technische universität
dortmund

Bachelor Thesis

Examining societal biases in word vector models trained on German language corpora

Sebastian Gerard
July 2017

Reviewers:
Prof. Dr. Kristian Kersting
M.Sc. Elena Erdmann

University of Technology Dortmund
Computer Science VIII:
Artificial Intelligence Unit
http://www-ai.cs.uni-dortmund.de

Contents

1 Introduction **1**

1.1 Objective of this thesis . . . 2

1.2 Structure of this thesis . . . 3

2 Theoretical background **5**

2.1 Natural language processing . . . 5

2.1.1 NLP foundations . . . 5

2.1.2 Machine learning . . . 6

2.1.3 Artificial neural networks . . . 8

2.1.4 Word2Vec . . . 9

2.1.5 GloVe . . . 14

2.1.6 Comparing word2vec and GloVe . . . 15

2.2 Societal biases are present in word vector models . . . 16

2.2.1 The implicit association test . . . 16

2.2.2 Measuring bias in word embeddings . . . 17

2.2.3 Biases found in word embeddings . . . 19

2.3 Debiasing word vector models . . . 20

3 Scientific protocol **23**

3.1 Software tools . . . 23

3.2 Text corpora used . . . 24

3.3 Preprocessing of corpora . . . 25

3.4 Semantic and syntactic tests . . . 26

3.5 Initial parameter selection . . . 27

3.5.1 Experimental settings . . . 27

3.5.2 Results . . . 28

3.6 Testing for biases in word embeddings . . . 29

3.7 Robustness of bias tests . . . 30

3.7.1 Robustness to subsampling . . . 31

3.7.2 Robustness to Permutation . . . 36

4 Language structure within word embeddings **37**
4.1 The natural gender subspace 37
4.2 Relationship between natural and grammatical gender 39
4.3 Investigating the generic masculine 41
4.4 Summary 44

5 Societal biases in German text corpora **45**
5.1 Which biases are present in German corpora? 45
5.2 Comparison with English biases 47

6 Discussion **49**
6.1 Summarizing our results 49
6.2 Methodical limitations 51
6.3 Future research 52
6.4 Dealing with societal biases in machine learning 53

A Further experimental results **55**
A.1 Preliminary tests 55

B Bias test **57**
B.1 Bias tests on public dataset 57
B.2 Word lists for IAT tests 58

Bibliography **61**

Chapter 1

Introduction

With the digitalization of every aspect of our lives, more and more data is produced, recorded and available for analysis. A type of data that contains a large amount of information is text data. However, it is notoriously difficult to analyze. Computers are very capable of solving complex mathematical tasks. It is very difficult for them to algorithmically understand the contents of a simple article written by humans, though. A main reason for this is that computers do not possess the knowledge what a given word means and especially how this meaning relates to other words. Words that look similar on a character-level often have completely different meanings. And words with similar meanings can be spelled entirely differently.

A set of methods that facilitate the understanding of text data is called word embeddings. These methods take in large corpora of text and compute vector representations of the words, phrases or documents that contain information about the relationship of these entities among each other. For this, they only use the information which words occur close to the word of interest. An assignment of words to real-valued vectors is called a word embedding. The model that appears to be the most popular at the moment is called word2vec[MCCD13]. It is able to deal with corpora consisting of several billions of words. It associates each word in the vocabulary with a vector with potentially several hundred dimensions. These vectors can then be used to process text data with a certain understanding of the meaning implicit in the text.

The computed vectors have two important features that help us in natural language processing. First, words with similar meanings are mapped to vectors that are close to each other in the vector space. Second, we can use these vectors to solve analogy tasks. We can, for example, compute the answer to the query 'man is to king as woman is to x.' We compute king - man + woman = x using only the vector representations of the words and find that the closest vector to x is the vector representing the word queen[MYZ13]. This means that we are able to make statements regarding the semantic meaning of words, even

though we have never explicitly defined them. We are only using mathematical operations on vectors.

However, these powerful methods also bring with them a potentially very serious problem. Because we construct the models based on texts written by humans, they reproduce the meanings of words as we humans use and understand them in our everyday life. The vector representations of the words contain many of the biases and stereotypes that are present in everyday language[CBN17]. Using these methods then poses the risk of not only discriminating against people based on the stereotypes that they already face in society. It also possibly perpetuates and maybe even aggravates the discrimination.

Real-world applications for word vector models that might involve such problems include applications in sentiment analysis. Dos Santos et al. have used word2vec as part of their sentiment analysis model that is supposed to predict the sentiment contained within a given short text (e.g. a tweet)[DSG14]. Problems could occur if terms like 'Muslim' are relatively close to terms like 'terror' or 'fear'. This should be expected if the training dataset for the word vector models consists of current news articles. Texts that include the word 'Muslim' might then be classified as less positive than the exact same texts with the word 'Christian' or 'Hindu'.

It should be noted that whether the presence of human biases in these models is considered a problem very much depends on the application at hand. If the models are used to learn something about how we use language, maybe how language can perpetuate stereotypes, or how society understands certain concepts or words, then this is very much a feature, not a bug. However, if these models are used in systems where we would expect and advocate for non-discriminatory behavior, then being aware of this problem and investigating how to mitigate its effects is an important issue.

1.1 Objective of this thesis

Previous work has focused on word vector models trained on English corpora. It was demonstrated that many biases present in society are implicitly contained in the word embeddings[CBN17][BCZ+16] (see Section 2.2). However, it is not clear how the structural differences between languages influence this phenomenon. Therefore we will investigate whether the biases reported in previous research are also present in word vector models trained on German corpora and how the specific properties of the language influence them. In this context, gender biases should be rather interesting to explore, since the German language is more strongly gendered than English. For example, it does not offer any gender-neutral words to describe occupations. A term describing an occupation is either explicitly male or explicitly female. Additionally, German uses the generic masculine. When talking about a person whose gender is not known or groups who are not exclusively female, the respective male terms are used.

To measure the strength and prevalence of biases and stereotypes contained in our vector models, we will employ a method called the Word Embedding Association Test (WEAT)[CBN17] adapted from a psychological test that is used to measure prejudice in humans, the Implicit Association Test (IAT)[GMS98]. To make sure that the results of the adapted version of this method are meaningful, we will test how robust the test results are to random influences in the input data.

Since it is not a priori clear which kind of word embedding is able to capture the specific structure of the German language best, we will not only use the popular word2vec model[MCCD13] but also compare it to a newer model called GloVe[PSM14] to construct word vectors for our tests. It might be the case that one model is significantly better able to capture the structure of the German language and this way we are making sure to receive the most useful word vectors so that our tests are as informative as possible.

After having verified the method, we conduct qualitative research with regard to the structure with which gender information is represented in German word embeddings. This gives us information about how the specific structural properties of the German language might influence the bias tests.

Finally, we use the bias tests to determine which biases are present in a big dataset of German online news articles ranging over six years and compare our findings with findings from English word embeddings and psychological research.

1.2 Structure of this thesis

In Chapter 2 we explain the theoretical background of our methods, both the algorithmic backgrounds of the word embedding models, as well as results from related research about biases in word embeddings. We introduce the methodological foundations of our main experiments in Chapter 3, including the description of our data sets and software tools. We also conduct preliminary experiments to determine parameter settings for all of our following experiments. Then we experimentally investigate the robustness of the bias tests towards two distinct random influences. In Chapter 4 we examine the influence of the structural characteristics of the German language pertaining to gender on the word embedding models. In Chapter 5 we then determine which societal biases are present in German word embeddings. Finally, we discuss our findings in Chapter 6, putting them into the context of existing research, discussing consequences and possible future research.

Chapter 2

Theoretical background

In this section, we introduce the theoretical background for the methods we use, based on existing research. First, we introduce the basics of natural language processing and different kinds of word embeddings. Next, we present the related research showing that these word embeddings contain biases akin to biases present in society. Finally, we discuss an approach that aims to remove those biases from word embeddings.

2.1 Natural language processing

Natural language processing (NLP) deals with processing text data produced by humans. This provides many difficult challenges since text data is inherently hard to understand for computers. While numbers possess meaning by their mathematical relationship to each other, each word in the human language has its own inherent meaning. Synonymy, the feature of several words having the same meaning, and polysemy, the feature that a word itself can have various different meanings, mostly depending on the context, are problems inherent to any NLP applications. Aspects like metaphors or irony pose even more complex challenges. Expressing a language's semantics appropriately and in ways that a computer can understand is therefore far from trivial.

2.1.1 NLP foundations

In this section, we introduce the basic terms used in the context of natural language processing that we use from here on.

In NLP, a body of text is referred to as a **document**. A collection of several documents is called a **corpus** (plural: corpora). For our use cases we do not need this distinction, so we use the word corpus to mean the entirety of all the textual data we are using in the respective context. We can think of a corpus T simply as a sequence of words. By listing each unique word in the corpus, we can construct our **vocabulary** $(t_1, t_2 \dots t_V)$ with $t_i \in T, 1 \leq i \leq V$, where V is the number of elements in the vocabulary.

It is very impractical for computers to handle words on a per-character basis if we are only interested in their meanings. Therefore we now move to representing words as vectors of numbers. We refer to a mapping that assigns a real-valued vector to each word in a vocabulary as a **word embedding**.

The simplest word embedding is called **bag-of-words**(BOW)[Joa02]. It produces binary vectors with V dimensions, where $BOW(t_i)$, the bag-of-words vector assigned to the word t_i, is 1 in dimension i and 0 everywhere else. For example, in a vocabulary of 5 words, the fourth word would be assigned the vector 00010. Note that this embedding does not add any information, it just numbers the words. If we want to represent multiple words in the bag-of-words embedding, we simply add the vectors representing these words. Then each dimension of the vector represents how often the respective word occurs in the given sentence or document.

The basic assumption of the BOW representation is that the order of the words in a sentence does not matter. Therefore throwing the words in a bag (and thereby discarding the sentence structure) can be used just as well as the full sentence.

A more complex embedding is **TF-IDF**[RU11]. Assume we have a collection of documents, each consisting of a sequence of words. Then let the **frequency** f_{ij} be the number of times that term t_i occurs in document j. We define the **term frequency** as $\mathrm{TF}_{ij} = \frac{f_{ij}}{\max_k f_{kf}}$, normalizing the frequency with that of the most frequent term in document j. It is exactly 1 for the most frequent term and smaller for less frequent terms. Next, let the **inverse document frequency** be $\mathrm{IDF}_i = \log_2 \frac{N}{n_i}$ where N is the number of documents and n_i is the number of different documents that the term t_i appears in. It is 0 for terms that occur in every document and bigger for terms that occur in less documents. The **TF-IDF score** for term t_i in document j is then $\text{TF-IDF}_{ij} = \mathrm{TF}_{ij} * \mathrm{IDF}_i$. It is low for words that occur in most documents or very infrequently and high for words that occur very frequently, but not in many documents. It can therefore be seen as assigning a kind of importance measure to each term in a document.

2.1.2 Machine learning

Many methods used in NLP belong to the area of machine learning (ML), including the word vector models this thesis focuses on. Machine learning aims to develop methods to find patterns and relationships within data. Its two main subsections are supervised and unsupervised machine learning. This section introduces basic machine learning terminology based on [RN03].

In **supervised learning**, we are given a **training dataset** $(\vec{x}_i, \vec{y}_i) \in X \times Y$. We call $\vec{x}_i$ **features** (or observations), $\vec{y}_i$ **labels** (or target values). Our goal is to correctly predict $\vec{y}_i$ given $\vec{x}_i$, even for observations which are not part of the training data. The underlying assumption is that there exists an unknown function $f : X \to Y$ which we

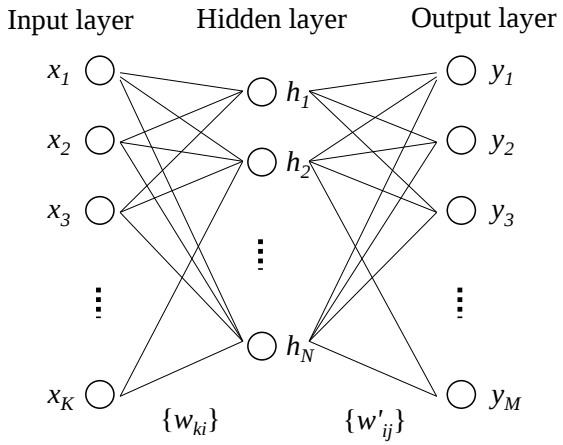

Figure 2.1: A multi-layer neural network with one hidden layer. Reprinted from [Ron14]

try to approximate, using the pointwise information given by the training data. We can therefore also represent the labels as $y_i = f(x_i)$.

The structure or entity that computes the approximated function values or **predictions** is called the (machine learning) **model**. We measure the quality of our predictions using an **error function**. Guided by the error function we can adjust the parameters of our model to improve the quality of our predictions. This process is referred to as **training** the model and it constitutes the learning aspect of machine learning.

For most methods of supervised learning, we need the input data to be represented as vectors of a fixed dimensionality for every observation. This is where word embeddings like bag-of-words are useful since they enable us to turn words or texts of different lengths into vectors of a fixed dimensionality that we can then use as input for our machine learning algorithms.

An example for supervised learning in NLP would be to predict which word comes next in a sentence, given the previous words. We make use of something similar to this when introducing word2vec.

In **unsupervised learning**, we are also given a set of observations $(\vec{x}_i)$, but we do not have any target values associated with them. The training data is called **unlabeled**. Instead, our goal is to find structures within this data. However, since we do not have any target values, it is often difficult to judge the quality of the result of these computations, because we do not know what the 'correct' result should look like. An example of this in NLP could be to group a set of articles based on their topics, without previously specifying which topics exist. The algorithm has to process the text and find topics on its own.

2.1.3 Artificial neural networks

Since the word2vec model uses an artificial neural network (from here on referred to as a neural network) as a core component, we give a short overview of the aspects of neural networks that we need. The information in this section is based on [RN03].

A neural network is a supervised machine learning model. The basic building block in a neural network is called a **node** or neuron. Nodes are connected to each other via directed links. Each node n_i receives an input vector $\vec{x_i} = (x_{i1}, x_{i2}, \ldots)^T$ via the incoming links and produces an output value y_i. Each incoming link is assigned a weight that determines its influence on the output value. The input vector is therefore multiplied with a **weight vector** $\vec{w_i} = (w_{i1}, w_{i2}, \ldots)^T$ to produce the inner product u_i:

$$u_i = \vec{w_i}^T \vec{x_i} = \sum_{j=1}^{N} w_{ij} * x_{ij} \tag{2.1}$$

where N is the number of input values for node i. Using the inner product, the output y_i of node i is then computed as

$$y_i = f_i(u_i) = f_i(\vec{w_i}^T \vec{x_i}) \tag{2.2}$$

where f_i is called the **activation function** of node i. A typical choice for the activation function is the sigmoid function σ:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

Neural networks are composed of an arbitrary amount of nodes, each receiving inputs either as the initial input to the network or as the output of other nodes. The nodes are arranged in **layers** of parallel nodes. See figure 2.1 for an example. Different layers in the same network can consist of different numbers of nodes.

The basic variant of neural networks in which connections can only go from a node in one layer to another node in a subsequent layer is called a **feed-forward neural network**. The alternative would be to allow cyclic structures with nodes feeding their values back into themselves or previous layers. Such networks are called **recurrent neural networks**. If the output of every node of one layer is fed into every node of the subsequent layer, we call these layers **fully connected**.

The layer of nodes that receives the initial inputs of the neural network is called the **input layer**, the one that outputs the final values computed by the network is called the **output layer** and every layer between those is called a **hidden layer**.

The output layer often uses the **softmax function** as the activation function for all its nodes. It is defined as:

$$\text{softmax}(u_i) = \frac{e^{u_i}}{\sum_j e^{u_j}} \tag{2.4}$$

where $u_i = \vec{w_i}^T \vec{x_i}$, as previously defined. In contrast to the sigmoid activation function, the softmax function does not only consider the local information of the current node's inner product u_i. Instead, we first compute the inner product $u_i = \vec{w_i}^T \vec{x_i}$ of the input vector and the weight vector for *every* node in the output vector. Then we take those results and feed them into the softmax function, using the inner product of every output node in the normalization, instead of only the local information. This results in the network's output vector $\vec{y} = (y_1, y_2 \dots y_n)^T$ being able to be interpreted as a probability distribution, since the normalization step ensures that $\sum_{i=1}^{n} y_i = 1$. The values are not real probabilities, since they are not connected to any events and their respective probabilities of occurrence, but they are often interpreted to represent a probability distribution.

