\text{PMI}(x_1,x_2) + \log p(x_1,x_2) = \text{PMI}(y_1,y_2) + \log p(y_1,y_2)$ for any two pairs $(x_1,y_1),(x_2,y_2) \in S$. In section 1.5, we provide empirical support of this finding by showing that there is a moderately strong correlation (Pearson's $r > 0.50$) between csPMI(x,y) and γ', in both normalized and unnormalized SGNS embedding spaces.

2.3.5 Robustness to Noise

The csPMI Theorem does not explain why, in practice, linear word analogies hold in embedding spaces that have some reconstruction error. There are two reasons for this: the looser definition of vector equality in practice and the lower variance in reconstruction error associated with more frequent word pairs. For one, in practice, a word analogy task *(a,?)::(x,y)* is solved by finding the *most similar* vector to $\vec{a} + (\vec{y} - \vec{x})$, where dissimilarity is defined in terms of Euclidean or cosine distance. The correct solution to a word analogy can be found even when that solution is not exact.

The second reason is that the variance of the noise $\varepsilon_{x,y}$ for a word pair (x,y) (i.e., $\langle \vec{x}, \vec{y}_c \rangle - (\text{PMI}(x,y) - \log k)$) is a strictly decreasing function of the frequency $X_{x,y}$: more frequent word pairs are associated with less reconstruction error. This is because the cost of deviating from the optimal value is higher for more frequent word pairs: this is implicit in the SGNS objective [Levy and Goldberg, 2014] and explicit in GloVe objective [Pennington et al., 2014]. We also show that this holds empirically in section 1.5. Assuming $\varepsilon_{x,y} \sim \mathcal{N}(0, h(X_{x,y}))$, where δ is the Dirac delta distribution:

$$\begin{aligned} \lim_{X_{x,y} \to \infty} h(X_{x,y}) = 0 &\implies \lim_{X_{x,y} \to \infty} \mathcal{N}(0, h(X_{x,y})) = \delta \\ &\implies \lim_{X_{x,y} \to \infty} \varepsilon_{x,y} = 0 \end{aligned} \tag{2.7}$$

As the frequency of a word pair increases, the probability that the noise is close to zero increases; when the frequency is infinitely large, the noise is sampled from the Dirac delta distribution and is therefore zero. Thus even without the assumption of zero reconstruction error, an analogy that satisfies the identity in the csPMI Theorem will hold over a set of ordered pairs in practice as long as the frequency of each pair is sufficiently large.

A possible benefit of h mapping lower frequencies to larger variances is that it reduces the probability that a linear analogy f will hold over rare word pairs. One way of interpreting this is that h essentially filters out the word pairs for which there is insufficient evidence, even if the identities in the csPMI Theorem are satisfied. This would explain why reducing the dimensionality of word vectors – up to a point – actually improves performance on word analogy tasks [Yin and Shen, 2018]. Representations with the optimal dimensionality have enough noise to preclude spurious analogies that satisfy the csPMI Theorem, but not so much noise that non-spurious analogies (e.g., *(king,queen)::(man,woman)*) are also precluded.

2.4 Vector Addition as a Word Analogy

2.4.1 Formalizing Addition

Corollary 1 *Let $\vec{z} = \vec{x} + \vec{y}$ be the sum of words x, y in an SGNS word embedding space with no reconstruction error. If z were a word in the vocabulary, where δ is a model-specific constant, csPMI$(x,z) = \log p(y) + \delta$.*

To frame the addition of two words x, y as an analogy, we need to define a set of ordered pairs S such that a linear analogy holds over S iff $\vec{x} + \vec{y} = \vec{z}$. To this end, consider the set $\{(x,z), (\emptyset, y)\}$, where z is a placeholder for the composition of x and y and the null word $\emptyset$ maps to $\vec{0}$ for a given embedding space. From Definition 2:

$$\begin{aligned}
&(\vec{x} + \vec{r} = \vec{z}) \wedge (\vec{\emptyset} + \vec{r} = \vec{y}) \\
\iff &\vec{z} - \vec{x} = \vec{y} - \vec{\emptyset} \\
\iff &\vec{x} + \vec{y} = \vec{z}
\end{aligned} \tag{2.8}$$

Even though $\emptyset$ is not in the vocabulary, we can map it to $\vec{0}$ because its presence does not affect any other word vector. To understand why, consider the shifted word-context PMI matrix M that does not have $\emptyset$, and the matrix M' that does, of which M is a submatrix. Where W and C are the word and context matrices, $WC^T = M \iff [W\ \vec{0}][C\ \vec{0}]^T = M'$. Even if the null word does not exist for a given corpus, the embeddings we would get by training on a corpus that did have the null word would otherwise be identical. An inner product with the zero vector is always 0, so we can infer from the SGNS identity in (2.3) that $\text{PMI}(\emptyset, \cdot) - \log k = 0$ for every word in the vocabulary. From the csPMI Theorem, we know that if a linear analogy holds over $\{(x,z), (\emptyset, y)\}$, then

$$\begin{aligned}
&\text{PMI}(x,z) + \log p(x,z) \\
=&2\ \text{PMI}(\emptyset, y) + \log p(y) + \log p(\emptyset) \\
=&\log p(y) + \delta \\
\text{where } &\delta = \log k^2 + \log p(\emptyset)
\end{aligned} \tag{2.9}$$

Thus the csPMI of the sum and one word is equal to the log probability of the other word shifted by a model-specific constant. If we assume, as in section 1.3.5, that the noise is normally distributed, then even without the assumption of zero reconstruction error, the csPMI of the sum and one word is *on average* equal to the log probability of the other word shifted by a constant. We cannot repeat this derivation with GloVe because it is unclear what the optimal values of the learned biases would be, even with perfect reconstruction.

2.4.2 Automatically Weighting Words

Corollary 2 *In an SGNS word embedding space, on average, the sum of two words has more in common with the rarer word, where commonality is measured by csPMI.*

For two words x, y, assume without loss of generality that $p(x) > p(y)$. By (2.9):

$$\begin{aligned} p(x) > p(y) &\iff \log p(x) + \delta > \log p(y) + \delta \\ &\iff \text{csPMI}(z,y) > \text{csPMI}(z,x) \end{aligned} \tag{2.10}$$

Therefore addition automatically down-weights the more frequent word. For example, if the vectors for x = *'the'* and y = *'apple'* were added to create a vector for z = *'the apple'*, we would expect csPMI(*'the apple'*, *'apple'*) $>$ csPMI(*'the apple'*, *'the'*); being a stopword, *'the'* would on average be heavily down-weighted. While the rarer word is not always the more informative one, weighting schemes like inverse document frequency (IDF) [Robertson, 2004] and unsupervised smoothed inverse frequency (uSIF) [Ethayarajh, 2018] are all based on the principle that more frequent words should be down-weighted because they are typically less informative. The fact that addition automatically down-weights the more frequent word thus provides novel justification for using addition to compose words.

2.4.3 Interpreting Euclidean Distance

Corollary 3 $\exists\, \lambda \in \mathbb{R}^+, \delta \in \mathbb{R}^-$ *such that for any two words x and y in an SGNS or GloVe embedding space with no reconstruction error,* $\lambda\, \|\vec{x} - \vec{y}\|_2^2 = -csPMI(x,y) + \delta'$.

From (2.6), we know that for some $\lambda, \alpha, \gamma' \in \mathbb{R}$, $\text{csPMI}(x,y) = \lambda\gamma' + \alpha$, where $\gamma' = -\|\vec{x} - \vec{y}\|_2^2$. Rearranging this identity, we get

$$\begin{aligned} \|\vec{x} - \vec{y}\|_2^2 &= -\gamma' \\ &= (-1/\lambda)(\text{csPMI}(x,y) - \alpha) \\ \lambda\, \|\vec{x} - \vec{y}\|_2^2 &= -\text{csPMI}(x,y) + \delta' \\ \text{where } \delta' &= \alpha \end{aligned} \tag{2.11}$$

Thus the squared Euclidean distance between two word vectors is simply a linear function of the *negative* csPMI. Since $\text{csPMI}(x,y) \in (-\infty, 0]$ and $\|\vec{x} - \vec{y}\|_2^2$ is non-negative, λ is positive. This identity is intuitive: the more similar two words are (as measured by csPMI), the smaller the distance between their word embeddings. In section 1.5, we provide empirical evidence of this, showing that there is a moderately strong correlation (Pearson's $r > 0.50$) between $-\text{csPMI}(x,y)$ and $\|\vec{x} - \vec{y}\|_2^2$, in both normalized and unnormalized SGNS embedding spaces.

2.4.4 Are Relations Ratios?

[Pennington et al., 2014] conjectured that linear relationships in the embedding space – which we call displacements – correspond to ratios of the form $p(w|x)/p(w|y)$, where (x,y) is a pair of words such that $\vec{y}-\vec{x}$ is the displacement and w is some word in the vocabulary. This claim has since been repeated in other work [Arora et al., 2016]. For example, according to this conjecture, the analogy *(king,queen)::(man,woman)* holds iff for every word w in the vocabulary

$$\frac{p(w|king)}{p(w|queen)} \approx \frac{p(w|man)}{p(w|woman)} \tag{2.12}$$

However, as noted earlier, this idea was neither derived from empirical results nor rigorous theory, and there has been no work to suggest that it would hold for models other than GloVe, which was designed around it. We now prove this conjecture for SGNS using the csPMI Theorem.

Pennington et al. Conjecture Let S be a set of ordered pairs (x,y) with vectors in an SGNS word embedding space with no reconstruction error. A linear analogy holds over S iff $\forall\ (x_1,y_1),(x_2,y_2) \in S, p(w|x_1)/p(w|y_1) = p(w|x_2)/p(w|y_2)$ for every word w in the vocabulary.

Given that there is no reconstruction error, we can rewrite the identity in the conjecture using (2.3):

$$\begin{aligned}
&\frac{p(w|x_1)}{p(w|y_1)} = \frac{p(w|x_2)}{p(w|y_2)} \\
\Longleftrightarrow\ &\text{PMI}(w,x_1) - \text{PMI}(w,y_1) = \\
&\text{PMI}(w,x_2) - \text{PMI}(w,y_2) \\
\Longleftrightarrow\ &\langle \vec{w}_c, \vec{x}_1 \rangle - \langle \vec{w}_c, \vec{y}_1 \rangle = \langle \vec{w}_c, \vec{x}_2 \rangle - \langle \vec{w}_c, \vec{y}_2 \rangle \\
\Longleftrightarrow\ &\langle \vec{w}_c, (\vec{x}_1 - \vec{y}_1) - (\vec{x}_2 - \vec{y}_2) \rangle = 0
\end{aligned} \tag{2.13}$$

The same equation appears in the derivation in (2.4). This holds iff $\vec{x}_1 - \vec{y}_1 = \vec{x}_2 - \vec{y}_2$ (i.e., iff, by Definition 2, an analogy holds over $\{(x_1,y_1),(x_2,y_2)\}$) or if $\vec{w}_c$ is orthogonal to non-zero $(\vec{x}_1 - \vec{y}_1) - (\vec{x}_2 - \vec{y}_2)$. Even if the context vector of some word is orthogonal to the difference between the relation vectors, not all are – as noted in section 1.3.4, not all word or context vectors lie in the same hyperplane. Therefore, a linear word analogy holds over $\{(x_1,y_1),(x_2,y_2)\}$ iff for every word w, $p(w|x_1)/p(w|y_1) = p(w|x_2)/p(w|y_2)$. If this applies to every $(x_1,y_1),(x_2,y_2) \in S$, as stated in the conjecture, then the same analogy holds over S.

2.5 Experiments

Measuring Noise We uniformly sample word pairs in Wikipedia and estimate the noise (i.e., $\langle \vec{x}, \vec{y}_c \rangle - [\text{PMI}(x,y) - \log k]$) using SGNS vectors trained on the same corpus. As seen in Figure 2.2, the noise has an approximately zero-centered Gaussian distribution and the variance of the noise is lower at higher frequencies, supporting our assumptions in section 1.3.5. As previously mentioned, this is partly why linear word analogies are robust to noise: at high frequencies, the amount of noise is simply negligible.

Analogy	Mean csPMI	Mean PMI	Median Word Pair Frequency	csPMI SD	Accuracy
capital-world	−9.294	6.103	980.0	0.704	0.965
capital-common-countries	−9.818	4.339	3436.5	0.587	0.998
city-in-state	−10.127	4.003	4483.0	1.726	0.802
gram6-nationality-adjective	−10.691	3.733	3147.0	1.285	0.977
family	−11.163	4.111	1855.0	1.702	0.847
gram8-plural	−11.787	4.208	342.5	0.768	0.946
gram5-present-participle	−14.530	2.416	334.0	1.723	0.769
gram9-plural-verbs	−14.688	2.409	180.0	1.463	0.786
gram7-past-tense	−14.840	1.006	444.0	1.011	0.747
gram3-comparative	−15.111	1.894	194.5	1.077	0.923
gram2-opposite	−15.630	2.897	49.0	1.733	0.723
gram4-superlative	−15.632	2.015	100.5	1.641	0.793
currency	−15.900	3.025	19.0	2.002	0.275
gram1-adjective-to-adverb	−17.497	1.113	46.0	1.411	0.763

Table 2.1: The mean csPMI for analogies in [Mikolov et al., 2013a] over the word pairs for which they should hold (e.g., *(Paris, France)* for *capital-world*). As implied by the csPMI Theorem, similar analogies have a similar mean csPMI and algebraic solutions are less accurate at higher csPMI variances. The type of analogy gradually changes with the csPMI, from geography (*capital-world*) to verb tense (*gram7-past-tense*) to adjectives (*gram2-opposite*).

