Rotate *King* to get *Queen*: Word Relationships as Orthogonal Transformations in Embedding Space

Kawin Ethayarajh*
Stanford University
kawin@stanford.edu

Abstract

A notable property of word embeddings is that word relationships can exist as linear substructures in the embedding space. For example, *gender* corresponds to $\vec{woman} - \vec{man}$ and $\vec{queen} - \vec{king}$. This, in turn, allows word analogies to be solved arithmetically: $\vec{king} - \vec{man} + \vec{woman} \approx \vec{queen}$. This property is notable because it suggests that models trained on word embeddings can easily learn such relationships as geometric translations. However, there is no evidence that models *exclusively* represent relationships in this manner. We document an alternative way in which downstream models might learn these relationships: orthogonal and linear transformations. For example, given a translation vector for *gender*, we can find an orthogonal matrix R, representing a rotation and reflection, such that $R(\vec{king}) \approx \vec{queen}$ and $R(\vec{man}) \approx \vec{woman}$. Analogical reasoning using orthogonal transformations is almost as accurate as using vector arithmetic; using linear transformations is more accurate than both. Our findings suggest that these transformations can be as good a representation of word relationships as translation vectors.

1 Introduction

Word embeddings are a cornerstone of current methods in NLP. A notable property of these vectors is that word relationships can exist as linear substructures in the embedding space (Mikolov et al., 2013a). For example, *gender* can be expressed as the translation vectors $\vec{woman} - \vec{man}$ and $\vec{queen} - \vec{king}$; similarly, *past tense* can be expressed as $\vec{thought} - \vec{think}$ and $\vec{talked} - \vec{talk}$. This, in turn, allows word analogies to be solved arithmetically. For example, *'man is to woman as king is to ?'* can be solved by finding the vector closest to $\vec{king} - \vec{man} + \vec{woman}$, which should be $\vec{queen}$ if one excludes the query words.

*Work partly done at the University of Toronto.

Ethayarajh et al. (2019a) proved that when there is no reconstruction error, a word analogy that can be solved arithmetically holds exactly over a set of ordered word pairs iff the co-occurrence shifted PMI is the same for every word pair and across any two word pairs. This means that strict conditions need to be satisfied by the training corpus for a word analogy to hold exactly, and these conditions are not necessarily satisfied by every analogy that makes intuitive sense. For example, most analogies involving countries and their currency cannot be solved arithmetically using Wikipedia-trained skipgram vectors (Ethayarajh et al., 2019a).

The fact that word relationships *can* exist as linear substructures is still notable, as it suggests that models trained with embeddings can easily learn these relationships as geometric translations. For example, as we noted earlier, it is easy to learn a translation vector $\vec{b}$ such that $\vec{queen} = \vec{king} + \vec{b}$ and $\vec{woman} = \vec{man} + \vec{b}$. However, Ethayarajh et al.'s proof and corresponding empirical evidence suggest that models do not *exclusively* represent word relationships in this manner. While past work has acknowledged that downstream models can capture relationships as complex non-linear transformations (Murdoch et al., 2018), it has not studied whether there are simpler linear alternatives that can also describe word relationships.

In this paper, we first document one such alternative: orthogonal transformations. More specifically, given the mean translation vector $\vec{b}$ for a word relationship (e.g., *gender*), we can find an orthogonal matrix that represents the relationship just as well as $\vec{b}$. For example, there is an orthogonal matrix R such that $R(\vec{king}) \approx \vec{queen}$ and $R(\vec{man}) \approx \vec{woman}$. To find R for a word relationship, we first create a source matrix X of randomly sampled word vectors and a target matrix Y by shifting X by $\vec{b}$. We then use the closed-form solution to orthogonal Procrustes (Schönemann, 1966)

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 3503–3508, Hong Kong, China, November 3–7, 2019.

to determine R, which is the orthogonal matrix that most closely maps X to Y. If we broaden our search to include all linear transformations – not just orthogonal ones – we can find a matrix A to represent the relationship by using the analytical solution to ordinary least squares.

We find that using orthogonal transformations for analogical reasoning is almost as accurate as using vector arithmetic, and using linear transformations is more accurate than both. However, given that finding the orthogonal matrix analytically is much more computationally expensive, we do not recommend using our method to solve analogies in practice. Rather, our key insight is that there are parsimonious representations of word relationships between the two extremes of simple geometric translations and complex nonlinear transformations. Our empirical finding offers novel insight into how downstream NLP models, both large and small, may be inferring word relationships. It suggests that a simple linear regression model or a single attention head of a Transformer (Vaswani et al., 2017) can adequately represent many word relationships, ranging from the one between a country and its capital to the one between an adjective and its superlative form.