To complete the supervised learning task, we need to adjust the weights of the neural network, so the outputs of the network become closer to the desired output given by the learning task. To this end, a method called **backpropagation** is used. It computes the prediction error at every output node and propagates these error values back through the network, while adjusting the weights, thus the name backpropagation. Mathematically, it computes the gradient of the error function by partially differentiating it with respect to the weights of the respective layers. The gradient then shows us in which direction the weights need to be changed to improve the output with respect to the error function.

2.1.4 Word2Vec

Word2vec is a method that computes word embeddings, based on an input corpus, internally utilizing an artificial neural network. For each word in the vocabulary, it computes a d-dimensional vector, based solely on information about its co-occurrence with other words in the corpus. A typical choice would be $d = 300$[MSC+13]. Since we do not give the algorithm any examples of what the vectors are supposed to look like word2vec is considered a method of unsupervised learning. However, the model internally uses a neural network, employing supervised learning that produces the vectors as a byproduct.

In this section, we define the word2vec model, including the two different architecture variants and important optimization schemes. All the information in this section is compiled from the original word2vec paper by Mikolov et al. [MCCD13] and the explanatory paper by Rong [Ron14] that goes more into the details and the background that Mikolv et al. assume to be known for their paper.

Skip-Gram

The Skip-Gram model uses a neural network that is trained to predict the next C words before and after the current word. The parameter C is called the **context window**. It takes as input a V-dimensional bag-of-words vector representing the currently observed word w_i and outputs a $V \times C$-dimensional vector, where V is again the number of words

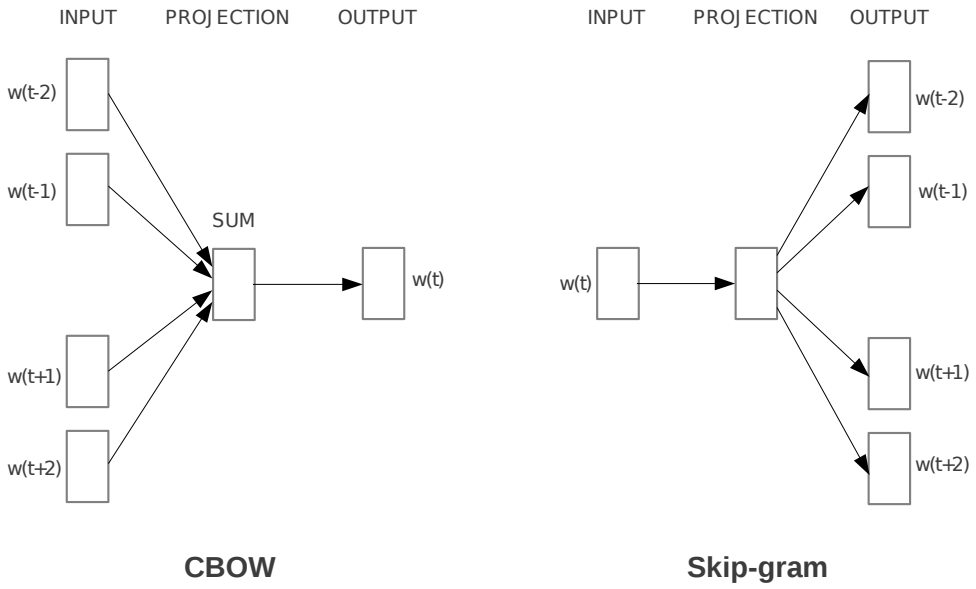

Figure 2.2: The CBOW architecture predicts the current word based on the context, and the Skip-gram predicts surrounding words given the current word. w(t) represents the currently observed word[MCCD13]. Reprinted with permission.

in our vocabulary. This vector represents C "panels" of V probabilities. Let p_{ij} be the i-th component of the j-th panel. It then represents the probability that the j-th context word is t_i, the i-th term in the vocabulary. The word embeddings we are interested in are actually a byproduct of this process, as explained below.

The Skip-Gram network consists of an input layer with V nodes, a hidden or projection layer of size d, and an output layer of size $V \times C$. C and d are parameters that we may choose freely. The connections between the fully-connected input layer and hidden layer are typically represented as a $V \times d$ matrix. By setting one of the input nodes to 1 and the others to 0, we are basically selecting one of the rows of the matrix to be the output of the hidden layer. As mentioned before, the output consists of the C probability distributions, each of size V. These panels share a weight matrix of size $d \times V$ that is applied when feeding data from the hidden layer to each of the panels. Since the weights are shared among those panels, the order in which the context words appear in the text has no effect on the training process.

Both weight matrices can be interpreted as a set of vector representations of the words from our vocabulary since each of them contains exactly V (column or row) vectors with d dimensions. These are referred to as the input vectors v_i and output vectors v'_i respectively. Usually, the input vectors are chosen as the vector representation that is output from the word2vec model.

The error function that is minimized for each training example during the machine learning process is:

$$\begin{aligned} E &= -\log \prod_{c=1}^{C} \frac{\exp(u_{c,j_c^*})}{\sum_{j'=1}^{V} \exp(u_{j'})} \\ &= -\sum_{c=1}^{C} u_{j_c^*} + C * \log \sum_{j'=1}^{V} \exp(u_{j'}) \end{aligned} \tag{2.5}$$

where j_c^* is the vocabulary index of the actually observed c-th context word and $u_j = \vec{v_j}'^T \vec{h}$.

Continuous bag-of-words (CBOW)

The continuous bag-of-words model is basically the reverse of the Skip-Gram model. Instead of predicting the C context words from the currently observed word, it predicts the currently observed word from the context words. The model has an input layer of $C \times V$ nodes, representing the bag-of-words vectors of the C context words and an output player of size V, representing the probability distribution over words in the vocabulary.

As in the Skip-Gram model, we again have two vector representations of the vocabulary within our model. The input vectors v_i are represented in the weight matrix of size $V \times d$ that is shared between all C input panels and is applied when feeding the input words into the hidden layer. The hidden layer simply computes the average of the input vectors:

$$\vec{h} = \frac{1}{C} * \sum_{i=1}^{C} \vec{v_{c_i}} \tag{2.6}$$

where c_i is the i-th context word and $\vec{v_{c_i}}$ is the input vector representation of the i-th context word. Since the context vectors are averaged and therefore the order does not matter anymore, this is similar to the bag-of-words embedding. However, bag-of-words is a discrete representation, where each component can either be 0 or 1. This model is continuous since each component is a fraction between 0 and 1 inclusively because we are computing the average values. Therefore it is called the continuous bag-of-words model.

The output vectors v_i' are represented by the weight matrix of size $d \times V$ that is applied when feeding values from the hidden layer into the output layer. As in the Skip-Gram model, the word embeddings to be output by the word2vec model when trained with this architecture are the input vectors.

The error function that is minimized during the machine learning process is:

$$E = -u_{j*} + \log \sum_{j'=1}^{V} \exp(u_{j'}) = -\vec{v}_{j*}'^T \vec{h} + \log \sum_{j'=1}^{V} \exp(\vec{v}_{w_j}'^T \vec{h}) \tag{2.7}$$

where $j*$ is the index of the actually observed word in the vocabulary.

Optimization methods

While both architectures are capable of producing useful vector embeddings, they require extensive computation that limits how big the input corpora can be. The most costly part of the computation is that for each training data pair of observed word and context words, we need to adjust the input and output vector representations of *every* word in the vocabulary to improve the prediction of the neural network. The first two methods to reduce the computational load, therefore, try to reduce the number of updates we need to make. Furthermore, we introduce a method to deal with the fact that certain words occur very frequently and therefore take up a lot of the computation time. After adding these methods, we are able to handle corpora in the order of billions of words.

Negative sampling

Negative sampling uses a very basic idea to reduce the number of updates mentioned above. Instead of updating all V input and output vectors, we simply randomly select a subset of words W_{neg} from our vocabulary, excluding the currently observed word, and update those. Of course, we also need to update the currently observed word. Since it is the correct prediction, we call it a positive example. All the words in W_{neg} were not observed, therefore we call them negative examples and consequently the method is called negative sampling since we are sampling negative examples. The error function then becomes:

$$E = -\log \sigma(\vec{v'}_{w_o}^T \vec{h}) - \sum_{w_i \in W_{neg}} \log \sigma(-\vec{v'}_{w_i}^T \vec{h}) \tag{2.8}$$

Hierarchical softmax

The approach of the hierarchical softmax is to replace the computation of the softmax function for all the terms in the vocabulary with a faster one. The terms t_i in the vocabulary are represented by the leaves in a binary tree. To be exact, we construct a Huffman tree, but the details of this are not discussed here.

Instead of having one output vector per term, we now have one output vector per inner node in the binary tree. We now use the notation v'_i to represent the output vector of an inner node of the binary tree. Instead of computing the probability of the output word being t_i using the softmax, we compute the probability of a random walk starting at the root of the binary tree ending at the leaf t_i and output this.

We define the probability of continuing the path towards the left child of the i-th node (as opposed to the right child) as:

$$p(i, left) = \sigma(\vec{v_i}'^T \vec{h}) \tag{2.9}$$

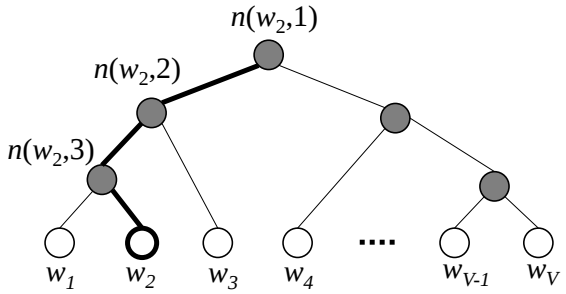

Figure 2.3: An example binary tree for the hierarchical softmax model. The white units are words in the vocabulary, and the dark units are inner units. An example path from the root to w_2 is highlighted. In the example shown, the length of the path $L(w_2) = 4$. $n(w, j)$ means the j-th unit on the path from the root to the word w. Reprinted from [Ron14]

where $\vec{h}$ is the output of the hidden layer and σ is the sigma function introduced in 2.1.3. Then the probability to take the right child instead is:

$$p(i, right) = 1 - p(i, left) = 1 - \sigma(\vec{v_i}'^T \vec{h}) = \sigma(-\vec{v_i}'^T \vec{h}) \tag{2.10}$$

since $1 - \sigma(x) = \sigma(-x)$.

Following the path from the root to our current term t_i, we simply need to multiply the respective probabilities of following the branches that lead to the leaf and receive the probability that we are looking for.

The probability for a word to be the output word is then:

$$p(w) = \prod_{j=1}^{L(w)-1} \sigma([.] * \vec{v'}_j^T \vec{h}) \tag{2.11}$$

where $L(w)$ is the length of the path from the root of the binary tree to the leaf representing w, $[.]$ is 1 if the left child of node j is the correct path to take and -1 otherwise, $\vec{v'}_j$ is the output vector associated with node j and $\vec{h}$ is the output of the hidden layer.

The error function becomes:

$$E = -\log p(w) = -\sum_{j=1}^{L(w)-1} \log \sigma([.]\vec{v'}_j^T \vec{h}) \tag{2.12}$$

where $p(w|w_I)$ is the predicted probability for words w that we want to maximize. Minimizing this error function achieves this objective.

During the training process we now do not need to update every output vector v'_i, but only those belonging to nodes on the path from the root to t_i. The prediction task at hand is now for each involved node to predict whether the left or the right branch should be taken. Because of the way the tree is constructed, this path has at most length $\lceil \log(V) \rceil$. Therefore we now have reduced the number of updates to output vectors from $O(V)$ to $O(\log(V))$, which significantly reduces the computation time.

Subsampling

Since we are dealing with large corpora, very frequent words like articles, pronouns etc. can make up a significant amount of the whole text. While it is nice to have a very good vector representation of such words, they are usually not what we are mainly interested in when training these models. Additionally, they take up a large amount of computation time that we could spend more productively. Therefore we randomly discard the word w_i from the corpus with probability

$$P(w_i) = 1 - \sqrt{\frac{t}{f(w_i)}} \tag{2.13}$$

where $f(w_i)$ is the number of times the word w_i occurs in the corpus and t is typically in the range of 10^{-5}. This results in words with higher frequency than t to be discarded often, while low-frequency words are usually preserved. This process is called subsampling, since we are randomly sampling words to discard from our corpus, thereby reducing its size[MSC+13].

2.1.5 GloVe

GloVe[PSM14] is another word embedding model that produces a vector for each word in the vocabulary of a given corpus. It starts operating on the word-word-cooccurrence matrix X where X_{ij} is the number of times that the word t_j appears in the context of the word t_i. The basic idea of GloVe is to train a regression model that approximates $\log(X_{ij})$ for terms t_i and t_j and have the vector representations be computed as the weights of this regression model. The desired property is the following:

$$\vec{v_i}^T \vec{v_j} + b_i + b_j = \log(X_{ij}) \tag{2.14}$$

where v_i, v_j are the vector representations of t_i, t_j and b_i, b_j are bias terms associated with the respective terms.

As mentioned in Section 2.1.2, every supervised model needs an error function according to which its internal parameters can be tuned. Regression models typically try to minimize the least-squares error (LSE):

$$LSE = \sum_{i=1}^{n} (y_i' - y_i)^2 \tag{2.15}$$

where y_i is the target value for our observation x_i and y_i' is the prediction of the model to be trained when given the input x_i[RN03]. Substituting our values, we receive:

$$LSE = \sum_{i,j=1}^{V} (\vec{v_i}^T \vec{v_j} + b_i + b_j - \log(X_{ij}))^2 \tag{2.16}$$

One problem that using this function leads to, is that the weights are adjusted to fit all word-word-cooccurrences equally well, even though some occur a lot more often than others. Cooccurrences that appear once in the whole corpus are given the same weight during the training process as those that appear several thousand times, even though the latter are obviously much more important. Therefore the authors introduce a weighting function f that addresses this problem.

$$f(x) = \begin{cases} (\frac{x}{x_{max}})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \tag{2.17}$$

with x_{max} and α being free parameters. x_{max} is the point that determines how often a cooccurrence has to appear to not get assigned a lower weight via the weighting function.

The final error function to train the regression model thereby becomes a weighted least-squares error function:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(\vec{v_i}^T \vec{v_j} + b_i + b_j - \log(X_{ij}))^2 \tag{2.18}$$

The weights and thereby the vector representations can again be learned by gradient descent, just as the weights for the neural networks.

2.1.6 Comparing word2vec and GloVe

The question of whether word2vec or GloVe produce the better word embeddings is controversial. As GloVe was published after word2vec, the authors, of course, compared their model to the existing models, including word2vec. They concluded that "The GloVe model outperforms all other methods on all evaluation metrics,"[PSM14] except for one special case. This makes it sound like GloVe would be the superior method to use.

However, further analysis by Levy et al.[LGD15] resulted in the exact opposite. Their main objective was to compare more traditional count-based NLP methods, that do not use any supervised machine learning algorithms, with word embeddings like GloVe and word2vec. Their main result is that the embedding methods are not generally better but mainly benefit from hyperparameter choices, both implicit in the construction of the model, and explicit in tunable parameters. However, another important result of their experiments is that, "in fact, SGNS[Skip-Gram with negative sampling] outperforms GloVe in every task." They attribute this strong contradiction to the original GloVe paper to the facts that they allowed for more hyperparameters to be varied than the GloVe authors, more analogy tests to be used to judge the quality of the vector models and the corpus to be the same for all of Levy et al.'s experiments.

Regarding the question of whether CBOW or Skip-Gram is the better architecture of word2vec, they acknowledge that CBOW has the potential to construct better word vector

models, but also report that the model did not realize this potential throughout their experiments.

Having investigated a lot of hyperparameter configurations, the authors are able to make recommendations on parameter settings to achieve the best results. Skip-Gram with negative sampling is recommended as a good baseline model. It shows good performance in every task while being fast to train and requiring the lowest amount of memory and disk space. Using negative sampling with at least two negative samples is always preferable to using no negative sampling.

