UPPSALA
UNIVERSITET

U.U.D.M. Project Report 2019:15

Word Embeddings and Gender Stereotypes in Swedish and English

Rasmus Précenth

Examensarbete i matematik, 30 hp
Handledare: David Sumpter
Examinator: Denis Gaidashev
Maj 2019

Department of Mathematics
Uppsala University

Word Embeddings and Gender Stereotypes in Swedish and English

Rasmus Précenth*

March 2019

Abstract

A word embedding is a representation of words as vectors. After Mikolov et al. introduced the algorithm WORD2VEC in 2013, the popularity of word embeddings exploded. Not only was the new algorithm much more efficient, but it also produced embeddings that exhibited an interesting property allowing for reasoning with analogies such as "he is to king as she is to queen". Later it was discovered that the embeddings contained different types of biases, such as gender bias. We first look at how word embeddings are constructed and then investigate what it means mathematically to create an analogy between words. By creating a new way of measuring how good analogies are we make it possible to extend and improve earlier methods. For creating word embeddings we use the techniques earlier applied to English to Swedish. We find that Swedish can be represented just as well as English in an embedding and exhibits many of the same biases. We also compare gender bias in Swedish and English embeddings directly using a new method.

*rasmus@precenth.eu

Contents

1 Introduction

A word embedding, or word representation, is in its most basic form a map from a vocabulary to a vector space. The history of word embeddings goes back to a concept called *distributed representations* [35] in which a word is represented by "distributing" its meaning across many dimensions. Thus words like *cat* and *dog* might both be present in the "animal" dimension. Contrast this with using a standard one-hot encoding of the vocabulary words, i.e. where each vector is 0 for all dimensions except the one corresponding to the word represented, for which it is 1. In such a representation no meaning would be preserved by the word vectors. The common way of creating a distributed representation is by following the *distributional hypothesis* [19]; words can be understood from the contexts they occur in and by the words they can be replaced by [19, 36].

Word embeddings were originally used for various natural language processing (NLP) tasks where they act as an intermediate representation between words and the target application. More recently they have also started to be studied in and of themselves after the discovery that word embeddings exhibited the structure for word analogies such as "he is to king as she is to queen" as simple vector arithmetic $\vec{\texttt{king}} - \vec{\texttt{he}} \approx \vec{\texttt{queen}} - \vec{\texttt{she}}$ [32].

Online systems have been shown to exhibit a diverse range of biases and stereotypes, such as racism in ad delivery networks [41, 7] and exaggerated gender stereotypes in image search results [23]. As the popularity and usage of algorithms trained no natural language grows, there is a great risk that biases present in the training data, like how men and women are treated differently in Wikipedia articles [45, 46], will remain in the result and help propagate stereotypes.

Word embeddings display bias in that sexist analogies such as "he is to computer programmer as she is to homemaker" make sense according to vector arithmetic [3]. This discovery lead to the subsequent findings that word embeddings show gender stereotypes [3] and that the gender bias contained within correlates with occupational data [14]. Moreover, a version of the Implicit Association Test (IAT) [18] was developed for word embeddings and all of the results from the original tests where reproduced for the embeddings studied [5]. For example, the embeddings considered male names to be closer to career words and female names to be closer to family words and at the same time they considered European-American names more pleasant than African-American names.

It is well known that language use affects the view on men and women in society, e.g. "[..] languages with grammatical gender promote sexist attitudes" [47]. One of the countries that has put a lot of effort towards making its language gender-neutral is Sweden, with multiple successful reforms [33]. In general, Sweden and the rest of the Nordic countries are considered at the forefront of gender equality, something that is well reflected in the Global Gender Gap Report [24]. Out of the top five countries, four are Nordic: Iceland (1st), Norway (2nd), Finland (3rd) and Sweden (5th). As a comparison, the United Kingdom is ranked 15th and the United States 49th. Can this be seen in NLP algorithms? Are algorithms run on Swedish corpora less biased than for example ones run

English language data?

2 Outline

The main question that will be answered in this report is: why and in what way do word embeddings, and word analogies in particular, express sexism and other biases? We start off by considering word embeddings themselves in Section 3.

The definitions of word analogies and the many ways of computing them lead us to wonder: what do they really mean? Can we find a better way of computing them? Furthermore, analogies are used to evaluate embeddings, how does that work? Section 4 considers all of these questions and provides a deeper background together leading to the discovery of a novel way of evaluating an analogy.

As stated in the introduction, the prevalence of bias in embeddings can be considered harmful depending on context. Mathematical ways of measuring and removing bias are explored in Section 5 resulting in a new method of measuring bias across languages.

A lot of the literature focuses on the English language as the target of word embeddings, with standardized tests of an embedding's quality and so forth. What happens when we apply word embeddings to other languages? Are there any tests for those languages? We try to answer these questions in Section 6 and provide embeddings for Swedish.

Following the reasoning in the introduction, is it possible to see the results of the effort, to reduce gender inequality in language, in Swedish word embeddings? Is Swedish less sexist than English? We perform some tests of these hypotheses in Section 7.

3 Word Embeddings

A word embedding, or word representation, is a mapping of words from a vocabulary V to vectors in d-dimensional space, while having semantically similar words be represented by similar vectors in the embedding. Similarity can be defined in different ways but in general we consider close vectors to be similar and distant vectors to be dissimilar. We typically write $\vec{w}$ to represent the word vector of the word w, but since there is no ambiguity most of the time we will simply write w to represent both the word and its corresponding vector.

3.1 Explicit Representations

The most basic version of word representations is by using a one-hot encoding of the words in a vocabulary, i.e. given an ordered vocabulary $V = (w_1, \ldots, w_n)$, define the representation of word w_i as the vector with 0's in all components, except the ith one which is set to 1. Visualized as a matrix it would be the identity matrix on $\mathbb{R}^n$. This is also known as a *local* representation [35]. While being simple, any semantic similarity between two words w_i and w_j would be lost

in the representation, or equivalently, all words are considered equally similar (similarity measures are discussed in Section 4.1).

What one can do instead is to create a *distributed* representation [35] that distributes the contribution of each word over the dimensions in the representation, trying to make the representations of semantically similar words similar as vectors.

Many types of representations of this form start out by measuring the frequencies of words in contexts in a large corpus, where contexts can be defined in many different ways, e.g. as words, phrases, sentences or syntactic relations [44, 25]. One example of how one can define contexts is by using words as contexts and consider another word to lie in a context if the distance between the words is below some threshold k. In the sentence `he is a man and she is a woman` the word `man` lies in the contexts of `he`, `is` (twice), `a`, `man`, `and` and `she`, with $k = 3$. Another way of counting contexts is by using k-shifted words, i.e. words offset by a distance between 1 and k. In the example sentence above, the word `man` appears in the contexts $\texttt{is}^{-2}$, $\texttt{a}^{-1}$, $\texttt{and}^{+1}$ and $\texttt{she}^{+2}$, with $k = 2$.

By counting all occurrences of word-context pairs throughout a corpus one can build a co-occurrence matrix M where M_{ij} is the number of times word i occurs in context j. To illustrate the process we show the co-occurrence matrix for the first example above (a simple window based context with $k = 3$) in the following table.

	he	she	is	a	and	woman	man
he	1	0	1	1	0	0	1
she	0	1	1	2	1	1	1
is	1	1	2	2	2	1	1
a	1	2	2	2	2	1	1
and	0	1	2	2	1	0	1
woman	0	1	1	1	0	1	0
man	1	1	2	1	1	0	1

A possible representation of the word w_i would thus be the row M_{i*} corresponding to the word w_i, i.e. the number of times w_i occurs in the different contexts. As is common in machine learning, one should normalize this before using it. A good approach is to let $P_{ij} = p(j|i) = M_{ij} / \sum_k M_{ik}$ be the (observed) probability that context j appears around word i. We then use P_{i*} as the representation for word w_i. The following is the result of normalizing the example matrix.

	he	she	is	a	and	woman	man
he	0.25	0.00	0.25	0.25	0.00	0.00	0.25
she	0.00	0.14	0.14	0.29	0.14	0.14	0.14
is	0.10	0.10	0.20	0.20	0.20	0.10	0.10
a	0.09	0.18	0.18	0.18	0.18	0.09	0.09
and	0.00	0.14	0.29	0.29	0.14	0.00	0.14
woman	0.00	0.25	0.25	0.25	0.00	0.25	0.00
man	0.14	0.14	0.29	0.14	0.14	0.00	0.14

As with the previous approaches, the somewhat naïve approach of using $\vec{w_i} = P_{i*}$ still has some issues. Amongst others, common function words such as *the*, *and*, *a* and *an* tend to dominate the representation. To combat this issue one can use a method known as *positive pointwise mutual information*, or *PPMI*. Using PPMI one tries to lower the impact of common words and increase the impact of rare words by looking at how often a word appears in a context compared to random chance [6]. PPMI is defined by [26]

$$\text{PPMI}(w, c) = \max(\text{PMI}(w, c), 0) \tag{1}$$

$$\text{PMI}(w, c) = \log \frac{p(w, c)}{p(w)p(c)} = \log \frac{\#(w, c)|\text{corpus}|}{\#(w) \cdot \#(c)} \tag{2}$$

where w is a word, c is a context, $\#(w, c)$ is the number of times w and c occur together, $\#(w)$ is the number of times w occurs and $\#(c)$ is the number of times c occurs. Finally, let $S_{ij} = \text{PPMI}(w_i, c_j)$ be the new representation. Note that we set $\text{PPMI}(w, c) = 0$ if $\#(w, c) = 0$. The PPMI matrix of the example is shown below.

	`he`	`she`	`is`	`a`	`and`	`woman`	`man`
`he`	2.64	0.00	1.63	1.63	0.00	0.00	2.24
`she`	0.00	1.52	1.07	1.77	1.52	2.08	1.68
`is`	1.73	1.17	1.41	1.41	1.86	1.73	1.32
`a`	1.63	1.77	1.31	1.31	1.77	1.63	1.23
`and`	0.00	1.52	1.77	1.77	1.52	0.00	1.68
`woman`	0.00	2.08	1.63	1.63	0.00	2.64	0.00
`man`	2.08	1.52	1.77	1.07	1.52	0.00	1.68

Although we now have a representation that captures and preserves semantic similarity between words, we have N vectors of dimension d, where N is the size of the vocabulary and d the number of contexts, which means that our representations are still infeasible for many applications. For instance, with 2 million words and using 5-shifted words as contexts, we would have $N = 2 \cdot 10^6$ and $d = (5-1)N = 8 \cdot 10^6$. In total there would be a need to store $Nd = 1.6 \cdot 10^{13}$ values. We thus need to find a way to reduce the size of the representation.

3.2 Implicit Representations

In contrast to the explicit representations described above where the dimension of the co-domain is quite big, the more implicit embeddings tend to be focused around dimensionality reduction. However, the result is often opaque and we no longer know what each dimension means. Early versions of these embeddings were low-rank approximations of matrices similar to the ones described in section 3.1.

To understand what that entails, we first note that the word vectors in the explicit case are $W = M$ and that we can obtain *context* vectors $C = M^\top$ for free. Most of the methods below talk about factorizing M, which means finding W and C such that $WC^\top = M$. In other words, word i is represented

by a vector $w_i \in W$ and context j is represented by a vector $c_j \in C$ such that $\langle w_i, c_j \rangle = M_{ij}$.

One of the early methods of this type is *latent semantic analysis (LSA)* [8] which uses singular-value decomposition (SVD) to factorize M as $U\Sigma V^\top$. By picking the d most significant singular values and their corresponding rows and columns in U and V we obtain the d-dimensional word vectors $W = U_d\Sigma_d$ and context vectors $C = V_d$. If we continue our example from the previous section we get the following word vectors for $d = 3$. Note that we no longer know what each dimension means.

	?	?	?
`he`	-1.54	0.98	0.46
`she`	-2.82	-0.74	0.47
`is`	-3.94	0.03	-0.45
`a`	-4.29	-0.31	-0.63
`and`	-3.16	0.18	0.78
`woman`	-1.61	-0.80	0.32
`man`	-2.82	0.89	-0.19

Now, $M' = WC^\top$ only approximates M but it is the best d-dimensional representation in the sense that it minimizes the approximation error $\|M - M'\|_F$ over all rank d matrices M'. This way LSA can be seen as noise reduction of the original co-occurrence matrix (see e.g. [44] for alternative perspectives of what LSA does). There is a downside to using LSA, however; the computational complexity of performing SVD is high.