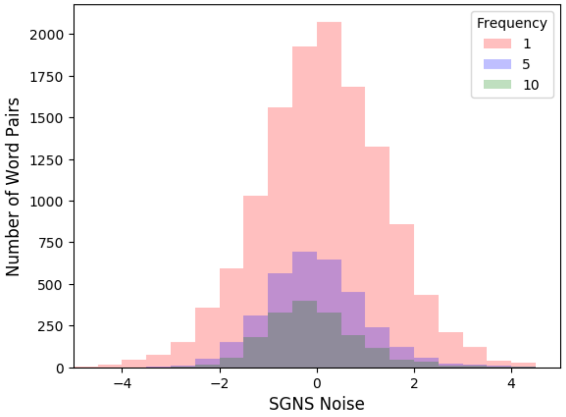

Figure 2.2: The noise distribution for an SGNS embedding model (i.e., $\langle \vec{x}, \vec{y}_c \rangle - [\text{PMI}(x,y) - \log k]$) at various frequencies. The noise is normally distributed and the variance decreases as the frequency increases.

Estimating csPMI The csPMI Theorem implies that if an analogy holds exactly over a set of word pairs when there is no reconstruction error, then each word pair has the same csPMI value. In Table 2.1, we provide the mean csPMI values for various analogies in [Mikolov et al., 2013a] over the set of word pairs for which they should

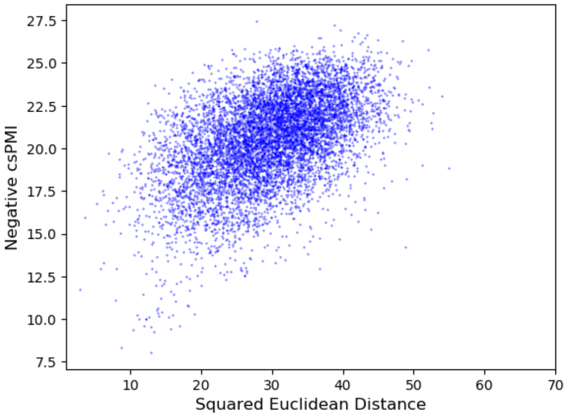

Figure 2.3: The negative csPMI for a word pair against the squared Euclidean distance between its SGNS word vectors. There is a positive correlation (Pearson's r = 0.502); the more similar two words are, the smaller the Euclidean distance between their vectors. In the normalized SGNS word space, the correlation is just as strong (Pearson's r = 0.514).

hold (e.g., {*(Paris, France), (Berlin, Germany)*} for *capital-world*). We also provide the accuracy of the vector algebraic solutions for each analogy, found by minimizing cosine distance over the set of all words in the analogy task.

As expected, when solutions to word analogies are more accurate, the variance in csPMI tends to be lower. This is because an analogy is more likely to hold over a set of word pairs when the displacement vectors are the same, and thus when the csPMI values are the same. Similar analogies (e.g., *capital-world* and *capital-common-countries*) also have similar mean csPMI values – our theory implies this, since similar analogies have similar displacement vectors. As the csPMI changes, the type of analogy gradually changes from geography (*capital-world*, *city-in-state*) to verb tense (*gram5-present-participle*, *gram7-past-tense*) to adjectives (*gram2-opposite*, *gram4-superlative*). We do not witness a similar gradation with the mean PMI, implying that analogies correspond uniquely to csPMI but not PMI.

Euclidean Distance Because the sum of two word vectors is not in the vocabulary, we cannot calculate co-occurrence statistics involving the sum, precluding us from testing Corollaries 1 and 2. We test Corollary 3 by uniformly sampling word pairs and plotting, in Figure 2.3, the negative csPMI against the squared Euclidean distance between the SGNS word vectors. As expected, there is a moderately strong and positive correlation (Pearson's r = 0.502): the more similar two words are (as measured by csPMI) the smaller the Euclidean distance between their vectors. The correlation is just as strong in the normalized SGNS word space, where Pearson's r = 0.514. As mentioned earlier in section 1.3.4, our assumption that $p(w,w)$ follows the Zipf distribution of $p(w)$ scaled by some $\rho \in (0,1)$ is justified here, since there is a strong positive correlation between the two (Pearson's r = 0.825).

Unsolvability The csPMI Theorem reveals two reasons why an analogy may be unsolvable in a given embedding space: polysemy and corpus bias. Consider senses $\{x_1,...,x_M\}$ of a polysemous word x. Assuming perfect reconstruction, a linear analogy f whose displacement has csPMI γ does not hold over (x,y) if $\gamma \neq \text{PMI}(x,y) + \log p(x,y) = \log\left[p(x_1|y) + ... + p(x_M|y)\right] p(y|x)$. The Theorem applies over all the senses of x, even if only a particular sense is relevant to the analogy. For example, while *(open,closed)::(high,low)* may make intuitive sense, it is unlikely to hold in an embedding space, given that all four words are highly polysemous.

Even if *(a,b)::(x,y)* is intuitive, there is also no guarantee that csPMI$(a,b) \approx$ csPMI(x,y) and csPMI$(a,x) \approx$ csPMI(b,y) for a given training corpus. The less frequent a word pair is, the more sensitive its csPMI to even small changes in frequency. Infrequent word pairs are also associated with more reconstruction error (see section 1.3.5), making it even more unlikely that the analogy will hold in practice. This is why the accuracy for the *currency* analogy is so low (see Table 2.1) – in Wikipedia, currencies and their country co-occur with a median frequency of only 19.

2.6 Conclusion

In this paper, we answered several open questions about gender bias in word embeddings, and word associations more broadly. We proved that when there is no reconstruction error, for any embedding model that implicitly does matrix factorization (e.g., SGNS, GloVe), debiasing with the subspace projection method is equivalent to training on an unbiased corpus. This was the first theoretical guarantee of this method, which is popular due to its empirical success. We then proved that WEAT and WEFAT, the most common tests of word embedding association, have theoretical flaws that exaggerate the extent of gender bias. By contriving the attribute sets for WEFAT, virtually any word can be classified as gender-biased. We then derived a new measure of association in word embeddings called the relational inner product association (RIPA). Using RIPA, we found that SGNS does not, on average, make most words any more gendered than they are in the training corpus. However, for words that are gender-biased or gender-specific by definition, SGNS actually amplifies the genderedness in the corpus.

Chapter 3

Understanding Undesirable Word Embedding Associations

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst

3.1 Introduction

A common criticism of word embeddings is that they capture undesirable associations in vector space. In addition to gender-appropriate analogies such as *king:queen::man:woman*, gender-biased analogies such as *doctor:nurse::man:woman* also hold in SGNS embedding spaces [Bolukbasi et al., 2016]. [Caliskan et al., 2017] created association tests for word vectors called WEAT and WEFAT, which use cosine similarity to measure how associated words are with respect to two sets of attribute words (e.g., 'male' vs. 'female'). They claimed that the vector for 'science' was significantly more associated with male attributes and the vector for 'art' with more female ones, for example. Since these associations are socially undesirable, they were described as gender bias.

Despite these remarkable findings, gender bias in word embeddings is still poorly understood. For one, what causes it – is it biased training data, the embedding model itself, or just noise? Why should WEAT and WEFAT be the tests of choice for measuring associations in word embeddings? [Bolukbasi et al., 2016] found that word vectors could be debiased by defining a "bias subspace" in the embedding space and then subtracting from each vector its projection on this subspace. But what theoretical guarantee is there that this method actually debiases vectors?

In this paper, we answer several of these open questions. We begin by proving that for any neural embedding model that implicitly does matrix factorization (e.g., GloVe, SGNS), assuming no reconstruction error, debiasing vectors *post hoc* via subspace projection is equivalent to training on an unbiased corpus. We find that contrary to what [Bolukbasi et al., 2016] suggested, word embeddings *should not* be normalized before debiasing, as vector length can contain important information [Ethayarajh et al., 2018]. If applied this way, the subspace projection method can be used to provably debias embeddings generated by SGNS and GloVe, among other models.

Using this notion of a "bias subspace", we then prove that WEAT and WEFAT, the most common association tests for word embeddings, have theoretical flaws that exaggerate the extent of gender bias. At least for SGNS and GloVe, these tests implicitly require the two sets of attribute words (e.g., 'male' vs. 'female') to occur with equal frequency in the training corpus; when they do not, even gender-neutral words can be classified as gender-biased. The test statistics for WEAT and WEFAT can also be easily manipulated by contriving the attribute word sets, allowing virtually any word – even a gender-neutral one – to be classified as male- or female-biased.

Given that subspace projection removal provably debiases embeddings, we use it to derive a new measure of association in word embeddings called the *relational inner product association* (RIPA). Given a set of ordered word pairs (e.g., $\{$('man', 'woman'), ('male', 'female')$\}$), we take the first principal component of all the difference vectors, which we call the *relation vector* $\vec{b}$. In [Bolukbasi et al., 2016]'s terminology, $\vec{b}$ would be a one-dimensional bias subspace. Then, for a word vector $\vec{w}$, the relational inner product is simply $\langle \vec{w}, \vec{b} \rangle$. Because RIPA is for embedding models that implicitly do matrix factorization, it has an information theoretic interpretation, allowing us to directly compare the association in the vector space with the association in the training corpus. This yields novel insights into the origin of gender bias in embedding spaces:

1. SGNS does not, *on average*, make the vast majority of words any more gendered in the vector space than they are in the training corpus; individual words may be slightly more or less gendered due to reconstruction error. However, for words that are gender-stereotyped (e.g., *nurse*) or gender-specific by definition (e.g., *queen*), SGNS amplifies the gender association in the training corpus.

2. To use the subspace projection method, one must have prior knowledge of which words are gender-specific by definition, so that they are not also debiased. Debiasing all vectors can preclude gender-appropriate analogies such as *king:queen::man:woman* from holding in the vector space. Since [Bolukbasi et al., 2016] use a supervised method to identify gender-specific words, we propose an unsupervised method. Ours is much more effective at preserving gender-appropriate analogies and precluding gender-biased ones.

To allow a fair comparison with prior work, our experiments in this paper focus on gender association. However, we ground our claims in theory so that they extend to other types of word associations as well, which we leave as future work.

3.2 Related Work

Word Embeddings Word embedding models generate distributed representations of words in a low-dimensional continuous space. This is generally done using: (a) neural networks that learn embeddings by predicting the contexts words appear in, or vice-versa [Bengio et al., 2003, Mikolov et al., 2013b, Collobert and Weston, 2008]; (b) low-rank approximations of word-context matrices containing a co-occurrence statistic [Landauer and Dumais, 1997, Levy and Goldberg, 2014]. The objective of SGNS is to maximize the probability of observed word-context pairs and to minimize the probability of k randomly sampled negative examples. Though no co-occurrence statistics are explicitly calculated, [Levy and Goldberg, 2014] proved that SGNS is implicitly factorizing a word-context PMI matrix shifted by $-\log k$. Similarly, GloVe implicitly factorizes a log co-occurrence count matrix [Pennington et al., 2014].

Word Analogies A word analogy *a:b::x:y* asserts that "*a is to b as x is to y*" and holds in the embedding space iff $\vec{a} + (\vec{y} - \vec{x}) = \vec{b}$. [Ethayarajh et al., 2018] proved that for GloVe and SGNS, *a:b::x:y* holds exactly in an embedding space with no reconstruction error iff csPMI(*a,b*) = csPMI(*x,y*) and csPMI(*a,x*) = csPMI(*b,y*), where $\text{csPMI}(a,b) = \text{PMI}(a,b) + \log p(a,b)$. Word analogies are often used to signify that semantic and grammatical properties of words (e.g., verb tense, gender) can be captured as linear relations.

Measuring Associations [Caliskan et al., 2017] proposed what are now the most commonly used association tests for word embeddings. The word embedding factual association test (WEFAT) and word embedding association test (WEAT) use cosine similarity to measure how associated a given set(s) of target words are with respect to two sets of attribute words (e.g., 'male' vs. 'female'). WEFAT is used for testing a single set of target words; WEAT for two sets. For example, [Caliskan et al., 2017] claimed that the vector for 'science' was more associated with 'male' than 'female' attributes, and that this was statistically significant. However, aside from some intuitive results (e.g., that female names are associated with female attributes), there is little evidence that WEAT and WEFAT are good measures of association.

Debiasing Embeddings [Bolukbasi et al., 2016] claimed that the existence of gender-biased analogies such as *doctor:nurse::man :woman* constituted gender bias. To prevent such analogies from holding in the vector space, they subtracted from each biased word vector its projection on a "gender bias subspace". This subspace was defined by the first m principal components for ten gender relation vectors (e.g., $\vec{man} - \vec{woman}$). Each debiased word vector was thus orthogonal to the gender bias subspace and its projection on the subspace was zero. While this subspace projection method precluded gender-biased analogies from holding in the embedding space, [Bolukbasi et al., 2016] did not provide any theoretical guarantee that the vectors were unbiased (i.e., equivalent to vectors that would be obtained from training on a gender-agnostic corpus with no reconstruction error). Other work has tried to learn gender-neutral embeddings from scratch [Zhao et al., 2018], despite this approach requiring custom changes to the objective of each embedding model.

3.3 Provably Debiasing Embeddings

Experiments by [Bolukbasi et al., 2016] found that debiasing word embeddings using the subspace projection method precludes gender-biased analogies from holding. However, as we noted earlier, despite this method being intuitive, there is no theoretical guarantee that the debiased vectors are perfectly unbiased or that the debiasing method works for embedding models other than SGNS. In this section, we prove that for any embedding model that does implicit matrix factorization (e.g., GloVe, SGNS), when there is no reconstruction error, debiasing *unnormalized* embeddings *post hoc* using the subspace projection method is equivalent to training on a perfectly unbiased corpus.