2.2 Societal biases are present in word vector models

The presence of biases and stereotypes in society is not a new phenomenon. When talking about a bias we mean an association of a concept with other concepts that are not part of the intrinsic definition of this concept. For example, the definition of 'football' will include that it is a game played by a certain number of players with a ball, goals and a certain set of rules. The association that football is manly or mainly played by males is not part of the definition, therefore it is a bias. Note that this is just a morally neutral observation, biases are not per se problematic. Whether we regard a bias as problematic or not is a question of societal consensus. Morally neutral biases present in society include that instruments and flowers are generally considered to be pleasant, while weapons and insects are considered unpleasant[GMS98]. Morally relevant biases include those that reference a person's gender, origin, outer appearance etc. We call these kinds of biases stereotypes.

In this section, we introduce methods to test for biases in word embeddings. First, we introduce a psychological bias test used for humans, then we adapt it to assert biases in word embeddings and report the results that other researchers have found using this methodology.

2.2.1 The implicit association test

One way to measure biases in people is the so-called **implicit association test** (IAT). It was developed by Greenwald et al.[GMS98] and intends to address the problem that it is difficult to determine the presence and extent of prejudice and stereotypes by asking a person about these things directly. Social pressure, one's own conscience and simply not being aware of one's own subconscious attitudes make it difficult to answer such questions accurately. The IAT, therefore, tries to measure a subconscious response to judge how strong certain associations are.

An IAT consists of two target concepts and two attributes, each represented by a set of words. It is then measured how quickly the test subject is able to associate each of the concepts with each of the attributes. If, for example, the target concepts are Christian

and Muslim and the two attributes are pleasant and unpleasant, then a relatively quick response to the association of Christian terms with pleasant terms and Muslim terms with unpleasant terms would indicate positive prejudice towards Christians and negative prejudice towards Muslims.

2.2.2 Measuring bias in word embeddings

Caliskan et al.[CBN17] adapt the IAT to measure biases in word embeddings. They call the resulting test the **Word-Embedding Association Test** (WEAT). We use this method to test for biases in word embeddings trained on German corpora, therefore we explain their methodology in detail in this section.

Since the vector embeddings of words with similar meanings are close to each other, we can simply compare the vectors of the target concepts and attributes and see how far apart they are from each other. This is a measure that is analogous to the time measurement in the original IAT.

More formally, let X, Y be the sets of vectors belonging to the target concepts and A, B the sets of vectors belonging to the attribute words. The distance measure the authors used to compare two vectors $\vec{x}, \vec{y}$ is the cosine similarity:

$$\cos(\vec{x}, \vec{y}) = \frac{\vec{x}^T \vec{y}}{|\vec{x}| * |\vec{y}|} \tag{2.19}$$

where $|\vec{x}|$ and $|\vec{y}|$ are the length of the vectors. Since the vectors in our word embeddings have unit length, the denominator becomes 1, so we only need to care about the inner product of the two vectors. A high value means that the vectors are similar to each other, a low value means that the vectors are dissimilar.

The association of word $\vec{w} \in X \cup Y$ with the attribute sets A and B is measured by the following formula. We refer to the resulting value as the gender association value of w for given sets A and B.

$$s(\vec{w}, A, B) = \frac{1}{|A|} \sum_{\vec{a} \in A} \cos(\vec{w}, \vec{a}) - \frac{1}{|B|} \sum_{\vec{b} \in B} \cos(\vec{w}, \vec{b}) \tag{2.20}$$

If $s(\vec{w}, A, B)$ is close to zero, $\vec{w}$ is not associated with one of the attributes more strongly than the other. If it is positive, $\vec{w}$ has a stronger association with the attribute represented by the words in A, else it is more strongly associated with the attribute represented by B.

To now measure how the target concepts represented by X and Y are associated with A and B, which is the goal of the IAT, we compute the following value:

$$s(X, Y, A, B) = \sum_{\vec{x} \in X} s(\vec{x}, A, B) - \sum_{\vec{y} \in Y} s(\vec{y}, A, B) \tag{2.21}$$

The interpretation of the resulting value is the same as for $s(\vec{w}, A, B)$, except that we are now gaining knowledge about the association of the whole sets X and Y with A and B, instead of only about the word $\vec{w}$.

To measure the significance of the results, we compute the p-value and the effect size. The null hypothesis for the experiments is that the target concepts X and Y do not differ in their association with the attributes. The p-value, therefore, represents the probability that a random permutation of the sets X and Y result in a greater association difference $s(X, Y, A, B)$. Let $Z = \{(X_i, Y_i)\}_i$ be the set of partitions of $X \cup Y$ with $|X_i| = |Y_i|$ and $X_i \cap Y_i = \varnothing$. The p-value is then given by:

$$p = P_i[s(X_i, Y_i, A, B) > s(X, Y, A, B)] \text{ with } (X_i, Y_i) \in Z \tag{2.22}$$

Since the way to compute this value was not outlined in [CBN17], we assume that the following formula could be used:

$$\begin{aligned} p &= P_i[s(X_i, Y_i, A, B) > s(X, Y, A, B)] \\ &= \frac{1}{|Z|} \sum_{(X_i, Y_i) \in Z} \mathbb{1}[s(X_i, Y_i, A, B) > s(X, Y, A, B)] \end{aligned} \tag{2.23}$$

where $\mathbb{1}$ is an indicator function that is 1 if its argument is true and 0 if it is not. The formula therefore simply counts the relative frequency of sets that fulfill the condition.

While the original paper assumes $|X| = |Y|$, this is not the case in many of our experiments. We use small corpora for some of our experiments which only contain few of the elements of X and Y. Oftentimes this results in $|X| \neq |Y|$. We therefore modify the test method by setting

$$s(X, Y, A, B) = \frac{1}{|X|} \sum_{\vec{x} \in X} s(\vec{x}, A, B) - \frac{1}{|Y|} \sum_{\vec{y} \in Y} s(\vec{y}, A, B) \tag{2.24}$$

$$Z = \{(X_i, Y_i) \,|\, ((X_i \cup Y_i) = (X \cup Y)) \wedge (|X_i| = |X|) \wedge (|Y_i| = |Y|)\} \tag{2.25}$$

Since the exact computation of this value is not feasible for one of our tests (the Pleasantness/Flowers test), we follow the approach published in the original paper's replication code by sampling only a random subset of partitions and computing the p-value from those. We choose to sample 10^6 random partitions $(X_i, Y_i) \in Z$ for our robustness tests (see Chapter 3) and 10^9 for our bias tests (see Chapter 5). The reason for this is that while accidentally changing the state of the random number generator responsible for generating the random partitions, we observed one p-value changing from 0.11 to 0.0. While we assume that this is an extreme edge case (see the robustness tests in Section 3.7 for variations in p-values), we want to make sure that the biases we report are as accurate as possible. Therefore we increase the number of sampled partitions to 10^9 for our important bias tests.

The replication code for the original paper was only made available while working on this thesis, therefore we had already implemented this method and only changed the exact computation of the p-value to an stochastic approximation by only drawing a fixed

number of samples from the possible space of permutations. The original code then offers the possibility to estimate either a normal or an empirical distribution from these samples and computes the p-value as 1 minus the cumulative probability given by the distribution for the concrete value $s(X, Y, A, B)$ of the test statistic. We believe that the estimated empirical distribution results in the same values as our formulation. Since neither the paper nor the code suggest a reason to use the normal distribution instead we stick with our approach that we derived from the formula above.

The effect size used for the experiments is Cohen's d and computed as follows:

$$d = \frac{\frac{1}{|X|}\sum_{\vec{x}\in X} s(\vec{x}, A, B) - \frac{1}{|Y|}\sum_{\vec{y}\in Y} s(\vec{y}, A, B)}{\operatorname*{std}_{\vec{w}\in X\cup Y} s(\vec{w}, A, B)} \tag{2.26}$$

As a second test to measure bias, the authors tried to find a correlation of real-world statistics with bias values in the word embeddings. They call this test the **Word-Embeddings Factual Association Test** (WEFAT). For this setting, they considered a single set of target words W, two sets of attributes A and B and a property $p_{\vec{w}}$ associated with each word $\vec{w} \in W$. The bias value for $\vec{w}$ is then computed as

$$s(\vec{w}, A, B) = \frac{\frac{1}{|A|}\sum_{\vec{a}\in A} \cos(\vec{w}, \vec{a}) - \frac{1}{|B|}\sum_{\vec{b}\in B} \cos(\vec{w}, \vec{b})}{\operatorname*{std}_{\vec{x}\in A\cup B} \cos(\vec{w}, \vec{x})} \tag{2.27}$$

They then try to predict $p_{\vec{w}}$ using $s(\vec{w}, A, B)$ using a linear regression analysis and report the resulting Pearson's correlation coefficient and p-value.

2.2.3 Biases found in word embeddings

Using the methodology described above, Caliskan et al. have been able to detect a wide range of biases in word embeddings that are also present in society. They first replicated two of the original IATs, showing that flowers and instruments are perceived as pleasant, while insects and weapons are perceived as unpleasant. They then went on to replicate two studies showing racial discrimination of people with African names in comparison to people with European names. Furthermore, they showed that the word embeddings associate women more strongly with family and arts, while men were associated more strongly with career and science.

Using their second method, they showed a strong correlation between how strongly a word vector is associated with male or female terms(see eq. 2.27) and how many women actually work in these occupations, according to government statistics. Using the statistics of how many people with an androgynous name are male or female, they found a similar association with the bias value computed from the word embedding.

The authors conclude: "We have demonstrated that word embeddings encode not only stereotyped biases but also other knowledge, such as the visceral pleasantness of flowers or the gender distribution of occupations."[CBN17]

2.3 Debiasing word vector models

Bolukbasi et al.[BCZ+16] concern themselves exclusively with gender bias. Through several experiments, they come to the same conclusion as Caliskan et al.[CBN17], namely that societal biases are present in word embeddings. Their methodology is slightly different and their tests are less extensive, but the results agree with the results in [CBN17]. While Bolukbasi et al. provide methods with which these biases can be neutralized or at least weakened, Caliskan et al. "focus instead on rigorously demonstrating human-like biases in word embeddings."[CBN17] With respect to Bolukbasi et al.'s work they argue that their "methods do not require an algebraic formulation of bias, which may not be possible for all types of bias."[CBN17] In this section we introduce Bolukbasi et al.'s methodology, based on [BCZ+16].

An important conceptual distinction that Bolukbasi et al. introduce is that between direct and indirect bias. **Direct bias** is a strong association between gender-specific and gender-neutral words, for example, 'volleyball' being closer to 'woman' than to 'man', although volleyball is a gender-neutral term. **Indirect bias** manifests itself in an association between gender-neutral words, that is caused by those gender-neutral words being implicitly associated with gender-specific words. For example, the word 'receptionist' is closer to the word 'volleyball' than 'football' because both 'receptionist' and 'volleyball' have a more female bias.

To study gender bias, the authors first identify the vector subspace that represents the gender aspect in the word vectors. For this, they identified ten pairs of gender-specific words like (he, she) or (man, woman). They then compute the center of each pair, subtract it from each of the vectors constituting the pair and apply principal component analysis (PCA) to the resulting list of 20 vectors. This method allows them to identify the directions in the vector space in which those vectors vary the most. They observe that changes in gender can be described by variations in a one-dimensional subspace, which they call the gender subspace $\vec{g}$. They note that in other cases the gender subspace might have more than one dimension and all their methods work for subspaces of more than one dimension.

To debias the word embeddings, the authors introduce two methods: hard debiasing and soft debiasing. To counteract direct bias in a given set of words they perform what they call **hard debiasing**.

The first method of hard debiasing is used to remove associations like that between 'volleyball' and 'woman' and make sure that 'volleyball' is actually considered gender-neutral. To **neutralize** the bias implicit in gender-neutral terms, they simply set the

component of the word that points into the gender direction to zero. Let G be the gender subspace, defined by the orthogonal unit vectors $\vec{g_1}, \vec{g_2} \dots \vec{g_k} = G$. In their study the authors found that $k = 1$. The part of a word vector $\vec{w}$ that represents the gender direction is then simply the projection of $\vec{w}$ onto G, which we represent as $\vec{w_G}$. We can compute it as $\vec{w_G} = \sum_{j=1}^{k} (\vec{w}^T \vec{g_j}) * \vec{g_j}$. To neutralize a word vector $\vec{w}$, we can simply compute the new embedding $\vec{w'} = \frac{\vec{w} - \vec{w_G}}{|\vec{w} - \vec{w_G}|}$.

The second method of hard debiasing is to **equalize** sets of words that are supposed to have the same meaning, apart from their gender association. For example, aunt and uncle are supposed to have the exact same meaning apart from their gender aspect. Given a set of equality sets $\mathcal{E} = E_1, E_2 \dots E_m$ with $E_i \subset V$, where V is a vocabulary, we compute for each equality set $E \in \mathcal{E}$:

$$\vec{\mu} = \sum_{\vec{w} \in E} \frac{\vec{w}}{|E|} \tag{2.28}$$

$$\vec{\nu} = \vec{\mu} - \vec{\mu_G} \tag{2.29}$$

We then compute the new embedding $\vec{w'}$ for each word $\vec{w}$:

$$\vec{w'} = \vec{\nu} + \sqrt{1 - |\nu|^2} \frac{\vec{w_G} - \vec{\mu_G}}{|\vec{w_G} - \vec{\mu_G}|} \tag{2.30}$$

This means that the vectors are set to the average of the equality set outside of the gender subspace, while the components within the subspace are centered. This leads to the vectors in the equality set being exactly equal apart from the gender aspect, while the direction in the gender dimension becomes more meaningful due to the centering.

To weaken the gender bias on the whole corpus, they introduce a method called **soft debiasing** that tries to find a transformation for all of the word embeddings. It is supposed to minimize the projection of the gender-neutral words onto the gender subspace while preserving the pairwise inner product between word vectors, therefore preserving the cosine similarity between vectors (see equation 2.19).

Let d be the number of dimensions of the word embedding vectors, V the size of the vocabulary, then $W \in \mathbb{R}^{d \times V}$ is the matrix of word vectors corresponding to the words in the vocabulary. Further, let $N \in \mathbb{R}^{d \times N}$ be the matrix of word vectors corresponding to a set of neutral words. Then the debiasing transformation $T \in \mathbb{R}^{d \times d}$ is the solution to the following optimization problem:

$$\min_T ||(TW)^T (TW) - W^T W||_F^2 + \lambda ||(TN)^T (TG)||_F^2 \tag{2.31}$$

where G is the matrix made up from the orthogonal normalized vectors making up the gender subspace identified earlier and λ is a parameter that represents the trade-off between preserving the cosine similarity and weakening the gender bias. The transformed embeddings also need to be normalized, so the new embedding becomes

$$\hat{W} = \{T\vec{w} / ||T\vec{w}||_2 \ | w \in W\} \tag{2.32}$$

Chapter 3

Scientific protocol

In this chapter, we describe the data, tools, and methods we use for our experiments. We determine the parameter settings for the later experiments, introduce the methods to test the quality of word embeddings and the biases inherent to them and examine the robustness of these bias tests.

3.1 Software tools

To compute the word2vec models, we use the gensim implementation of word2vec[ŘS10]. Since they do not offer a GloVe implementation we use the original GloVe implementation published by the authors[PSM]. All glue code to manage the different parts of the experiments was written in Python. We use scipy[JOP$^+$] to compute linear regression models and scikit-learn[PVG$^+$11] to conduct principal component analysis.

The original word2vec code[Mik] supplies a range of different semantic and syntactic tests to determine and compare the quality of word embeddings. We first tried to simply translate them, using the Glosbe API [Mor] to help with semi-automatic translations. This approach was also employed by Morik et al. while comparing German and English word embeddings[MJW$^+$15]. However, it quickly became apparent that the automatic translation often failed to provide a proper translation, so a lot of manual work was needed to get useful test lists. While this is possible to manually correct for, we also found that the English grammar tested for in the word2vec tests was not easily transferable to German tests. For example "quick:quickly - sad:sadly" would basically translate to "quick:quick - sad:sad", since German adverbs often are the same as the adjectives. The same goes for the use of the gerund, forms like "sell:selling" can be translated, but result in expressions like "verkaufen:verkaufend" that will be rare, even in very big corpora, so that we can expect tests involving the gerund to be more or less useless for our research. Lastly, Morik et al. already suggested to leave out another grammatical category, since the superlative

expression is usually split up into two words in German, which would require to at least deal with 2-grams, which neither they nor we are using.