Skip-gram with negative-sampling (SGNS) [30, 31] or WORD2VEC, from the name of the software used to produce the embeddings, is a very efficient method that creates an embedding by scanning the corpus trying to predict the contexts surrounding each word. It has been shown that SGNS implicitly factorizes a shifted version of the PMI matrix where the contexts are k-shifted words [27];

$$M_{ij} = \mathrm{PMI}(w_i, c_j) - \log k. \tag{3}$$

The introduction of SGNS made it possible to quickly produce embeddings on large corpora while retaining a high accuracy (see Section 3.3 on how the accuracy is computed). The introduction of SGNS was the starting point of a new wave of algorithms. Together with the observations made regarding word analogies [32] it opened up new areas where embeddings could be applied.

Global Vectors for Word Representation (GloVe) [34] is a method that constructs word embeddings using a few criteria regarding the desired relationship between the resulting vectors and co-occurrence probabilities of the words. The main criterion is that differences of vectors should correspond to ratios of probabilities, thus trying to preserve the linguistic regularities (word analogies etc.) observed in [32]. More concretely, the algorithm creates word vectors w_i, and context vectors, $\tilde{w}_j$, from the co-occurrence matrix M, trying to minimize the error in the equation

$$\langle w_i, \tilde{w}_j \rangle + b_i + \tilde{b}_j = \log M_{ij} \tag{4}$$

man	n-grams
3-grams	<ma, man, an>
4-grams	<man, man>
5-grams	<man>
6-grams	
word	<man>

woman	n-grams
3-grams	<wo, wom, oma, man, an>
4-grams	<wom, woma, oman, man>
5-grams	<woma, woman, oman>
6-grams	<woman, woman>
word	<woman>

Table 1: The n-grams of the words *man* and *woman*. The symbols < and > indicate the beginning and the end of the word respectively. Note that man, an> and man> occur in both collections.

where b_i and $\tilde{b}_j$ are learned bias vectors that are discarded after training has completed. GloVe thus factorizes the element-wise logarithm of the matrix M shifted by the learned biases.

A method based on SGNS is *Subword Information Skip-Gram (SISG)* or fastText [2], in which words are represented by collections of substrings of specific lengths, so called n-grams. The representation of the word is then the sum of the representations of the n-grams. The default collection of n-grams used for the method is the set of all 3-grams, 4-grams, 5-grams and 6-grams of the word together with the whole word. For example, the words *man* and *woman* have the collections of n-grams shown in Table 1. In this case, some n-grams are shared between the words, meaning that the representations are linked such that adjusting one might also adjust the other. It was shown in [2] that SISG produces good embeddings using less data than previous algorithms. The reason being that it can not only train multiple words at the same time but it can also create representations of out-of-vocabulary words by simply summing up the representations of the n-grams, thus obtaining fairly accurate estimations of the would-be representations. It should also be noted that for languages with more inflection than English it is especially helpful to share the base meaning between the different variations of the same words. An example in English would be the words *run*, *runner* and *running* which all contain the 4-gram <run.

3.3 Evaluating Word Embeddings

Now that we have some word embeddings, how do we know how good they are? What does good mean? The goal of a word embedding is to provide a more feasible input to NLP algorithms with increased performance compared to earlier methods. Tests that evaluate the performance of an embedding on NLP tasks are called *extrinsic*. *Intrinsic* tests, on the other hand, are those that test the structure of the embedding by, for instance, evaluating how well word similarity correlates with human judgment (e.g. SemEval [22] and SimLex-999 [20]) or how well the embedding allows for analogical reasoning (see Section 4.6).

Because of the simplicity of running intrinsic tasks, the methods quickly became the *de facto* standard way of evaluating word embeddings, see e.g. [30,

28] which both exclusively use intrinsic measures for evaluating the accuracy of the trained embeddings. Performance on the intrinsic tasks are thus used as a proxy for performance on extrinsic tasks.

The problem with the above way of evaluating embeddings is that there is no strong correlation between performance on intrinsic and extrinsic tasks [39, 42]. Moreover, there is an apparent risk of overfitting and there are no tests for statistical significance [13]. There has been an attempt to remedy these problems with the introduction of QVec [42] which is an intrinsic measure that correlates more strongly with extrinsic measures. The general recommendation is however to train the embedding with an objective that suits the application [39].

4 Word Analogies

Word analogies in the context of word embeddings are statements of the form "a is to a^* as b is to b^*", usually written as $a : a^* :: b : b^*$. Although the concept of analogies has been known for some time [15], it wasn't until [30, 32] discovered that word embeddings exhibited these analogies as simple linear relations that they became popular. A method based on word analogies was developed to measure the quality of word embeddings and this method is now the *de facto* standard method of evaluating embeddings together with various word similarity tasks [12].

4.1 Word Similarity

The similarity of two words in the representation is most commonly measured by computing the *cosine similarity* of their vectors [44];

$$\cos(x, y) = \cos\theta = \frac{\langle x, y\rangle}{\|x\|\|y\|}, \tag{5}$$

where θ is the angle between x and y. The range of cosine similarity is from -1 (opposite vectors, highly dissimilar) to 1 (parallel vectors, highly similar). Most of the time one normalizes the word vectors before using them. In those cases we have

$$\cos(x, y) = \langle x, y\rangle. \tag{6}$$

Words that are similar semantically should thus be mapped to vectors with a high cosine similarity. Revisiting our example from Section 3 we compute some of the cosine similarities of the word vectors produced by the LSA method below.

$$\cos(\texttt{he}, \texttt{man}) = 0.92,\ \cos(\texttt{she}, \texttt{woman}) = 0.98,\ \cos(\texttt{he}, \texttt{woman}) = 0.54$$

As can be seen from this basic example, *he* is more similar to *man* than to *woman*.

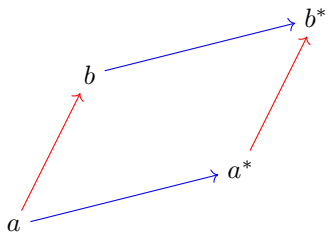

Figure 1: A typical visualization of an analogy. By adding the difference $b - a$ to a^* we get b^*, or equivalently, by adding $a^* - a$ to b.

There is no apparent reason for choosing specifically cosine similarity as the similarity measure, and indeed there are many other options [4]. In the normalized case, cosine similarity and Euclidean distance are related in the following way;

$$\begin{aligned} \|x - y\|^2 &= \langle x, x\rangle - 2\langle x, y\rangle + \langle y, y\rangle \\ &= 2 - 2\langle x, y\rangle \\ &= 2(1 - \cos(x, y)) \end{aligned}$$

For a better understanding of why cosine similarity works so well, consider the explicit representation from Section 3.1 where we use $\vec{w_i} = P_{i*}$ as our vectors; we find that the cosine similarity is a sum of products of probabilities, namely $\cos(w_i, w_j) = \sum_k p(k|i)p(k|j)$. Each term of the sum is thus high if both $p(k|i)$ and $p(k|j)$ are high and low if either of them are. In other words, each term contains information on how similar w_i and w_j are when it comes to the context k. The full cosine similarity is then a sum over all contexts resulting in a measure of how similar the words are.

4.2 Definitions and Intuition

The key observation made in [30, 32] was that an analogy can be stated as an equation of the form $b^* - b = a^* - a$. What this means is that the way we go from a to a^* is the same as from b to b^*. This relation – or rather relation of relations – can be interpreted geometrically as a parallelogram (see Figure 1).

Using a measure of word similarity, one can derive different methods of solving word analogies. Given words a, a^* and b, the original method introduced in [32] searches for the word b^* that solves the equation $b^* - b = a^* - a$. It is unlikely that a word b^* solves it exactly, so they search for the b^* that maximizes the similarity of the sides in the equation $b^* = a^* - a + b$, i.e.

$$b^* = \text{argmax}_{c \in V} \cos(c, a^* - a + b). \tag{7}$$

The goal is thus to find the word in the vocabulary with a representation as similar as possible to the optimal solution $a^* - a + b$.

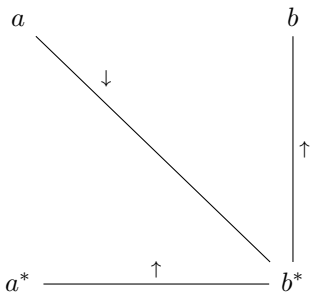

Figure 2: Interpretation of analogies according to 3CosAdd and 3CosMul where ↑ represents a high similarity between the words and ↓ a low similarity.

The RHS of Equation (7) can be expanded into the form

$$b^* = \text{argmax}_{c \in V} \cos(c, a^*) - \cos(c, a) + \cos(c, b) \tag{8}$$

and is therefore called 3CosAdd [26]. One interpretation of this method of analogy solving is thus "find the b^* that is the most similar to a^* and b but dissimilar to a". This interpretation is visualized in Figure 2. As a concrete example, `Sweden` : `Stockholm` :: `Germany` : x should have the solution $x =$ `Berlin` since *Berlin* is similar to *Stockholm* (both are capital cities) and similar to *Germany* (both concerns things that are German) but dissimilar to *Sweden*.

The above method was later improved by [26] by replacing the additive operations with multiplicative ones to remove the issue of a "soft-or" behavior where one term dominates the others. The resulting objective is

$$b^* = \text{argmax}_{c \in V} \frac{\cos(c, a^*) \cos(c, b)}{\cos(c, a) + \varepsilon} \tag{9}$$

and is aptly named 3CosMul. Since the range of the cosine similarity is $[-1, 1]$ it is suggested that one transforms the similarities to the range $[0, 1]$ by $(x+1)/2$ before computing (9). We call this version of the similarity shifted cosine similarity. The value ε is a small positive value, usually 0.001, whose sole purpose is to prevent division by zero. Due to showing better performance 3CosMul has since its introduction mostly replaced 3CosAdd for solving analogies.

A method related to 3CosAdd is PairDirection which is based on the original equation $a^* - a = b^* - b$ instead of the equation $b^* = a^* - a + b$. The objective becomes

$$b^* = \text{argmax}_{c \in V} \cos(c - b, a^* - a) \tag{10}$$

thus looking for the b^* that makes the differences $b^* - b$ and $a^* - a$ the most similar. Note that the norms of the differences are ignored since values are normalized under cosine similarity. Therefore, PairDirection only seeks the b^* that produces the most similar direction of the offset. Albeit a bit unclear from the literature, it seems that PairDirection is better than at least 3CosAdd at solving syntactical analogies [26].

4.3 Generalizing the Objectives

The methods for solving analogies from the previous section are based around the implicit assumption that the first part of the analogy makes sense. It is unreasonable to expect a word embedding to complete a nonsense analogy such as `he` : `food` :: `woman` : x. There is thus a need to check that a completed analogy makes sense, or more generally, there is a need to compare how good analogies are. Our contribution towards solving this problem follows below.

Consider the analogy $a : a^* :: b : b^*$. We can view this analogy as a parallelogram, or equivalently, as the equation $b^* - b = a^* - a$. It is then possible to use basic arithmetic manipulations to yield equivalent equations which then correspond to equivalent analogies. For instance, $a - b = a^* - b^*$ represents the analogy $b^* : a^* :: b : a$. There are eight such analogies in total;

$$\begin{array}{ll} a : a^* :: b : b^* & a : b :: a^* : b^* \\ a^* : a :: b^* : b & a^* : b^* :: a : b \\ b : b^* :: a : a^* & b : a :: b^* : a^* \\ b^* : b :: a^* : a & b^* : a^* :: b : a \end{array}$$

All of the above analogies are thus equivalent, if one holds so should the other seven. To illustrate that point, consider the analogy `he` : `king` :: `she` : `queen`, the equivalent analogies are as follows.

`he` : `king` :: `she` : `queen`	`he` : `she` :: `king` : `queen`
`king` : `he` :: `queen` : `she`	`king` : `queen` :: `he` : `she`
`she` : `queen` :: `he` : `king`	`she` : `he` :: `queen` : `king`
`queen` : `she` :: `king` : `he`	`queen` : `king` :: `she` : `he`

As the reader can confirm, the above analogies all make sense from a human perspective.

By taking all of the eight analogies above into account and by extending the intuition from the previous section we can complete the diagram in Figure 2 to get the one in Figure 3 in which all symmetries are present. Translating this diagram back into an expression, using 3CosMul as a template, leads us to define the function

$$S(a, a^*, b, b^*) = \frac{\cos(a, b)\cos(a, a^*)\cos(b, b^*)\cos(a^*, b^*)}{\cos(a, b^*)\cos(b, a^*) + \varepsilon} \tag{11}$$

where ε is the same as for 3CosMul and where we use the shifted version of cosine similarity[1]. We refer to this function as the *score* of an analogy.

This function S is invariant under all eight symmetries, e.g. $S(a, a^*, b, b^*) = S(a, b, a^*, b^*)$. Moreover, if we fix a, a^* and b and look for the b^* that maximizes

[1] An analogous additive version of S could be defined if needed. We will however focus on the multiplicative one since 3CosMul is generally preferred to 3CosAdd.