Definition 1 *Let M denote the Hermitian word-context matrix for a given training corpus that is implicitly or explicitly factorized by the embedding model. Let S denote a set of word pairs. A word w is* unbiased *with respect to S iff $\forall\,(x,y) \in S, M_{w,x} = M_{w,y}$. M is* unbiased *with respect to S iff $\forall\, w \notin S$, w is unbiased. A word w or matrix M is* biased *wrt S iff it is not unbiased wrt S.*

Note that Definition 1 does not make any distinction between socially acceptable and socially unacceptable associations. A word that is gender-specific by definition and a word that is gender-biased due to stereotypes would both be considered biased by Definition 1, although only the latter is undesirable. For example, by Definition 1, ‘door’ would be unbiased with respect to the set {(‘male’, ‘female’)} iff the entries for $M_{\text{door,male}}$ and $M_{\text{door,female}}$ were interchangeable. The entire corpus would be unbiased with respect to the set iff $M_{w,\text{male}}$ and $M_{w,\text{female}}$ were interchangeable for any word w. Since M is a word-context matrix containing a co-occurrence statistic, unbiasedness effectively means that the elements for $(w, \text{‘male’})$ and $(w, \text{‘female’})$ in M can be switched without any impact on the embeddings. M is factorized into a word matrix W and context matrix C such that $WC^T = M$, with the former giving us our word embeddings.

Theorem 1 *For a set of word pairs S, let B denote the bias subspace spanned by $\{\vec{x}-\vec{y}\,|\,(x,y) \in S\}$. For every word $w \notin S$, let $\vec{w}_d \triangleq \vec{w} - proj_B\vec{w}$. Assuming no reconstruction error, where $WC^T = M$, the reconstructed matrix $W_dC^T = M_d$ is unbiased with respect to S.*

Lemma 1 *A word embedding $\vec{w}$ is* unbiased *with respect to a set of word pairs S iff $\forall\,(x,y) \in S, \langle \vec{w}, \vec{x}_c - \vec{y}_c \rangle = 0$, where $\vec{x}_c, \vec{y}_c$ are the context vectors of x, y respectively.*

Assuming perfect reconstruction, this follows from Definition 1. The word w is unbiased with respect to S iff $\forall\,(x,y) \in S, M_{w,x} = M_{w,y} \iff \langle \vec{w}, \vec{x_c} \rangle = \langle \vec{w}, \vec{y_c} \rangle \iff \langle \vec{w}, \vec{x_c} - \vec{y_c} \rangle = 0$.

Lemma 2 *Assuming there is no reconstruction error, for any word w and any* $(x,y) \in S$, $\exists\, \lambda \in \mathbb{R}$, $\langle \vec{w}, \vec{x_c} - \vec{y_c} \rangle = \lambda \, \langle \vec{w}, \vec{x} - \vec{y} \rangle$.

For a detailed explanation, we refer the reader to the proof of Lemma 2 of [Ethayarajh et al., 2018], which shows that under perfect reconstruction, $\exists\, \lambda \in \mathbb{R}, C = \lambda W$. In short, if a linear word analogy holds over S (i.e., the word pairs have the same difference vector), then there exists a real symmetric matrix A that maps W to C. A can only have non-distinct eigenvalues because any word analogy that holds in the word space must also hold in the context space. All word vectors must therefore lie in the same eigenspace, with eigenvalue λ. Then $\langle \vec{w}, \vec{x_c} - \vec{y_c} \rangle = \langle \vec{w}, \lambda (\vec{x} - \vec{y}) \rangle$.

Proof of Theorem 1 Each debiased word vector w_d is orthogonal to the bias subspace in the word embedding space, so $\forall\,(x,y) \in S, \langle \vec{w_d}, \vec{x} - \vec{y} \rangle = 0$. Then by Lemma 2, $\forall\,(x,y) \in S, \langle \vec{w_d}, \vec{x_c} - \vec{y_c} \rangle = \lambda \, \langle \vec{w_d}, \vec{x} - \vec{y} \rangle = 0$. By Lemma 1, every debiased word vector $\vec{w_d}$ is therefore unbiased with respect to S. This implies that the co-occurrence matrix M_d that is reconstructed using the debiased word matrix W_d is also unbiased with respect to S.

The subspace projection method is therefore far more powerful than initially stated in [Bolukbasi et al., 2016]: not only can it be applied to any embedding model that implicitly does matrix factorization (e.g., GloVe, SGNS), but debiasing word vectors in this way is equivalent to training on a perfectly unbiased corpus when there is no reconstruction error. However, word vectors *should not* be normalized prior to debiasing, since the matrix that is factorized by the embedding model cannot necessarily be reconstructed with normalized embeddings, at least when there is no reconstruction error.

3.4 The Flaws of WEAT and WEFAT

Given *attribute word sets* X and Y (e.g., {'male', 'man'} vs. {'female', 'woman'}), WEFAT and WEAT use cosine similarity-based statistics to capture whether a given set(s) of *target words* are more associated with X or Y [Caliskan et al., 2017]. For the sake of simplicity, we focus on WEFAT – which applies to a single set of target words – though our criticisms extend to WEAT as well. Given w, X, Y, the statistic d for WEFAT is:

$$d = \frac{\mu(\{\cos(\vec{w},\vec{x}) | x \in X\}) - \mu(\{\cos(\vec{w},\vec{y}) | y \in Y\})}{\sigma(\{\cos(\vec{w},\vec{z}) \mid z \in X \cup Y\})} \tag{3.1}$$

This statistic is calculated for every word w in the target set, which also has a value p_w that denotes some property. The null hypothesis is that d is unrelated to p_w, which is tested using linear regression [Caliskan et al., 2017].

Proposition 1 *Let $X = \{x\}, Y = \{y\}$, and w be unbiased with respect to $\{(x,y)\}$ by Definition 1. According to WEFAT, an SGNS vector $\vec{w}$ is equally associated with X and Y under perfect reconstruction iff $p(x) = p(y)$.*

Both theoretical and empirical work have found the squared word embedding norm $\|\vec{w}\|_2^2$ to be linear in the log probability of the word, $\log p(w)$ [Arora et al., 2016, Ethayarajh et al., 2018]. Then where $\alpha_1, \alpha_2 \in \mathbb{R}$,

$$\begin{aligned} 0 &= \cos(\vec{w},\vec{x}) - \cos(\vec{w},\vec{y}) \\ &= \frac{1}{\|\vec{w}\|_2}\left(\frac{\langle\vec{w},\vec{x}\rangle}{\|\vec{x}\|_2} - \frac{\langle\vec{w},\vec{y}\rangle}{\|\vec{y}\|_2}\right) \\ &= \frac{\langle\vec{w},\vec{x}\rangle}{\sqrt{\alpha_1 \log p(x) + \alpha_2}} - \frac{\langle\vec{w},\vec{y}\rangle}{\sqrt{\alpha_1 \log p(y) + \alpha_2}} \end{aligned} \tag{3.2}$$

By Theorem 1, w is unbiased with respect to the set $\{(x,y)\}$ iff $\langle\vec{w},\vec{x}\rangle = \langle\vec{w},\vec{y}\rangle$. Therefore (3.2) holds iff $p(x) = p(y)$. Because SGNS implicitly factorizes a shifted word-context PMI matrix [Levy and Goldberg, 2014] and the context matrix is a scalar multiple of the word matrix when there is no reconstruction error [Ethayarajh et al., 2018], if the squared vector norm were not linear in $\log p(w)$, then the condition that would need to be satisfied is even stricter: $\text{PMI}(x,x) = \text{PMI}(y,y)$.

Thus for w to be equally associated with both sets of attribute words, not only must w be unbiased with respect to $\{(x,y)\}$ by Definition 1, but words x,y must also occur with equal frequency in the corpus. If the embedding model were GloVe instead of SGNS, the equal frequency requirement would still apply, since GloVe implicitly factorizes a log co-occurrence count matrix [Pennington et al., 2014] while SGNS implicitly factorizes the shifted PMI matrix [Levy and Goldberg, 2014]. Despite this being implicitly required, it was not stated as a requirement in [Caliskan et al., 2017] for using WEAT and WEFAT. In practice, this issue often goes unnoticed because each word in the attribute set, at least for gender association, has a counterpart that appears with roughly equal frequency in most training corpora (e.g., ‘man’ vs. ‘woman’, ‘boy’ vs. ‘girl’). However, for attribute sets that are more nebulous (e.g., ‘pleasant’ vs. ‘unpleasant’ words), this is not guaranteed to hold.

Proposition 2 *Assume that for a word w, $\exists\, c \in \mathbb{R}^+$, $\forall\, x \in X, \cos(\vec{w},\vec{x}) = c$ and $\forall\, y \in Y, \cos(\vec{w},\vec{y}) = -c$. Then regardless of how small $|c|$ is, the WEFAT statistic $d = 2$.*

If $\exists\, c \in \mathbb{R}^+$, $\forall\, x \in X, \cos(\vec{w},\vec{x}) = c$ and $\forall\, y \in Y, \cos(\vec{w},\vec{y}) = -c$, then $\mu(\{\cos(\vec{w},\vec{a}) | a \in A\}) - \mu(\{\cos(\vec{w},\vec{b}) | b \in B\}) = 2c$. Because the standard deviation is also c, $d = 2c/c = 2$. In this scenario, the WEFAT statistic is at its maximum regardless of how small $|c|$ is. For example, even if the cosine similarity is 0.01 between $\vec{door}$ and male attribute words while being -0.01 between $\vec{door}$ and female attribute words, $\vec{door}$ would still be male-associated according to WEFAT. Even though this exact scenario does not always apply, it evinces that attribute sets can be easily contrived to manipulate the test statistic, such that even words that are marginally closer to one attribute set in the vector space end up being classified as biased. In Table 3.1, we show how we can easily contrive the

Target Word	Male Attribute Words	Female Attribute Words	μ_{male}	μ_{female}	WEFAT Statistic
door	male, masculine, manly	she, womanly, girl	0.115	0.226	-1.425
	male, masculine, man	she, woman, female	0.152	0.197	-0.707
	male, masculine	female, feminine	0.108	0.109	-0.040
	male, manly	female, feminine	0.138	0.109	1.356
	male, boyish	female, feminine	0.177	0.109	1.582
bowtie	male, masculine, manly	she, womanly, girl	0.119	0.199	-1.297
	male, masculine	female, feminine	0.100	0.121	-0.723
	male, manly	female, feminine	0.118	0.121	-0.087
	male, masculine, man	she, woman, female	0.134	0.118	0.393
	male, boyish	female, feminine	0.206	0.121	0.825
hairpin	male, masculine, manly	she, womanly, girl	0.106	0.147	-1.282
	male, masculine	female, feminine	0.084	0.097	-0.954
	male, masculine, man	she, woman, female	0.116	0.129	-0.338
	male, manly	female, feminine	0.110	0.097	0.421
	male, boyish	female, feminine	0.148	0.097	0.821

Table 3.1: By contriving the 'male' and 'female' attribute words, we can manipulate the WEFAT statistic to claim that any target word is female-biased (i.e., negative statistic), male-biased (i.e., positive statistic), or gender-neutral. For example, the values for 'door' (top row) range from -1.425 (strong female bias) to 1.582 (strong male bias).

attribute sets to claim that any given target word is female-biased, male-biased, or gender-neutral.

Broadly speaking, cosine similarity is a useful measure of embedding similarity and hypothesis tests are useful for testing differences between samples. Because of this, WEAT and WEFAT seem, at first glance, to be intuitive. However, as shown in Propositions 1 and 2, there are two key theoretical flaws to WEAT and WEFAT that exaggerate the degree of association and ultimately make them inappropriate for word embeddings. The only other metric of note quantifies association as $|\cos(\vec{w},\vec{b})|^c$, where $\vec{b}$ is the bias subspace and $c \in \mathbb{R}$ the "strictness" of the measurement [Bolukbasi et al., 2016]. For the same reason discussed in Proposition 1, this too is exaggerative.

3.5 Relational Inner Product Association

Given the theoretical flaws of WEAT and WEFAT, we derive a new measure of word embedding association using the subspace projection method, which provably debiases embeddings (section 3.3).

Definition 2 *The* relational inner product association $\beta(\vec{w};\vec{b})$ *of a word vector* $\vec{w} \in V$ *with respect to a relation vector* $\vec{b} \in V$ *is* $\langle \vec{w},\vec{b} \rangle$. *Where S is a non-empty set of ordered word pairs (x,y) that define the association, $\vec{b}$ is the first principal component of* $\{\vec{x}-\vec{y} \mid (x,y) \in S\}$.

Our metric, the *relational inner product association* (RIPA), is simply the inner product of a relation vector describing the association and a given word vector in the same embedding space. To use the terminology in [Bolukbasi et al., 2016], RIPA is the scalar projection of a word vector onto a one-dimensional bias subspace defined by the unit vector $\vec{b}$. When $\vec{w}$ is unbiased with respect to S (by Definition 1), the projection of $\vec{w}$ on the subspace is $\vec{0}$, meaning that RIPA will also be 0. In their experiments, [Bolukbasi et al., 2016] defined $\vec{b}$ as the first principal component for a set of ten gender difference vectors (e.g., $\vec{man} - \vec{woman}$). This would be the means of deriving $\vec{b}$ for RIPA as well. As we show in the rest of this section, the interpretability of RIPA, its robustness to how the relation vector is defined, and its derivation from a method that provably debiases word embeddings are the key reasons why it is an ideal replacement for WEAT and WEFAT. Given that RIPA can be used for any embedding model that implicitly does matrix factorization, it is applicable to most common embedding models, such as SGNS and GloVe.