Therefore we decided to give up on this approach of translating the English tests and instead use tests that were created specifically for German semantical and syntactical relationships. To this end, we use the GermanWordEmbeddings toolkit by Andreas Müller[Mü], which provides exactly the kinds of tests we need.

3.2 Text corpora used

In our experiments, we want to gain knowledge about biases in the German language. For this, we need as much text as possible that is representative of the German language as a whole. Therefore we base our analysis on a category of text that constitutes a big part of the natural language that people are exposed to each day: news articles. In addition to representing an important part of everyday language, online news articles are also very readily available and produced each day by various sources of various political backgrounds, ages, and regional origin, so that we can expect a certain variety in the data that incorporates the diversity of natural language.

We use two different sources for news articles collections. First, we use a private dataset, kindly provided to us by the Wortschatz project[GEQ12] at Universität Leipzig. We refer to this dataset as the Wortschatz corpus. It was compiled from 9 873 327 articles from 2748 different web domains (including sub domains). The amount of articles per website and sentences per article are illustrated in Figure 3.1. Before applying processing, it contains 2 728 449 741 words and 171 464 986 sentences. After preprocessing it still contained 2 662 413 110 words (97.58 % of the unprocessed corpus) and 171 461 085 sentences. The final vocabulary of our model is constructed by removing any words that occur less than 20 times in our corpus. This leads to a vocabulary of 1 091 080 words.

The other source we use are articles provided within the context of the Conference on Machine Translation (WMT)[BCF$^+$16]. The 2017 conference provides article collections for each year from 2007 to 2016[sta], varying strongly in size per year. We refer to the corpus consisting of the entirety of these collections as the WMT corpus. The articles were collected from 152 different news sites. Since we do not have any information on how many articles were used to create the dataset or how many articles were used per site, we do not know anything about possible biases deriving from certain websites contributing much more to the dataset than others. Therefore we conduct our main experiment of determining how strong certain biases are in the German language on the Wortschatz corpus. Since that dataset is private, however, we also run the experiments determining the biases in German corpora on the WMT corpus and report the results in the appendix, to allow reproducibility.

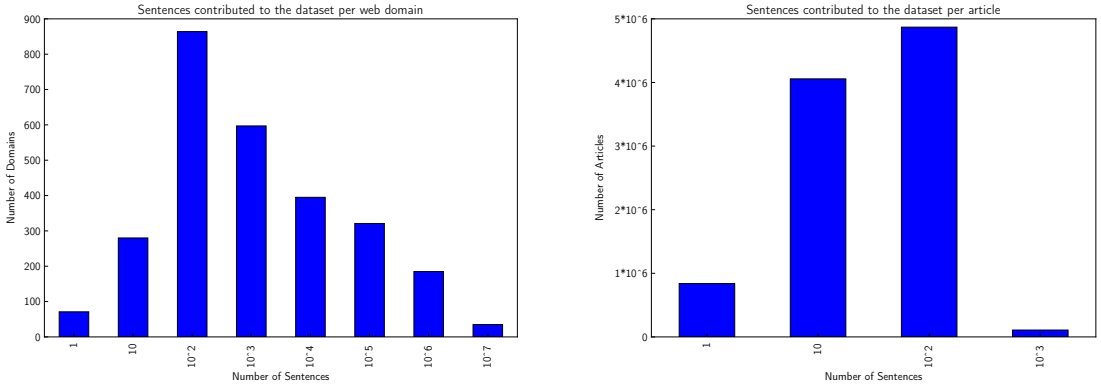

Figure 3.1: Distribution of the number of sentences contributed to the dataset by a single web domain or a single article respectively. Note the logarithmic scale for the number of sentences.

Before preprocessing, the WMT corpus contains 3 370 498 299 words and 221 810 812 sentences, afterwards 3 274 891 951 words (97.16 % of the unprocessed corpus) and 220 067 538 sentences. The final vocabulary of our model contains 1 294 710 words.

We also want to determine the robustness of our bias tests. For this, we have to train several models for several configurations of the input corpus. Running these experiments on one of the aforementioned corpora would cause substantial computation times that are beyond the scope of this thesis. Therefore we use only a subset of the data for these experiments. Since the distribution of the test data is not important for this methodological experiments, we do not need to use the private corpus. Therefore we use the 2013 article collection provided for WMT. We refer to this corpus as the WMT2013 corpus. It consists of 532 391 817 words and 35 014 404 sentences before preprocessing and 517 094 642 words (97.13 % of the unprocessed corpus) and 34 813 941 sentences afterwards. The resulting vocabulary for this corpus contains 426 568 words, although this value of course changes for experiments in which we subsample the corpus so that some words do not occur at least 20 times anymore and therefore be discarded during the word embedding training procedure.

3.3 Preprocessing of corpora

Before feeding a corpus into the algorithm to produce word embeddings, we perform the following preprocessing steps, mainly based on a preprocessing script provided by [Mü]. We transform all letters to lower case. This way we lose some information because we can not distinguish between cases where a verb is nominalized and those where it is simply a verb. However, we also avoid introducing a new word into the vocabulary every time a word is capitalized as the first word in the sentence. We do not require these fine-grained distinctions for our research and merging those special cases with the non-capitalized cases

reduces computational load, therefore we use the lower case of every word. Since we are dealing with German texts, the texts contain Umlauts, which we replace with their respective two-character representation (e.g. "ä" becomes "ae", "ß" becomes "ss"). Furthermore, we remove every non-alphanumeric character and any remaining words that only consist of numbers. This also results in compound words made of hyphens being collapsed into one word, which is desirable. We do not remove stop words for our analysis. While they do not provide any semantic content, the additional information of which stop words are used in a word's context should allow the algorithms to find a more precise embedding. We also do not remove duplicates, since this would distort the distribution of how language is used in the real world. If a certain phrase is repeated in ten different articles, it has a stronger influence on how the words in it are perceived than a sentence that only occurs in one article. Since we do want our model to capture this, we do not remove duplicate sentences. This also means that feed aggregators used in the dataset may introduce duplicate documents, which should be noted.

3.4 Semantic and syntactic tests

To judge how good a word embedding is, we use a set of semantic and syntactic analogy tests. As mentioned in the introduction, the word embeddings are able to solve analogies like king - man + woman = queen. We can also write this example as king:man - woman:queen, read as: "king is to man as woman is to queen." To test whether the embeddings capture this structure, we compute $v_{\vec{king}} - v_{\vec{man}} + v_{\vec{woman}} = \vec{x}$ where $v_{\vec{w}}$ is the word vector representation of the word w. We then search for the word vector that is the closest to $\vec{x}$ and take it as the model's answer to the analogy query. The percentage of correct answers is the final metric by which we judge the quality of the embedding. The code to run the tests is the evaluation script included in the GermanWordEmbeddings project[Mü], which internally uses the gensim package[ŘS10].

The analogy tests cover the following semantic categories:

- Semantic analogies with similar relationships, e.g.: Deutschland:Euro - Japan:Yen
- Semantic analogies with opposites, e.g.: Norden:Süden - positiv:negativ

Furthermore, the following syntactic tests are included, as well as all their respective inverse versions:

- Nouns: singular/plural
- Adjectives: base form/comparative
- Adjectives: base form/superlative
- Adjectives: comparative/superlative
- Verbs: infinitive/first person singular present
- Verbs: infinitive/second person plural present
- Verbs: first person singular present/second person plural present
- Verbs: infinitive/third person singular past
- Verbs: infinitive/third person plural past
- Verbs: third person singular past/third person plural past

Additionally, GermanWordEmbeddings provides a set of tests for which the task is to find the word that does not fit the other three words, i.e. the one that is the furthest away from the mean of all four word vectors. For example: Twitter, Facebook, Instagram, *App*.

3.5 Initial parameter selection

Our robustness experiments consist of training word vector models on different subsets and permutations of our corpus. Therefore we need to choose a consistent parameter setting to use for all these models. To determine these parameters, we perform a set of preliminary experiments.

3.5.1 Experimental settings

The main parameters to tune in word2vec are the following:

- sg: skip-gram or CBOW architecture. $sg \in [0, 1]$
- d: dimensionality of the produced word vectors. $d \in [100, 200, 300, 400, 500]$
- C: size of context window. $C \in [3, 5, 7, 10, 15]$

For the rest of the parameters, we set fixed values. Since Levy et al.[LGD15] mention that deleting rare words only has a small impact on performance, we arbitrarily set the threshold to deleting words that occur less than 20 times in the corpus and the number

of negative samples to 20. We set the subsampling parameter to $t = 10^{-5}$, which is the value used by Mikolov et al.[MSC+13] We always use the hierarchical softmax instead of the regular softmax function to speed up training[MCCD13]. Since the original word2vec implementation [Mik] sets the default number of iterations to five, we also apply this value. For the same reason, we set the learning rate of the neural network to $\alpha = 0.025$.

For GloVe we need to tune the following parameters:

- d: dimensionality of the produced word vectors. $d \in [100, 300, 500]$
- C: size of context window. $C \in [5, 10, 15]$

We set the parameters for the weighting function f to the values that the GloVe authors determined empirically: $\alpha = 0.75, x_{max} = 100$[PSM14]. We also stick to the default configuration of five training epochs and an initial learning rate of 0.05. Note: This is the initial set of parameter values we started with for both models. For word2vec we extended our experiments to further parameter values as reported above, but since GloVe turned out to be clearly inferior (see next section), we did not extend our initial experiments for GloVe.

The experiments on the WMT13 dataset (see Section 3.2) after applying the preprocessing described in Section 3.3.

3.5.2 Results

The GloVe model reaches a maximum accuracy of 31%, which is far below every result of the word2vec models. We conclude that the GloVe model is not well suited to capture the structure of the German language under the test settings provided in our experiments. Therefore we do not use the GloVe model in any of our further tests. The results of these experiments for the GloVe model can be seen in Appendix A.1.

Regarding the Word2Vec models, we find every model using the Skip-Gram architecture to achieve higher accuracy scores than all models but one using the CBOW architecture. Except for one remarkable outlier in the CBOW architecture, the Skip-Gram architecture clearly performs better. For the following analysis, we ignore this outlier since we can not assume that its extreme outlying performance is transferable when training other models with these parameters. Context windows of size $c = 3$ consistently perform the worst. No single context window size always performs better than the others. The vector dimensionality of $d = 300$ generally performed slightly better than that of $d = 500$. In the CBOW model the performance for $d = 200$ and $d = 400$ is significantly lower than for the other values. While we can see differences among different d-values for the Skip-Gram model too, the variation is much less strong than for the CBOW model. The highest accuracy score is achieved by the Skip-Gram architecture trained with $d = 300, c = 5$ and

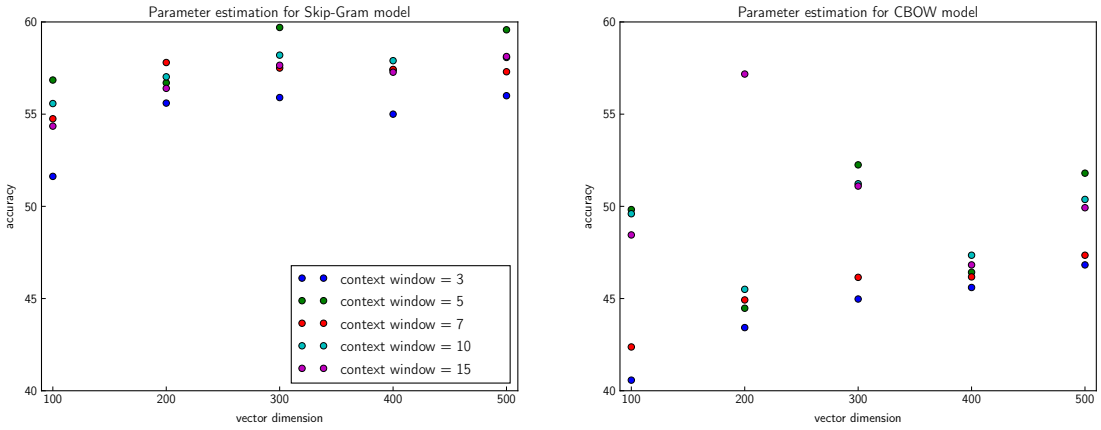

Figure 3.2: Accuracy for word2vec Skip-Gram (left) and CBOW models (right) trained with vector dimension 100, 300 or 500. The different graphs show configurations with context windows of size 5, 10 or 15. Higher accuracy is more desirable.

achieved 59.7% accuracy. The left plot in Figure 3.2 illustrates these test results for the Skip-Gram models, while the right plot shows the result for the CBOW models.

Our results align well with those reported by Levy et al.[LGD15] GloVe does not outperform the Word2Vec models in any task. While CBOW does not perform much worse than Skip-Gram, it does not manage to outperform the Skip-Gram models, except for a single configuration which resulted in an extreme outlying accuracy score. We were however surprised to see that lower context window sizes, except $c = 3$, were not regularly outclassed by higher window sizes. Since bigger context windows offer more information, we expected the models to perform better. However, the effect of more distant words being less related and therefore adding noise seems to outweigh the greater amount of information. This effect could possibly be mitigated by a higher number of training iterations, which we did not vary in our experiments. The low variability in accuracy scores for the Skip-Gram model compared to the CBOW model is another positive indicator, suggesting that Skip-Gram performs well on other corpora, too. We therefore simply pick the parameter setting that resulted in the highest accuracy scores and use it for our further experiments.

3.6 Testing for biases in word embeddings

To test for biases in word embeddings, we use Caliskan et al.'s methodology, described in Section 2.2.2. Since we remove any words that occur less often than 20 times in our respective corpus before training the model, we often have cases where a word we want to test is not contained in our model. In such a case we simply ignore the word. Since we always take the average association value over all words in a set, this does not falsify our results, even though the variance can be expected to increase. To still be able to use the

information of how many words were actually used in a test, we internally documented coverage rates for each of the four sets of words involved in each test. The coverage rate is the ratio of how many terms that are part of the WEAT are actually available in the model. If one of the four sets involved has none of its words represented in the model, we can not conduct the test.

As a simple, non-controversial baseline for biases, we replicate the original IAT by Greenwald et al.[GMS98] that measures whether insects or flowers are perceived as pleasant or unpleasant. We translate the terms for insects and flowers and take the pleasant and unpleasant terms from [Gaw02].

To measure gender bias, we use two IATs from Project Implicit[pro] with their permission. These IATs measure whether men or women are associated more strongly with social or natural sciences, and career or family respectively. To make use of the strongly-gendered aspect of the German language, we create an IAT-like test that measures whether men or women are perceived as more competent. Therefore, we use the respective male and female occupation terms describing the same occupations and test whether either the male or female terms are more strongly associated with terms related to competence. The competence and occupation terms were chosen arbitrarily.

We adopt the IAT conducted by Gawronski et al. [Gaw02] to check for racial biases by testing whether German names are considered more pleasant than Turkish names or Asian names.

Because it is a contemporarily relevant subject, we also want to test for biases with respect to religions. Therefore we create a test that measures whether religious terms pertaining to Christianity or Islam are perceived as more or less pleasant or unpleasant than the other. The terms for (un)pleasantness are again taken from Gawronski et al.[Gaw02], while the religious terms were chosen by us. The exact word lists used can be found in Appendix B.2.

Since both the publication in Science[CBN17] as well as the respective replication code were only published while we were working on this thesis, we had to implement the WEAT ourselves, based on the pre-print paper. We verified the correctness of our implementation by replicating the original results reported in [IBN16] exactly, except of course for the p-values which are stochastic approximations and therefore not expected to be the same for any two evaluations.

3.7 Robustness of bias tests

To detect biases inherent to the vector models, we can use the methodology detailed in Section 2.2.2, applying the WEAT to a word embedding model. However, it is not clear how robust the WEAT results are. They might vary strongly according to random influences during the training of the word vectors. This would mean that when applying the test to

different corpora that we want to compare, the test results could be very different, although both corpora are similar with regard to the biases contained. Such results would make the WEAT useless.