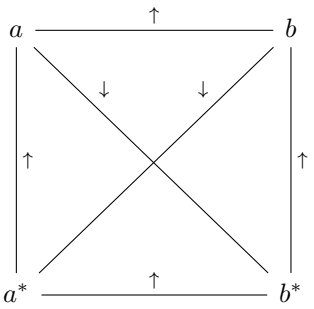

Figure 3: Interpretation of analogies according to 3CosAdd and 3CosMul extended to include all eight symmetries (compare Figure 2). Again, ↑ represents high similarity and ↓ low similarity.

the score we find

$$\begin{aligned} b^* &= \text{argmax}_{c \in V}\, S(a, a^*, b, c) \\ &= \text{argmax}_{c \in V}\, \frac{\cos(a,b)\cos(a,a^*)\cos(b,c)\cos(a^*,c)}{\cos(a,c)\cos(b,a^*) + \varepsilon} \\ &= \text{argmax}_{c \in V}\, \frac{\cos(a,b)\cos(a,a^*)}{\cos(b,a^*)} \frac{\cos(b,c)\cos(a^*,c)}{\cos(a,c) + \varepsilon'} \\ &= \text{argmax}_{c \in V}\, \frac{\cos(b,c)\cos(a^*,c)}{\cos(a,c) + \varepsilon'} \end{aligned}$$

which is equivalent to 3CosMul.

In short, we have thus created an absolute measure of analogy quality (for a particular embedding) that captures all symmetries present in analogies and that generalizes the familiar objective 3CosMul. The score S is also minimal in the sense that removing any of the factors would not make it invariant under all symmetries. We can now use this measure to score completed analogies, but we still don't know what that score tells us about the quality of the analogies.

To solve that problem, consider the trivial analogy $a : a :: b : b$. The score[2] for this analogy is $S(a, a, b, b) = 1$ meaning that analogies with a score close to 1 can be considered particularly appropriate. Moreover, if we fix a and b we have $S(a, x, b, x) = \cos(a, b)$ for all words x. This provides a lower bound for analogies of the form $a : x :: b : y$ since if $S(a, x, b, y) \leq \cos(a, b)$ then $a : x :: b : x$ and $a : y :: b : y$ would both have been better analogies, none of which are proper analogies. We can thus say that analogies for which

$$S(a, a^*, b, b^*) \geq \max\{\cos(a, a^*), \cos(a, b), \cos(b, b^*), \cos(a^*, b^*)\} \qquad (12)$$

[2]The constant ε is by design negligible so we will exclude it from our computations here.

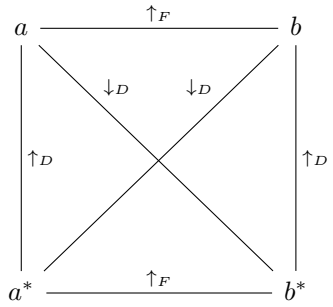

Figure 4: Diagram from [43] which shows an alternative interpretation of word analogies. Here $\uparrow_D$ means high domain similarity, $\downarrow_D$ low domain similarity and $\uparrow_F$ means high function similarity.

does not hold can be considered inappropriate or nonsensical[3]. We will refer to this as the lower bound of an analogy and later use it to improve an existing algorithm in Section 5.2.

4.4 Alternative Ways of Understanding Word Analogies

Although the way of reasoning with analogies described so far works very well, it is based on the idea of similarities visualized in Figure 2 and fails to capture *how* words relate to each other. Ideally one would like *Sweden* and *Stockholm* to be related *in the same way* as *Germany* and *Berlin* are related. The underlying assumption of vector addition does capture this idea, but it is lost in all objectives. Turney [43] suggests using two adjoined embeddings, one that captures *function similarity* and one that captures *domain similarity*. A high domain similarity means that the words mostly lie in the same domain while a high function similarity implies that the words fulfill a similar function in their respective domains. *Sweden* and *Germany* are both countries while *Stockholm* and *Berlin* are both capital cities. *Sweden* and *Stockholm* both concern things Swedish while *Germany* and *Berlin* concern German things. A corresponding diagram for this version is shown in Figure 4.

However, this way of thinking has its own problems. For example, `Stockholm` : `Berlin` :: `Sweden` : `Germany` is a valid analogy, but the domains and functions involved are a lot less intuitive compared to the previous example. Would the domain of *Stockholm* and *Berlin* be capital cities and thus *Stockholm* and *Sweden* fill the function of being the "Swedish" elements of their respective domains?

4.5 Realizations in GloVe and SGNS Embeddings

After having studied the intuition behind analogies one might start to wonder: in terms of word co-occurrences, what does it mean for an analogy to hold?

[3]Note that we are speaking about this from the view of the embedding, the analogy might make sense to humans.

Recent work [16, 10] has started studying this in the cases of GloVe and SGNS leading to the following theorem in [10].

Theorem 4.1 (Co-occurrence Shifted PMI Theorem[4])**.** *An analogy $a : a^* :: b : b^*$ with interpretation $a^* - a = b^* - b$ holds in an SGNS or GloVe embedding with no reconstruction error if and only if*

$$\mathrm{PMI}(a, a^*) + \log p(a, a^*) = \mathrm{PMI}(b, b^*) + \log p(b, b^*). \tag{13}$$

The quantity $\mathrm{PMI}(x, y) + \log p(x, y)$ *is called co-occurrence shifted PMI, or* csPMI *for short.*

Equation (13) provides a way of linking an analogy to the distribution of its constituent words in the training corpus. While a bit hard to understand intuitively, it is at least clear that the csPMI of a and a^* describes a relationship of the distribution of the two words that has to be the same for b and b^*.

Some corollaries are provided to the theorem in the paper for which the most important one is the proof that the fundamental criteria specified during the derivation of the GloVe method [34] does hold for SGNS embeddings, namely that ratios of probabilities do correspond to differences of vectors (here represented as an analogy).

Corollary 4.2. *A linear analogy $a : a^* :: b : b^*$ holds in an SGNS embedding with no reconstruction error if and only if*

$$\frac{p(w|a)}{p(w|a^*)} = \frac{p(w|b)}{p(w|b^*)} \tag{14}$$

for all words w in the vocabulary.

The above theorem and corollary only holds in embeddings without any reconstruction error. That means that they only hold in SGNS embeddings if there is no error in Equation (3) and no error in Equation (4) for GloVe embeddings. However, the authors of [10] also prove that the above statements are robust to noise since the more frequent a word pair is the lower the reconstruction error associated with that pair.

4.6 Limitations of Analogies in Measuring Embedding Quality

Understanding a word analogy requires knowledge of at least two different aspects of each word, which is a desirable property for a word embedding to have. We want word embeddings to understand as much as possible when it comes to what words mean. A test set based on this idea was introduced in [30, 32] and quickly became a popular way of measuring embedding quality. The test consists of analogies where the task is to find the missing fourth word, e.g.

[4]We present a less general version of the theorem here, for the full version and proof see the original paper [10].

Kind	Category	Example
Semantic	Common capital city (506)	`Berlin` : `Germany` :: `Stockholm` : `Sweden`
	All capital cities (4524)	`Abuja` : `Nigeria` :: `Astana` : `Kazakhstan`
	Currency (866)	`Denmark` : `krone` :: `Brazil` : `real`
	City-in-state (2467)	`Phoenix` : `Arizona` :: `Dallas` : `Texas`
	Man-Woman (506)	`brother` : `sister` :: `sons` : `daughters`
Syntactic	Adjective to adverb (992)	`calm` : `calmly` :: `cheerful` : `cheerfully`
	Opposite (812)	`aware` : `unaware` :: `known` : `unknown`
	Comparative (1332)	`young` : `younger` :: `bad` : `worse`
	Superlative (1122)	`dark` : `darkest` :: `easy` : `easiest`
	Present Participle (1056)	`go` : `going` :: `walk` : `walking`
	Nationality adjective (1599)	`France` : `French` :: `Germany` : `German`
	Past tense (1560)	`dancing` : `danced` :: `falling` : `fell`
	Plural nouns (1332)	`mango` : `mangoes` :: `woman` : `women`
	Plural verbs (870)	`increase` : `increases` :: `write` : `writes`

Table 2: The categories of the standard test set introduced by [30, 32], split into syntactic and semantic groups. An example is provided for each category and the total number of analogies per category is shown in parenthesis.

what is x in `moving` : `moved` :: `paying` : x. Each analogy is scored as either correct or incorrect, synonyms are considered incorrect. In the example the answer is expected to be $x =$ `paid`. The analogies are split into semantic (e.g. "Man-Woman") and syntactic (e.g. "Adjective to adverb") categories and can be found in Table 2.

Accuracies on the test set are often quoted when new algorithms (see e.g. [34, 2]) or configurations (for example [28]) are tried. To mention one of many examples, one 300-dimensional GloVe embedding had an accuracy of 75.0% (81.9% on the semantic and 69.3% on the syntactic sets) when it was introduced [34].

Unfortunately, despite being very popular and used by many researchers, a quick glance at the test set reveals errors and ambiguities that should make one hesitant to blindly trust the results.

Few Dimensions and Overfitting

The first issue is one of dimensionality. In the test introduced by [32] there are only 5 semantic categories being tested, of which two overlap completely, together with 9 syntactic categories. Word embeddings are commonly 300-dimensional, because empirical evidence shows diminishing returns for higher dimensions (see e.g. [34]). How come only 300 dimensions are enough? Surely natural language is more than 300-dimensional? By a rough count the standard test set only covers 26 dimensions, two for each analogy type (remember the parallelogram in Figure 1). This reasoning leads to the question: would 300 dimensions be enough if we had more test categories?

Same	dance, decrease, describe, enhance, feed, generate, hit, implement, increase, jump, listen, look, move, pay, play, predict, read, say, scream, sell, sleep, slow, spend, strike, think, vanish, walk
Different	fall, fly, go, hide, know, run, see, shrink, sing, sit, swim, take, write

Table 3: The verbs from the "Past tense" category separated on the condition that they have the same or different simple past and past participle forms.

In addition to the above, a quarter[5] of all analogies are in the category relating capital cities to their respective countries risking that the embeddings are overfitting on that category. The result is even lower dimensionality of the results.

Furthermore, contrary to common practice in the field of machine learning, there is no standardized split into different sets for training, validation and testing. As has been noted in the similar case of word similarity tasks: "Therefore, optimizing the word vectors to perform better at a word similarity task implicitly *tunes on the test set* and overfits the vectors to the task." [13]

Ambiguous Questions

Dimensionality aside, many analogies are completely ambiguous! Ambiguity appears for analogies $x : y :: z : w$ where y has more than one valid interpretation. There is no way to know which of the senses is meant from a single example. Many such examples can be found in the category "Past tense" where x and y are the present participle and the simple past of the same verb (and similarly for z and w), e.g. `dancing` : `danced` :: `jumping` : `jumped`. The problem is that for many of the verbs used the simple past and the past participle are identical, while they differ for some. For example, compare *dance-danced-danced* to *hide-hid-hidden*. Thus the analogy `dancing` : `danced` :: `hiding` : x has solutions $x =$ `hid` and $x =$ `hidden`, both equally valid from only observing this single example. Of the 40 verbs included, 27 have the same simple past and past participle, leading to 351 / 1560 (22.5%) analogies in that category being ambiguous[6]. The verbs are listed in Table 3.

Further ambiguities can be found in the "Nationality adjective" category where the task is to complete analogies of the form $c_1 : a_1 :: c_2 : a_2$ where c_i and a_i are the country together with the corresponding adjectival form of pair i, e.g. `Sweden` : `Swedish` :: `Germany` : `German`. For most countries, the adjectival form, the demonymic form and the name of the dominant language all coincide, such as for Germany, *German is the language spoken in Germany by Germans*

[5]The categories "All capital cities" and "Common capital city" contain 4524 and 506 analogies respectively out of 19544 in total, which is 25.7%. Note that "Common capital city" is a subset of "All capital cities".

[6]A total of 40 verbs, of which 27 have the same simple past and past participle and 13 that have different forms yields $27 \cdot 13 = 351$ ambiguous analogies out of $40 \cdot 39 = 1560$ in total.

Country	Adjective	Demonym	Language
Albania	Albanian	Albanian	Albanian
Germany	German	German	German
Israel	Israeli	Israeli	**Hebrew**
Mexico	Mexican	Mexican	**Spanish**
Slovakia	Slovakian	Slovak(ian)	**Slovak**
Spain	Spanish	**Spaniard**	Spanish

Table 4: Some countries and their corresponding demonymic form, adjectival form and dominant/official language from the "Nationality adjective" category.