3.5.1 Interpreting RIPA

If only a single word pair (x,y) defines the association, then the relation vector $\vec{b} = (\vec{x}-\vec{y})/\|\vec{x}-\vec{y}\|$, making RIPA highly interpretable. Given that RIPA is for embedding models that factorize a matrix M containing a co-occurrence statistic (e.g., the shifted word-context PMI matrix for SGNS), if we assume that there is no reconstruction error, we can rewrite $\beta(\vec{w};\vec{b})$ in terms of M. Where x and y have context vectors $\vec{x}_c$ and $\vec{y}_c$, $\lambda \in \mathbb{R}$ is such that $C = \lambda W$ (see Lemma 2, [Ethayarajh et al., 2018]), $\delta' \in \mathbb{R}^-$ is a model-specific constant, and there is no reconstruction error:

$$\begin{aligned}
\beta_{\text{SGNS}}(\vec{w};\vec{b}) &= \frac{(1/\lambda)\langle \vec{w}, \vec{x}_c - \vec{y}_c \rangle}{\|\vec{x}-\vec{y}\|} \\
&= \frac{(1/\lambda)(\text{PMI}(x,w) - \text{PMI}(y,w))}{\sqrt{(1/\lambda)(-\text{csPMI}(x,y)+\delta')}} \\
&= \frac{1/\sqrt{\lambda}}{\sqrt{-\text{csPMI}(x,y)+\delta'}} \log \frac{p(w|x)}{p(w|y)}
\end{aligned} \tag{3.3}$$

Here, $\text{csPMI}(x,y) \triangleq \text{PMI}(x,y) + \log p(x,y)$ and is equal to $-\lambda\,\|\vec{x}-\vec{y}\|_2^2 + \delta'$ under perfect reconstruction [Ethayarajh et al., 2018]. There are three notable features of this result:

1. [Ethayarajh et al., 2018] proved the conjecture by [Pennington et al., 2014] that a word analogy holds over a set of words pairs (x,y) iff for every word w, $\log[p(w|x)/p(w|y)]$ is the same for every word pair (x,y). The expression in (3.3) is a multiple of this term.

2. Assuming no reconstruction error, if a linear word analogy holds over a set of ordered word pairs (x,y), then the co-occurrence shifted PMI (csPMI) should be the same for every word pair [Ethayarajh et al., 2018]. The more x and y are unrelated, the closer that $\text{csPMI}(x,y)$ is to $-\infty$ and $\beta(\vec{w};\vec{b})$ is to 0. This prevents RIPA from exaggerating the extent of the association simply because x and y are far apart in embedding space.

3. Because $\vec{b}$ is a unit vector, $\beta(\vec{w};\vec{b})$ is bounded in $[-\|\vec{w}\|, \|\vec{w}\|]$. This means that one can calculate a word's association with respect to multiple relation vectors and then compare the resulting RIPA values.

These points highlight just how robust RIPA is to the definition of $\vec{b}$. As long as a word analogy holds over the word pairs that define the association – i.e., as long as the word pairs have roughly the same difference vector

– the choice of word pair does not affect $\log[p(w|x)/p(w|y)]$ or $\text{csPMI}(x,y)$. Using ('king', 'queen') instead of ('man', 'woman') to define the gender relation vector, for example, would have a negligible impact. In contrast, as shown in section 3.4, the lack of robustness of WEAT and WEFAT to the choice of attribute sets is one reason they are so unreliable.

We can also interpret $\beta(\vec{w};\vec{b})$ for other embedding models, not just SGNS. Where $X_{x,y}$ denotes the frequency of a word pair (x,y) and z_x, z_y denote the learned bias terms for GloVe:

$$\begin{gathered}\beta_{\text{GloVe}}(\vec{w};\vec{b}) = C\left(\log\frac{p(x,w)}{p(y,w)} - z_x + z_y\right)\\ \text{where } C = \frac{1/\sqrt{\lambda}}{\sqrt{-\text{csPMI}(x,y)+\delta'}}\end{gathered} \tag{3.4}$$

Because the terms z_x, z_y are learned, $\beta(\vec{w};\vec{b})$ is not as interpretable for GloVe. However, [Levy et al., 2015] have conjectured that, in practice, z_x, z_y may be equivalent to the log counts of x and y respectively, in which case $\beta_{\text{GloVe}} = \beta_{\text{SGNS}}$.

3.5.2 Statistical Significance

Unlike WEAT and WEFAT, there is no notion of statistical significance attached to RIPA. There is a simple reason for this. Whether a word vector $\vec{w}$ is spuriously or non-spuriously associated with respect to a relation vector $(\vec{x}-\vec{y})/\|\vec{x}-\vec{y}\|$ depends on how frequently (w,x) and (w,y) co-occur in the training corpus; the more co-occurrences there are, the less likely the association is spurious. As shown in experiments by [Ethayarajh et al., 2018], the reconstruction error for any word pair (x,y) follows a zero-centered normal distribution where the variance is a decreasing function of $X_{x,y}$. Word embeddings alone are thus not enough to ascribe a statistical significance to the association. This also implies that the notion of statistical significance in WEAT and WEFAT is disingenuous, as it ignores how the spuriousness of an association depends on co-occurrence frequency in the training corpus.

3.6 Experiments

With our experiments, we address two open questions. For one, how much of the gender association in an embedding space is due to the embedding model itself, how much is due to the training corpus, and how much is just noise? Second, how can we debias gender-biased words (e.g., 'doctor', 'nurse') but not gender-appropriate ones (e.g., 'king', 'queen') without *a priori* knowledge of which words belong in which category?

Word Type	Word	Genderedness in Corpus	Genderedness in Embedding Space	Change (abs.)
Gender-Appropriate (n = 164)	mom	−0.163	−0.648	0.485
	dad	0.125	0.217	0.092
	queen	−0.365	−0.826	0.462
	king	0.058	0.200	0.142
	Avg (abs.)	**0.231**	**0.522**	**0.291**
Gender-Biased (n = 68)	nurse	−0.190	−1.047	0.858
	doctor	−0.135	−0.059	−0.077
	housekeeper	−0.132	−0.927	0.795
	architect	−0.063	0.162	0.099
	Avg (abs.)	**0.253**	**0.450**	**0.197**
Gender-Neutral (n = 200)	ballpark	0.254	0.050	−0.204
	calf	−0.039	0.027	−0.012
	hormonal	−0.326	−0.551	0.225
	speed	0.036	−0.005	−0.031
	Avg (abs.)	**0.125**	**0.119**	**−0.006**

Table 3.2: On average, SGNS makes gender-appropriate words (e.g., 'queen') and gender-biased words (e.g., 'nurse') *more* gendered in the embedding space than they are in the training corpus. As seen in the last column (in bold), the average change in absolute genderedness for these categories is 0.291 and 0.197 respectively. However, for gender-neutral words, the average change is −0.006: SGNS does not make the words any more gendered.

3.6.1 Setup

For our experiments, we use SGNS embeddings trained on Wikipedia, since RIPA is highly interpretable for SGNS (see section 3.5). This means that for any given word in the vocabulary, we can compare its gender association in the training corpus to its gender association in the embedding space, which should be equal under perfect reconstruction. Words are grouped into three categories with respect to gender: *biased*, *appropriate*, and *neutral*. We create lists of biased and appropriate words using the [Bolukbasi et al., 2016] lists of gender-biased and gender-appropriate analogies. For example, *doctor:nurse::man:woman* is biased, so we classify the first two words as biased. The last category, *neutral*, contains uniformly randomly sampled words that appear at least 10K times in the corpus and that are not in either of the other categories, and which we therefore expect to be gender-agnostic.

3.6.2 Breaking down Gender Association

For any given word, the gender association in the training corpus is what the gender association in the embedding space would be if there were no reconstruction error. By comparing these two quantities, we can infer the change induced by the embedding model. Let $g(w;x,y)$ denote the RIPA of a word w with respect to the gender relation vector defined by word pair (x,y), let $\hat{g}(w;x,y)$ denote what $g(w;x,y)$ would be under perfect reconstruction for an SGNS embedding model, and let Δ_g denote the change in absolute gender association from corpus to embedding space. Where S is a set of gender-defining word pairs (e.g., ('man', 'woman')) and λ, δ' are the model-specific constants defined in section 3.5,

$$\begin{aligned}
g(w;x,y) &= \frac{\langle \vec{w}, \vec{x}-\vec{y} \rangle}{\|\vec{x}-\vec{y}\|} \\
\hat{g}(w;x,y) &= \frac{1/\sqrt{\lambda}}{\sqrt{-\text{csPMI}(x,y)+\delta'}} \log \frac{p(w|x)}{p(w|y)} \\
\Delta_g(w;S) &= \left| \sum_{(x,y)\in S} \frac{g(w;x,y)}{|S|} \right| - \left| \sum_{(x,y)\in S} \frac{\hat{g}(w;x,y)}{|S|} \right|
\end{aligned} \tag{3.5}$$

We take the absolute value of each term because the embedding model may make a word more gendered, but in the direction opposite of what is implied in the corpus. $\lambda \leftarrow 1$ because we expect $\lambda \approx 1$ in practice [Ethayarajh et al., 2018, Mimno and Thompson, 2017]. Similarly, $\delta' \leftarrow 1$ because it minimizes the difference between $\|\vec{x}-\vec{y}\|$ and its information theoretic interpretation over the gender-defining word pairs in S, though this is an estimate and may differ from the true value of δ'. The set S itself is taken from [Bolukbasi et al., 2016]. In Table 3.2, we list the gender association in the training corpus ($g(w)$), the gender association in embedding space ($\hat{g}(w)$), and the absolute change ($\Delta_g(w)$) for each group of words.

On average, the SGNS embedding model does not make gender-neutral words any more gendered than they are in the training corpus. Given that much of the vocabulary falls into this category, this means that the embedding model does not systematically change the genderedness of most words. However, because of reconstruction error, individual words may be more or less gendered in the embedding space, simply due to chance. In contrast, for words that are either gender-biased or gender-appropriate, on average, the embedding model actually amplifies the gender association in the corpus. For example, for the word 'king', which is gender-specific by definition, the association is 0.058 in the corpus and 0.200 in the embedding space – it becomes more male-associated. For the word 'nurse', which is gender-biased, the association is -0.190 in the corpus and -1.047 in the embedding space – it becomes more female-associated. On average, the amplification is much greater for gender-appropriate words than it is for gender-biased ones, although the latter are more gendered in the corpus itself.

This amplification effect is unsurprising and can largely be explained by second-order similarity. Two words can be nearby in a word embedding space if they co-occur frequently in the training corpus (first-order similarity) or if there exists a large set of context words with which they both frequently co-occur (second-order similarity). The latter explains why words like 'Toronto' and 'Melbourne' are close to each other in embedding space; both are cities that appear in similar contexts. In an environment with some reconstruction error, such as low-dimensional embedding spaces, second-order similarity permits words to be closer in embedding space than would be the case if only first-order similarity had an effect. As a result, $\langle \vec{king}, \vec{man} \rangle > (1/\lambda)(\text{PMI}(king, man) - \log k)$ for SGNS, for example. What is often treated as a useful property of word embeddings can have, with respect to gender bias, a pernicious effect.

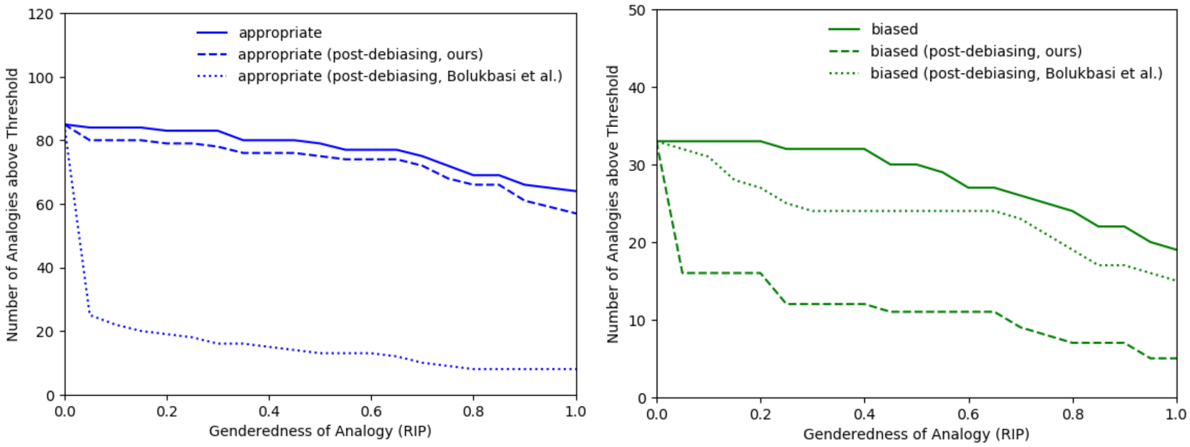

Figure 3.1: The [Bolukbasi et al., 2016] method of identifying gender-appropriate words – to avoid debiasing them – is ineffective: it ends up precluding most gender-appropriate analogies (dotted line, left) while preserving most gender-biased analogies (dotted line, right). Our unsupervised method (dashed line) does much better in both respects. The x-axis is a threshold for how gendered the first word pair (x,y) in an analogy is (i.e., $|\beta(\vec{x}-\vec{y};\vec{b})|$).

3.6.3 Debiasing without Supervision

To use the subspace projection method [Bolukbasi et al., 2016], one must have prior knowledge of which words are gender-appropriate, so that they are not debiased. Debiasing all vectors can preclude gender-appropriate analogies such as *king:queen: :man:woman* from holding in the embedding space. To create an exhaustive list of gender-appropriate words, [Bolukbasi et al., 2016] started with a small, human-labelled set of words and then trained an SVM to predict more gender-appropriate terms in the vocabulary. This bootstrapped list of gender-appropriate words was then left out during debiasing.

