Effect of Debiasing on Information Retrieval

Emma J. Gerritse(✉) and Arjen P. de Vries

Institute for Computing and Information Sciences, Radboud University, Nijmegen, The Netherlands
emma.gerritse@ru.nl, a.devries@cs.ru.nl

Abstract. Word embeddings provide a common basis for modern natural language processing tasks, however, they have also been a source of discussion regarding their possible biases. This has led to a number of publications regarding algorithms for removing this bias from word embeddings. Debiasing should make the embeddings fairer in their use, avoiding potential negative effects downstream. For example: word embeddings with a gender bias that are used in a classification task in a hiring process. In this research, we compare regular and debiased word embeddings in an Information Retrieval task. We show that the two methods produce different results, however, this difference is not substantial.

Keywords: Query expansion · Word embeddings · Bias

1 Introduction

Word embeddings have been used for many downstream Natural Language Processing (NLP) tasks lately. They are a method of presenting words in a high dimensional vector space, learned by applying machine learning on large text corpora. It has been shown that these embeddings can be very useful in many tasks, hence their wide-spread usage. However, this method is not without any critique. One of the most influential critique papers demonstrates gender bias in pre-trained word embeddings derived from Google News [1].

The authors of that work claim that having a gender bias in word embeddings can be damaging for downstream tasks like information retrieval. Imagine the scenario where a user wants to retrieve documents of people working in a male-dominated field, like computer science. If the embeddings of male names are closer to the embedding of computer science than the embeddings of female names, it could be that John's page gets a higher ranking than Jane's, even when the contents of their pages are otherwise similar.

While this scenario would be very alarming, to our knowledge no experiments have shown this to happen in a practical setting. That is why, in this research, we investigate this. We empirically show the difference in retrieval outcomes when performing a retrieval task with or without debiased embeddings.

L. Boratto et al. (Eds.): BIAS 2020, CCIS 1245, pp. 35–42, 2020.
https://doi.org/10.1007/978-3-030-52485-2_4

For this, we perform a retrieval experiment on TREC Robust. We incorporate biased and debiased embeddings for query expansion, using a method based on [2]. We compare the difference in expanded terms, and also the difference in the effectiveness measurements obtained for the different embeddings.

2 Related Work

Word embeddings are a vector representation of vocabulary. To compute these vectors, many methods have been proposed. One of the best-known methods is Word2Vec [7]. This method works by training a neural network that predicts words considering their context. The Skip-Gram variant of the method predicts a word's context from its observation, while the Continuous Bag of Words variant predicts the word occurrence from its context. Of the two variants, the Skip-Gram is used most widely. The resulting word representations (called word embeddings) have been successfully used in a range of NLP tasks, including sentence classification [5] and text classification [10].

Word embeddings can be used for document retrieval as well. In [2], query terms are expanded with terms found by using word embeddings. The idea here is that you can use the embedding space to find words similar to the other words in the query. This paper shows that using locally trained word embeddings will always perform better than globally trained embeddings for document retrieval. The retrieval is done by combining the expanded terms with a language model. An updated language model is computed for the language model p_q of the query q. This expansion, p_{q^+}, is combined with p_q by a linear combination:

$$p_q^1(w) = \lambda p_q(w) + (1 - \lambda) p_{q^+}(w) \qquad \lambda \in (0, 1) \tag{1}$$

p_{q^+} is computed in the following way. Let U be the embedding matrix of size $|D| \times k$. Let q be the $|D| \times 1$ vector describing the query. Then the query expansion can be computed by taking the top k terms from the resulting $|D| \times 1$ vector $UU^T q$. This is identical to computing $\text{argmin}_{w' \in U} \sum_{w \in Q} w \cdot w'$.

While very useful, word embeddings have also triggered controversy. Pretrained embeddings have been shared by researchers to be easily used, however, researchers have exposed inherent biases. In [1] for example, the pretrained word embeddings trained on Google News by [7] are shown to exhibit common gender stereotypes on well-known analogy tasks. One of the appealing examples of analogies in [6] is $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$; it looks like word embeddings capture semantic linguistic knowledge! In [1] however, it is shown that less desirable analogies *also* exist in the embedding space, like $\overrightarrow{computer\_programmer} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{homemaker}$ (a particularly shocking example for the computer science field, where many researchers actively try to overcome such prejudices and work toward a better gender balance). They found many more examples for similarly biased analogies, and then asked mechanical Turkers to rate the level of bias in these examples. It turns out that many of these analogies have some degree of gender bias, which is why they propose two methods (hard and soft debias) to remove this bias.