We, therefore, investigate two such possible confounding influences, namely how permutations and random subsampling of the input corpus influence the bias test results. Ideally, the test results should show little variability under both influences. Strong variance would indicate that the results of the tests are less meaningful and need to be interpreted more carefully or discarded as a methodology entirely since they would not measure a true characteristic of the test, but only indicate random noise in the data.

First, we take ten random subsets of a specified proportion of the input corpus and train a word2vec model on each of them (subsampling). We repeat this for several different proportions. Second, we create ten random permutations of the sentences in the corpus and train a word2vec model for each of the resulting corpora (permutation). For both experiments we test each of the resulting models with respect to the contained biases and report the variation in p-values and effect sizes.

3.7.1 Robustness to subsampling

To test how strong the results of a bias test vary when we only have access to a small dataset we train models on only a portion of the original input corpus. This helps us to determine whether a given corpus is big enough for us to draw conclusions about the bias contained in it with the bias tests.

After initial testing on subsets of 90% of the data showed little variation for most bias tests, we decided to conduct our subsampling experiments with much steeper subsampling rates of 1, 5, 10 and 50%. The base corpus from which we randomly draw subsamples of the specified sizes is again the WMT13 corpus (see Section 3.2).

Most bias tests prove surprisingly stable with regards to their p-value, even when working only on 1% of the input corpus (see Figure 3.3). The tests Gender/Career, Unpleasantness/Islam, and Unpleasantness/Turkish all exhibit very stable p-values below 0.05 for all subsampling rates. Gender/Competence was not avaiable for subsampling rate $s = 0.01$ since the subsampled corpus did not contain enough occurrences of the words used in the test. For the other subsampling rates it always had a p-value > 0.8.

Both Pleasantness/Flowers and Unpleasantness/Asian show a clear change in behavior with regard to their p-values when comparing $s = 0.01$ with the higher subsampling rates. The latter test shows a high variation in p-values for $s = 0.01$, and low variation and also much lower p-values for higher rates. The former, on the contrary, shows a very stable and low p-value for $s = 0.01$ and much higher variance and values for $s = 0.05$. However, both variance and p-value become lower when increasing the subsampling rate from 0.05 to 0.1 to 0.5. We believe that these observations concerning these two tests can be explained

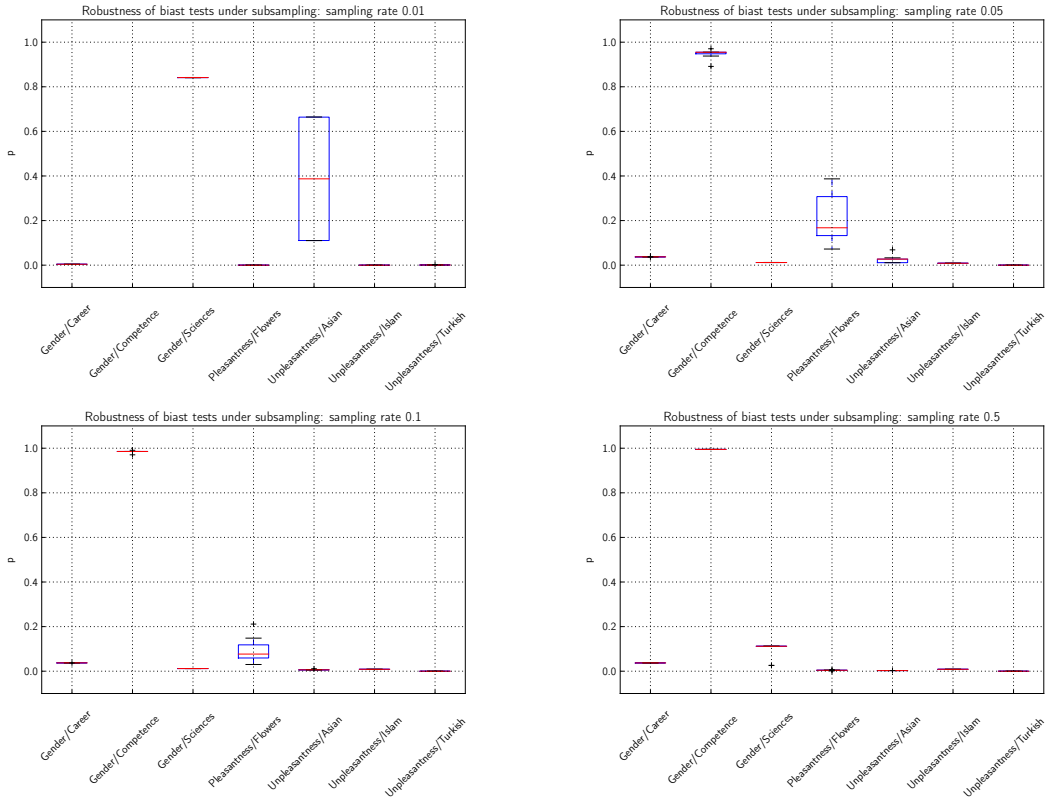

Figure 3.3: p-values of bias tests on word2vec models trained on random subsets of 1, 5, 10 and 50% of the input corpus respectively. A low value means more statistically significant results. Note that Gender/Competence was left out for the 1% subsamples because the subsets of the corpus did not contain enough occurrences of the words used in this bias test.

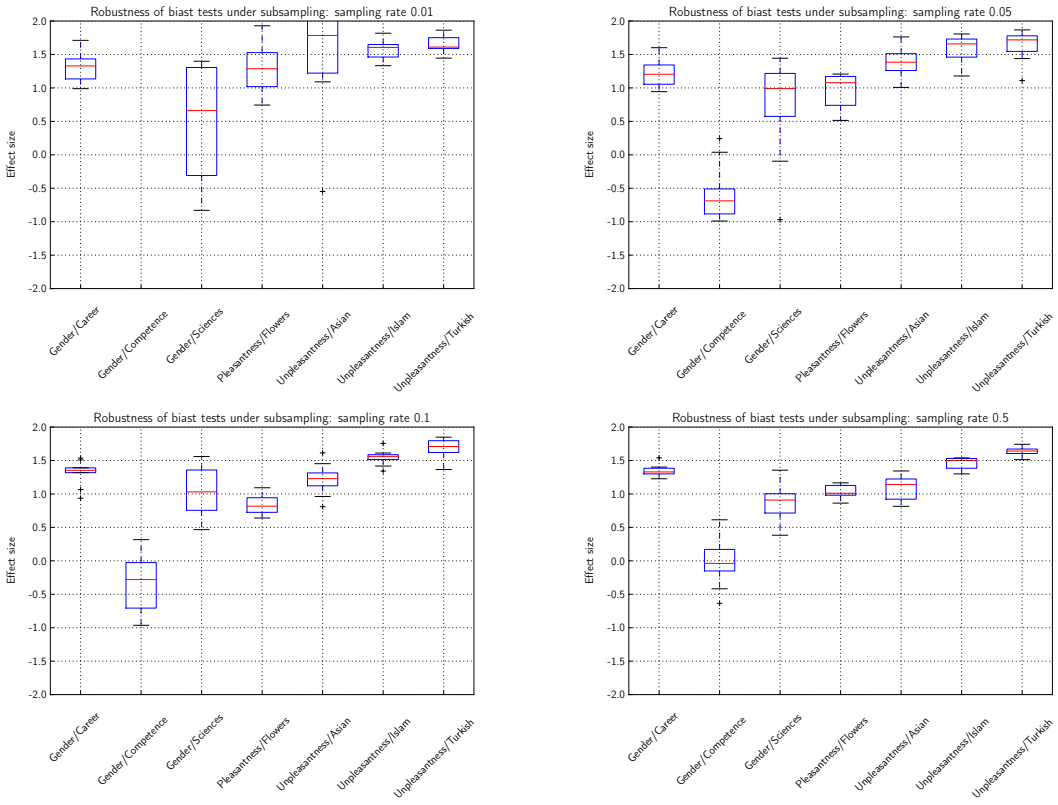

Figure 3.4: Effect sizes of bias tests on word2vec models trained on random subsets of 1, 5, 10 and 50% of the input corpus respectively. A value near 0 indicates no bias, a positive value indicates a bias in the expected direction, a negative value indicates a bias in the unexpected direction. More extreme values indicate stronger biases. Note that Gender/Competence was left out for the 1% subsamples because the subsets of the corpus did not contain enough occurrences of the words used in this bias test.

by the low coverage rates for the tests at lower subsampling rates. Both tests had some iterations in which a word set used for the test had only one of its words represented in the model. The low amount of input data caused by the subsampling introduces a lot of random fluctuation depending on which part of the corpus we sample. With growing subsampling rate these fluctuations are balanced out more and more by the larger amount of data. From these results, it seems like low coverage rates indicate lower robustness of p-values.

However, the Gender/Sciences test raises questions. It has clearly insignificant p-values of $p > 0.8$ for $s = 0.01$ but clearly significant p-values of $p < 0.02$ for sampling rates 0.05 and 0.1. In the case of Pleasantness/Flowers and Unpleasantness/Asian we were able to explain inconsistencies between the first and the other experiments by the effects of low coverage rates. However, the Gender/Sciences test has a coverage of 4/6 at its lowest, while Unpleasantness/Asian and Pleasantness/Flowers had a minimal coverage rate of 1/10 and 1/25 respectively. Additionally Gender/Sciences shows $p > 0.1$ for eight of the ten iterations with $s = 0.5$ and only for two iterations a value of $p < 0.03$. In our experiments on the full corpus (see Section 3.7.2) we find p-values for this test to be consistently at $p > 0.2$. For the examined methodology to be considered robust, we would have wanted to see low, significant p-values. With this result, we can not say that biases that were determined to be significant on smaller collections of text can not be determined to be insignificant when adding more data. While the WEAT seems to be robust to subsampling for most tests, under the constraint of sufficiently large coverage rates, it does not seem to be so for the Gender/Sciences test.

When looking at the effect sizes, we see the general pattern that tests with low and stable p-value also show a stable and high effect size. Especially looking at the results for $s = 0.05$ we can see much lower variance in the effect sizes. On the contrary, Gender/Competence and Gender/Sciences show a high variance in effect sizes, fitting their high variance in p-values. Unpleasantness/Asian shows a high variance for $s = 0.01$ and low variance for the higher subsampling rates, just as we would have expected from looking at its p-values, which varied largely for $s = 0.01$ and were stable for the higher rates. With growing subsampling rate and therefore growing amount of input data, the variance in effect size does not increase for any tests. For most tests we can indeed see that the variance decreases with growing subsampling rate. Again the Gender/Science test shows results that do not fit the patterns indicated by the other tests. While the high variance for $s = 0.01$ fits to the high variance in the respective p-values, the effect sizes for $s = 0.05$ still show great variance, even though the corresponding p-values are very stable and low.

From these results, we draw the conclusion that p-values and effect sizes are not robust to subsampling if the coverage rates are low. This probably happens because we only include words in our model that occur at least 20 times in the corpus. When taking different subsamples this means that different words are included in each of the models. Therefore some test runs are based on one set of words and other test runs are based on other sets of words, since the words included in the model change. For high coverage rates p-values are stable for all but one test. Effect sizes become more stable with more data and are less stable for larger p-values. We were not able to explain the unusual behavior of the Gender/Science test, which represents a clear exception to the mentioned regularities.

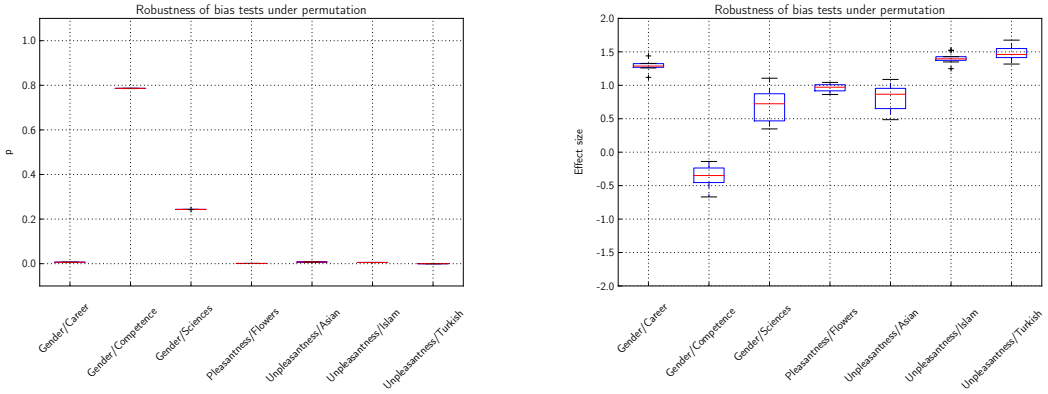

Figure 3.5: p-values and effect sizes of bias tests on word2vec models trained on ten random permutations of the input corpus. Low variation means the test had similar results on all random samples. A low p-value means more statistically significant results. A high positive effect size indicates a strong bias in the expected direction. An effect size near zero indicates no bias being present. + signs indicate outliers.

3.7.2 Robustness to Permutation

Apart from the size of the input corpus, the word2vec model also depends on the order of the sentences in the input corpus. Therefore it is reasonable to assume that this factor might also influence the bias tests. We, therefore, assess the influence of the order of sentences within the input corpus on the bias tests by training a model on each of ten random permutations of the sentences in the input corpus.

The results indicate very stable p-values over all permutations (see Figure 3.5). Gender/Competence and Gender/Sciences have very stable insignificant p-values for all ten permutations, while Gender/Career, Pleasantness/Flowers, Unpleasantness/Islam, Unpleasantness/Turkish and Unpleasantness/Asian show consistent p-values below 0.05.

Regarding the effect sizes, we see more variation again. The tests with lowest variation in effect size also have stable and low p-values, while those with high p-values exhibit comparatively high variance in effect size. However, Gender/Competence and Unpleasantness/Asian show similarly strong variations of slightly more than 0.5 standard deviations, even though the former has p-values around 0.8, while the latter has p-values close to 0.

We therefore conclude that p-values for the WEAT are very robust to permutation, while effect sizes show non-negligible variations even for tests with very low and stable p-values. Test results must therefore be interpreted with care.

Chapter 4

Language structure within word embeddings

In this chapter, we start to approach biases in German word2vec models by investigating the structure by which gender information is represented in vector models. We exploit the gender-related aspects that are specific to the German language to guide our research. First, we determine the vector subspace that contains most of the information about the natural gender of words, utilizing the strong genderedness of German nouns. Next, we try to find further structures by examining a relationship between the grammatical gender (genus) and the natural gender (sexus) of words. Finally, we inspect the influence of the generic masculine on gender information contained within the vector models.

4.1 The natural gender subspace

Bolukbasi et al.[BCZ+16] were able to show that the information about the natural gender is mainly contained in one dimension in the word embeddings trained on English texts. We replicate their experiments and examine whether this structure is the same for German texts. We expect to find a vector subspace of higher dimensionality since German also uses the concept of grammatical gender, which to a certain degree correlates with the biological gender.

We determine the vector subspace based on the methods by Bolukbasi et al. which we introduced in section 2.3. Therefore we choose the ten pairs of gender-defining words in Table 4.1 to be the basis for our computations. For grammatical reasons, these words are not exact translations of the original English words, but they are functionally equivalent. To verify our implementation we conducted the original experiment on the original data and were able to reproduce the authors' reported results.