The way in which [Bolukbasi et al., 2016] evaluated their method is unorthodox: they tested the ability of their debiased embedding space to generate new analogies. However, this does not capture whether gender-appropriate analogies are successfully preserved and gender-biased analogies successfully precluded. In Figure 3.1, we show how the number of appropriate and biased analogies changes after debiasing. The x-axis captures how strongly gendered the analogy is, using the absolute RIPA value $|\beta(\vec{w};\vec{b})|$ but replacing $\vec{w}$ with the difference vector defined by the first word pair (e.g., $\vec{king}-\vec{queen}$). The y-axis captures the number of analogies that meet that threshold. As seen in Figure 3.1, using [Bolukbasi et al., 2016]'s bootstrapped list of gender-appropriate words yields the opposite of what is intended: it is much better at preserving biased analogies and precluding appropriate ones.

We propose an unsupervised alternative. We first create a gender-defining relation vector $\vec{b}^*$ by taking the first principal component of gender-defining difference vectors such as $\vec{man}-\vec{woman}$. Using difference vectors from biased analogies, such as $\vec{doctor}-\vec{midwife}$, we then create a gender-bias relation vector $\vec{b}'$ the same way. We then debias a word w iff it satisfies $|\beta(\vec{w};\vec{b}^*)| < |\beta(\vec{w};\vec{b}')|$. As seen in Figure 3.1, this simple condition is sufficient to preserve almost all gender-appropriate analogies while precluding most gender-biased ones. In our debiased embedding space, 94.9% of gender-appropriate analogies with a strength of at least 0.5 are preserved while only 36.7% of gender-biased analogies are. In contrast, the [Bolukbasi et al., 2016] method preserves only 16.5% of appropriate analogies with a strength of at least 0.5 while preserving 80.0% of biased ones. Combining our simple heuristic with other approaches may yield even better results, which we leave as future work.

3.7 Conclusion

In this paper, we answered several open questions about gender bias in word embeddings, and word associations more broadly. We proved that when there is no reconstruction error, for any embedding model that does matrix factorization (e.g., SGNS, GloVe), the subspace projection method [Bolukbasi et al., 2016] provably debiases embeddings. This was the first theoretical guarantee of this method, which is popular due to its empirical success. We then proved that WEAT and WEFAT, the most common tests of word embedding association, have theoretical flaws that exaggerate the extent of gender bias. By contriving the attribute sets for WEFAT, virtually any word can be classified as gender-biased. We then derived a new measure of association in word embeddings called the relational inner product association (RIPA). Using RIPA, we found that SGNS does not, on average, make most words any more gendered than they are in the training corpus. However, for words that are gender-biased or gender-specific by definition, SGNS actually amplifies the genderedness in the corpus.

Chapter 4

Unsupervised Random Walk Sentence Embeddings: A Strong but Simple Baseline

Kawin Ethayarajh

4.1 Introduction

Distributed representations of words, better known as word embeddings, have become fixtures of current methods in natural language processing. Word embeddings can be generated in a number of ways [Bengio et al., 2003, Collobert and Weston, 2008, Pennington et al., 2014, Mikolov et al., 2013b] by capturing the semantics of a word using the contexts it appears in. Recent work has tried to extend that intuition to sequences of words, using methods ranging from a weighted average of word embeddings to convolutional, recursive, and recurrent neural networks [Le and Mikolov, 2014, Kiros et al., 2015, Luong et al., 2013, Tai et al., 2015]. Still, [Wieting et al., 2016b] found that these sophisticated architectures are often outperformed, particularly in transfer learning settings, by sentence embeddings generated as a simple average of tuned word embeddings.

[Arora et al., 2017b] provided a more powerful approach: compute the sentence embeddings as weighted averages of word embeddings, then subtract from each one the vector projection on their first principal component. The weighting scheme, *smoothed inverse frequency* (SIF), is derived from a random walk model where words in a sentence s are produced by the random walk of a latent discourse vector c_s. A word unrelated to c_s can be produced by chance or if it is part of frequent discourse such as stopwords. This approach evens outperforms more complex models such as LSTMs on textual similarity tasks. Arora et al. argued that the simplicity and effectiveness of their method make it a tough-to-beat baseline for sentence embeddings. Though they call their approach unsupervised, others have noted that it is actually ‘weakly supervised’, since it requires hyperparameter tuning [Cer et al., 2017].

In this paper, we first propose a class of worst-case scenarios for Arora et al.’s random walk model. Specifically, given some sentence g that is dominated by words with zero similarity, and some sentence h that is dominated by identical words, we show that their approach can return two discourse vectors c_g and c_h such that $p(g|c_g) \approx p(h|c_h)$, provided that the word vectors for g have a sufficiently greater length than those for h. In other words, word vector length has a confounding effect on the probability of a sentence being generated, and this effect can be strong enough to yield completely unintuitive results. This problem is not endemic to these scenarios, though they are the most illustrative of it; because of the underlying log-linear word production model, Arora et al.’s model is fundamentally sensitive to word vector length.

Our contributions in this paper are three-fold. First, we propose a random walk model that is robust to distortion by vector length, where the probability of a word vector being generated by a discourse vector is inversely related to the angular distance between them. Second, we derive a weighting scheme from this model and compute a MAP estimate for the sentence embedding as follows: normalize the word vectors, take a weighted average of them, and then subtract from each weighted average vector the projection on their first m principal components. We call the weighting scheme derived from our random walk model *unsupervised smoothed inverse frequency* (uSIF). It is similar to SIF [Arora et al., 2017b] in practice, but requires no hyperparameter tuning at all – it is completely unsupervised, allowing it to be used when there is no labelled data. Lastly, we show that our approach outperforms Arora et al.’s by up to 44.4% on textual similarity tasks, and is even competitive with state-of-the-art methods. Given the simplicity, effectiveness, and unsupervised nature of our method, we suggest it be used as a baseline for computing sentence embeddings.

4.2 Related Work

Word Embeddings Word embeddings are distributed representations of words, typically in a low-dimensional continuous space. These word vectors can capture semantic and lexical properties of words, even allowing some relationships to be captured algebraically (e.g., $v_{\text{Berlin}} - v_{\text{Germany}} + v_{\text{France}} \approx v_{\text{Paris}}$) [Mikolov et al., 2013b]. Word embeddings are generally obtained in two ways: (a) from internal representations of words in shallow neural networks [Bengio et al., 2003, Mikolov et al., 2013b, Collobert and Weston, 2008]; (b) from low rank approximations of co-occurrence matrices [Pennington et al., 2014].

Word Sequence Embeddings Embeddings for sequences of words (e.g., sentences) are created by composing word embeddings. This can be done simply, by doing coordinate-wise multiplication [Mitchell and Lapata, 2008] or taking an unweighted average [Mikolov et al., 2013b] of the word vectors. More sophisticated architectures can also be used: for instance, recursive neural networks [Socher et al., 2011, Socher et al., 2013], LSTMs [Tai et al., 2015], and convolutional neural networks [Kalchbrenner et al., 2014] can be defined and trained on parse and dependency trees.

Other approaches are based on the presence of a latent vector for the entire sequence. Paragraph vectors [Le and Mikolov, 2014] are latent representations that influence the distribution of words. Skip-thought vectors [Kiros et al., 2015] are hidden representations of a neural network that encodes a sentence by trying to reconstruct its surrounding sentences. [Conneau et al., 2017] leverage transfer learning by using the hidden representation of a sentence in an LSTM trained for another task, such as textual entailment. The inspiration for [Arora et al., 2017b] is [Wieting et al., 2016b], who use word averaging after updating word embeddings by tuning them on paraphrase pairs. A later work [Wieting and Gimpel, 2017a] tried trigram-averaging and LSTM-averaging in addition to word-averaging. In that approach, vectors were tuned on the ParaNMT-50M dataset, created by using neural machine translation to translate 51M Czech-English sentence pairs into English-English pairs. This yielded state-of-the-art results on textual similarity tasks, beating the previous baseline by a wide margin.

4.3 Approach

4.3.1 The Log-Linear Random Walk Model

In Arora et al.'s original model [Arora et al., 2016], words are generated dynamically by the random walk of a time-variant discourse vector $c_t \in \mathbb{R}^{\mathrm{d}}$, representing "what is being talked about". Words are represented as $v_w \in \mathbb{R}^{\mathrm{d}}$. The probability of a word w being generated at time t is given by a log-linear production model [Mnih and Hinton, 2007]:

$$p(w|c_t) \propto \exp\left(\langle c_t, v_w \rangle\right) \tag{4.1}$$

Assuming that the discourse vector c_t does not change much over the course of the sentence, Arora et al. replace the sequence of discourse vectors $\{c_t\}$ across all time steps with a single discourse vector c_s. The MAP estimate of c_s is then the unweighted average of word vectors (ignoring any scalar multiplication).

Arora et al.'s improved random walk model [Arora et al., 2017b] allows words to also be generated: (a) by chance, with probability $\alpha \cdot p(w)$, where α is some scalar and $p(w)$ is the frequency; (b) if the word is correlated with the common discourse vector, which represents frequent discourse such as stopwords. We use c_0 to denote the common discourse vector, to be consistent with the literature. Among other things, these changes help explain words that appear frequently despite being poorly correlated with the discourse vectors — words like *the*, for example. The probability of a word w being generated by a discourse vector c_s is then given as:

$$p(w|c_s) = \alpha \cdot p(w) + (1-\alpha) \cdot \frac{\exp\left(\langle \widetilde{c_s}, v_w \rangle\right)}{Z_{\widetilde{c_s}}},$$

$$\text{where } \widetilde{c_s} \triangleq \beta \cdot c_0 + (1-\beta) \cdot c_s, c_0 \perp c_s$$

$$Z_{\widetilde{c_s}} \triangleq \sum_{w' \in \mathcal{V}} \exp\left(\langle \widetilde{c_s}, v_{w'} \rangle\right)$$

where α, β are scalar hyperparameters, $\mathcal{V}$ is the vocabulary, $\widetilde{c_s}$ is a linear combination of the discourse and common discourse vectors parameterized by β, and $Z_{\widetilde{c_s}}$ is the partition function.

The sentence embedding for a sentence is defined as the MAP estimate of the discourse vector c_s that generated the sentence. To compute this tractably, Arora et al. assume that word vectors v_w are roughly uniformly dispersed in the latent space. This implies that the partition function $Z_{\widetilde{c_s}}$ is roughly the same for all $\widetilde{c_s}$, allowing it to be replaced with a constant Z. Assuming a uniform prior over $\widetilde{c_s}$, the maximum likelihood estimator for $\widetilde{c_s}$ on the unit sphere (ignoring normalization) is then approximately proportional to:

$$\frac{1}{|s|} \sum_{w \in s} \frac{a}{a + p(w)} \cdot v_w, \text{where } a \triangleq \frac{1-\alpha}{\alpha \cdot Z}$$

Since Z cannot be evaluated, and α is not known, a is a hyperparameter that needs tuning. This weighting scheme is called *smoothed inverse frequency* (SIF) and places a lower weight on more frequent words. The first principal component of all $\{\widetilde{c_s}\}$ in the corpus is used as the estimate for the common discourse vector c_0. The final discourse vector c_s is then produced by subtracting the projection of the weighted average on the common component (*common component removal*):

$$c_s \triangleq \widetilde{c_s} - \text{proj}_{c_0} \widetilde{c_s}$$

Arora et al. call their approach unsupervised, but others [Cer et al., 2017] have correctly noted that it is weakly supervised, since the hyperparameter a needs to be tuned on a validation set.

4.3.2 The Confounding Effect of Vector Length

We now propose worst-case scenarios where word vector length clearly distorts $p(s|c_s)$ due to the underlying log-linear word production model. Note that we discuss these scenarios because they are illustrative, not because they circumscribe the universe of all scenarios in which word vector length has a confounding effect.

Consider a sentence g comprising two rare words x and y, where x and y have zero similarity. Also consider some sentence h, where the only word z appears twice. g might not occur naturally, but its weighted average $\widetilde{c_g}$ would be similar to that of some longer sentence where x,y are the only non-stopwords (i.e., those with non-negligible weight). For simplicity, further assume that common component removal has negligible effect:

$$\begin{aligned}\langle v_x, v_y \rangle &= 0 \\ c_g = \widetilde{c_g} &= \frac{1}{2}\left(\frac{a}{a+p(x)} \cdot v_x + \frac{a}{a+p(y)} \cdot v_y\right) \\ c_h = \widetilde{c_h} &= \frac{a}{a+p(z)} \cdot v_z\end{aligned} \tag{4.2}$$

Words x,y,z are so infrequent that the probability of them being produced by chance or by the common discourse vector is negligible; the likelihood of them being produced is therefore proportional to the inner product of the discourse and word vectors. Given that the words $x,y \in g$ have zero similarity, and given that the only word $z \in h$ is identical to its discourse vector, we would expect:

$$p(h|c_h) \gg p(g|c_g) \tag{4.3}$$

However, (4.3) does not always hold. Suppose that the word embeddings lie in $\mathbb{R}^2$. Then any scalar k can be used to create a valid set of assignments for word embeddings v_x, v_y, v_z that satisfy (4.2):

$$v_x = \begin{bmatrix} 2k \\ 0 \end{bmatrix}, v_y = \begin{bmatrix} 0 \\ 2k \end{bmatrix}, v_z = \begin{bmatrix} k \\ k \end{bmatrix} \tag{4.4}$$

Assuming the words x,y,z have roughly the same frequency, they should have the same SIF-weight. Then the weighted averages, and by extension the discourse vectors (4.2), are the same:

$$\begin{aligned} c_g = c_h &= \frac{a}{a+p(x)} \begin{bmatrix} k \\ k \end{bmatrix} \\ \Rightarrow \langle c_g, x \rangle = \langle c_g, y \rangle = \langle c_h, z \rangle &= \frac{a}{a+p(x)} \cdot 2k^2 \\ \Rightarrow p(g|c_g) = p(h|c_h) \end{aligned}$$

Thus it is possible for g to be generated by discourse vector c_g with roughly the same probability as h by c_h, contradicting (4.3). How is this possible, given that the words in g have zero similarity with each other while those in h are identical to each other? The answer can be found in the word vector lengths. Because $||v_x||_2 = \sqrt{2}||v_z||_2$, and $p(w|c_s)$ depends on the inner product of the word and discourse vectors (4.1), words with longer word vectors are more likely to be produced. In fact, if v_x and v_y were multiplied by some scalar greater than 1, then $p(h|c_h)$

would be less than $p(g|c_g)$.