Rank	Answer	Score
1	`Belarusian`	1.009
2	`Belarussian`	0.986
3	`Ukrainian`	0.964
4	`Russian`	0.960
5	`Moldovan`	0.928
14	`Belorussian`	0.868

Table 5: The output from the model trained in [30, 31] on the analogy question `Sweden` : `Swedish` :: `Belarus` : x using the 3CosMul method. The test set expects the answer $x =$ `Belorussian`.

in German cities. There are however some exceptions as in the cases with (1) Slovakia where the name of the language (*Slovak*) differs from the adjectival form (*Slovakian*) and (2) Spain where the adjectival from (*Spanish*) and the demonym (*Spaniard*) differ. Analogies containing these words suffer from the same problems as those in the "Past tense" category. A few examples can be found in Table 4.

There is also the tangentially related issue with the country Belarus. The most common[7] adjectival form is *Belarusian* and not *Belorussian* as is included in the test set. This causes models to achieve a lower score than they should. Indeed, the model produced by the original authors [31] answers the analogy `Sweden` : `Swedish` :: `Belarus` : x correctly with $x =$ `Belarusian`, but is considered incorrect (see Table 5).

No Flexibility

The third and final point is that of flexibility and human judgement. In contrast to the similarity tasks (e.g. [22, 20]) no human evaluation has been performed on the analogies. The score for the similarity tasks is based on how much the similarities computed with the embedding correlate to human judgement, resulting in a high flexibility. The word analogy task, however, is binary. The

[7] According to both the corpus used in [30, 31] and Google Ngram Viewer (`https://books.google.com/ngrams`) [29].

answer is either correct or incorrect whereas a human might deviate from the "correct" answer, as in the case with Belarus.

Evaluating Analogies

Word analogies can be – as was described above – used to evaluate word embeddings. But the opposite is also true; word embeddings can be used to evaluate word analogies. By using the score function S from Equation (11) we can measure how good an analogy is in a given embedding.

In light of the issues highlighted in this section we evaluate the analogies of the standard test set of [31] in Section 7.4.

5 Bias in Word Embeddings

Bias, in the context of word embeddings, is the systematic tendencies of word embeddings to exhibit inappropriate[8] relations. For example, the word vector `man` being more similar to `engineer` than `woman` is an instance of gender bias. In general, this is something to be avoided in order to not reinforce gender stereotypes.

Word analogies played a significant role during the discovery of biases in word embeddings. In [3] it was found, by using word analogies, that many previously published embeddings were "blatantly sexist", illustrated by the analogy "man is to computer programmer as woman is to homemaker". By creating methods based on word analogies they were able to quantitatively show that these biases exist.

5.1 Methods For Measuring and Removing Bias

How can we measure how biased word embeddings are? It turns out that there are quite a few ways. The realization that vector differences can contain enough information to encode analogies was most likely the main motivation that lead the authors of [3] to think of meaning as subspaces. If `man` : `king` :: `woman` : `queen` holds, then the vector offset $g = \vec{\texttt{woman}} - \vec{\texttt{man}}$, and thus the subspace (line) spanned by g, should capture this gender information.

Using a Bias Subspace

Using the idea of a gender line, we can measure gender bias in words by projecting them onto the line; a method that was introduced in [3] as a way to measure gender bias in occupations. They used the line spanned by $\vec{\texttt{woman}} - \vec{\texttt{man}}$ and found that the most extreme female occupations for the embedding studied were *homemaker*, *nurse* and *receptionist* while the most male occupations were *maestro*, *skipper* and *protege*.

[8] This is of course very subjective, but the variants discussed in this text are generally considered unwanted.

Concretely, and more generally, to use this method one starts from a single seed pair (x, y) that differs only by the bias one wants to capture. Using our example of *man* and *woman* – two words similar in almost every way except for the gender – we should expect the difference $\vec{\texttt{woman}} - \vec{\texttt{man}}$ to express the gender difference between the words. Let g be the (normalized) difference $(y-x)/\|y-x\|$. Given a word w we compute its bias score by projecting its vector onto g, i.e. $\langle w, g\rangle g$. The further along the positive g direction the word vector lies the more biased the word is towards y and the further along the negative g direction the more biased it is towards x. The actual value of $\langle w, g\rangle$ is only interesting in a comparative sense and can be used to find the most extreme words, like what was described above regarding occupations.

In order to make the subspace more stable in case the seed words have multiple sense, one can instead collect many pairs of seed words and perform Principal Component Analysis (PCA) on the pairs (as [3] did). For example, instead of only using *man* and *woman*, for which *man* can be used as a verb in *Man the stations!* and as a part of the interjection *Oh man!*, we add other similar pairs like *father-mother*, *he-she* etc. The more pairs we add the better we can isolate the gender components with the PCA.

For seed pairs $P = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, start by computing the mean $\mu_i = \frac{x_i - y_i}{2}$ of each pair (x_i, y_i). Then compute the matrix C by

$$C = \sum_{i=1}^{n} \sum_{w \in \{x_i, y_i\}} (w - \mu_i)^\top (w - \mu_i)/2$$

and perform singular-value decomposition on C. The top k (empirically chosen) components will make up the basis of the bias subspace. In most cases, a single component and thus a one-dimensional subspace seems to be enough to capture the bias. A one-dimensional gender subspace generated in this way was used in [3] to remove bias from a word embedding, a method called debiasing which we will come back to later.

Building on the idea of projecting words onto a bias line one can compute how much bias a set of words contain. It too was introduced in [3] and works as follows. The direct bias of a set of words N according to a subspace spanned by a single vector g is defined as

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} |\cos(w, g)|^c \tag{15}$$

where c is a parameter that tunes the strictness of the bias measurement.

The term $|\cos(w, g)|$ measures how much of w that lies in g. However, that is not clear from the equation itself. We therefore suggest that one should simplify and generalize (15) to the following form, for a subspace spanned by orthonormal vectors $B = (b_1, \ldots, b_n)$,

$$\text{DirectBias}_c = \frac{1}{|N|} \sum_{w \in N} \left(\frac{\|w_B\|}{\|w\|} \right)^c \tag{16}$$

where w_B is the orthogonal projection of w onto B, i.e.

$$w_B = \langle w, b_1 \rangle b_1 + \cdots + \langle w, b_n \rangle b_n.$$

Note that (16) generalizes (15), for normalized w and g, since

$$\frac{\|\langle w, g \rangle g\|}{\|w\|} = \frac{|\langle w, g \rangle| \|g\|}{\|w\|} = |\langle w, g \rangle| = |\cos(w, g)|.$$

A value of DirectBias_c close to 0 signifies that the set of words N contains very little bias, values closer to 1 means that N almost entirely lies in B.

Using Relative Similarity and Distance

A different way of measuring bias, or rather similarity, is to evaluate how close a word, or a set of words, is to two target sets. The method was introduced in [5] and imitates the Implicit Association Test (IAT) [18], which is used to detect biases such as sexism and racism in humans. For instance, one might want to measure whether European-American names or African-American names are considered more pleasant, something tested both on humans and in word embeddings [18, 5].

There are two main and closely related methods of doing this. Using the terminology in [5], let X and Y be the target sets and A and B the attribute sets. The first method is to measure the *differential association* [5] of the two sets X and Y with an attribute represented by A and B.

$$s(X, Y, A, B) = \sum_{x \in X} s(x, A, B) - \sum_{y \in Y} s(y, A, B) \tag{17}$$

where

$$s(x, A, B) = \text{mean}_{a \in A} \cos(x, a) - \text{mean}_{b \in B} \cos(x, b) \tag{18}$$

In our example, *European-American names* and *African-American names* would be the target sets and *pleasant* and *unpleasant* would be the attribute sets. The value of $s(X, Y, A, B)$ would thus be positive and larger the closer X is to A and Y is to B. The value will be negative if the opposite is true, i.e. X is close to B and Y is close to A. The word embedding tested in [5] considered European-American names to be more pleasant than the African-American ones, which reflects the results from when the IAT was performed on humans in [18].

The second method is to measure the *relative norm distance* [14] of X and Y with a single attribute set A:

$$r(X, Y, A) = \sum_{a \in A} \|a - v_X\| - \|a - v_Y\| \tag{19}$$

where v_X and v_Y are the normalized average vectors of the words in X and Y;

$$v_X = \frac{\sum_{x \in X} x}{\|\sum_{x \in X} x\|}, \; v_Y = \frac{\sum_{y \in Y} y}{\|\sum_{y \in Y} y\|}. \tag{20}$$

A positive value of $r(X, Y, A)$ signifies a bias of A towards X and a negative value a bias towards Y. This version of the test has been used to show a correlation between gender bias in word embeddings and US occupational data [14].

By Generating Analogies

Rather than just drawing inspiration from word analogies, it can be helpful in analyses to generate them. Inspired by the PairDirection objective, the authors of [3] devised a method of generating analogies without any human input based on a pair of words. Given the seed pair (a, b), we seek a new pair (x, y) that tries to maximize the similarity of the directions $a - b$ and $x - y$. The goal of course being that if we find a pair that yields a high similarity, then $a : x :: b : y$ will most likely make sense as an analogy. In order to ensure that the words x and y don't drift too far away in meaning, the condition is added that the distance between them be kept below a certain threshold δ. The search is performed over all possible pairs $x \in V$, $y \in V$, but only the best analogy for each word x is output. The algorithm can thus be summarized as follows.

1. Select a seed pair (a, b) and a threshold δ.
2. For each word x, find the y that maximizes $s_{xy} = \cos(a - b, x - y)$ while satisfying $\|x - y\| < \delta$.
3. Output the pairs from step (2) sorted in descending order according to the similarity s_{xy}.

By using the seed pair (`she`, `he`), [3] generated 150 analogies that they let crowd workers evaluate. They found that 19% of the analogies were considered to contain gender stereotypes[9].

Removing Bias

Suppose that we have an embedding with bias, what should we do about it? We would like to use it, but without the bias. Two related ideas were had by [3] regarding this. Both start out in the same way[10]: one first identifies the words that are expected to be neutral and the ones one expects to contain bias. For example, one would like *engineer* to be neutral while *man* and *woman* should differ only in gender. Let the set of neutral words be N and the pairs of gendered words be $P = \{P_1, \ldots, P_n\}$.

Using these pairs, one now identifies the gender subspace using the PCA method described earlier, call this subspace B. The goal is to remove the bias from the neutral words by making them lie entirely in $B^{\perp}$, the orthogonal complement of the bias subspace B, while keeping the gendered words gendered. Here the algorithm diverges to *soft* and *hard* debias.

For soft debias, a linear transformation is computed that tries to minimize the projection of neutral words onto the bias subspace while preserving inner

[9] It should be noted that only 10 people were used for the evaluations.

[10] We present a simplified version of the algorithm here, for full generality see [3].

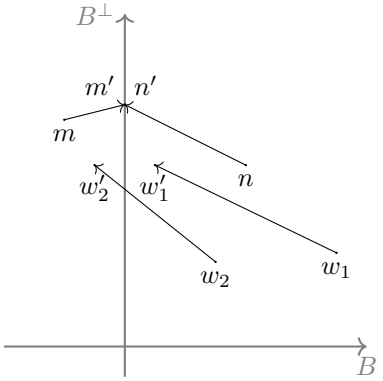

Figure 5: Visualization of the hard debiasing algorithm. The words w_1 and w_2 are gendered while n and m are neutral. The resulting n' and m' contain no bias (they lie outside of B) while w_1' and w_2' are equally biased in opposite directions. Note that n' and m' not necessarily coincide since $B^\perp$ is multidimensional.

products. The result was shown to be inferior to the hard debias algorithm, so we will focus on that instead.

Hard debiasing instead matches the intuition described above. Neutral words are simply projected onto the orthogonal complement $B^\perp$ of the bias subspace and normalized to unit length. The paired gender words are instead centered around $B^\perp$ and also normalized to unit length. By performing this projection one completely removes any influence of the subspace B from the neutral words N, thus resulting in an embedding without that particular kind of bias. Figure 5 illustrates this version of the algorithm.

While intuitively simple, the hard debiasing algorithm is very effective at eliminating bias while preserving the accuracy on the three benchmarks used in [3]. Indeed, they even showed that it eliminates indirect bias in words such as *softball* and *football*. Unfortunately, in order to debias an embedding one has to find the set of neutral words N, which might be infeasible for some embeddings and languages because of its size. Although we are not aware of any such tests, it does seem likely that hard debiasing would remove any bias from downstream tasks and we recommended debiasing as a precaution before using word embeddings.

5.2 New Methods for Measuring Bias

Some of the methods discussed in the previous section are generally used with a discrete result; they simply return the most extreme words, such in the case of projecting words onto a line, or the best generated analogies (see also Section 4.6 for a longer discussion on this topic). This discretization of the answer makes the embeddings seem overly sexist. There might not be a big difference between the similarities of *man* and *engineer* and *woman* and *engineer*. We know that

there will probably be a difference, since in general, more men than women are engineers[11] and the word *engineer* will thus appear more often near words such as *he* and *him* than *she* and *her*. This association will make *engineer* more male than female, albeit only slightly, which will show up in the results. It is therefore important to include the values in the results and not just the most extreme words.