The results of this experiment are much less impressive than in Bolukbasi et al.'s case. While we do see that the first principal components explain a lot more of the variance than

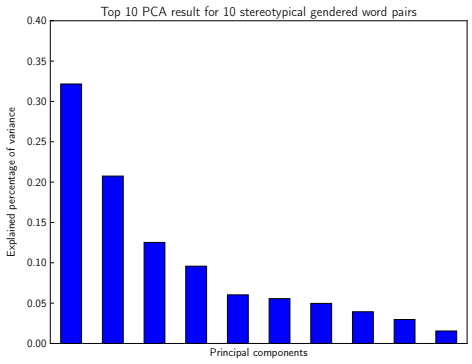

Figure 4.1: Percentage of variance explained by first ten principal components, computed from the word pairs in Table 4.1

sie	er	Oma	Opa
Frau	Mann	Großmutter	Großvater
ihr	sein	Mädchen	Junge
Tocher	Sohn	weiblich	männlich
Mutter	Vater	sie	ihn

Table 4.1: Word pairs of gender-defining terms used to determine the gender subspace

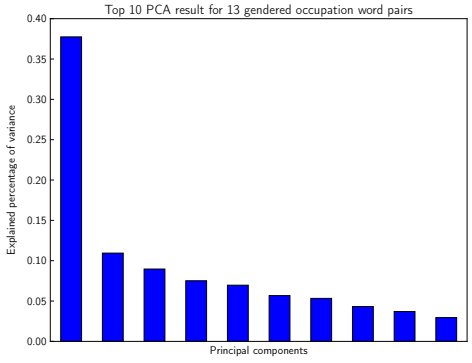

Figure 4.2: Percentage of variance explained by first ten principal components, computed from the 13 pairs of gendered occupation terms used in the Gender/Competence bias test

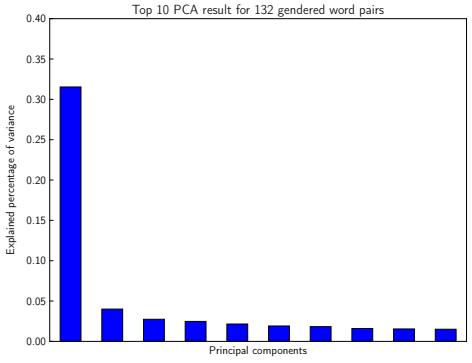

Figure 4.3: Percentage of variance explained by first ten principal components, computed from 132 pairs of gendered occupation terms or stereotypical gender terms

the later ones, we can not identify a clear cut-off point (see Figure 4.1). It looks like there might be a multi-dimensional gender subspace as we assumed. However, it could also just be that the data contains too much non-gender information.

To see if we can find a more clearly defined gender subspace, we use another set of gendered word pairs. For this, we use the 13 gendered occupation title pairs from our Gender/Competence bias test. The results look much more conclusive, as can be seen in Figure 4.2. The explained variance of the first component is clearly larger than that of the other components and the decline in explained variance seems to be rather steady from the second to the tenth principal component.

Since we only used a small number of possible occupation terms, we might be able to get even better results by adding more word pairs to our experiment. While the number of possible basic stereotypical word pairs like "Mutter-Vater" or "er-sie" is limited, we can make use of the German language not having gender-neutral occupation descriptors. We, therefore, increase the number of word pairs by adding some more stereotypical word pairs and a big, arbitrarily chosen set of male and female occupation terms like "Journalistin-Journalist."
Repeating the experiment with the resulting 132 word pairs slightly decreases the variance explained by the first principal component, but also greatly increases the distance to the percentage of variance explained by the second principal component (see Figure 4.3), thereby identifying the gender direction more clearly. Of course, we expect all of the percentages to go down if we increase the number of principal components anyway, but providing more examples from which to identify the direction of strongest variation should give us a more robust estimate of the gender direction.

Based on these results, we identify the first principal component as the direction in the vector space that contains most of the information about the gender associated with word vectors. We, therefore, discard the hypothesis of the multi-dimensional gender subspace.

4.2 Relationship between natural and grammatical gender

While inanimate objects can be assigned one of the three genera (grammatical genders), terms describing people generally have a genus conforming to the sexus (natural gender) of the described person. For example, the term describing a male kindergarten teacher has a male genus: "der Kindergärtner." Therefore, it is an interesting question to examine whether the word embeddings also model a relationship between sexus and genus. We expect no such correlation for terms describing inanimate objects, since their sexus is neutral, but their genus is often not. However, if the modeling is imperfect, it might, for example, occur that "die Tür" (the door) is associated with female terms because it has a female genus.
For terms describing people, we expect a strong correlation, because genus and sexus

generally correlate for these phrases, so the model should be able to capture this. One problem that might occur is that when designating possession the German language can use a pronoun that is identical to the male article, even when the people being talked about are female. For example: "Die Leistung *der Fußballspielerinnen* ist sehr gut." Furthermore, plural forms use the female article, independent of sexus or genus. Therefore we should expect a certain amount of noise that weakens the correlation.

To test this we would like to use the WEAT, using male and female terms as the attribute sets A and B. The target words are words with neutral sexus and male genus for target set X and female genus for target set Y. Using words with male and female sexus as sets X and Y does not seem to make sense, since we would not be able to distinguish between the obvious correlation of X and A (resp. Y and B) having the same sexus from the sexus-genus correlation.

However, while working on this problem it became clear that it is very difficult to actually conduct this experiment in a way that gives us reliable results. The reason is that any way we choose to identify a gender direction in our model necessarily includes using a lot of words that have equivalent sexus and genus. The way we identify the sexus is by measuring the cosine distance to several stereotypically male/female terms. However, these also mostly use the respective male/female genus. Therefore if we were to find out that words of female genus are more strongly associated with words of the female sexus and male genus terms with male sexus terms, we could not say that this shows us that sexus and genus correlate. The explanation might just be that both sets of words share a genus and the association between both sets might be caused by this commonality.

A solution would be to find gender-stereotypical words that do not have an identical genus and sexus like "das Mädchen", which is associated with a female sexus but has a neutral genus. However, there are only few terms like this, so this route does not seem feasible. Another solution would be to pick stereotypically male or female terms instead of the sexus terms. Football and rugby could be used to replace the male terms, while volleyball and cheerleading could replace the female terms. However, this would require us to make assumptions about our model sharing these biases. Since our results in section 2.2 show that the model does not mirror all of the society's biases, this also does not seem like a reasonable approach. Therefore we conclude that our attempts to identify a correlation between grammatical and natural gender in the word2vec model can not be successful with our current methodology. It is likely that grammatical and biological gender are not separable.

4.3 Investigating the generic masculine

In English, terms describing a person are generally gender-neutral (e.g. doctor, traveler, companion). On the contrary, German usually only offers gendered terms, using the male form for any subjects that are not explicitly and exclusively female. This grammatical construct is called the generic masculine.

Are men more manly than women are womanly?

We want to determine the influence of the generic masculine on the way gender information is represented in the word2vec model. Therefore we create a list of female and a list of male terms, describing the exact same occupations and compare them with respect to how closely they are associated with their respective gender. We expect the female terms to be clearly designated as female, while the male terms should be less clearly identified as male. Using the generic masculine means that male terms are used to describe both male and female individuals, therefore a perfect embedding would contain the information that the male terms are not exclusively representing males. Experimentally, we determine the mean of the gender association values $s(\vec{w}, A, B)$, computed as described in section 2.2.2, separately for male and female terms. We use the male and female occupation terms from the gender vs. competence IAT as the target concepts X, Y and the male and female terms from the natural vs. social science IAT as the attribute words A, B.

We find that the average association value for the male occupation terms is $\bar{s}_m = 0.05$, while the average association value for the female occupation terms is $\bar{s}_f = -0.11$. The signs indicate that male occupation terms are more strongly associated with male terms and female occupation terms with female terms. As predicted the mean association is stronger for female terms. The average male association is less than half as strong. The left plot in Figure 4.4 shows the individual gender association values, illustrating that this result is not a distortion caused by extreme outliers, but is a trend over all values. We interpret this as demonstrating the expected effect of the generic masculine, as outlined above. Generic male terms are implicitly associated with being less male because they are also used in the generic sense and not exclusively as male. It is important to note that this is only true for the *generic* male terms, which can be used to describe both male and female individuals. Our result does not say anything about the strength of association of purely male terms like "Vater" or "männlich".

To determine whether this effect also holds true for the pure male terms, which we do not expect, we repeat the above experiment, replacing the occupation terms with the male and female names from the Gender/Career IAT. Since these names are all unambiguous with respect to their gender, we interpret this test as showing the average strength of association between purely male terms on the one hand and purely female terms on the other hand. To our surprise, we get similar results as in the previous test, with the average

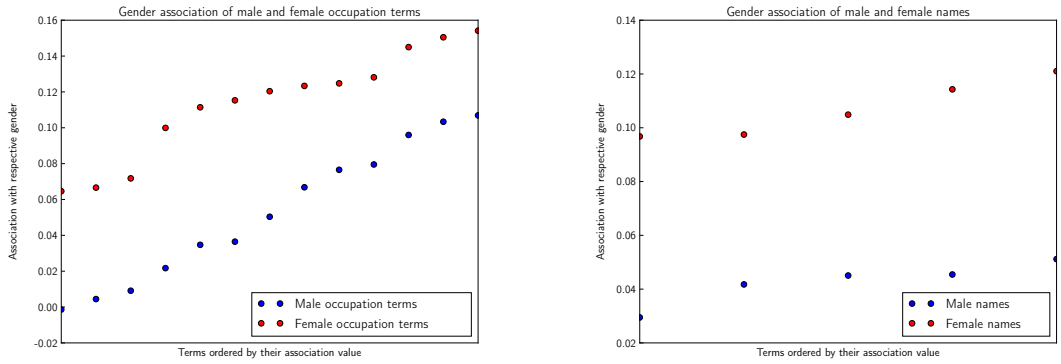

Figure 4.4: Gender association values of male and female occupations and names. Female association values were negated and both female and male association values were sorted amongst themselves to allow for easier visual comparison.

male association being $\bar{s}_m = 0.04$ and the average female association being $\bar{s}_f = -0.11$. Again we find that the female terms have an average association value that is more than twice as high as the male one. Since we used purely male and female terms this time, we conclude that male terms show a weaker association with other male terms than female terms with other female terms. The individual gender association values for this experiment can be seen in the right plot of Figure 4.4.

Since this result is both true for the generic male terms, as well as the purely male ones, we must assume that it is also true for stereotypically male or female ones, like "Fußball" being male and "Volleyball" being female. However, we can not reliably test this, since that would require us to make the assumption that these terms are gender-biased in our model. Furthermore, we would need to find two sets of terms of male-gendered and female-gendered terms that have the same strength of gender-biases. For names this was no problem, since we can find names that are unambiguously male or female, but for stereotypes, this approach does not seem feasible.

This result should have implications for our bias tests. We expect gender association values to be skewed towards female associations since these seem to be generally stronger in our model. However, since we have no reference point, we can not say what this means for our bias test results. If we could say that the male association values are lower than we would expect, while female values are unaffected, then we could say that the effect size should be lower than expected, since it is computed via the mean association values of both sets. If we could say that the female values are bigger to the same extent that male values are smaller, then we could expect the effect size to stay the same. Since we do not have a reference point from which we could judge whether these values are bigger or smaller, we can not make any such statements, though. The p-values should not be affected in any case.

Correlating occupation statistics with the vector model

As a second approach to investigate the generic masculine we try to replicate Caliskan et al.'s result of correlating the gender association values of occupations with the actual gender distribution as reported by government statistics, using the WEFAT method[CBN17]. The generic masculine offers several different options for our experimental design here. We simply realize all of them.

For all of our experiments, we use a set of 57 occupation terms and their respective ratio of male employees. These occupation terms were chosen in a way that would offer a wide range of different occupation ratios. The occupation rates were originally published on the website of the Institut für Arbeitsmarkt und Berufsforschung[fAuB]. However, their website is both slow and very cumbersome to use for our purpose. Therefore we used the Wikipedia's compilation of this data[wik] and verified them by comparing a random sample of their reported values with the original source. The alternative would have been to write a program that scrapes the original website, which is outside the scope of this thesis. We compute the gender association value s for each word as described in section 2.2.2 and compute a linear regression model to find a correlation between these gender association values and the occupation statistics. For each linear regression, we report Pearson's r and the corresponding p-value. The respective results can be found in Table 4.2.

First, we choose the most direct way to replicate Caliskan et al.'s results by using the male singular descriptions for the occupations. Because of the generic masculine, this is the most general description the German language offers, similar to the generic gender-neutral terms used in English. For this setting, we achieve a p-value of 0.02 and Pearson's r = 0.32.

Second, we use the male plural occupation terms. While a singular female person would still be described with a singular female term, any group including at least one male is referred to with the male plural. Therefore we expect the male plural form to more closely represent the female individuals described by it. We report p = 0.07 and r = 0.24. Third, we use female singular terms and correlate them with the female occupation ratios, computed from the male ratios by subtracting them from 1. Our results show p = 0.20 and r = 0.15.

Finally, we use female plural occupation terms and correlate them with female occupation ratios. We find p = 0.32 and r = 0.15.

We can see hat the male singular descriptors weakly correlate with the real occupation statistics. This makes sense as this is the generic form in which people refer to an occupation. The male plural form is a lot less significant and shows an even lower effect size than the first experiment. The male plural forms, therefore, do not seem to offer much information about the true gender distribution of the respective occupations.

Occupation terms	Pearson's r	p
Male singular	0.32	0.02
Male plural	0.24	0.07
Female singular	0.15	0.20
Female plural	0.15	0.32

Table 4.2: Correlation of occupation statistics with gender association values in word2vec model

The experiments with female terms do not show any significant results. This is easy to explain, as the female terms are exclusively used for female individuals. Therefore we should not expect any correlation with occupation statistics. While this result does not add any new knowledge, it confirms what we expected and excludes the possibility of unexpected results which would challenge our assumptions and approach.

Summarizing our results, we can not replicate Caliskan et al.'s results of strong correlation between occupation statistics and gender association in the word vector model. We can only find a weak correlation between male singular occupation terms and the respective occupation statistics.

4.4 Summary

Concluding this chapter, we find that gender information is represented in German word2vec models in a one-dimensional subspace. We are able to detect the influence of the generic masculine and find that it changes the way that gender information is represented in the model in comparison to English models. We also find that associations between male terms are weaker than between female terms, which is presumably an effect of the generic masculine. This effect could have a strong impact on the effects of our bias tests pertaining to gender.

Chapter 5

Societal biases in German text corpora

Caliskan et al. have demonstrated the presence of a wide range of societal biases in word2vec models trained on English corpora(see Section 2.2.2). In this chapter we investigate whether models trained on German corpora exhibit similar biases, how strong they are and where they differ.

5.1 Which biases are present in German corpora?

In this section, we try to answer our central question: How biased are German word embeddings? For this, we train a word vector model using the parameters determined in our preliminary experiments in Section 3.5. Since Mikolov et al.[MCCD13] reported that the quality of the word vectors depends on the size of the input corpus, we use a large corpus to train this model. This should both result in vectors that represent the language structures and the contained biases well, as well as mitigating low-quality embeddings of words in the bias tests, because they would occur too infrequently in a smaller corpus. Therefore we use the Wortschatz corpus, described in Section 3.2.

First, we compare the bias tests we used during our robustness tests with psychological studies that investigated similar biases. Since we created some tests by ourselves, this is not always perfectly possible, but we try to compare our results with studies that are as close to our tests as possible.

The results of the bias tests on this big model can be seen in Table 5.1. We are able to closely replicate Greenwald et al.'s test that measures whether flowers or insects are perceived as more pleasant with a high effect size of 1.37 and a p-value of 10^{-7}. Greenwald et al.[GMS98] reported an effect size of 1.35 and a p-value of 10^{-8}. Nosek et al.[NBG02a] found that female names were more strongly associated with family than career with an effect size of 0.72 and $p < 10^{-2}$. We found the same relationship with a bigger effect size

Target words	Attribute words	Effect size	p
Flowers vs. insects	Pleasant vs. unpleasant	1.37	10^{-7}
Male vs. female terms	Career vs. family	1.24	10^{-2}
Male vs. female occupations	Competence vs. incompetence	0.49	0.11
Natural vs. social sciences	Male vs. female terms	1.20	10^{-2}
German vs. Asian names	Pleasant vs. unpleasant terms	1.04	10^{-2}
German vs. Turkish names	Pleasant vs. unpleasant terms	1.57	10^{-5}
Christian vs. Islamic terms	Pleasant vs. unpleasant	1.29	10^{-2}

Table 5.1: Results of bias tests trained on big corpus of German news articles from 2011 to 2016

of 1.24 and $p = 10^{-2}$. While we used the word lists of Gawronskis et al.'s experiments regarding biases against Asian and Turkish people[Gaw02], the author's do not provide effect sizes for their experiments. Therefore we compare our results with similar experiments by other authors. Crespillo measured negative bias towards Turkish names as opposed to German names and detects them with an effect size of 0.9 and $p < 10^{-3}$[Cre09]. We obtained a much larger effect size of 1.57 and $p < 10^{-5}$. Rudman et al.[RA07] report a racial bias against Asian names versus Western names with effect size 1.41 and $p < 0.01$. We detected a weaker effect with an effect size of 1.04 and $p = 10^{-3}$. This discrepancy might be explained by cultural differences between the test subjects in Rudman et al.'s study and the authors of the articles that our model is based on or simply the variation in effect size we saw during our robustness experiments (see Section 3.7). While we could not find an IAT directly measuring biases towards Christian or Muslim religious terms, Parker et al.[PFL07] demonstrated a bias towards Western names as compared to Arab-Muslim names with an effect size of 1.56 and $p = 10^{-13}$. Our test comparing bias towards Christian or Islamic religious terms resulted in a slightly smaller effect size of 1.29 and $p = 10^{-2}$.