Generalizing Worst-Case Scenarios By manipulating the word vector length, we can also come up with a more general class of assignments that can contradict (4.3):

$$v_x = \begin{bmatrix} \beta k_1 \sigma \\ \beta k_2 (1-\sigma) \end{bmatrix} v_y = \begin{bmatrix} \beta k_1 (1-\sigma) \\ \beta k_2 \sigma \end{bmatrix} v_z = \begin{bmatrix} k_1 \\ k_2 \end{bmatrix} \tag{4.5}$$

where $\sigma \in [0,1], \beta \in \mathbb{R}, \beta \geq 2$. For convenience, we replace $\frac{a}{a+p(x)}$ with C below:

$$c_g = C \begin{bmatrix} \frac{1}{2}\beta k_1 \\ \frac{1}{2}\beta k_2 \end{bmatrix}, c_h = C \begin{bmatrix} k_1 \\ k_2 \end{bmatrix}$$

For simplicity, we assume that the words x, y, z across the two sentences are so infrequent that the probability of them being generated by chance is zero. Then the conditional probabilities of the sentences being generated are:

$$\begin{aligned} p(g|c_g) &\propto \exp\left(\langle c_g, v_x \rangle + \langle c_g, v_y \rangle\right) \\ &= \exp\left(\frac{1}{2}\beta^2 C \left({k_1}^2 + {k_2}^2\right)\right) \\ p(h|c_h) &\propto \exp\left(\langle c_h, v_z \rangle + \langle c_h, v_z \rangle\right) \\ &= \exp\left(2C\left({k_1}^2 + {k_2}^2\right)\right) \\ \therefore \beta \geq 2 &\Rightarrow p(g|c_g) \geq p(h|c_h) \end{aligned} \tag{4.6}$$

In this general formulation, not all scenarios are worst-case. This describes a spectrum of scenarios ranging from acceptable (e.g., $v_x = v_y = v_z$ when $\beta = 2, \sigma = 0.5$) to completely counter-intuitive (see (4.4)). Though these assignments only apply for word vectors in $\mathbb{R}^2$, they can easily be extended to higher-dimensional spaces.

The confound of vector length persists for longer, naturally occurring sentences. Ultimately, the underlying log-linear word production model (4.1) means that words with longer word vectors are more likely to be generated. Because this confound is due to model design, rather than the MLE, removing it requires redesigning the model. The exact degree of the confound varies across sentences, but in theory, it is unbounded.

4.3.3 An Angular Distance–Based Random Walk Model

To address the confounding effect of word vector length, we propose a random walk model where the probability of observing a word w at time t is inversely related to the angular distance between the time-variant discourse vector $c_t \in \mathbb{R}^{\mathrm{d}}$ and the word vector $v_w \in \mathbb{R}^{\mathrm{d}}$:

$$\begin{aligned} p(w|c_t) &\propto 1 - \frac{\arccos\left(\cos\left(v_w, c_t\right)\right)}{\pi}, \\ \text{where } \cos\left(v_w, c_t\right) &\triangleq \frac{v_w \cdot c_t}{\|v_w\|_2 \cdot \|c_t\|_2} \end{aligned} \tag{4.7}$$

where $\arccos\left(\cos\left(v_w, c_t\right)\right)$ is the angular distance. For the intuition behind the use of this distance metric, note that the angular distance between two vectors is equal to the geodesic distance between them on the unit sphere. Thus the angular distance can also be interpreted as the length of the shortest path between the L_2 normalized word vector and the L_2 normalized discourse vector on the unit sphere. Since the angular distance lies in $[0, \pi]$, we divide it by π to bound it in $[0, 1]$. Our choice of angular distance – as opposed to, say, the exponentiated cosine similarity – is critical to avoiding hyperparameter tuning.

Assuming that the discourse vector c_t does not change much over the course of the sentence, the sequence of discourse vectors $\{c_t\}$ across all time steps can be replaced with a single discourse vector c_s for the sentence s. To model sentences more realistically, we allow words to be generated in two additional ways, as proposed in [Arora et al., 2017b]: (a) by chance, with probability $\alpha \cdot p(w)$, where α is some scalar and $p(w)$ is the frequency; (b) if the word is correlated with one of m common discourse vectors $\{c'_m\}$, which represent various types of frequent discourse, such as stopwords. The probability of a word w being generated by discourse vector c_s is then:

$$\begin{aligned} p(w|c_s) &= \alpha \cdot p(w) + (1-\alpha) \cdot \frac{d\left(\widetilde{c_s}, v_w\right)}{Z_{\widetilde{c_s}}}, \\ \text{where } \widetilde{c_s} &\triangleq (1-\beta)\, c_s + \beta \sum_{i=1}^{m} \lambda_i\, c'_i, \;\; c_s \perp c'_i \\ d\left(\widetilde{c_s}, v_w\right) &\triangleq 1 - \frac{\arccos\left(\cos\left(v_w, c_t\right)\right)}{\pi}, \\ Z_{\widetilde{c_s}} &\triangleq \sum_{w' \in \mathcal{V}} d\left(\widetilde{c_s}, v_{w'}\right) \end{aligned} \tag{4.8}$$

where $\alpha, \beta, \{\lambda_i\}$ are scalar hyperparameters, $\mathcal{V}$ is the vocabulary, $\widetilde{c_s}$ is a linear combination of the discourse and common discourse vectors parameterized by β and $\{\lambda_i\}$, and $Z_{\widetilde{c_s}}$ is the partition function. Instead of searching for the optimal hyperparameter values over some large space, as [Arora et al., 2017b] did, we make some simple assumptions to directly compute them.

We define the sentence embedding for some sentence s to be the MAP estimate of the discourse vector c_s that generates s. Assuming a uniform prior over possible c_s, the MAP estimate is also the MLE estimate for c_s. The log-likelihood of a sentence s is:

$$\log p(s|c_s) = \sum_{w \in s} \log p(w|c_s)$$

To maximize $\log p(s|c_s)$, we can approximate $\log p(w|c_s)$ using a first-degree Taylor polynomial:

$$\begin{aligned}
f_w(\widetilde{c_s}) &\triangleq \log p(w|\widetilde{c_s}) \\
\nabla f_w(\widetilde{c_s}) &= \left[\frac{1-\alpha}{\pi \cdot Z_{\widetilde{c_s}} \cdot \exp\left(f_w(\widetilde{c_s})\right)}\right] \frac{\frac{\partial}{\partial \widetilde{c_s}} \cos\left(v_w, \widetilde{c_s}\right)}{\sqrt{1-\cos^2\left(v_w, \widetilde{c_s}\right)}}, \\
\frac{\partial}{\partial \widetilde{c_s}} \cos &= \frac{v_w}{\|v_w\|_2 \cdot \|\widetilde{c_s}\|_2} - \cos\left(v_w, \widetilde{c_s}\right) \frac{\widetilde{c_s}}{\|\widetilde{c_s}\|_2^2}
\end{aligned}$$

Where $a \triangleq (1-\alpha)/(\alpha Z_{\widetilde{c_s}})$, C is a constant, and v'_w is a vector orthogonal to v_w with length $\|v_w\|^{-1}$:

$$\begin{aligned}
f_w(\widetilde{c_s}) &\approx f_w(v'_w) + \nabla f_w(v'_w)^{\intercal}\left(\widetilde{c_s} - v'_w\right) \\
&= C + \frac{a}{\pi \cdot \left(p(w) + \frac{1}{2} \cdot a\right)} \cdot v_w \left(\widetilde{c_s} - v'_w\right) \\
&= C + \frac{1}{\pi}\left(\frac{a}{p(w) + \frac{1}{2} \cdot a} \langle \widetilde{c_s}, v_w \rangle\right)
\end{aligned}$$

The MLE for $\widetilde{c_s}$ on the unit sphere (ignoring normalization) is then approximately proportional to:

$$\frac{1}{|s|} \sum_{w \in s} \frac{a}{p(w) + \frac{1}{2} a} \cdot v_w \tag{4.9}$$

The MLE of $\widetilde{c_s}$ is approximately a weighted average of word vectors, where more frequent words are downweighted. In fact, it very closely resembles the SIF weighting scheme [Arora et al., 2017b]! However, there are two key differences. For one, as we show later in this subsection, we have derived this weighting scheme from a model that is robust to the confounding effect of word vector length. Secondly, in SIF, a is a hyperparameter that needs to be tuned on a validation set. We now show that in our approach, we can calculate a directly as a function of the vocabulary $\mathcal{V}$ and the number of words in the sentence, $|s|$.

Normalization Before weighting the word vectors, we normalize them along each dimension: we construct a matrix $[v_{w_1} ... w_{w_{|s|}}]$ and take the L_2 norm of each row, which corresponds to a single dimension in $\mathbb{R}^{\mathrm{d}}$. We then multiply this d-dimensional vector element-wise with every vector in the sentence. This helps reduce the difference in variance across the dimensions.

Partition Function To calculate $Z_{\widetilde{c_s}}$, we borrow the key assumption from [Arora et al., 2017b] that the word vectors v_w are roughly uniformly dispersed in the latent space. Then the expected geodesic distance between a latent discourse vector and a word vector on the unit sphere is $\pi/2$, so $\mathbb{E}_{w' \in \mathcal{V}} \left[d\left(\widetilde{c_s}, v_{w'}\right) \right] = \frac{1}{2}$. Then:

$$\begin{aligned} Z_{\widetilde{c_s}} &= \sum_{w' \in \mathcal{V}} d\left(\widetilde{c_s}, v_{w'}\right) \\ &= |\mathcal{V}| \; \mathbb{E}_{w' \in \mathcal{V}} \left[d\left(\widetilde{c_s}, v_{w'}\right) \right] = \frac{1}{2} |\mathcal{V}| \end{aligned} \tag{4.10}$$

Odds of Random Production α is the probability that a word w will be produced by chance instead of by the discourse or common discourse vectors. To estimate α, we first consider the probability that a random word w will be produced by a discourse vector c_s at least once over n steps of a random walk:

$$\begin{aligned} p(w|c_s^1, ..., c_s^n) &= 1 - \prod_{t=1}^{n} \left[1 - \frac{d(c_s^t, v_w)}{Z_{c_s}} \right] \\ \mathbb{E}_{w \sim \mathcal{V}}[p(w|c_s^1, ..., c_s^n)] &= 1 - \left(1 - \frac{1}{|\mathcal{V}|} \right)^n \end{aligned}$$

The number of steps taken during the random walk is itself a random variable, so we let $n = \mathbb{E}_{s \in S}|s|$. We assume that if the frequency is greater than this expectation, then the word is always produced by chance; less than this expectation, and it is always produced by the discourse or common discourse vectors. α is the proportion of the vocabulary with $p(w)$ above this threshold:

$$\alpha = \frac{\sum_{w \in \mathcal{V}} \mathbb{1} \left[p(w) > \mathbb{E}_{w \sim \mathcal{V}}[p(w|c_s^1, ..., c_s^n)] \right]}{|\mathcal{V}|} \tag{4.11}$$

Since we can directly calculate $Z_{\widetilde{c_s}}$ and α, we can also directly calculate $a = (1-\alpha)/(\alpha Z_{\widetilde{c_s}})$.

Common Discourse Vectors We estimate the m common discourse vectors as the first m singular vectors from the singular value decomposition of the weighted average vectors. $\{\lambda_i\}$ are the weights on the common discourse vectors. In reality, these are unique to the word for which $p(w|c_s)$ is being evaluated. However, we let λ_i be:

$$\lambda_i = \frac{\sigma_i^2}{\sum_j^m \sigma_j^2}$$

where σ_i is the singular value for c_i'. λ_i can be interpreted as the proportion of variance explained by $\{c_1', ..., c_m'\}$ that is explained by c_i'. If removing the common discourse vectors is a form of denoising [Arora et al., 2017b], increasing m, in theory, should improve results. Because the variance explained by a singular vector falls with every additional vector that is included, the choice of m is thus a trade-off between variance explained and computational cost. When $m = 1$, this is equivalent to the removal in [Arora et al., 2017b]. We fix m at 5, since we find empirically that singular vectors beyond that do not explain much more variance. To get c_s, we subtract from

Algorithm 1 uSIF Sentence Embedding

Input: vocabulary $\mathcal{V}$, word vectors $\{v_w : w \in \mathcal{V}\}$, frequencies $\{p(w) : w \in \mathcal{V}\}$, sentences $\mathcal{S}$
Output: sentence embeddings $\{c_s : s \in \mathcal{S}\}$

```
procedure EMBED
    $m \leftarrow 5$
    $n \leftarrow \mathbb{E}_{s \in S}|s|$
    for all $s \in \mathcal{S}$ do
        $\alpha \leftarrow \frac{\sum_{w \in \mathcal{V}} \mathbb{1}\left[p(w) > 1 - \left(1 - \frac{1}{|\mathcal{V}|}\right)^n\right]}{|\mathcal{V}|}$
        $Z \leftarrow |\mathcal{V}|/2$
        $a \leftarrow (1-\alpha)/(\alpha \cdot Z)$
        $\widetilde{c_s} \leftarrow \frac{1}{|s|} \sum_{w \in s} \frac{a}{p(w) + \frac{1}{2}a} v_w$
    end for
    $\mathcal{A} \leftarrow \left(\widetilde{c_{s_1}} \ldots \widetilde{c_{s_n}}\right)$
    for all $i$ in 1...m do
        $c'_i \leftarrow i^{\text{th}}$ singular vector of $\mathcal{A}$
        $\sigma_i \leftarrow i^{\text{th}}$ singular value of $\mathcal{A}$
    end for
    for all $i$ in 1...m do
        $\lambda_i \leftarrow \frac{\sigma_i^2}{\sum_j^m \sigma_j^2}$
    end for
    for all $s \in \mathcal{S}$ do
        $c_s \leftarrow \widetilde{c_s} - \sum_{i=1}^m \lambda_i \, \text{proj}_{c'_i} \widetilde{c_s}$
    end for
end procedure
```

$\widetilde{c_s}$ the weighted projection on each singular vector:

$$c_s \triangleq \widetilde{c_s} - \sum_{i=1}^{m} \lambda_i \, \text{proj}_{c'_i} \widetilde{c_s}$$

We call this *piecewise common component removal*. Because our weighting scheme requires no hyperparameter tuning, it is completely unsupervised. For this reason, we call it *unsupervised smoothed inverse frequency* (uSIF). The full algorithm is given in Algorithm 1.