Using analogies in the ways discussed so far has other problems as well. Let us first consider how the original statement, "man is to computer programmer as woman is to homemaker", might be misleading. Something we did not mention in Section 4 is that the input words are excluded when one searches for the last word in an analogy. For example, in the analogy question `Sweden` : `Stockholm` :: `Germany` : x, we search for the word x *different from Sweden, Stockholm and Germany* that best satisfies the analogy. Under normal analogy solving, this does not constitute a problem, so it is unclear why the input words are excluded. Consider the analogy $a : a^* :: b : b^*$ and the objective 3CosMul. We have seen in Section 4 that the analogy $a : b :: a^* : b^*$ should then hold by symmetry. Further consider the factors $\cos(a^*, b^*)$, $\cos(b, b^*)$ and $1/\cos(a, b^*)$ of 3CosMul[12]. We want all of these to be high for an analogy to hold, and if we find a good b^*, they will be. Replacing b^* with any of a, a^* or b will at most yield two high values out of three, which can be seen by looking at the arrows in Figure 2. To continue the above example, suppose $x =$ `Stockholm`. It is trivially maximally similar to itself, so $\cos(a^*, x)$ is high; similar to *Sweden* since both concern Sweden and things Swedish, so $1/\cos(a, x)$ is low and finally not similar at all to *Germany*, so $\cos(b, x)$ is low.

This of course also ties into the problems that lead us to introduce the score S in Section 4.3. If we asked someone to answer the analogy question "man is to computer programmer as woman is to *what*?" we would probably not get a clear answer, because it is not a good analogy to begin with. In cases like this, the option of picking b^* equal to either a^* or b starts to look much more inviting, since at least one term will be maximized in the objective. Furthermore, if there is a high similarity between a and b the lower bound from Equation (12) becomes high enough that a^* turns into the most natural answer. As a concrete example, `he` : `engineer` :: `she` : x would most likely have the answer $x =$ `engineer` rather than any other, possibly sexist, occupation. We provide empirical evidence of this in Section 7.3 by performing an analysis on stereotypical occupations.

Considering all of the above, we thus recommend that if one includes analogies as a part of an analysis then make sure (1) that the input words are allowed to appear in the solution and (2) that more than one answer is considered, ideally displaying the actual scores of the solutions.

Generating Analogies

In addition to the problems mentioned above, the method of generating analogies from Section 5.1 has a few downsides. Most important is the fact that the

[11] This is true for Sweden at least [38].

[12] It works similarly for 3CosAdd.

base objective is PairDirection, which as the authors themselves noted, only takes the direction into consideration. This problem is partly solved by the check $\|x - y\| < \delta$, which is an ad hoc solution that does have some justification outlined in the section introducing the method. Ideally, the objective should be replaced by one based on 3CosMul and the parameter δ should be removed. However, 3CosMul was disregarded by the authors because it was made to find solutions to analogies and because it is not symmetric in x and y. We will now address these short-comings and introduce a new method based on the function S from Section 4.3.

We start by giving another reason why the plain 3CosMul is insufficient. In step 2 of the original algorithm, we first pick a word x and then find the best y under our objective. But 3CosMul was designed under the implicit assumption that the analogy is supposed to make sense! Only a few words in the vocabulary can produce valid starting triples of analogies, so the objective breaks down even before we start searching for y. We introduced the score S to solve this particular problem; by including all of the similarities we make sure that the starting triple makes sense. It also contains all possible symmetries. Finally, it removes the need for the check $\|x - y\| < \delta$ since $\cos(x, y)$ is included in the expression[13].

By including the check from Equation (12) we make sure that we only return analogies that are valid according to the embedding. Our version of the algorithm thus looks like follows.

1. Select a seed pair (a, b).
2. For each word x, find the y that maximizes $S(a, x, b, y)$.
3. Output the pairs (x, y) from step (2) satisfying

$$S(a, x, b, y) \geq \max\{\cos(a, x), \cos(a, b), \cos(b, y), \cos(x, y)\}$$

sorted in descending order according to the score $S(a, x, b, y)$.

Finally, make sure to include the score in the results to prevent the problems of discretization discussed previously. We use this version of the algorithm to generate analogies for English and Swedish in Section 7.4.

Cross-Language Methods

In the introduction and outline we introduced the interest in comparing English and Swedish word embeddings to see if one language is more biased than the other. One can compare the results of all the previously discussed methods and algorithms and try to qualitatively check whether they are more or less biased. Ideally, however, a method that quantitatively compares bias across embeddings should exist. We now take the first steps towards such a method.

All of the previously mentioned methods do measure bias, but mostly for a single language or corpus. It is possible, as in [14], to measure changes over time

[13] Note that $\|x - y\| < \delta$ is equivalent to $\cos(x, y) > 1 - \delta^2/2$.

for which the problems exist to a lesser degree. However, it is much harder to measure differences between languages since the values of the different measures depends heavily on: the gender subspace g and the choice of neutral words N in the case of direct bias; the target and attribute sets X, Y, A, B in the case of relative cosine similarity; the choice of target sets and attribute set X, Y, A in the case of relative norm distance; and the bias line g in the case of projecting words onto a line. The more words we have to pick the harder it will be to make a reasonable comparison across languages.

In order to derive a method for comparison, let us assume that we have an embedding without any gender bias. Let X be a set of female words and Y as set of male words, then we would, for instance, expect $s(X, Y, A, B) = 0$ and $r(X, Y, A) = 0$ for all attribute sets $A, B \subseteq V$ consisting of neutral words. Consider a pair of words $(x, y) \in X \times Y$. We consider this pair *matched* if the words only differ in gender. Thus, for any matched pair of female-male words $x \in X$ and $y \in Y$, the solution to the analogy $x : w :: y : z$ should be $z = w$ for all $w \in V$. For example, if there was no bias *nurse* would be equally male and female and the analogy *she* is to *nurse* as *he* is to x would have the solution $x = \texttt{nurse}$.

Using this information, let P be a set of matched gender[14] pairs, e.g. *he* and *she* is considered one such pair. Define *paired similarity* as

$$\text{PairedSim}_P = \text{mean}_{(x,y)\in P} \cos(x, y). \tag{21}$$

We claim that the value of the paired similarity can be viewed as a proxy for bias in a word embedding.

Proposition 5.1. *The value of* PairedSim_P *is a good proxy for bias in the sense that, if* $\text{PairedSim}_P = 1$ *then* $s(X, Y, A, B) = 0$, $r(X, Y, A) = 0$ *and* $x : w :: y : z$ *has solution* $w = z$ *for all* $A, B \subseteq V, w \in V$ *where* $X = \{x \mid (x, y) \in P\}$ *and* $Y = \{y \mid (x, y) \in P\}$. *In other words, the higher the paired similarity the less biased the embedding.*

Proof. Assume that $\text{PairedSim}_P = 1$ holds, then $\cos(x, y) = 1$ for all $(x, y) \in P$. Thus, if all vectors are normalized, $x = y$. It follows that $s(x, A, B) = s(y, A, B)$ for all $(x, y) \in P$ and $A, B \subseteq V$, and thus $s(X, Y, A, B) = 0$. The result $r(X, Y, A) = 0$ follows similarly. Finally, let $(x, y) \in P$ and $w \in V$ and look at the analogy $x : w :: y : z$. Using 3CosAdd as the objective function (3CosMul works similarly), the solution is

$$\begin{aligned} z &= \text{argmax}_{z\in V} \cos(z, w) + \cos(z, x) - \cos(z, y) \\ &= \text{argmax}_{z\in V} \cos(z, w) + \cos(z, x) - \cos(z, x) \\ &= \text{argmax}_{z\in V} \cos(z, w) \\ &= w. \end{aligned}$$

□

[14] We will focus on gender bias, but it is straightforward to generalize the method to other types of bias.

Although one still needs to provide a set of matched gender pairs in order to evaluate the bias of an embedding, this set is much more tangible than for instance the set of neutral words used to measure direct bias. A small set is easy to translate and check for polysemy (i.e. words with multiple senses). We therefore propose the following test for comparing bias in two embeddings, in two different languages l_1 and l_2.

1. Find sets of matched gender pairs P_1 and P_2 in l_1 and l_2 respectively. Each pair in $p \in P_1$ should translate to one pair in $q \in P_2$ and vice versa (a bijection). Furthermore, the translation should be unique in the sense that there should be no other (or only negligible other) possible translations of p and q in l_1 and l_2.

2. Compute the cosine similarity of each pair, i.e. let $x_i = \cos(p_i)$ and $y_i = \cos(q_i)$, where $q_i \in P_2$ is the translation of $p_i \in P_1$.

3. Perform a one-sided sign test on the pairs of cosine similarities $\{(x_i, y_i)\}_{i=1}^n$.

4. A significant result can thus be interpreted as one embedding being more biased than the other.

Since common words might lie closer together in some embeddings than others it is helpful to divide the values in the test above with baseline values μ_1 and μ_2 computed in step 2 below. As cosine similarity has a range of $[-1, 1]$ we instead use the regular Euclidean distance[15]. The rest of the algorithm stays the same.

1. Find sets of matched words W_1 and W_2 in l_1 and l_2 respectively. The words should cover as many aspects of the language as possible.

2. Compute the baseline distance $\mu_i = \text{mean}_{(u,v)\in W_i} \|u - v\|$.

3. Find sets of matched gender pairs P_1 and P_2 in l_1 and l_2 respectively.

4. Compute the distance between the elements of each pair and divide by μ_i, i.e. let $x_k = \|p_{k,1} - p_{k,2}\|/\mu_1$ and $y_k = \|q_{k,1} - q_{k,2}\|/\mu_2$, where $(q_{k,1}, q_{k,2}) \in P_2$ is the translation of $(p_{k,1}, p_{k,2}) \in P_1$

5. Perform a one-sided sign test on the results from step 4 on the pairs $\{(x_k, y_k)\}_{k=1}^n$.

6. A significant result can thus be interpreted as one embedding being more biased than the other.

Tests of this form were performed on a few Swedish and English embeddings and the results can be found in Section 7.3.

[15]Remember that $\|x - y\| = \sqrt{2(1 - \cos(x, y))}$.

6 Swedish Word Embeddings

Not a lot can be found in the literature regarding word embeddings for the Swedish language. There are some pre-trained embeddings available online as a part of different sets of embeddings for languages other than English, most of which are trained on Wikipedia data or data collected for the Common Crawl project[16] (see e.g. [1, 17]).

We did find one reference specifically aimed at producing Swedish embeddings. However, it mostly concerns preliminary empirical tests at producing good embeddings [11]. Moreover, the embeddings were trained on a fairly small corpus. We therefore trained our own in combination with using the pre-trained embedding from [17].

We use the Swedish Culturomics Gigaword Corpus [9] to train our embeddings[17]. The corpus contains data from different sources spanning the years 1950-2015. It is a mix of five different categories: newspaper, fiction, government, social media and science. In total, the corpus contains around 1 billion tokens, 152 million of which comes from Swedish Wikipedia.

The methods we used to train our embeddings were GloVe and fastText, we excluded word2vec since it has largely been superseded by fastText which uses the same underlying algorithm. Moreover, by using subword information fastText should be able to handle languages with compound words and inflections. Indeed, it was shown that the embeddings produced for German – which will act as a substitute for Swedish in this discussion as they are fairly similar – did perform very well and that compound nouns were correctly understood [2, 17].

We thus end up with the following embeddings:

- The pre-trained fastText embedding from [17] which was trained on the Common Crawl corpus and Wikipedia.
- A 300-dimensional GloVe model trained on the subset corresponding to the years 1980-2015 of the Swedish Gigaword corpus. The vocabulary size was limited to 1 000 000 words.

The GloVe embedding uses the default values for all parameters unless explicitly stated. Some limits were however put in place to limit the vocabulary size to make it feasible to train the embedding on a modest PC.

Due to the lack of tests for Swedish embeddings we have not performed any on the trained embedding. However, the results from the analyses in Sections 7.3 and 7.4 show a satisfactory understanding of similarities and analogies.

[16] `https://commoncrawl.org/`

[17] The corpus is available from Språkbanken at `https://spraakbanken.gu.se/swe/resurs/gigaword`.

7 Results

For our evaluation of gender bias in Swedish and English we have performed some qualitative tests based on methods used in previous work. This section will outline the embeddings and data used for the tests followed by the results.

Finally we also evaluate the standard test set for word embeddings based on analogies that was described in Section 3.3.

7.1 Corpora and Embeddings

For Swedish we used the embeddings described in Section 6 and for English we use the following embeddings.