While our effect sizes are sometimes bigger or smaller than those in the reference studies, this is to be expected due to the different test settings, especially culturally, in the original studies and our own. Additionally our robustness tests already indicated a variance in effect size of 0.5 to be expected. All in all we can find all of the mentioned biases that the original studies found.

However, we were not able to detect a bias towards women being associated with incompetence and men with competence, finding an effect size of 0.49 and $p = 0.11$. Richeson et al. showed that while men do show the bias we tested for, women exhibit the opposite bias, associating men more strongly with incompetence and women with competence[RA01]. This may explain the *p*-value being relatively close to significance. While our model should reflect both male and female opinions and biases, since we did not filter our input data by the authors' genders, we might assume that the male-focused language structure (generic masculine) influences the biases expressed in the language. Furthermore societal power

		Original results			Caliskan et al.'s results[CBN17]		Our results	
Target words	Attribute words	Ref	Effect size	p	Effect size	p	Effect size	p
Flowers vs. insects	Pleasant vs. unpleasant	[GMS98]	1.35	$10^{-}8$	1.50	10^{-7}	1.33	10^{-7}
Instruments vs. weapons	Pleasant vs. unpleasant	[GMS98]	1.66	10^{-10}	1.53	10^{-7}	1.72	0.0
Male vs. female terms	Career vs. family	[NBG02a]	0.72	$< 10^{-2}$	1.81	10^{-3}	1.24	10^{-2}
Math vs. arts	Male vs. female terms	[NBG02a]	0.82	$< 10^{-2}$	1.06	0.18	-0.10	0.50
Science vs. arts	Male vs. female terms	[NBG02b]	1.47	10^{-24}	1.24	10^{-2}	-0.14	0.49
Mental vs. physical disease	Temporary vs. permanent	[MP11]	1.01	10^{-3}	1.38	10^{-2}	1.54	10^{-4}
Young vs. old people's names	Pleasant vs. unpleasant	[NBG02a]	1.42	10^{-2}	1.21	10^{-2}	0.09	0.43

Table 5.2: Comparison of original IAT studies, Caliskan et al.'s experiments on English GloVe vectors and our experiments on German word2vec vectors

structures might play a role here. Since women are still disadvantaged in various areas of society, we might expect the language to also reflect the biases of the more dominant male group.

5.2 Comparison with English biases

To compare the biases that we detect in our model with those found in the English model by Caliskan et al.[CBN17], we directly translate the respective IATs and apply them to our model. We present the results in Table 5.2. It should be noted again that our p-values are computed using a stochastic approximation, since the exact computation is not feasible. Therefore p-values of exactly 0 are only an approximation and do not indicate absolutely certainty.

We see that we can replicate the results for flowers vs. insects, instruments vs. weapons and career vs. family and achieve effect sizes closer to the original psychological studies than the experiments on the English model did. For mental vs. physical disease we report an effect size of 1.54, which is both higher than the original result of 1.01 and Caliskan et al.'s result of 1.38. Our p-values indicate significance for all of the mentioned experiments.

Contrary to Caliskan et al. and the original studies we do not achieve significant results for math vs. arts, science vs. arts and young vs. old people's names. It should be noted that while it is common practice for the construction of IATs to conduct preliminary studies to determine which words are used, we only translated the IATs without any such tests. Therefore especially aspects like which German names we picked for the old vs. young people's names IAT can have a strong influence on the results. We assume that this is the reason for the insignificance of this specific test. The insignificance of the association between women and arts or social sciences, as compared to math and natural sciences, is most probably not the result of such translation issues, because these specific translations leave little room for error. We believe that this insignificance might be caused by the training corpus containing reports about the various initiatives that try to encourage more women to study and work in the areas of natural sciences, technology and math. This would

result in female terms being more strongly associated with these subjects than would be expected otherwise.

Since our main data set is not publicly available, we repeated the experiments on the publicly available WMT corpus described in Section3.2 to make replication possible. The results are qualitatively the same and the exact numbers are reported in Appendix B.

Chapter 6

Discussion

In this chapter we discuss our results and put them into the context of existing research. We mention limitations of our experiments and possible future research. Finally we discuss the general problem posed by societal biases in machine learning and its impact on society.

6.1 Summarizing our results

We have shown that the WEAT produces p-values that are robust towards subsampling and permutation for most tests, as long as the coverage rate of the word sets used are high enough. Apart from one exception we found that all tests became more robust with growing size of input data. We found that the effect size varied by an order of half a standard deviation while only changing the order of the input sentences, even for tests with very low p-values. This variation should be considered when interpreting the results of bias tests. One of the tests showed inconsistent behavior when varying the size of the input corpus. We were not able to find a definitive explanation for this behavior, though. This inconsistency must be examined further before we can use the results of the WEAT for definitive statements about biases in corpora. All such statements in this thesis should therefore be interpreted accordingly.

If we find an explanation for the behavior of the outlier test we will be able to use this method to detect and compare biases in different corpora. We could, for example, compare different web sites, forums or news papers. We might even try to conduct regional or temporal comparisons, assuming the availability of corresponding datasets.

Afterwards, we investigated how gender information is represented in German word vector models. We found that we can replicate the findings of Bolukbasi et al.[BCZ+16] by identifying a single-dimensional gender subspace. However, to find the gender subspace we had to change the terms we used to identify the subspace, as compared to the ones used in the original research. Using the traditional gender-specific terms like "he:she" or "mother:father" did not result in a clearly cut-off gender subspace. Instead, we needed

to use gendered pairs of words describing the same occupation. While our final approach used 132 word pairs, the qualitative result was already the same as Bolukbasi et al.'s result when using only 13 occupation word pairs. It is unclear why the first approach did not succeed, since the phrases we used are very clearly associated with their respective gender. A possible explanation would be that for example the words "mother" and "father" do not only differ in their gender. They also differ in a lot of other biases that are stereotypically associated with them. Mothers are stereotypically perceived as more loving and supportive, while fathers are perceived as more distant and strict.
However, these biases should not be too different in the culture reflected by the English word2vec models. If the presence of societal biases would explain why our first approach did not work, then the original authors' approach should not have succeeded either. Therefore this attempt to explain this difference between the two models is probably not the correct reason.

Our following approach to find a relationship between grammatical and natural gender, unfortunately, failed. Any approach to measure the association with natural gender is almost inevitably strongly connected to the respective grammatical gender. Therefore we are not able to separate the influences of the two different gender aspects.

Next, we surprisingly found that both generic, as well as purely male terms, have a lower average gender association with other male values than female terms with other female terms. Put simply: Male terms are less manly than female terms are womanly. We assume that this is caused by the influence of the generic masculine extending also to the non-generic terms and thereby weakening their gender associations. This would mean that when comparing the similarity of terms the non-gender components would be more impactful. This of course leads to less similarity among male terms since they are generally different from each other, except for their gender component. Another attempt to explain this would be that, also caused by the generic masculine, the male case is always the default. Therefore the male terms are less strongly gendered since the context of a word does not need to indicate that a word is male. It is always assumed to be male if not explicitly stated otherwise.

Our last result concerning gender information also differs from Caliskan et al.'s results. They found a very strong correlation between gender occupation statistics and the gender association in the model with correlation coefficient 0.9. We were only able to detect a weak correlation with coefficient 0.32. One reason for this is surely that English occupation terms are fully generic. They do not assume any gender but are used to describe both genders. Therefore the frequency with which they are used in a male or female context can directly be detected by the model in the training process. In German, the generic

term is only partly generic. A separate term exists if we only refer to female individuals, distributing the gender associations of the occupation among multiple words.

Finally, we found that we can detect four out of seven different biases found in the English word2vec models in our German models too. While we can explain one of the results with possible translation issues, the other results seem to indicate true differences between languages or at least underlying datasets. We found that both science and math are not significantly more strongly associated with men than women, as compared to art being associated with women instead of men. Since we also found that women are more strongly associated with family and men with career, the reason is probably not one of gender structure within the model. Seeing that we detected a significant bias in the natural vs. social sciences test, the reason for this is probably that art is not more closely related to either men or women. It might even be more strongly associated with men than women. An explanation for women being more closely associated with natural sciences and math could be the presence of news reports about corresponding initiatives. In that case we would expect the natural vs. social sciences test to also show insignificant results, which it did not. However, the test comparing the gender association of natural and social sciences also showed inconsistent results during the robustness experiments, so its result might be unreliable.
In addition, we also detected the expected racial biases against Asian and Turkish people and against Islam, as compared to Christianity.

We, therefore, conclude that we can detect most of the societal biases we tested for in our word2vec model, thereby confirming the results of Caliskan et al.[CBN17] for German texts.

6.2 Methodical limitations

A strong limitation of the methodology followed from sticking closely to the IATs from which Caliskan et al. adapted the bias tests we used. We mostly used the original word lists (or their translations) of existing IATs. However, these tests were designed to be applied to human test subjects, resulting in a rather low number of words per test category. Since computers have no limitations regarding how many words we can test them for, we believe that increasing the number of words used per test category should yield even more reliable bias tests. If we only use five words per category (see Gender/Career test), an unusual embedding for one of them could already have a big influence on the whole test result. Especially when using news reports as a data basis, it is easy to imagine cases in which stories spread across several days and news outlets, leading to specific words being over-represented in unusual contexts. Choosing which words to include in a given bias test should then of course not be decided based on word frequency of a given corpus to prevent

selection bias. Instead, the preliminary studies that are done in psychological research to determine words to be included in an IAT should be consulted. We note that such problems as outlined above can not be ruled out for our experiments. The big size of the dataset, ranging over several years, should counteract these effects to some degree, though.

Furthermore, we only used one parameter setting for all our experiments. The robustness behavior of the tests might vary for other settings, especially when using word embeddings of higher or lower dimensionality.

It should also be noted that we exclusively operated on word2vec models, since the performance of the GloVe models we tested during our preliminary parameter optimization were significantly worse. Vector models trained with other approaches might show different behavior.

6.3 Future research

We were not able to clearly identify whether the differences between German and English word vectors with regard to biases are caused by the structural differences between the languages. Therefore future research might try to investigate this question by identifying several languages that differ in the way they represent certain aspects of reality in their language. Gender seems to be a good candidate for such comparisons, since it can be represented with a low dimensionality and is a very basic aspect of human life. Ideally, this kind of machine learning research could be combined with psychological research, so that the biases measured in text corpora can be compared to the biases measured in the inhabitants of the respective countries.

Another possibility would be to take a more in-depth and case-based look at gender-representation in German word embeddings. Instead of computing the gender association value by [CBN17] to determine how male or female a term is perceived to be, we can also use the projection into the gender subspace. This method is a lot cheaper, so we could, for example, compute the projection of every single word vector in our model into the gender subspace. We could then examine the terms that are perceived by the model as extremely male or female are surprising or if any of the words perceived as gender-neutral are actually gendered in human understanding. Furthermore, we could inspect which of the male or female occupation terms input into the PCA in Section 4.1 have surprising gender associations. Our results when determining the gender association of male and female terms (Figure 4.4) indicated that the differences between how strongly male or female occupation terms are associated with their gender vary greatly.

Lastly, an interesting case-study can be conducted by looking at the few phrases that are used in a gender-neutral way in the German language. "Studierende" is the prime example, as it refers to university students of any gender and has been widely adopted by the universities themselves. There are also efforts to establish a new word ending to

indicate gender-neutrality, as proposed for example in [dHUzB15]. The only such word that got some popularity was "Professx", referring to a professor, which is also present in our model. However, it was only ever used in the report about this initiative and never widely adopted in normal texts. Therefore the embedding can not be expected to reflect the word's meaning in everyday use and is therefore not useful for us. The widely adopted method to be gender-inclusive is to merge male and female terms, for example: Lehrer + Lehrerin = LehrerIn / Lehrer_in / Lehrer*in, although there is no consensus on which form to use.

A very interesting methodical approach to solving the problem of biased machine learning models would be to develop a variation of the word2vec model that takes bias constraints into consideration while training the neural network. While Bolukbasi et al.[BCZ+16] modified the vectors after training was completed, we could also try to modify the training process instead. Such constraints might take the form that we identify a set of bias-relevant words, for example any terms for which a gender is relevant, like 'mother', male', 'girlfriend'. We then demand that for any two words w_{i1}, w_{i2} in this set and any one word w_o not within this set, the distance between w_{i1} and w_o is equal to the distance between w_{i2} and w_o. This would, for example, solve the problem of "man is to programmer as woman is to homemaker"[BCZ+16]. However, it also makes painfully clear the difficulty of having to explicitly enumerate each of the bias-relevant words that may be included in the corpus. Anything less than this extensive enumeration potentially leads to partially biased vector models, which would make the detection of such biases even harder. Furthermore, even knowing which kinds of biases are relevant is already a very difficult problem. As Caliskan et al. mention: "[...] societal understanding of prejudice is constantly evolving, along with our understanding of humanity and human rights, and also varies between cultures."[IBN16] Therefore this approach in the suggested form does not yet seem to be a solution to the problem, but it may be a reasonable stepping stone on the way to bias-free models.

6.4 Dealing with societal biases in machine learning

The problems indicated by our research might not be very severe in practical applications yet. The worst thing that might currently happen when applying biased word embeddings is that the web search[NMCC16] ranks a competitor more favorably or the résumé screening software[HTG+15] discards the application. While these consequences can lead to monetary loss, in other areas the consequences of algorithmic biases are already visible in everyday procedures and have a lot more severe consequences.

The US judicial system often uses algorithms to determine the risk of a criminal defendant re-offending. A study by ProPublica[ALMKa] investigated one of the commercial algorithms that are used to assess such risk scores. The system was found to be heavily

biased against black people. While white and black people had similar misclassification rates, the system regularly misclassified white people as more harmless than they actually were, while black people were regularly misclassified as more dangerous than they were[ALMKb].

This example demonstrates that societal biases in machine learning pose very serious risks to the freedom and justice in our societies. While humans are able to consciously decide to act contrary to biases that they are aware of having, algorithms are often applied without any such safeguards. As we can expect the application of machine learning systems in everyday life to become more and more prevalent in the future, it is of high importance to not only be aware of the biases we are introducing by the choice of the data we are training a model on. We also need to develop methods to counteract the biases in these specific models. Different approaches for this problem exist already by Dwork et al.[DHP+12], Feldmann et al.[FFM+15] and Hardt et al.[HPS16] These approaches focus on classification, however, so they can only be used to reduce the bias in applications that use word2vec as a way to generate features from texts. They can not be used to reduce the biases in the word2vec model itself. They are also not applicable to all classification settings that employ word2vec features. For example when dealing with web search the documents are not annotated as belonging to a protected group and doing so is not trivial, so the mentioned approaches would not be able to establish non-discrimination of the protected class.

Moving forward, we must actively investigate our models for unwittingly introduced biases if they are ever supposed to be used outside of research settings. The risks that machine learning brings with itself are big, but so are the benefits. If we apply these powerful techniques with enough caution and deliberation we will be able to realize their great potential and use them to change society for the better.