Note that while it is certainly *possible* to tune the hyperparameters in our model to achieve optimal results, it is not necessary to do so, which allows our method to be used when there is no labelled data. By contrast, in Arora et al.'s model [Arora et al., 2017b], hyperparameter tuning is a necessity.

Confound of Vector Length To understand why this model is not prone to the confound of word vector length, we reconsider the class of assignments for v_x, v_y, v_z in (4.5) and the resulting values for $\widetilde{c_g}$ and $\widetilde{c_h}$. Recall that in our example, sentence g comprises words x, y and sentence h comprises two instances of the word z. Under our new weighting scheme, C in (4.5) is replaced with $C' = \frac{a}{p(x) + \frac{1}{2}a}$. Note that we use $p(x)$ in C' because of the simplifying assumption that $p(x) = p(y) = p(z)$. Assuming again that $p(x) \approx 0$ and that piecewise common component removal has negligible effect, we can see how $p(g|c_g)$ and $p(h|c_h)$ change in our random walk model:

$$p(g|c_g) \propto \prod_{w \in \{x,y\}} \left(1 - \frac{\arccos(\cos(c_g, v_w))}{\pi}\right)$$
$$p(h|c_h) \propto \left(1 - \frac{\arccos(\cos(c_h, v_z))}{\pi}\right)^2 = 1$$

Because $p(g|c_g)$ is ultimately based on the cosine similarities between the discourse vector and word vectors, it is a function of the parameter $\sigma \in [0,1]$ that controls the degree of similarity between v_x and v_y. For example, for the worst-case assignments (4.4), $p(g|c_g) \propto 9/16$. Conversely, when $v_x = v_y = v_z$, we get $p(g|c_g) = p(h|c_h) \propto 1$. Recall that in Arora et al.'s model [Arora et al., 2017b], $\beta \geq 2$ was sufficient to ensure the counter-intuitive result of $p(g|c_g) \geq p(h|c_h)$ (4.6), where β was a scalar that controlled the word vector length. In contrast, in our random walk model, the effect of β – and thus the confound of vector length – is entirely absent; only the similarity between the word vectors is influential.

4.4 Results and Discussion

4.4.1 Textual Similarity Tasks

We test our approach on the SemEval semantic textual similarity (STS) tasks (2012-2015) [Agirre et al., 2012, Agirre et al., 2013, Agirre et al., 2014, Agirre et al., 2015], the SemEval 2014 Relatedness task (SICK'14) [Marelli et al., 2014], and the STS Benchmark dataset [Cer et al., 2017]. In these tasks, the goal is to determine the semantic similarity between a given pair of sentences; the evaluation criterion is the Pearson correlation coefficient between the predicted and actual similarity scores. To predict the similarity score, we simply encode each sentence and take the cosine similarity of their vectors. The individual scores for STS tasks are in Table 4.1 and the average scores are in Table 4.2. The STS benchmark scores are in Table 4.3. We compare our results with those from several methods, which are categorized by [Cer et al., 2017] as 'unsupervised', 'weakly supervised', or 'supervised'.

4.4.2 Experimental Settings

For a fair comparison with [Arora et al., 2017b], we use the unigram probability distribution used by them, based on the enwiki dataset (Wikipedia, 3B words). Our preprocessing of the sentences is limited to tokenization. We try our method with three types of word vectors: GloVe vectors [Pennington et al., 2014], PARAGRAM-SL999 (PSL) vectors [Wieting et al., 2015], tuned on the SimLex999 dataset, and ParaNMT-50 vectors [Wieting and Gimpel, 2017a], tuned on 51M English-English sentence pairs translated from English-Czech sentence pairs. The value of n in (4.11) is $\mathbb{E}_{s \in S}|s| \approx 11$ and was estimated using sentences from all corpora. The value of a in (4.9) is then $1.2 \times$

	[Wieting and Gimpel, 2017b]			[Wieting et al., 2016b]			[Arora et al., 2017b]		Our Approach		
Supervision	Weakly Supervised			Unsupervised			Weakly Supervised		Unsupervised		
Tasks	LSTM AVG	AVG	GRAN	PP-XXL	tfidf-GloVe	skip-thought	GloVe+WR	PSL+WR	GloVe+UP	PSL+UP	Czeng+UP
MSRpar	49.0	45.9	47.7	44.8	50.3	16.8	35.6	43.3	51.4	55.2	**58.3**
MSRvid	84.3	85.1	85.2	79.6	77.9	41.7	83.8	84.1	86.4	85.4	**90.1**
SMT-eur	51.2	47.5	49.3	49.5	54.7	35.2	49.9	44.8	**54.9**	51.3	54.7
OnWN	71.5	71.2	71.5	70.4	64.7	29.7	66.2	71.8	72.7	**74.3**	74.2
SMT-news	**68.0**	58.2	58.7	63.3	45.7	30.8	45.6	53.6	59.3	62.8	64.1
STS'12	64.8	61.6	62.5	61.5	58.7	30.8	56.2	59.5	64.9	65.8	**68.3**
headlines	77.3	76.9	76.1	73.9	69.2	34.6	69.2	74.1	74.8	77.7	**81.2**
OnWN	81.2	72.8	81.4	73.8	72.9	10.0	82.8	82.0	85.5	85.4	**87.5**
FNWN	53.2	50.2	55.6	47.7	36.6	30.4	39.4	52.4	55.0	**56.9**	53.6
SMT	40.7	38.0	40.3	40.4	29.6	24.3	37.9	38.5	39.1	41.0	**42.0**
STS'13	63.1	59.4	63.4	58.9	52.1	24.8	56.6	61.8	63.6	65.2	**66.1**
deft forum	56.6	55.6	55.7	53.4	37.5	12.9	41.2	51.4	52.3	55.0	**58.5**
deft news	78.0	**78.5**	77.1	74.4	68.7	23.5	69.4	72.6	74.6	74.5	78.4
headlines	74.5	75.1	72.8	71.5	63.7	37.8	64.7	70.1	69.7	73.3	**78.7**
images	84.7	85.6	85.8	80.4	72.5	51.2	82.6	84.8	84.3	84.9	**87.1**
OnWN	84.9	81.4	85.1	81.5	75.2	23.3	82.8	84.5	86.9	87.6	**88.9**
tweet news	76.3	78.7	78.7	77.4	65.1	39.9	70.1	77.5	78.5	**80.0**	78.8
STS'14	75.8	75.8	75.9	73.1	63.8	31.4	68.5	73.5	74.4	75.9	**78.4**
answers-forum	71.8	70.6	**73.1**	69.1	45.6	36.1	63.9	70.1	72.4	72.2	72.0
answers-student	71.1	75.8	72.9	**78.0**	63.9	33.0	70.4	75.9	72.7	75.1	73.5
belief	75.3	76.8	78.0	78.2	75.3	24.6	71.8	75.3	76.5	77.3	**79.1**
headline	79.5	80.3	78.6	76.4	70.9	43.6	70.7	75.9	75.9	78.8	**83.1**
images	85.8	86.0	85.8	83.4	72.9	17.7	81.5	84.1	83.0	84.6	**87.5**
STS'15	76.7	77.9	77.7	77.0	60.6	31.0	71.7	76.3	76.1	77.6	**79.0**
SICK14	71.3	72.4	72.9	72.7	69.4	49.8	72.2	72.9	73.0	72.3	**73.5**

Table 4.1: Results (Pearson's $r \times 100$) on textual similarity tasks. The highest score in each row is in bold. "GloVe+UP" is the application of uSIF-weighting (U) and piecewise common component removal (P) to GloVe word vectors; "PSL+UP" to PSL word vectors; "ParaNMT+UP" to ParaNMT word vectors.

10^{-3}. Our results are denoted as **X+UP**, where $\mathbf{X} \in \{\text{`GloVe'}, \text{`PSL'}, \text{`ParaNMT'}\}$, **U** denotes uSIF-weighting, and **P** denotes piecewise common component removal.

4.4.3 Results

Our model outperforms Arora et al.'s by up to 44.4% on individual tasks (see GloVe+UP vs. GloVe+WR for the STS'12 MSRpar task in Table 4.1) and by up to 15.5% on yearly averages (see GloVe+UP vs. GloVe+WR for STS'12 in Table 4.2). Our approach proves most useful in cases where Arora et al. underperform others, such as for STS'12, where our models – GloVe+UP and PSL+UP – outperform their equivalents in Arora et al.'s results by 15.5% and 10.6% respectively. On average, our approach outperforms Arora et al.'s by around 7.6%, but the improvement is highly variable. This may be because the hyperparameter values we derived may be closer to the optima for some corpora more than others or because our other improvements – normalization and piecewise common component removal – are more effective for certain datasets.

Our best model, ParaNMT+UP, is also competitive with the state-of-the-art model, ParaNMT Trigram-Word, an average of trigram and word embeddings tuned on the ParaNMT-dataset. ParaNMT+UP outperforms ParaNMT Trigram-Word on STS'12, STS'13, and STS'14; it is narrowly outperformed on STS'15 and the STS benchmark. ParaNMT Trigram-Word's inclusion of trigram embeddings gives it an edge over our model for out-of-vocabulary words [Wieting and Gimpel, 2017a]. It should be noted that ParaNMT+UP outperforms both ParaNMT Word Avg. and ParaNMT BiLSTM Avg., implying that our model composes words better than both simple averaging and BiLSTMs. Similarly, our model PSL+UP outperforms PP-XXL [Wieting et al., 2016b], despite the latter using the same word vectors and a learned projection instead.

Model	STS'12	STS'13	STS'14	STS'15	SICK14
[Wieting et al., 2016b] - unsupervised					
PP	58.7	55.8	70.9	75.8	71.6
PP-XXL	61.5	58.9	73.1	77.0	72.7
tfidf-GloVe	58.7	52.1	63.8	60.6	69.4
skip-thought	30.8	24.8	31.4	31.0	49.8
[Arora et al., 2017b] - weakly supervised					
GloVe+WR	56.2	56.6	68.5	71.7	72.2
PSL+WR	59.5	61.8	73.5	76.3	72.9
[Wieting and Gimpel, 2017b] - weakly supervised					
LSTM AVG	64.8	63.1	75.8	76.7	71.3
AVG	61.6	59.4	75.8	77.9	72.4
GRAN	62.5	63.4	75.9	77.7	72.9
[Conneau et al., 2017] - unsupervised (transfer learning)					
InferSent (AllSNLI)	58.6	51.5	67.8	68.3	-
InferSent (SNLI)	57.1	50.4	66.2	65.2	-
[Wieting and Gimpel, 2017a] - unsupervised					
ParaNMT Word Avg.	66.2	61.8	76.2	79.3	-
ParaNMT BiLSTM Avg.	67.4	60.3	76.4	79.7	-
ParaNMT Trigram-Word	67.8	62.7	77.4	**80.3**	-
Our Approach - unsupervised					
GloVe+UP	64.9	63.6	74.4	76.1	73.0
PSL+UP	65.8	65.2	75.9	77.6	72.3
ParaNMT+UP	**68.3**	**66.1**	**78.4**	79.0	**73.5**

Table 4.2: Average results (Pearson's $r \times 100$) on textual similarity tasks. The highest score in each column is in bold. "Glove+UP" is the application of uSIF-weighting (U) and piecewise common component removal (P) to GloVe word vectors; "PSL+UP' to PSL word vectors; "ParaNMT+UP", to ParaNMT word vectors.

Ablation Study On average, our weighting scheme alone is responsible for a roughly 4.4% improvement over Arora et al. The piecewise common component removal alone is responsible for a roughly 5.1% improvement, and the normalization alone is responsible for a roughly 6.7% improvement. This suggests that the benefits of our individual contributions have much overlap. The choice of tuned word vectors (e.g., ParaNMT over GloVe) can also improve results by up to 11.2%.