This paper has lead to quite some discussion among academics. The paper [8] points out that the method of detecting biased analogies might not be fair, because of the way the `GENSIM` packages handles these analogies. The analogy function in this package can never return one of the input words, for example in the example given above, when giving 'king', 'man' and 'woman' as input, these three words can not be given as output. When removing these constraints, it turns out that many of the analogies discussed in [1] do not hold anymore. Most noticeably, without this constraint, the result of the analogy $\overrightarrow{computer\_programmer} - \overrightarrow{he} + \overrightarrow{she}$ is $\overrightarrow{computer\_programmer}$.

Further exploration of the validity of biased analogies is reported by [3]. Here, robustness of analogies is defined in the following way. If for example $\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$, then the reverse should also hold:

$\overrightarrow{queen} - \overrightarrow{woman} + \overrightarrow{man} = \overrightarrow{king}$. If the reverse does not hold, the analogy is not robust. Several of the analogies in [1] were tested. Most importantly, the title-giving analogy is not robust: the answer to $\overrightarrow{computer\_programmer} - \overrightarrow{he} + \overrightarrow{she}$ is indeed homemaker. But when computing the reverse, the answer to the analogy is $\overrightarrow{homemaker} - \overrightarrow{she} + \overrightarrow{he} \approx \overrightarrow{carpenter}$. It seems illogical that these analogies are not always robust, since they are often denoted with an '=' sign instead of an '≈' sign. However, it is important to consider that the embedding space is very sparse. Analogies are computed in `GENSIM` by finding the closest word-vector to for example the result of $\overrightarrow{computer\_programmer} - \overrightarrow{he} + \overrightarrow{she}$. Looking at the results, it seems that the closest neighbour can still be relatively far away in the embedding space. In this example, $\cos(\overrightarrow{computer\_programmer} - \overrightarrow{he} + \overrightarrow{she}, \overrightarrow{homemaker}) = 0.57$ while with a robust analogy the result is $\cos(\overrightarrow{king} - \overrightarrow{he} + \overrightarrow{she}, \overrightarrow{queen}) = 0.73$. Because the answer to the analogy is relatively distant, it is not surprising that the reverse sequence of operations would identify a different word-vector as the most similar result. This process of reversing the analogy can be repeated until the results are robust. For the home maker example, the analogies converge at $\overrightarrow{carpenter} - \overrightarrow{he} + \overrightarrow{she} \approx \overrightarrow{seamstress}$, and for this analogy is robust. While this analogy is still biased, it seems less severe than the *computer programmer* and *homemaker* combination.

Analogies also seem to depend heavily on the choice of words. When computing the analogy for $\overrightarrow{programmer}$ instead of $\overrightarrow{computer\_programmer}$, the result of the analogy is $\overrightarrow{programmers}$. Then when you look at the convergence of the analogy, it results in the names of two random people. This may be because of the sparseness of the embedding vectors, and because of the constraints of the analogies as discussed in [8].

Finally, people observed that the debiased word vectors still encode some degree of bias: biases can be recovered from the data. The authors of [4] first show that clusters of word embeddings, using k-means to assign the most biased words to two clusters, still align with the given gender with an accuracy of 92.5% for the debiased version. They also trained a Support Vector Machine to predict whether a word was a male or female word, and with an accuracy of 96.5% they were able to recover the gender information, even when debiased. So it seems

that debiasing only superficially covers up the bias. This result can be seen with debiasing methods applied before and after computing the embeddings.

3 Method

3.1 Debiasing Word Embeddings

In this paper, we investigate the effects of the hard debiasing method described in [1]. We give a description of the method here, but for exact details, we refer the reader to the original paper.

Debiasing the word embeddings works as follows. First, define a gendered set consisting of words with a clear gender component (e.g. *man, woman, male, female, brother, sister, etc.*). Use this set to compute the gender direction B in the vector space. Next, define the set N of words which need to be neutralized or debiased. Project the words in N onto the gender direction B, and normalize their length. Finally, define a set of equal pairs E, containing pairs like $(man, woman)$, which are also centered around the origin (to prevent vectors of one of the genders to have a greater length than the other).

After debiasing, for any neutralized word $w \in N$ and any equal pair $(e_1, e_2) \in E$, it should hold that $\overrightarrow{w} \cdot \overrightarrow{e_1} = \overrightarrow{w} \cdot \overrightarrow{e_2}$ and $||\overrightarrow{w} - \overrightarrow{e_1}|| = ||\overrightarrow{w} - \overrightarrow{e_2}||$. I.e., words that should be gender neutral have equal distance to the previously defined male and female words.