2 Related Work

Word Embeddings Word embeddings are distributed representations in a low-dimensional continuous space. They can capture semantic and syntactic properties of words as linear substructures, allowing relationships to be expressed as geometric translations (Mikolov et al., 2013b). Word vectors can be learned from: (a) neural networks that learn representations by predicting co-occurrences (Bengio et al., 2003; Mikolov et al., 2013b); (b) low-rank approximations of word-context matrices containing a co-occurrence statistic (Landauer and Dumais, 1997; Levy and Goldberg, 2014b).

Solving Word Analogies There are two main strategies for solving word analogies, `3CosAdd` and `3CosMul` (Levy and Goldberg, 2014a). The former is the familiar vector arithmetic method: given a word analogy task *a:b::x:?*, the answer is $\text{argmin}_w \cos(\vec{w}, \vec{x} + \vec{b} - \vec{a})$. For `3CosMul`, the answer is $\text{argmin}_w(\cos(\vec{w},\vec{x})\cos(\vec{w},\vec{b}))/(\cos(\vec{w},\vec{a}) + \varepsilon)$, where ε prevents null division. Although `3CosMul` is more accurate on average, we do not discuss it further because it does not create a distinct representation of the relationship. As noted previously, our goal is *not* to come up with a better strategy for solving analogies, but to show that there are parsimonious representations of word relationships other than translation vectors.

Orthogonal Maps Orthogonal transformations have been applied to word embeddings to achieve various objectives. Most famously, they have been used for the cross-lingual alignment of word embeddings trained on non-parallel data, for unsupervised machine translation (Conneau et al., 2018). Rothe et al. (2016) proposed a method for creating ultra-dense word embeddings in more meaningful subspaces by first learning an orthogonal transformation of the embeddings and then clipping all but the relevant dimensions. Park et al. (2017) built on this work, exploring several strategies for rotating embeddings to obtain more semantically meaningful dimensions. However, to our knowledge, orthogonal transformations themselves have not been used to represent word relationships; our work is novel in this respect.

3 Representing Word Relationships

To formalize the notion of a word relationship, we treat it as an invertible transformation that can hold over an arbitrary number of ordered pairs, following a similar framing proposed by Ethayarajh et al. (2019a) for word analogies. For example, the word pairs *{(Berlin, Germany), (Paris, France), (Ottawa, Canada)}* all express the same word relationship because some function f maps each capital city to its respective country. In this paper, we look at three specific types of transformations: *translative*, *orthogonal*, and *linear*.

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

Definition 2.1 *A translative word relationship f is a transformation of the form $\vec{x} \mapsto \vec{x} + \vec{b}$, where $\vec{b}$ is the translation vector. f holds over ordered pairs S iff* $\forall\ (x,y) \in S, \vec{x} + \vec{b} = \vec{y}$.

Definition 2.2 *An orthogonal word relationship f is a transformation of the form $\vec{x} \mapsto R\vec{x}$, where $R^T R = I$. f holds over ordered pairs S iff* $\forall\ (x,y) \in S, R\vec{x} = \vec{y}$.

Definition 2.3 *A linear word relationship f is a transformation of the form $\vec{x} \mapsto A\vec{x}$, where A is a non-degenerate square matrix. f holds over ordered pairs S iff* $\forall\ (x,y) \in S, A\vec{x} = \vec{y}$.

Analogy Category	Accuracy			Avg Cosine Similarity with Solution		
	Orthogonal	Linear	Translative	Orthogonal	Linear	Translative
capital-common-countries	**0.957**	**0.957**	**0.957**	0.802	0.844	0.847
capital-world	0.922	**0.966**	**0.966**	0.727	0.786	0.793
currency	0.300	**0.467**	0.267	0.517	0.515	0.511
city-in-state	0.529	0.897	**0.926**	0.705	0.775	0.802
family	**0.913**	**0.913**	**0.913**	0.840	0.840	0.859
gram1-adjective-to-adverb	0.438	**0.500**	**0.500**	0.667	0.670	0.678
gram2-opposite	**0.621**	0.586	0.517	0.629	0.607	0.632
gram3-comparative	0.865	0.865	**0.892**	0.791	0.768	0.812
gram4-superlative	**0.912**	0.882	**0.912**	0.747	0.705	0.764
gram5-present-participle	0.909	**0.939**	0.848	0.813	0.808	0.829
gram6-nationality-adjective	0.902	0.902	**0.927**	0.816	0.837	0.841
gram7-past-tense	0.600	**0.650**	0.625	0.768	0.770	0.775
gram8-plural	0.892	**0.919**	0.892	0.796	0.787	0.807
gram9-plural-verbs	**0.900**	0.733	0.800	0.786	0.773	0.803
Avg	0.761	**0.798**	0.782	0.743	0.749	0.768

Table 1: The accuracy on our word analogy task – detailed in section 4.1 – when word relationships are represented as orthogonal, linear, and translative functions. The highest accuracy for each category is in bold. As seen in the last row, on average, orthogonal transformations are almost as accurate as translations (0.761 vs. 0.782), and the average cosine similarity between a transformed vector and the solution is about the same for both.