- The 300-dimensional WORD2VEC embedding based on the Google News corpus from [31].
- The 300-dimensional FASTTEXT embedding from [17] trained on the Common Crawl corpus and Wikipedia.
- The 300-dimensional pre-trained GLOVE embedding from [34], which was trained on Wikipedia data and Gigaword 5.

Before using them, all word vectors were normalized to unit length, as is common practice. All embeddings and corpora can be found by following the links provided in Appendix A.4.

7.2 Gender Words and Words for Occupations

For the sign test based on paired similarity described in Section 5.2 we use the following matched gender pairs.

English		Swedish	
she	he	hon	han
her	him	henne	honom
daughter	son	dotter	son
daughters	sons	döttrar	söner
sister	brother	syster	bror
sisters	brothers	systrar	bröder
mom	dad	mamma	pappa
moms	dads	mammor	pappor
women	men	kvinnor	män
girl	boy	flicka/tjej	pojke/kille
girls	boys	flickor/tjejer	pojkar/killar
girlfriend	boyfriend	flickvän	pojkvän
aunt	uncle	faster/moster	farbror/morbror
grandmother	grandfather	farmor/mormor	farfar/morfar

In some cases there are multiple Swedish words that translate to a single English word, e.g. *farmor* (paternal grandmother) and *mormor* (maternal grandmother). In those cases we use the averages of the Swedish vectors for the measurements.

We use the list of occupations provided by [14] for English. For Swedish we started with the lists of occupations found on English[18] and Swedish[19] Wiktionary together with a list of vocations produced by Språkbanken [40]. This list was then reduced by eliminating out-of-vocabulary words. Further manual editing was performed to eliminate words with a clear gendered definition, such as *lärarinna* (female teacher); archaic or dated words, such as *sumprunkare*; words with homonyms, e.g. *torped* (hitman and torpedo); and some hyponyms where a clear hypernym exists in the vocabulary, e.g. we exclude *latinlärare* (Latin teacher) since *språklärare* (language teacher) is included. The final list is included in Appendix A.3. Future work should include a mapping to SSYK2012 (Swedish Standard Classification of Occupations [37]) codes to enable correlational analyses akin to the one performed by [14].

7.3 Gender Stereotypes in Swedish and English

We first note that we ran the tests on the 1 000 000 most common words of each embedding for performance reasons. This could not significantly affect the results since the words found after those are mostly noise and not really words at all.

Biased Occupations

Firstly, we computed the most extreme occupations in the Swedish embeddings according to the relative norm distance[20] method described in Section 5. We found that the most extreme words on both ends of the spectrum were quite stereotypical occupations for the two genders. On the male side we find occupations such as *civilingenjör* (civil engineer), *statsman* (statesman) and *fysiklärare* (physics teacher). On the female side we find *barnmorska* (midwife), *hembiträde* (housemaid) and *sjuksköterska* (nurse). The results are shown in Table 6 and Table 7 for the male and female occupations respectively[21].

Secondly, we took four of the most common male occupations and four of the most common female occupations according to the tests above and queried the embeddings for solutions to analogies of the form `she` : w :: `he` : ? and `he` : w :: `she` : ? for each occupation w. For Swedish we used the translations `hon` : w :: `han` : ? and `han` : w :: `hon` : ? respectively. We allowed the algorithm to output the input words, if they seemed fitting. The results for the English WORD2VEC embedding and the Swedish FASTTEXT embedding are shown in

[18] `https://en.wiktionary.org/wiki/Category:sv:Occupations`

[19] `https://sv.wiktionary.org/wiki/Kategori:Svenska/Yrken`

[20] We also ran the computation for the other methods, which produced similar results.

[21] We have simply recreated the analysis from [3] here without considering the actual difference between the male and female occupations. This is contrary to what we described at the beginning of Section 5.2 and needs to be followed up in the future.

GloVe	fastText
advokat	mekaniker
diktator	rörmokare
vaktmästare	grävmaskinist
bankdirektör	byggare
ämbetsman	trumslagare
rabbin	biskop
biskop	ingenjör
lokförare	kolare
trumpetare	tränare
endokrinolog	diktator

Table 6: Top male occupations in the different Swedish embeddings using the relative norm distance method.

GloVe	fastText
barnmorska	barnmorska
prostituerad	mannekäng
sjuksköterska	prostituerad
tandhygienist	sjuksköterska
hemkunskapslärare	uska
dietist	hembiträde
djuruppfödare	undersköterska
bärplockare	flygvärdinna
hembiträde	kvinnoläkare
ufolog	kontorist

Table 7: Top female occupations in the different Swedish embeddings using the relative norm distance method.

Tables 8 and 9. It is clear that the intuition from Section 5.2 was correct, only two of the analogies resulted in stereotypes if we allowed the method to output the input words. Those analogies are "*he* is to *mechanic* as *she* is to *beautician*" and "*he* is to *architect* as *she* is to *interior designer*" and appear in the English word2vec embedding. In terms of the score, this means that all analogies except the two above failed the bounds check from Equation (12) and were considered nonsensical to the embeddings. The results for the remaining English and Swedish embeddings were identical to the Swedish fastText embedding in that all analogies returned the input occupations[22].

Biased Analogies

We also generated analogies using the modified algorithm of Section 5.2, for both Swedish and English. Due to the quadratic nature of the algorithm we

[22]The complete results are included in Appendix B.

w	she : w :: he : ?	he : w :: she : ?
nurse	nurse	nurse
midwife	midwife	midwife
librarian	librarian	librarian
housekeeper	housekeeper	housekeeper
retired	retired	retired
mason	mason	mason
mechanic	mechanic	beautician
architect	architect	interior_designer

Table 8: The result of completing the analogies for English occupation words using 3CosMul in the English word2vec embedding.

w	hon : w :: han : ?	han : w :: hon : ?
barnmorska	barnmorska	barnmorska
mannekäng	mannekäng	mannekäng
sjuksköterska	sjuksköterska	sjuksköterska
uska	uska	uska
mekaniker	mekaniker	mekaniker
trumslagare	trumslagare	trumslagare
rörmokare	rörmokare	rörmokare
byggare	byggare	byggare

Table 9: The result of completing the analogies for Swedish occupation words using 3CosMul in the Swedish fastText embedding.

had to limit the vocabulary size to make the process feasible. We therefore use the 25 000 most common words for this part. To make the process even more feasible, we only ran the algorithm for some of the embeddings. Analogies for which the inequality in Equation (12) did not hold were discarded. For English we used the seed pair (`she`, `he`) and for Swedish the pair (`hon`, `han`) which is the translation of the English seed pair.

The resulting analogies for the English embeddings are shown in Tables 10, 11 and 12. For Swedish the same can be found in Tables 13 and 14.

For English we first of all note that none of the top 30 analogies for any of the embeddings contain stereotypes. All of them are words that have appropriate gender differences, like *spokeswoman* and *spokesman*. Looking at the WORD2VEC embedding in particular we find the pair *breast cancer* and *prostate cancer* also found by [3] in the same embedding. No tests have been performed to quantitatively evaluate how good the generated analogies are.

Curiously, many analogies from the WORD2VEC embedding involve names such as Alison and David. On the other hand, the GLOVE embedding generated analogies with last names such as Hingis, Sampras, Kuznetsova and Safin, all tennis players.

Among the Swedish analogies we mostly find words corresponding to ones from the English embeddings, but due to the higher number of noun forms in Swedish (number, definiteness and the possible additional genitive suffix) many of the analogies share the same base lemmas. There is one stereotype among the top 30 analogies produced by the FASTTEXT embedding: *klänning* (dress) vs. *skjorta* (shirt). The Swedish GLOVE embedding produced only 17 analogies, likely because of the high similarity of the seed words (0.9142 shifted cosine similarity).

Cross-Language Sign Test

For our direct comparison of gender bias in Swedish and English word embeddings we performed the sign test described in Section 5.2. To make the comparison as fair as possible we only compared embeddings generated via the same method and trained on similar corpora. The resulting pairs were thus the English and Swedish FASTTEXT embeddings as well as the English and Swedish GLOVE embeddings. As can be seen in the table below, none of the differences were significant.

Embedding Type	p
FASTTEXT	0.211975
GLOVE	0.211975

Table 15 summarizes how similar each pair of gender words are in the Swedish and English embeddings.

she	he	Score
herself	himself	1.0051
Her	His	1.0049
she	he	1.0000
She	He	0.9959
her	his	0.9868
spokeswoman	spokesman	0.9747
sisters	brothers	0.9633
woman	man	0.9595
actress	actor	0.9471
Ms.	Mr.	0.9444
niece	nephew	0.9422
daughter	son	0.9322
Mrs	Mr	0.9309
sister	brother	0.9302
granddaughter	grandson	0.9253
Ms	Mr	0.9197
Actress	Actor	0.9184
aunt	uncle	0.9138
girl	boy	0.9100
daughters	sons	0.9083
Mrs.	Mr.	0.9067
spokesperson	spokesman	0.9049
heroine	hero	0.9033
mothers	fathers	0.9024
Grandma	Grandpa	0.9022
Katie	Matt	0.8976
queen	king	0.8965
women	men	0.8964
grandmother	grandfather	0.8962
Megan	Matt	0.8958

Table 10: The best generated analogies for the English fastText embedding using the seed pair (`she`, `he`).

she	he	Score
her	his	1.0188
herself	himself	1.0139
she	he	1.0000
niece	nephew	0.9826
actress	actor	0.9646
daughter	son	0.9610
spokeswoman	spokesman	0.9550
granddaughter	grandson	0.9465
woman	man	0.9440
chairwoman	chairman	0.9427
mother	father	0.9409
mrs	mr	0.9382
aunt	uncle	0.9378
sister	brother	0.9374
daughters	sons	0.9320
girl	boy	0.9313
hingis	sampras	0.9307
mom	dad	0.9245
women	men	0.9188
kuznetsova	safin	0.9184
sharapova	federer	0.9157
heroine	hero	0.9122
capriati	agassi	0.9117
seles	agassi	0.9095
girls	boys	0.9095
princess	prince	0.9080
sisters	brothers	0.9041
wta	atp	0.9025
lesbian	gay	0.9020
actresses	actors	0.9016

Table 11: The best generated analogies for the English GloVe embedding using the seed pair (`she`, `he`).

she	he	Score
herself	himself	1.0566
her	his	1.0292
chairwoman	chairman	1.0254
Ms.	Mr.	1.0105
she	he	1.0000
Her	His	0.9976
spokeswoman	spokesman	0.9892
She	He	0.9892
woman	man	0.9876
heroine	hero	0.9729
sisters	brothers	0.9725
actress	actor	0.9641
Ms	Mr	0.9629
Alison	David	0.9568
Actress	Actor	0.9567
queen	king	0.9544
Rebecca	David	0.9537
breast_cancer	prostate_cancer	0.9534
Councilwoman	Councilman	0.9533
daughter	son	0.9519
sister	brother	0.9473
Ann	John	0.9413
Liz	Steve	0.9409
Julie	Steve	0.9408
Melanie	David	0.9399
girl	boy	0.9386
Pamela	David	0.9383
Amanda	Matt	0.9383
Mrs	Mr	0.9377
Katie	Matt	0.9370

Table 12: The best generated analogies for the English WORD2VEC embedding using the seed pair (`she`, `he`).

`hon`	`han`	Score
hennes	hans	1.0303
henne	honom	1.0141
Hon	Han	1.0103
hon	han	1.0000
tjejen	killen	0.9850
tjej	kille	0.9646
systrar	bröder	0.9632
syster	bror	0.9608
systrarna	bröderna	0.9576
dotter	son	0.9497
Hennes	hans	0.9464
kvinnan	mannen	0.9450
dottern	sonen	0.9449
storasyster	storebror	0.9448
flickan	pojken	0.9443
flicka	pojke	0.9429
lillasyster	lillebror	0.9420
tjejerna	grabbarna	0.9339
mamma	pappa	0.9329
damer	herrar	0.9298
döttrar	söner	0.9274
sångerska	sångare	0.9240
tjejer	killar	0.9234
Johanna	Johan	0.9218
farmor	farfar	0.9212
drottning	kung	0.9208
mormor	morfar	0.9200
Anna	Johan	0.9181
flickor	pojkar	0.9177
klänning	skjorta	0.9155

Table 13: The best generated analogies for the Swedish fastText embedding using the seed pair (`hon`, `han`).

`hon`	`han`	Score
hon	han	1.0000
hennes	hans	0.9867
henne	honom	0.9838
dottern	sonen	0.9507
damer	herrar	0.9505
dotter	son	0.9440
mamman	pappan	0.9431
kvinnan	mannen	0.9420
mormor	morfar	0.9326
tjej	kille	0.9289
syster	bror	0.9250
tjejen	killen	0.9230
tjejerna	killarna	0.9209
kvinnor	män	0.9197
systern	brodern	0.9188
flickan	pojken	0.9170
sångerska	sångare	0.9163

Table 14: The best generated analogies for the Swedish GloVe embedding using the seed pair (`hon`, `han`).