The authors of [1] have shared debiased pre-trained Google News embeddings, that we use in the empirical part of this work.

3.2 Retrieval Model and Experimental Setup

Having the debiased embeddings, we now explain how we use these in a retrieval experiment. We select two different sets of pre-trained word embeddings, the standard pre-trained Word2Vec embeddings on Google News as shared by [7] and the debiased version of these embeddings (as explained in the section above). For the dataset to test our model, we selected the TREC Robust 04 test collection consisting of news articles, matching the domain of our embeddings. This test collection consists of 250 queries (usually called topics in IR), with a total of 311410 relevance judgments.

We removed stopwords from these queries using the `NLTK` stopword list, and we cast query terms to all lower case. We expand each of these queries with $k = 5$ terms, by computing the five closest terms to the query embedding in the embedding space with each method regarding the cosine similarity. To compute these terms, we use the `GENSIM` `most_similar` function, where the input is the stopped lowercase query terms, and the output is the top-k closest words which are not in the input words. After this, we substitute the words of the query with the expanded terms and used these for retrieval. The score is based on the method used in [2], but not identical as we use cosine instead of the dot product, and we only expand with words that de not occur in the original query.

To run our experiment, we used Anserini [9]. We ranked the documents using RM3 and BM25. This gives us three ranking files, the one with the regular queries (*Standard*), with the biased expansions (*Biased*) and with the debiased expansions (*Debiased*).

To combine the biased or debiased word embeddings based score with the standard retrieval score, we used `Coordinate Ascent` from the `RankLib` package.

$$score_{total} = \lambda score_{standard} + (1 - \lambda) score_{(de)biased} \qquad \lambda \in [0, 1] \quad (2)$$

We used cross fold validation, where we trained with 5 folds, and we optimized regarding to the metrics of NDCG@10 and ERROR@10. This gave us, for all folds with both methods, the average λ score of 0.90 ($\sigma = 0.04$).

4 Results

As we can see in Table 1, there is no significant difference in score between biased and debiased query expansion. We also see no significant difference regarding the Expanded versus the Regular version. Table 1 has two columns, one where we evaluate with respect to the full Robust 04 qrels file, and one where we compare to only the 48 queries which got different expansions. The expansions only differ in about 20% of the queries, so differences are more clear if we confine ourselves to this subset. The two query sets are denoted as **Robust Full** and **Robust Changed**, respectively.

Table 1. Results of the retrieval documents. Both expansions did not lead to any significant improvement in P30 of MAP.

Model	Robust full		Robust changed	
Score	MAP	P30	MAP	P30
Expansion biased	0.106	0.135	0.126	0.156
Expansion debiased	0.105	0.135	0.117	0.158
Regular	0.290	0.337	0.303	0.372
Regular + Expansion biased	0.290	0.339	0.306	0.377
Regular + Expansion debiased	0.290	0.338	0.305	0.375

Of the 250 analyzed queries in TREC, 48 gain a different expansion. Of those 48, 16 have a substantial difference in MAP and 18 have a substantial difference in P_30. We denote a difference in score of 0.01 as substantial (an arbitrary number defined by the Anserini script to compare runs).

We show the queries with a substantial difference in Table 2, together with the difference in expanded terms with both methods. A positive number means that the biased method performs better, while a negative number means that the debiased version performed better. In some of these queries, the change in

Table 2. Difference between biased and debiased query expansion. The first term (in italics) is the query terms, the second term is the biased expansion and the third term is the debiased expansion. Only queries with a substantial change in either MAP or P30 are listed. Words of the original query might be repeated in expansion with different capitalization. Note that words often contain spelling errors ('anti_biotics' or 'prostrate').

P30	MAP	Query: query expansion difference
−0.067	−0.046	*international organized crime*: Organized → human_trafficking
0.1	0.080	*hubble telescope achievements*: astronomical_telescope → inch_refractor_telescope
−0.267	−0.061	*women in parliaments*: gender_equality → females
0.067	0.068	*adoptive biological parents*: mother → birthmother
−0.067	−0.018	*territorial waters dispute*: Diaoyutais → Spratleys
−0.033	0.008	*anorexia nervosa bulimia*: bulimic → binge_eating
−0.033	−0.013	*health insurance holistic*: healthcare → preventative_medicine
−0.033	−0.003	*mental illness drugs*: mental_disorders → alzheimer_disease
0.033	0.006	*teaching disabled children*: cognitively_disabled → nondisabled_peers
0.367	0.119	*sick building syndrome*: headaches_nausea_diarrhea → persistent_sexual_arousal
−0.1	−0.059	*behavioral genetics*: neurobiological → neurogenetics
−0.013	0.0	*osteoporosis*: rheumatoid_arthritis → osteoarthritis
−0.033	−0.043	*heroic acts*: heroic_feats → bravery
−0.033	0.010	*women clergy*: clergywomen → bishops
−0.067	0.029	*antibiotics ineffectiveness*: anti_biotics → antibiotic_overuse antibiotic_therapy → antifungal_medications
0.0	0.084	*human genetic code*: epigenetic_reprogramming → primate_genomes
0.033	0.020	*women ordained church of england*: clergy → priests
0.033	0.018	*doctor assisted suicides*: psychiatrist → prescribed_anti_depressants
0.1	0.054	*maternity leave policies*: Maternity_Matters → Policies
−0.1	−0.069	*prostate cancer detection treatment*: differentiated_thyroid → prostrate_cancer