Appendix A

Further experimental results

A.1 Preliminary tests

The GloVe model proved to be significantly worse than any of the word2vec models during our preliminary experiments (see Section 3.5). In contrast to the Word2Vec models, we observe that higher context window size results in higher accuracy in every case. Regarding the dimensionality of the word vectors, we see a similar behavior as with the Word2Vec models. Increasing the dimensionality of the word vectors from 100 to 300 results in a very big increase in accuracy, while further increasing it to 500 results in a slight decrease. The best accuracy of 31% was achieved with a context window size of $c = 15$ and a dimensionality of $d = 300$. Figure A.1 illustrates these results.

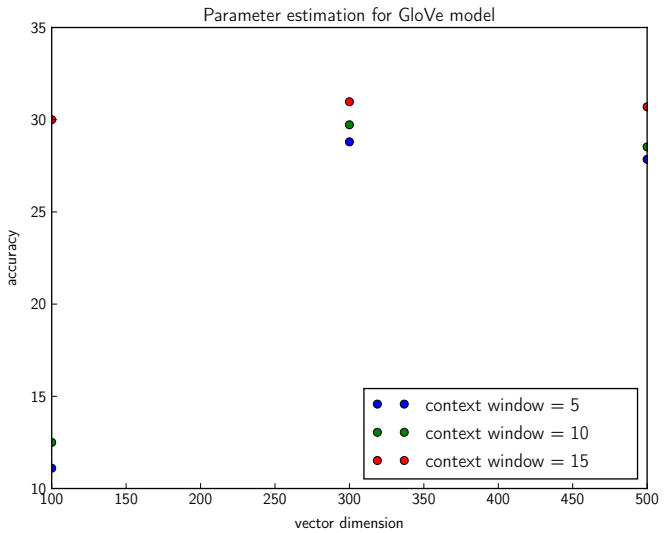

Figure A.1: Accuracy for GloVe models trained with vector dimension 100, 300 or 500. The different colored points show configurations with context windows of size 5, 10 or 15. Higher accuracy is more desirable.

Appendix B

Bias test

B.1 Bias tests on public dataset

To allow our results to be replicated, we executed the bias tests from Section 5 on the publicly available WMT corpus (see Section 3.2). The results are shown in Table B.2 and B.1. The results are qualitatively the same as for the Wortschatz corpus. The exact numbers are different but of course this is to be expected when using two corpora that are not exactly the same.

		Original results			Caliskan et al.'s results		Our results	
Target words	Attribute words	Ref	Effect size	p	Effect size	p	Effect size	p
Flowers vs. insects	Pleasant vs. unpleasant	[GMS98]	1.35	$10^{-}8$	1.50	10^{-7}	1.30	10^{-7}
Instruments vs. weapons	Pleasant vs. unpleasant	[GMS98]	1.66	10^{-10}	1.53	10^{-7}	1.64	0.0
Male vs. female terms	Career vs. family	[NBG02a]	0.72	$< 10^{-2}$	1.81	10^{-3}	1.08	0.02
Math vs. arts	Male vs. female terms	[NBG02a]	0.82	$< 10^{-2}$	1.06	0.18	-0.18	0.59
Science vs. arts	Male vs. female terms	[NBG02b]	1.47	10^{-24}	1.24	10^{-2}	0.22	0.35
Mental vs. physical disease	Temporary vs. permanent	[MP11]	1.01	10^{-3}	1.38	10^{-2}	1.05	0.03
Young vs. old people's names	Pleasant vs. unpleasant	[NBG02a]	1.42	10^{-2}	1.21	10^{-2}	0.24	0.34

Table B.1: Comparison of original IAT studies, Caliskan et al.'s experiments on English GloVe vectors and our experiments on German word2vec vectors

Target words	Attribute words	Effect size	p
Flowers vs. insects	Pleasant vs. unpleasant	1.01	10^{-4}
Male vs. female terms	Career vs. family	1.08	0.02
Male vs. female occupations	Competence vs. incompetence	0.42	0.15
Natural vs. social sciences	Male vs. female terms	0.78	0.08
German vs. Asian names	Pleasant vs. unpleasant terms	1.05	10^{-2}
German vs. Turkish names	Pleasant vs. unpleasant terms	1.41	10^{-4}
Christian vs. Islamic terms	Pleasant vs. unpleasant	1.44	10^{-3}

Table B.2: Results of bias tests trained on big corpus of German news articles from 2007 to 2016

B.2 Word lists for IAT tests

Flowers and Insects

- Flower terms: Aster, Klee, Hyazinthe, Ringelblume, Mohn, Azalee, Krokus, Schwertlilie, Orchidee, Rose, Blauchglöckchen, Narzise, Flieder, Stiefmütterchen, Tulpe, Butterblume, Gänseblümchen, Lilie, Pfingstrose, Pfeilchen, Nelke, Gladiole, Magnolie, Petunie, Zinnie
- Insect terms: Ameise, Raupe, Floh, Grashüpfer, Spinne, Bettwanze, Hundertfüßler, Fliege, Made, Vogelspinne, Biene, Kakerlake, Mücke, Moskito, Termiten, Käfer, Grille, Hornisse, Motte, Wespe, Libelle, Blattlaus, Pferdebremse, Küchenschabe, Borkenkäfer
- Pleasant terms: Heiterkeit, Spaß, Freundschaft, Glück, Freude, Gesundheit, Liebe, Paradies, Begeisterung, Entspannung
- Unpleasant terms: Ärger, Elend, Hass, Angst, Unglück, Verrat, Streit, Pest, Krankheit, Panik

Gender bias: Natural and social sciences

- Female: Mädchen, Weiblich, Tante, Tochter, Ehefrau, Frau, Mutter, Großmutter
- Male: Mann, Junge, Vater, Männlich, Großvater, Ehemann, Sohn, Onkel
- Social sciences: Philosophie, Kunst, Geschichte, Literaturwissenschaften, Sprachwissenschaften, Musik, Geschichte
- Natural sciences: Biologie, Physik, Chemie, Mathematik, Geologie, Ingenieurswissenschaften

Gender bias: Career and family

- Male: Johannes, Lukas, Daniel, Paul, Thomas
- Female: Julia, Michaela, Anna, Laura, Sofie
- Career: Verwaltung, Berufstätigkeit, Unternehmen, Gehalt, Büro, Verdienst, Karriere
- Family: Zuhause, Eltern, Kinder, Familie, Hochzeit, Ehe, Verwandte

Gender bias: Gendered job titles and competence

- Male job titles: Arzt, Rechtsanwalt, Professor, Ingenieur, Informatiker, Pfleger, Metzger, Lehrer, Frisör, Florist, Kindergärtner, Kaufmann, Berater
- Female job titles: Ärztin, Rechtsanwältin, Professorin, Ingenieurin, Informatikerin, Pflegerin, Metzgerin, Lehrerin, Frisörin, Floristin, Kindergärtnerin, Kauffrau, Beraterin
- Competence: kompetent, fähig, professionell, anspruchsvoll, Erfolg
- Incompetence: inkompetent, unfähig, unprofessionell, anspruchslos, Misserfolg

Racist bias: German and Asian names

- German names: Günther, Matthia,s Harald, Stefan, Dieter, Eberhard, Wolfgang, Volker, Michael, Konrad
- Asian names: Li, Wang, Zhao, Zhang, Peng, Chang, Wu, Quian, Feng, Jiang
- Pleasant terms: Heiterkeit, Spaß, Freundschaft, Glück, Freude, Gesundheit, Liebe, Paradies, Begeisterung, Entspannung
- Unpleasant terms: Ärger, Elend, Hass, Angst, Unglück, Verrat, Streit, Pest, Krankheit, Panik

Racist bias: German and Turkish names

- German names: Günther Matthias Harald Stefan Dieter Eberhard Wolfgang Volker Michael Konrad
- Turkish names: Mehmet Kemal Ahmed Erkan Özal Murat Abdullah Ali Mohammed Mustafa
- Pleasant terms: Heiterkeit, Spaß, Freundschaft, Glück, Freude, Gesundheit, Liebe, Paradies, Begeisterung, Entspannung
- Unpleasant terms: Ärger, Elend, Hass, Angst, Unglück, Verrat, Streit, Pest, Krankheit, Panik

Religious bias: Christianity and Islam

- Christian terms: Christentum, Bibel, Kirche, Jesus, Ostern, Christ
- Muslim terms: Islam, Koran, Moschee, Mohammed, Ramadan, Muslim
- Pleasant terms: Heiterkeit, Spaß, Freundschaft, Glück, Freude, Gesundheit, Liebe, Paradies, Begeisterung, Entspannung
- Unpleasant terms: Ärger, Elend, Hass, Angst, Unglück, Verrat, Streit, Pest, Krankheit, Panik

Correlation of male job titles with gender distribution

- Male occupation terms: arzthelfer, zahnarzthelfer, ernaehrungsberater, erzieher, kosmetiker, schneider, florist, friseur, kassierer, apotheker, masseur, sozialarbeiter, buchhalter, verkaeufer, bibliothekar, heilpraktiker, sozialpaedagoge, optiker, dolmetscher, steuerberater, kellner, tierarzt, zahnarzt, philosoph, koch, fahrkartenkontrolleur, psychologe, tankwart, fotograf, schauspieler, arzt, journalist, sportlehrer, jurist, konditor, biologe, musiker, artist, seelsorger, chemiker, architekt, baecker, manager, landwirt, pilot, gaertner, informatiker, fischer, physiker, raumausstatter, pfoertner, fleischer, elektroingenieur, schlosser, tischler, maurer, elektroniker
- Percentage of male workers: 0.8, 0.8, 2.4, 4.2, 4.6, 5.7, 6.1, 7.0, 19.3, 19.7, 19.8, 22.1, 24.1, 25.4, 25.9, 27.8, 27.9, 28.8, 29.6, 31.4, 34.6, 37.2, 44.9, 45.1, 46.3, 47.0, 50.8, 51.2, 51.2, 52.4, 54.6, 54.8, 56.2, 57.7, 59.3, 65.2, 67.8, 70.1, 72.0, 73.1, 73.6, 76.8, 78.1, 78.9, 79.5, 81.5, 82.1, 83.3, 85.2, 89.1, 90.5, 94.4, 96.3, 96.9, 99.5, 98.7

Bibliography

[ALMKa] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. How we analyzed the compas recidivism algorithm. `https://www.propublica.org/article/how-we-analyzed-the-compas-recidivism-algorithm` Accessed: 2017-07-08.

[ALMKb] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner. Machine bias - there's software used across the country to predict future criminals. and it's biased against blacks. `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing` Accessed: 2017-07-08.

[BCF+16] Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany, August 2016. Association for Computational Linguistics.

[BCZ+16] 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.

[CBN17] Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

[Cre09] Raquel Crespillo. Effekte der adressateneinstellung auf erinnerungen in der interaktion mit fremdgruppenangehörigen: die bedeutung einer sozialen realitätsbildung. 2009.

[DHP+12] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations*

in Theoretical Computer Science Conference, ITCS '12, pages 214–226, New York, NY, USA, 2012. ACM.

[dHUzB15] AG Feministisch Sprachhandeln der Humboldt-Universität zu Berlin. Was tun? sprachhandeln – aber wie? w_ortungen statt tatenlosigkeit! Brochure, 2015. `http://feministisch-sprachhandeln.org/wp-content/uploads/2015/04/sprachleitfaden_zweite_auflage.pdf` Accessed: 2017-07-08.

[DSG14] Cícero Nogueira Dos Santos and Maira Gatti. Deep convolutional neural networks for sentiment analysis of short texts. In *COLING*, pages 69–78, 2014.

[fAuB] Institut für Arbeitsmarkt und Berufsforschung. Berufe im Spiegel der Statistik. `http://bisds.infosys.iab.de/bisds/faces/Start.jsp` Accessed: 2017-07-08.

[FFM+15] Michael Feldman, Sorelle A Friedler, John Moeller, Carlos Scheidegger, and Suresh Venkatasubramanian. Certifying and removing disparate impact. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[Gaw02] Bertram Gawronski. What does the implicit association test measure? a test of the convergent and discriminant validity of prejudice-related iats. *Experimental psychology*, 49(3):171, 2002.

[GEQ12] Dirk Goldhahn, Thomas Eckart, and Uwe Quasthoff. Building large monolingual dictionaries at the leipzig corpora collection: From 100 to 200 languages. In *LREC*, pages 759–765, 2012.

[GMS98] 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.

[HPS16] Moritz Hardt, Eric Price, and Nathan Srebro. Equality of opportunity in supervised learning. *CoRR*, abs/1610.02413, 2016.

[HTG+15] C Hansen, M Tosik, G Goossen, C Li, L Bayeva, F Berbain, and M Rotaru. How to get the best word vectors for resume parsing. In *SNN Adaptive Intelligence/Symposium: Machine Learning*, 2015.

[IBN16] Aylin Caliskan Islam, Joanna J. Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora necessarily contain human biases. *CoRR*, abs/1608.07187, 2016.

[Joa02] Thorsten Joachims. *Learning to Classify Text Using Support Vector Machines: Methods, Theory and Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA, 2002.

[JOP+] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–. `http://www.scipy.org/` Accessed: 2017-07-08.

[LGD15] 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.

[MCCD13] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

[Mik] Tomas Mikolov. word2vec – tool for computing continuous distributed representati- ons of words. `https://code.google.com/archive/p/word2vec/` Accessed: 2017-07-08.

[MJW+15] Katharina Morik, Alexander Jung, Jan Weckwerth, Stefan Rötner, Sibylle Hess, Sebastian Buschjäger, and Lukas Pfahler. Untersuchungen zur Analyse von deutschsprachigen Textdaten. 2015.

[Mor] Manuela Moraes. Glosbe - the multilingual online dictionary. `https://glosbe.com/a-api` Accessed: 2017-07-08.

[MP11] Lindsey L Monteith and Jeremy W Pettit. Implicit and explicit stigmatizing attitudes and stereotypes about depression. *Journal of Social and Clinical Psychology*, 30(5):484–505, 2011.

[MSC+13] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[MYZ13] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. 2013.

[Mü] Andreas Müller. GermanWordEmbeddings. `http://devmount.github.io/GermanWordEmbeddings/` Accessed: 2017-07-08.

[NBG02a] Brian A Nosek, Mahzarin Banaji, and Anthony G Greenwald. Harvesting implicit group attitudes and beliefs from a demonstration web site. *Group Dynamics: Theory, Research, and Practice*, 6(1):101, 2002.

[NBG02b] Brian A Nosek, Mahzarin R Banaji, and Anthony G Greenwald. Math= male, me= female, therefore math≠ me. *Journal of personality and social psychology*, 83(1):44, 2002.

[NMCC16] Eric Nalisnick, Bhaskar Mitra, Nick Craswell, and Rich Caruana. Improving document ranking with dual word embeddings. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, pages 83–84, Republic and Canton of Geneva, Switzerland, 2016. International World Wide Web Conferences Steering Committee.

[PFL07] Jaihyun Park, Karla Felix, and Grace Lee. Implicit attitudes toward arab-muslims and the moderating effects of social information. *Basic and Applied Social Psychology*, 29(1):35–45, 2007.

[pro] Project Implicit - Impliziter Assoziationstest. `https://implicit.harvard.edu/implicit/germany/` Accessed: 2017-07-08.

[PSM] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation (code on github). `https://github.com/stanfordnlp/GloVe` Accessed: 2017-07-08.

[PSM14] 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.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[RA01] Jennifer A Richeson and Nalini Ambady. Who's in charge? effects of situational roles on automatic gender bias. *Sex Roles*, 44(9):493–512, 2001.

[RA07] Laurie A. Rudman and Richard D. Ashmore. Discrimination and the implicit association test. *Group Processes & Intergroup Relations*, 10(3):359–372, 2007.

[RN03] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2 edition, 2003.

[Ron14] Xin Rong. word2vec parameter learning explained. *CoRR*, abs/1411.2738, 2014.

[ŘS10] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.

[RU11] Anand Rajaraman and Jeffrey David Ullman. *Mining of Massive Datasets*. Cambridge University Press, New York, NY, USA, 2011.

[sta] WMT17 News Crawl corpora. `http://www.statmt.org/wmt17/translation-task.html` Accessed: 2017-07-08.

[wik] Wikipredia- Liste von Frauenanteilen in der Berufswelt. `https://de.wikipedia.org/wiki/Liste_von_Frauenanteilen_in_der_Berufswelt` Accessed: 2017-07-08.