Embedding	Base Distance (μ)	Avg. Gender Distance / μ
English		
GloVe	1.3098	0.5440
fastText	1.2997	0.4717
word2vec	1.3234	0.5190
Swedish		
GloVe	1.2689	0.4883
fastText	1.2449	0.4454

Table 15: A summary of the average distance between the gender pairs of the different embeddings. The base distance is the μ from Section 5.2 and the average distance is the mean of the distances between each pair of gender words.

7.4 Evaluating the Test Analogies

Using the score S from Section 4.3 we computed the scores for all analogies included in the test set from [31] – which was described in detail in Section 4.6 – for all English embeddings. The best and worst analogies for each embedding can be found in Tables 16, 17 and 18. We only output one analogy per equivalence class since the scores and lower bounds are invariant under symmetries.

In general, all embeddings scored the analogies from the "Nationality adjective" category very high, around or above 1. Those are analogies of the form discussed in Section 4.6 where we associate countries and their adjectival form. At the other end we find currencies and opposites. It is natural that opposites get a low score because, by definition, the words involved are dissimilar semantically. The similarity between, e.g. *clear* and *unclear*, would be syntactical only. The low score for the currency analogies is most probably a consequence of the low frequency of currency words in the training corpora [10]. Likewise, the high score for analogies involving countries and adjectival forms is likely due to high frequencies in the training corpora, in particular Wikipedia.

We further summarize the results by considering the score of an analogy either high, neutral or low depending on whether

- it was greater than or equal to 1 for "high",
- it was between 1 and the lower bound (see Equation (12)) for "neutral" and
- "low" otherwise.

As can be seen below, many analogies did not pass the threshold from Equation (12).

Embedding	High	Neutral	Low
GloVe	596 (3.0%)	6689 (34.2%)	12259 (62.7%)
fastText	745 (3.8%)	9524 (48.7%)	9275 (47.5%)
word2vec	210 (1.1%)	7065 (36.3%)	12211 (62.7%)

These results can be viewed from two different perspectives: either the test set contains analogies most embeddings consider invalid and should thus be improved or there is still a lot of work to be done before word embeddings truly understand analogies.

8 Discussion and Future Work

Although we have showed that bias is present in both English and Swedish word embeddings we still ask that one errs on the side of caution in interpreting the results. Due to the discretization of vectors one loses many nuances in the result. If used correctly it can however be quite valuable.

Word embeddings can be considered a way of compressing statistical information about a corpus for later study. It is possible to perform different

Analogy	Score	Lower Bound
Italy : Italian :: Sweden : Swedish	1.0802	0.8383
Cambodia : Cambodian :: Italy : Italian	1.0773	0.8750
Norway : Norwegian :: Italy : Italian	1.0714	0.8545
Korea : Korean :: Italy : Italian	1.0710	0.8690
Japan : Japanese :: Ukraine : Ukrainian	1.0703	0.8631
⋮	⋮	⋮
tasteful : distasteful :: possibly : impossibly	0.4225	0.7224
possibly : impossibly :: certain : uncertain	0.4217	0.6734
Korea : won :: Canada : dollar	0.4092	0.6628
USA : dollar :: Korea : won	0.4073	0.6972
clear : unclear :: possibly : impossibly	0.3970	0.7363

Table 16: The best and worst analogies from the test set according to the English fastText embedding using the score function S from Equation (11).

Analogy	Score	Lower Bound
colombia : colombian :: egypt : egyptian	1.0762	0.8792
egypt : egyptian :: croatia : croatian	1.0744	0.8680
australia : australian :: bulgaria : bulgarian	1.0714	0.8685
colombia : colombian :: australia : australian	1.0635	0.8792
peru : peruvian :: egypt : egyptian	1.0613	0.8680
⋮	⋮	⋮
usa : dollar :: korea : won	0.3125	0.6251
korea : won :: macedonia : denar	0.3119	0.6251
armenia : dram :: usa : dollar	0.3079	0.5459
armenia : dram :: korea : won	0.2851	0.6251
possibly : impossibly :: tasteful : distasteful	0.2531	0.6588

Table 17: The best and worst analogies from the test set according to the English GloVe embedding using the score function S from Equation (11).

analyses on the whole corpus without having to store it, something that gets increasingly difficult as more and more data becomes available. Furthermore, performing computations on the embeddings are much more efficient than scanning the whole corpus. For this purpose we believe that word embeddings will be used in the future, where one analyzes trends over time, like [14] did for stereotypes; compares corpora and how language usage differs in, for instance, newspapers and internet forums; and other uses not yet thought of. All of these analyses are orthogonal to the use of word embeddings in NLP, which makes us wonder: is it time to separate word embeddings used for NLP from those used to study language use?

There is still a lot that can be done for Swedish word embeddings. Firstly there needs to exist a standard way of quantitatively evaluating embeddings involving not only word analogies but also word similarities and extrinsic measures (or intrinsic ones like QVec [42]). Secondly, there is a lot of knowledge to be gained from comparing English and Swedish embeddings and more work should be put into that area. Using aligned word vectors [21] would most likely help.

Finally, the topic of word analogies is important since understanding them requires an understanding of the relationships between words, thus showing that word embeddings capture this rich structure. There are a couple of things of particular interest, mainly that there is still work to be done in describing analogies mathematically. Are there ways of computing them that show *how* words relate to each other? Cosine similarity compares vectors elementwise before collapsing the result into a single sum. We believe that the elementwise product holds the key to a better understanding of word similarities and, as a consequence, analogies. Also of interest is to understand what the current way of computing analogies means, work that has already been started with promising results [10].

References

[1] Rami Al-Rfou, Bryan Perozzi, and Steven Skiena. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 183–192, Sofia, Bulgaria, August 2013. Association for Computational Linguistics.

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

[3] Tolga Bolukbasi, Kai-Wei Chang, James Y. Zou, Venkatesh Saligrama, and Adam Kalai. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. *CoRR*, abs/1607.06520, 2016.

[4] John A Bullinaria and Joseph P Levy. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior research methods*, 39(3):510–526, 2007.

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

[6] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

[7] Amit Datta, Michael Carl Tschantz, and Anupam Datta. Automated experiments on ad privacy settings. *Proceedings on Privacy Enhancing Technologies*, 2015(1):92–112, 2015.

[8] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[9] Stian Rødven Eide, Nina Tahmasebi, and Lars Borin. The Swedish Culturomics Gigaword Corpus: A one billion word Swedish reference dataset for NLP, 2016.

[10] Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. Towards understanding linear word analogies. *arXiv preprint arXiv:1810.04882v5*, 2018.

[11] Per Fallgren, Jesper Segeblad, and Marco Kuhlmann. Towards a standard dataset of Swedish word vectors. In *Proceedings of the Sixth Swedish Language Technology Conference (SLTC)*, Umeå, Sweden, 2016.

[12] Manaal Faruqui and Chris Dyer. Community evaluation and exchange of word vectors at wordvectors. org. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 19–24, 2014.

[13] Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. Problems with evaluation of word embeddings using word similarity tasks. *CoRR*, abs/1605.02276, 2016.

[14] Nikhil Garg, Londa Schiebinger, Dan Jurafsky, and James Zou. Word embeddings quantify 100 years of gender and ethnic stereotypes. *CoRR*, abs/1711.08412, 2017.

[15] Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive science*, 7(2):155–170, 1983.

[16] Alex Gittens, Dimitris Achlioptas, and Michael W Mahoney. 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, 2017.

[17] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[18] Anthony G Greenwald, Debbie E McGhee, and Jordan LK Schwartz. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6):1464, 1998.

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

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

[21] Armand Joulin, Piotr Bojanowski, Tomas Mikolov, Hervé Jégou, and Edouard Grave. Loss in translation: Learning bilingual word mapping with a retrieval criterion. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

[22] David A Jurgens, Peter D Turney, Saif M Mohammad, and Keith J Holyoak. Semeval-2012 task 2: Measuring degrees of relational 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 356–364. Association for Computational Linguistics, 2012.

[23] Matthew Kay, Cynthia Matuszek, and Sean A. Munson. Unequal representation and gender stereotypes in image search results for occupations. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3819–3828, New York, NY, USA, 2015. ACM.

[24] Till Alexander Leopold, Vesselina Ratcheva, and Saadia Zahidi. The global gender gap report 2017. World Economic Forum, 2017.

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

[26] Omer Levy and Yoav Goldberg. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180, 2014.

[27] Omer Levy and Yoav Goldberg. Neural word embedding as implicit matrix factorization. In *Advances in neural information processing systems*, pages 2177–2185, 2014.

[28] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

[29] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2011.

[30] Tomáš Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[31] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc., 2013.

[32] Tomáš Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, 2013.

[33] Karin Milles. Feminist language planning in sweden. *Current issues in language planning*, 12(1):21–33, 2011.

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

[35] David E. Rumelhart and James L. McClelland. *Distributed Representations*, pages 77–109. MITP, 1987.

[36] Magnus Sahlgren, RISE Research Institutes of Sweden, ICT, and SICS. The distributional hypothesis. *Italian Journal of Disability Studies*, 20:33, 2008.

[37] Statistics Sweden (SCB). Swedish Standard Classification of Occupations 2012, 2012.

[38] Statistics Sweden (SCB). Anställda (yrkesregistret) 16-64 år efter yrke (ssyk 2012), födelseregion, kön och år. `http://www.statistikdatabasen.scb.se/pxweb/sv/ssd/START__AM__AM0208__AM0208E/YREG53/`, 2018. Accessed: 2019-01-21.

[39] Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. Evaluation methods for unsupervised word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 298–307, 2015.

[40] Språkbanken. Yrkesbeteckningar. `https://spraakbanken.gu.se/swe/resurs/vocation-list`. Accessed: 2018-10-12.

[41] Latanya Sweeney. Discrimination in online ad delivery. *Queue*, 11(3):10, 2013.

[42] Yulia Tsvetkov, Manaal Faruqui, Wang Ling, Guillaume Lample, and Chris Dyer. Evaluation of word vector representations by subspace alignment. In *Proc. of EMNLP*, 2015.

[43] P. D. Turney. Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, 44:533–585, 2012.

[44] P. D. Turney and P. Pantel. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188, 2010.

[45] Claudia Wagner, David Garcia, Mohsen Jadidi, and Markus Strohmaier. It's a man's Wikipedia? assessing gender inequality in an online encyclopedia. In *ICWSM*, pages 454–463, 2015.

[46] Claudia Wagner, Eduardo Graells-Garrido, David Garcia, and Filippo Menczer. Women through the glass ceiling: gender asymmetries in wikipedia. *EPJ Data Science*, 5(1):5, 2016.

[47] Benjamin D. Wasserman and Allyson J. Weseley. ¿qué? quoi? do languages with grammatical gender promote sexist attitudes? *Sex Roles*, 61(9):634, Sep 2009.

A Data and Code

All data composed for this report is, together with the trained Swedish embeddings and all code written, available at `https://precenth.eu/word-embeddings.html`.