4.4.4 Supervised Tasks

We also test our approach on three supervised tasks: the SICK similarity task (SICK-R), the SICK entailment task (SICK-E), and the Stanford Sentiment Treebank (SST) binary classification task [Socher et al., 2013]. To a large extent, performance on these tasks depends on the architecture that is trained with the sentence embeddings. We take the embeddings that perform best on the textual similarity tasks, ParaNMT+UP, and follow the setup in [Wieting et al., 2016b]. As seen in Table 4.4, both SIF-weighting with common component removal [Arora et al., 2017b] and uSIF-weighting with piecewise common component removal (ours) perform slightly better than simple word averaging, but not as well as more sophisticated models. Past work has found that tuning the word embeddings in addition to the parameters of the model yields much better performance [Wieting et al., 2016b], as does increasing the size of the hidden layer in the classifier [Arora et al., 2017b]. The results here, however, suggest that regardless of such changes, our approach would not be any more effective than

Unsupervised	
Doc2Vec DBOW [Le and Mikolov, 2014]	64.9
GloVe+UP	71.5
Charagram [Wieting et al., 2016a]	71.6
Paragram-Phrase [Wieting et al., 2016b]	73.2
PSL+UP	74.8
Sent2vec [Pagliardini et al., 2017]	75.5
InferSent (bi-LSTM trained on SNLI) [Conneau et al., 2017]	75.8
ParaNMT Word Avg. [Wieting and Gimpel, 2017a]	79.2
ParaNMT BiLSTM Avg. [Wieting and Gimpel, 2017a]	79.2
ParaNMT+UP	79.5
ParaNMT Trigram-Word Addition [Wieting and Gimpel, 2017a]	**79.9**
Weakly Supervised	
GloVe+WR [Arora et al., 2017b]	72.0
GRAN [Wieting and Gimpel, 2017b]	76.4
Supervised	
Constituency Tree-LSTM [Tai et al., 2015]	71.9
CNN (HCTI) [Shao, 2017]	78.4

Table 4.3: Results (Pearson's $r \times 100$) on the STS Benchmark dataset. The highest score is in bold. The scores of our approaches are underlined.

Model	SST	SICK-R	SICK-E
ParaNMT-based [Wieting and Gimpel, 2017a]			
ParaNMT Word Avg. (300d)	80.0	83.6	80.6
ParaNMT Trigram Avg. (300d)	73.6	79.3	78.0
ParaNMT LSTM Avg. (300d)	80.6	83.9	81.9
LSTM (600d)	80.0	85.2	82.6
LSTM (900d)	81.6	86.0	83.0
BiLSTM (600d)	79.1	85.4	84.3
BiLSTM (900d)	81.3	85.8	84.4
Trigram-Word (600d, concatenation)	79.7	84.6	82.0
Trigram-Word-LSTM (900d, concatenation)	82.0	85.4	83.8
BILSTM AVG (4096)	82.8	85.9	83.8
ParaNMT+WR† [Arora et al., 2017b]	80.5	83.9	80.9
ParaNMT+UP† (ours)	80.7	83.8	81.1
Other Approaches			
BiLSTM-Max (on AllNLI) [Conneau et al., 2017]	84.6	**88.4**	**86.3**
skip-thought [Kiros et al., 2015]	82.0	85.8	82.3
BYTE mLSTM [Radford et al., 2017]	**91.8**	79.2	-

Table 4.4: Results on the SST, SICK-R, and SICK-E tasks. The best score for each task is bolded. † indicates our implementation.

Arora et al.'s on these tasks. Still, our approach retains the advantage of being a completely unsupervised method that can be used when there is no labelled data.

4.5 Future Work

There are several possibilities for future work. For one, the values we derived for $Z_{\tilde{c_s}}, \alpha, a$ and $\{\lambda_i\}$ are not necessarily optimal. While they are based on reasonable assumptions, there are likely sentence-specific and task-specific values that yield better results. Hyperparameter search is one way of finding these values, but that would require supervision. It may be possible, however, to theoretically derive more optimal values.

4.6 Conclusion

We first showed that word vector length has a confounding effect on the log-linear random walk model of generating text [Arora et al., 2017b], the basis of a strong baseline method for sentence embeddings. We then proposed an angular distance–based random walk model where the probability of a sentence being generated is robust to distortion from word vector length. From this model, we derived a simple approach for creating sentence embeddings: normalize the word vectors, compute a weighted average, and then modify it using SVD. Unlike in Arora et al., our approach does not require hyperparameter tuning – it is completely unsupervised and can therefore be used when there is no labelled data. Our approach outperforms Arora et al.'s by up to 44.4% on textual similarity tasks and is even competitive with state-of-the-art methods. Because our simple approach is tough-to-beat, robust, and unsupervised, it is an ideal baseline for computing sentence embeddings.

Chapter 5

Conclusion

In this work, I provided an answer to two open questions on word embedding properties. The first of these questions was: why, and under what conditions, can vector algebra be used to solve word analogy tasks? Building on prior theoretical work that framed neural embedding models as matrix factorization, I showed that the geometry of word analogies could be exploited to derive the co-occurrence shifted PMI (csPMI) Theorem: a linear word analogy holds over a set of ordered word pairs iff the csPMI is the same for every word pair and across any two word pairs. This had three key implications: (1) this proved the [Pennington et al., 2014] conjecture, the intuitive explanation of this phenomenon; (2) the addition of two SGNS word vectors automatically downweights the more frequent word, as weighting schemes do *ad hoc*; (3) Euclidean distance between two words in embedding space is a good proxy of word dissimilarity because it is linear in the negative csPMI. Most importantly, unlike past theories, the csPMI Theorem did not make strong assumptions about the word distribution or embedding space.

The second open question that this work addressed was: why do socially undesirable associations such as gender bias exist in word embedding spaces and can they be provably removed? I proved that when there is no reconstruction error, debiasing word embeddings using subspace projection [Bolukbasi et al., 2016] is, in theory, equivalent to training on an unbiased corpus. Moreover, WEAT and WEFAT [Caliskan et al., 2017], the standard metrics of word embedding association that are used to measure gender bias, have theoretical flaws that exaggerate the extent of bias. Using the subspace projection method, I derived a new measure of word embedding association called the *relational inner product association* (RIPA), which does not suffer from the theoretical flaws of WEAT and WEFAT. Experiments with RIPA revealed that SGNS does not, *on average*, make the vast majority of words any more gendered in the vector space than they are in the training corpus; individual words may be slightly more or less gendered due to reconstruction error. However, for words that are gender-stereotyped (e.g., *nurse*) or gender-specific by definition (e.g., *queen*), SGNS amplifies the gender association in the training corpus.

Lastly, I designed a sentence embedding method based on a random walk model of sentences, which was itself based on the broader finding that the simple addition of two word vectors is a sound, albeit simplistic, means of composing words. This sentence embedding method, called *unsupervised smoothed inverse frequency* (uSIF), involves taking a weighted average of word embeddings that downweights more frequent words and then denoising with singular value decomposition (SVD). It is similar to SIF [Arora et al., 2017b], but does not require any hyperparameter tuning and achieves much better results on sentence similarity tasks, on par with state-of-the-art. The success of this simple method highlights how answering theoretical questions about word embeddings, while an important research direction in its own right, can also lead to improvements on empirical problems. While this work has taken major steps toward understanding word embedding phenomena, many properties of word embeddings remain poorly understood. For example, why is the optimal dimensionality of word embeddings so low relative to vocabulary size? Even as contextualized word embeddings [Peters et al., 2018] and deep end-to-end neural networks [Devlin et al., 2018] supplant the traditional paradigm of using pretrained word embeddings and a custom neural network architecture, there is still value in developing a deeper theoretical understanding of word embeddings, if only to better understand the new wave of contextualized representations.

Bibliography

[Agirre et al., 2015] Agirre, E., Banea, C., Cardie, C., Cer, D. M., Diab, M. T., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., et al. (2015). Semeval-2015 task 2: Semantic textual similarity, English, Spanish and pilot on interpretability. In *Proceedings SemEval@ NAACL-HLT*, pages 252–263.

[Agirre et al., 2014] Agirre, E., Banea, C., Cardie, C., Cer, D. M., Diab, M. T., Gonzalez-Agirre, A., Guo, W., Mihalcea, R., Rigau, G., and Wiebe, J. (2014). Semeval-2014 task 10: Multilingual semantic textual similarity. In *Proceedings SemEval@ COLING*, pages 81–91.

[Agirre et al., 2013] Agirre, E., Cer, D., Diab, M., Gonzalez-Agirre, A., and Guo, W. (2013). Sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics*. Association for Computational Linguistics.

[Agirre et al., 2012] Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 385–393. Association for Computational Linguistics.

[Arora et al., 2016] Arora, S., Li, Y., Liang, Y., Ma, T., and Risteski, A. (2016). A latent variable model approach to PMI-based word embeddings. *Transactions of the Association for Computational Linguistics*, 4:385–399.

[Arora et al., 2017a] Arora, S., Liang, Y., and Ma, T. (2017a). A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

[Arora et al., 2017b] Arora, S., Liang, Y., and Ma, T. (2017b). A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

[Bengio et al., 2003] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

[Bolukbasi et al., 2016] Bolukbasi, T., Chang, K.-W., Zou, J. Y., Saligrama, V., and Kalai, A. T. (2016). Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

[Caliskan et al., 2017] Caliskan, A., Bryson, J. J., and Narayanan, A. (2017). Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

[Cer et al., 2017] Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

[Chen and Manning, 2014] Chen, D. and Manning, C. (2014). A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

[Church and Hanks, 1990] Church, K. W. and Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

[Collobert and Weston, 2008] Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*, pages 160–167. ACM.

[Conneau et al., 2017] Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

[Ethayarajh, 2018] Ethayarajh, K. (2018). Unsupervised random walk sentence embeddings: A strong but simple baseline. In *Proceedings of The Third Workshop on Representation Learning for NLP*, pages 91–100.

[Ethayarajh et al., 2018] Ethayarajh, K., Duvenaud, D., and Hirst, G. (2018). Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882*.

[Firth, 1957] Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.

[Gittens et al., 2017] Gittens, A., Achlioptas, D., and Mahoney, M. W. (2017). Skip-gram – Zipf + uniform = vector additivity. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 69–76.

[Gong et al., 2018] Gong, Y., Luo, H., and Zhang, J. (2018). Natural language inference over interaction space. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

[Kalchbrenner et al., 2014] Kalchbrenner, N., Grefenstette, E., and Blunsom, P. (2014). A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655–665.

[Kiros et al., 2015] Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3294–3302.

[Landauer and Dumais, 1997] Landauer, T. K. and Dumais, S. T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological review*, 104(2):211.

[Le and Mikolov, 2014] Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196.

[Levy and Goldberg, 2014] Levy, O. and Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

[Levy et al., 2015] Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.

[Luong et al., 2013] Luong, T., Socher, R., and Manning, C. D. (2013). Better word representations with recursive neural networks for morphology. In *SIGNLL Conference on Computational Natural Language Learning (CoNLL)*, pages 104–113.

[Marelli et al., 2014] Marelli, M., Bentivogli, L., Baroni, M., Bernardi, R., Menini, S., and Zamparelli, R. (2014). Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of SemEval@ COLING*, pages 1–8.

[Mikolov et al., 2013a] Mikolov, T., Chen, K., Corrado, G. S., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

[Mikolov et al., 2013b] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

[Mimno and Thompson, 2017] Mimno, D. and Thompson, L. (2017). The strange geometry of skip-gram with negative sampling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2873–2878.

[Mitchell and Lapata, 2008] Mitchell, J. and Lapata, M. (2008). Vector-based models of semantic composition. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*, pages 236–244.

[Mnih and Hinton, 2007] Mnih, A. and Hinton, G. (2007). Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, pages 641–648. ACM.

[Pagliardini et al., 2017] Pagliardini, M., Gupta, P., and Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

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

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

[Piantadosi, 2014] Piantadosi, S. T. (2014). Zipf's word frequency law in natural language: A critical review and future directions. *Psychonomic Bulletin & Review*, 21(5):1112–1130.

[Radford et al., 2017] Radford, A., Jozefowicz, R., and Sutskever, I. (2017). Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

[Robertson, 2004] Robertson, S. (2004). Understanding inverse document frequency: on theoretical arguments for IDF. *Journal of Documentation*, 60(5):503–520.

[Rohde et al., 2006] Rohde, D. L., Gonnerman, L. M., and Plaut, D. C. (2006). An improved model of semantic similarity based on lexical co-occurrence. *Communications of the ACM*, 8(627-633):116.

[Shao, 2017] Shao, Y. (2017). Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

[Socher et al., 2011] Socher, R., Huang, E. H., Pennin, J., Manning, C. D., and Ng, A. Y. (2011). Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Advances in Neural Information Processing Systems*, pages 801–809.

[Socher et al., 2013] Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., and Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642.

[Tai et al., 2015] Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*.

[Wieting et al., 2016a] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016a). Charagram: Embedding words and sentences via character n-grams. *arXiv preprint arXiv:1607.02789*.

[Wieting et al., 2016b] Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016b). Towards universal paraphrastic sentence embeddings. In *International Conference on Learning Representations*.

[Wieting et al., 2015] Wieting, J., Bansal, M., Gimpel, K., Livescu, K., and Roth, D. (2015). From paraphrase database to compositional paraphrase model and back. *Transactions of the Association for Computational Linguistics*, 3:345–358.

[Wieting and Gimpel, 2017a] Wieting, J. and Gimpel, K. (2017a). Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. *arXiv preprint arXiv:1711.05732*.

[Wieting and Gimpel, 2017b] Wieting, J. and Gimpel, K. (2017b). Revisiting recurrent networks for paraphrastic sentence embeddings. *arXiv preprint arXiv:1705.00364*.

[Yin and Shen, 2018] Yin, Z. and Shen, Y. (2018). On the dimensionality of word embedding. In *Advances in Neural Information Processing Systems*, pages 894–905.

[Zhao et al., 2018] Zhao, J., Zhou, Y., Li, Z., Wang, W., and Chang, K.-W. (2018). Learning gender-neutral word embeddings. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4847–4853.