the expansion is as expected of a version without gender bias. For example, in the query 'women clergy', the expanded terms get changed from 'clergywomen' to 'bishops', which is a logical gender-neutral change of this word. We also see that the score here changes positively with the debiased terms.

However, in other cases, the changes in terms with the debiased version do not make much sense. For example, in the query 'sick building syndrome', query expansion 'headaches_nausea_diarrhea' changes into 'persistent_sexual_arrousal' (note the spelling mistake). Naturally, the biased version performs much better than the debiased version.

As for a possible explanation of why this might happen: If only one of the word vectors in either of the query terms changes, the aggregated query changes along, as do the 5 expanded query terms. Even if the input query changes ever so

slightly, due to the sparsity of the embedding space, completely different terms can become the closest ones.

It is interesting to see some queries are expanded with words with spelling mistakes (e.g. 'prostrate'). A possible explanation is that these words are so uncommon in the corpus, that they are not seen enough during training. This may result in words which are not properly embedded, leading to nonsensical expansions.

While gender bias is removed, some other versions of bias remain in both versions of embeddings. For example, for query number 316 'polygamy polyandry polygyny' gets expanded in both cases with 'incestuous_marriages', which can be considered a lifestyle bias. Removing all potential biases from embedding space seems infeasible with the proposed approach, because one would need to specify actual examples of every single bias that may be encoded in the data.

5 Conclusion

We carried out a comparative study on the effect of biased and debiased word embeddings on information retrieval. In about 20% of the queries, query expansions differed; where 38% of those queries that changed led to a substantial difference in documents retrieved. This corresponds to only 7% of the total number of queries. Retrieval results for debiased word embeddings may change for the better or for the worse. Taking only these experimental outcomes into account, we may conclude that the effect of debiasing word embeddings on retrieved results is not dramatic.

However, when looking at the expanded terms of a query, these terms can still be biased. Debiasing for gender will not remove other types of bias that may occur in the data from which the word embeddings have been derived. Sometimes, biases can be present of which the user is not even aware they exist. Based on our experience, we conclude that the more general problem of unfairness in document ranking cannot be addressed by the debiasing approaches found in the literature.

For further research, literature has proven that locally trained embeddings work better than globally trained embeddings for query expansion. It would be interesting to see if when training the embeddings ourselves, and debiasing the embeddings ourselves, if results will change.

References

1. Bolukbasi, T., Chang, K.W., Zou, J.Y., Saligrama, V., Kalai, A.T.: Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In: Advances in Neural Information Processing Systems 29, pp. 4349–4357 (2016)
2. Diaz, F., Mitra, B., Craswell, N.: Query expansion with locally-trained word embeddings. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 367–377 (2016)

3. Gerritse, E.: Impact of debiasing word embeddings on information retrieval. In: Proceedings of the 9th PhD Symposium on Future Directions in Information Access, CEUR Workshop Proceedings, pp. 54–59 (2019)
4. Gonen, H., Goldberg, Y.: Lipstick on a Pig: debiasing methods cover up systematic gender biases in word embeddings but do not remove them. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pp. 609–614 (2019)
5. Kim, Y.: Convolutional neural networks for sentence classification. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, pp. 1746–1751. ACL (2014)
6. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: 1st International Conference on Learning Representations, pp. 1–12 (2013)
7. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
8. Nissim, M., van Noord, R., van der Goot, R.: Fair is better than sensational: man is to doctor as woman is to doctor. arXiv preprint arXiv:1905.09866 (2019)
9. Yang, P., Fang, H., Lin, J.: Anserini: enabling the use of Lucene for information retrieval research. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1253–1256 (2017)
10. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification. In: Advances in Neural Information Processing Systems 28, pp. 649–657. Curran Associates, Inc. (2015)