A.1 Base Words

The following are the words used to compute the base distance for the embeddings: *måndag* (*Monday*), *tisdag* (*Tuesday*), *onsdag* (*Wednesday*), *torsdag* (*Thursday*), *fredag* (*Friday*), *lördag* (*Saturday*), *söndag* (*Sunday*), *jag* (*I*), *mig* (*me*), *vi* (*we*), *två* (*two*), *tre* (*three*), *fyra* (*four*), *fem* (*five*), *sju* (*seven*), *åtta* (*eight*), *nio* (*nine*), *tio* (*ten*), *elva* (*eleven*), *tolv* (*twelve*), *huvud* (*head*), *tand* (*tooth*), *mun* (*mouth*), *hår* (*hair*), *bil* (*car*), *bi* (*bee*), *geting* (*wasp*), *is* (*ice*), *Afghanistan* (*Afghanistan*), *Aten* (*Athens*), *Australien* (*Australia*), *Bagdad* (*Baghdad*), *Bangkok* (*Bangkok*), *Peking* (*Beijing*), *Berlin* (*Berlin*), *Bern* (*Bern*), *Kairo* (*Cairo*), *Canada* (*Canada*), *Canberra* (*Canberra*), *Kina* (*China*), *Kuba* (*Cuba*), *Egypten* (*Egypt*), *England* (*England*), *Finland* (*Finland*), *Frankrike* (*France*), *Tyskland* (*Germany*), *Grekland* (*Greece*), *Hanoi* (*Hanoi*), *Havanna* (*Havana*), *Helsingfors* (*Helsinki*), *Iran* (*Iran*), *Irak* (*Iraq*), *Islamabad* (*Islamabad*), *Italien* (*Italy*), *Japan* (*Japan*), *Kabul* (*Kabul*), *London* (*London*), *Madrid* (*Madrid*), *Moskva* (*Moscow*), *Norge* (*Norway*), *Oslo* (*Oslo*), *Ottawa* (*Ottawa*), *Pakistan* (*Pakistan*), *Paris* (*Paris*), *Rom* (*Rome*), *Ryssland* (*Russia*), *Spanien* (*Spain*), *Stockholm* (*Stockholm*), *Sverige* (*Sweden*), *Schweiz* (*Switzerland*), *Teheran* (*Tehran*), *Thailand* (*Thailand*), *Tokyo* (*Tokyo*), *Vietnam* (*Vietnam*), *banan* (*banana*), *bananer* (*bananas*), *fågel* (*bird*), *fåglar* (*birds*), *flaska* (*bottle*), *flaskor* (*bottles*), *byggnad* (*building*), *byggnader* (*buildings*), *bil* (*car*), *bilar* (*cars*), *katt* (*cat*), *katter* (*cats*), *färg* (*color*), *färger* (*colors*), *dator* (*computer*), *datorer* (*computers*), *ko* (*cow*), *kor* (*cows*), *hund* (*dog*), *hundar* (*dogs*), *åsna* (*donkey*), *åsnor* (*donkeys*), *dröm* (*dream*), *drömmar* (*dreams*), *örn* (*eagle*), *örnar* (*eagles*), *elefant* (*elephant*), *elefanter* (*elephants*), *öga* (*eye*), *ögon* (*eyes*), *finger* (*finger*), *fingrar* (*fingers*), *get* (*goat*), *getter* (*goats*), *hand* (*hand*), *händer* (*hands*), *häst* (*horse*), *hästar* (*horses*), *maskin* (*machine*), *maskiner* (*machines*), *mango* (*mango*), *melon* (*melon*), *meloner* (*melons*), *mus* (*mouse*), *möss* (*mice*), *apa* (*monkey*), *apor* (*monkeys*), *lök* (*onion*), *lökar* (*onions*), *gris* (*pig*), *grisar* (*pigs*), *ananas* (*pineapple*), *ananaser* (*pineapples*), *råtta* (*rat*), *råttor* (*rats*), *väg* (*road*), *vägar* (*roads*), *orm* (*snake*), *ormar* (*snakes*) .

A.2 Swedish Gender Words

Male words: *han, honom, son, sonen, söner, bror, bröder, kille, killen, killar, killarna, gubbe, gubben, pappa, pappan, pappor, mannen, män, farsa, farsan, pojke, pojken, pojkar, pojkvän, pojkvännen, herrar, herrarna* .

Female words: *hon, henne, dotter, dottern, döttrar, syster, systrar, tjej, tjejen, tjejer, tjejerna, gumma, gumman, mamma, mamman, mammor, kvinnan,*

kvinnor, morsa, morsan, flicka, flickan, flickor, flickvän, flickvännen, damer, damerna .

A.3 Swedish Occupations

The final list of Swedish occupations: *administratör, advokat, agronom, akademiker, aktuarie, allmänläkare, ambassadör, analytiker, antikvarie, apotekare, arborist, arkeolog, arkitekt, arkivarie, artist, astronaut, astronom, auktionsförrättare, badvakt, bagare, bagarmästare, balettdansare, balettmästare, bankdirektör, bankir, banvakt, barberare, bardskärare, barnbibliotekarie, barnläkare, barnmorska, bartender, basist, begravningsentreprenör, bergmästare, bergsfiskal, bergsfogde, beridare, bibliotekarie, bilmekaniker, bilskadetekniker, biografmaskinist, biografpianist, biologilärare, biomaskinist, bioteknolog, biskop, blåsare, bokhandlare, bokhållare, bonde, boxare, brandman, brandsoldat, brevbärare, byggmästare, busschaufför, bussförare, butikschef, byggare, byggherre, bärplockare, bödel, chaufför, chef, chefsåklagare, civilingenjör, coach, cykelhandlare, dansare, dekan, dekoratör, diakon, dietist, diktator, direktor, diskare, diskjockey, djuruppfödare, doktorand, ekolog, ekonom, eldare, elektriker, endokrinolog, fackman, fastighetsmäklare, filmskapare, filosofilärare, finansman, fiskare, fiskhandlare, flottare, flygvärdinna, folkhögskolelärare, folklivsforskare, folkskollärare, fotograf, frilansjournalist, friskvårdskonsulent, frisör, fritidspedagog, fysiker, fysiklärare, fysiolog, fågelfängare, fäktmästare, fältskär, fönsterputsare, författare, förläggare, förrådsförvaltare, församlingspräst, förskollärare, försvarsadvokat, försäkringshandläggare, gallerist, genetiker, geograf, geografilärare, geolog, geometriker, gesäll, golfarkitekt, gondoljär, gravör, grundläggare, grundskollärare, gruvarbetare, gränsvakt, grävmaskinist, grönsakshandlare, guldsmed, guvernör, gymnasielärare, gymnastiklärare, gympalärare, gynekolog, hamnarbetare, handelsman, handläggare, handsättare, hembiträde, hemkunskapslärare, historielärare, historiker, hovmästare, hunduppfödare, hushållsarbetare, hypnotisör, hårfrisör, härskare, högskolelärare, högstadielärare, idrottslärare, illustratör, informationschef, ingenjör, inköpare, inspektör, jaktlöjtnant, journalist, jurist, järnvägsarkitekt, kansler, kardiolog, kemilärare, kemist, kiropraktor, kirurg, klimatolog, kock, kolare, kombinatoriker, kompositör, koncernchef, konditor, konstnär, konsul, kontorist, kopparslagare, kormästare, kosmetolog, kosmolog, kriminalkommissarie, kriminalvårdsinspektör, kronofogde, krämare, kusk, kvinnoläkare, kyltekniker, kyrkoherde, kändisadvokat, kökschef, köksmästare, köpman, körskolelärare, lagman, lantbrukare, lantmätare, lastbilschaufför, levnadstecknare, lingvist, lokalvårdare, lokförare, lågstadielärare, låssmed, läkare, lärare, lönnmördare, mannekäng, massör, matematiker, matematiklärare, matros, mekaniker, merkonom, meteorolog, mjölnare, modellsnickare, monark, montör, murare, musikalartist, musiklärare, målare, målsman, möbelsnickare, naturforskare, neurolog, nämndeman, oftalmolog, optiker, ordningsman, organolog, ortodontist, ortoped, paleontolog, parkeringsvakt, pastor, pedagog, pizzabagare, platschef, plattsättare, poet, polis, politiker, portier, programledare, programmerare, programvärd, prostituerad, präst, psykolog, pugilist, pälshandlare, rabbin, rapsod, religionslärare, repslagare, revisor, rorsman, råttfångare, rörmokare, röst-*

skådespelare, samhällskunskapslärare, sekreterare, serviceman, simlärare, sjuksköterska, sjåare, sjöman, skoflickare, skogsman, skolbibliotekarie, skomakare, skorstensfejare, skräddare, slaktare, smed, småskollärare, snickare, sockerbagare, sophämtare, sotare, spelman, sportredaktör, språkforskare, språklärare, språkvetare, spårvagnsförare, stadionchef, stadsarkitekt, stadsfullmäktige, statist, statistiker, statsman, stenograf, steward, strateg, styrelseproffs, styrman, stålverksarbetare, städhjälp, svetsare, systemerare, sångare, tandhygienist, tandläkare, taxichaufför, taxiförare, teckningslärare, tekniker, telefonförsäljare, telefonväktare, televerkare, terapeut, textare, timmerman, tjänsteman, tolk, tonsättare, traktorförare, trumpetare, trumslagare, tränare, tullare, tunnbindare, tågförare, ufolog, undersköterska, undertextare, ungdomsbibliotekarie, universitetsadjunkt, universitetsbibliotekarie, universitetslektor, universitetslärare, uppfödare, urolog, uska, utgivare, vaktis, vaktmästare, vattenrallare, vd, webbdesigner, webbutvecklare, veterinär, vice-president, vägarbetare, vävare, yrkesboxare, yrkeschaufför, yrkesman, yrkesmilitär, åklagare, ämbetsman, ärkebiskop, ögonläkare, öronläkare, överläkare, översättare .

A.4 Sources

For convenience, this section contains links to all data sources used throughout this project.

Corpora

Both English and Swedish Wikipedia dumps are available at `https://dumps.wikimedia.org/`. The Swedish Gigaword corpus [9] can be found at `https://spraakbanken.gu.se/swe/resurs/gigaword`. The English Gigaword 5 corpus referenced by [34] can be found at `https://catalog.ldc.upenn.edu/LDC2011T07`. The Common Crawl project can be found at `https://commoncrawl.org/`.

Pre-trained Embeddings

The fastText embeddings for both Swedish and English as a part of [17] can be found at `https://fasttext.cc/docs/en/crawl-vectors.html`. The English word2vec embedding from [31] can be found at `https://code.google.com/archive/p/word2vec/`. The English GloVe embedding from [34] can be found at `https://nlp.stanford.edu/projects/glove/`.

Words

Lists of Swedish occupations can be found on Wiktionary at `https://en.wiktionary.org/wiki/Category:sv:Occupations` and `https://sv.wiktionary.org/wiki/Kategori:Svenska/Yrken` as well as Språkbanken `https://spraakbanken.gu.se/swe/resurs/vocation-list`.

B Complete Results

For completeness sake we include the results omitted from the main text here. Tables 19 and 20 show the results of the relative norm distance method applied to the different English embeddings. Furthermore, Tables 21, 22 and 23 consider the stereotypical occupations from the previous tables in the context of word analogies.

Analogy	Score	Lower Bound
Norway : Norwegian :: Brazil : Brazilian	1.0473	0.8806
Thailand : Thai :: Poland : Polish	1.0436	0.8768
Italy : Italian :: Norway : Norwegian	1.0404	0.8806
Brazil : Brazilian :: Sweden : Swedish	1.0400	0.8702
Manila : Philippines :: Moscow : Russia	1.0369	0.8788
⋮	⋮	⋮
possibly : impossibly :: clear : unclear	0.3683	0.7551
ethical : unethical :: possibly : impossibly	0.3602	0.7562
responsible : irresponsible :: decided : undecided	0.3468	0.6768
possibly : impossibly :: tasteful : distasteful	0.3317	0.6930
USA : dollar :: Korea : won	0.3242	0.6498

Table 18: The best and worst analogies from the test set according to the English WORD2VEC embedding using the score function S from Equation (11).

GLOVE	FASTTEXT	WORD2VEC
engineer	carpenter	carpenter
soldier	soldier	mechanic
architect	blacksmith	mason
guard	engineer	blacksmith
retired	surveyor	retired
manager	janitor	architect
surveyor	mason	engineer
sheriff	shoemaker	mathematician
blacksmith	laborer	shoemaker
police	smith	physicist

Table 19: Top male occupations in the different English embeddings using the relative norm distance method.

GLOVE	FASTTEXT	WORD2VEC
nurse	midwife	nurse
midwife	nurse	midwife
housekeeper	dancer	librarian
dancer	librarian	housekeeper
attendant	housekeeper	dancer
librarian	teacher	teacher
teacher	student	cashier
psychologist	designer	student
dentist	cook	designer
cashier	artist	weaver

Table 20: Top female occupations in the different English embeddings using the relative norm distance method.

w	she : w :: he : ?	he : w :: she : ?
midwife	midwife	midwife
nurse	nurse	nurse
librarian	librarian	librarian
dancer	dancer	dancer
carpenter	carpenter	carpenter
engineer	engineer	engineer
surveyor	surveyor	surveyor
soldier	soldier	soldier

Table 21: The result of completing the analogies for English occupation words using 3CosMul in the English fastText embedding.

w	she : w :: he : ?	he : w :: she : ?
midwife	midwife	midwife
nurse	nurse	nurse
housekeeper	housekeeper	housekeeper
dancer	dancer	dancer
engineer	engineer	engineer
architect	architect	architect
manager	manager	manager
surveyor	surveyor	surveyor

Table 22: The result of completing the analogies for English occupation words using 3CosMul in the English GloVe embedding.

w	hon : w :: han : ?	han : w :: hon : ?
barnmorska	barnmorska	barnmorska
prostituerad	prostituerad	prostituerad
tandhygienist	tandhygienist	tandhygienist
hemkunskapslärare	hemkunskapslärare	hemkunskapslärare
fiskhandlare	fiskhandlare	fiskhandlare
elektriker	elektriker	elektriker
tränare	tränare	tränare
ingenjör	ingenjör	ingenjör

Table 23: The result of completing the analogies for Swedish occupation words using 3CosMul in the Swedish GloVe embedding.