Given a set of word pairs S, how can we determine a translative, orthogonal, or linear transformation f such that $\forall\ (x,y) \in S, f(\vec{x}) \approx \vec{y}$? Fortunately, there are closed form solutions for each case. To define a translation, we can simply take the mean of the pairwise difference vectors as our translation vector:

$$\vec{b} = \frac{1}{|S|} \sum_{(x,y)\in S} \vec{y} - \vec{x} \tag{1}$$

For orthogonal transformations, we first uniformly randomly sample n words from the vocabulary and stack their word vectors to get a source matrix X. Then, we add $\vec{b}$ to each sampled word vector to get a target matrix Y. Finding the orthogonal matrix that most closely maps X to Y is called the orthogonal Procrustes problem:

$$R = \underset{\Omega}{\operatorname{argmin}} \|\Omega X - Y\|_F \text{ s.t. } \Omega^T\Omega = I \tag{2}$$

Orthogonal Procrustes has a closed-form solution that we can use to find R (Schönemann, 1966). We frame the problem similarly to find a linear map A that does not necessarily need to be orthogonal. If we assume that the linear transformation should minimize the ordinary least squares objective,

$$\begin{aligned} A &= \underset{\Omega}{\operatorname{argmin}} \|\Omega X - Y\|^2 \\ &= YX^T(XX^T)^{-1} \end{aligned} \tag{3}$$

Note that our approach to finding the orthogonal and linear transformations is to find those that best approximate the geometric translation by $\vec{b}$. The reasoning behind this is simple: in practice, the number of word pairs in S is much smaller than the embedding dimensionality d, so trying to find a map $x \mapsto y\ \forall\ (x,y) \in S' \subset S$ would be very conducive to over-fitting. Since we randomly sample n words to create X and Y, we can choose $n \gg d$ to prevent this problem.

Although we are ultimately learning to approximate a translation, representing a word relationship as an orthogonal matrix as opposed to a translation vector has some useful mathematical properties, such as preserving the inner product: $\forall\ (x,y) \in S, \langle\vec{x},\vec{y}\rangle = \langle R\vec{x}, R\vec{y}\rangle$. Ethayarajh et al. (2019a) proved that when there is no reconstruction error, the word-context matrix M that is implicitly factorized by models such as skipgram and GloVe can be recovered from the inner products of word vectors. Since the inner product is preserved under rotation, M can also be recovered from an orthogonally transformed word space. The same does not hold under translation.

4 Evaluating the Representations

4.1 Task and Setup

We evaluate the different representations of word relationships using analogy tasks. However, we do not aim to solve word analogies in the traditional sense, since that largely depends on which words are present in each analogy's 4-tuple. Instead, we first calculate the mean translation vector $\vec{b}$ by av-

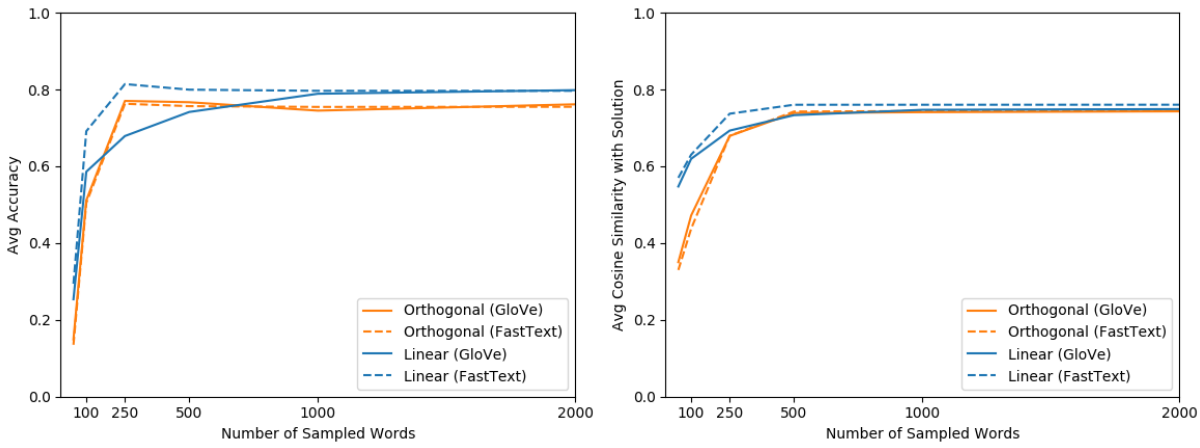

Figure 1: The accuracy on our word analogy task (left) and the average cosine similarity between the predicted and actual answers (right) as n, the number of sampled words used to learn the transformation, increases. The accuracy plateaus for $n \geq 250$ and the similarity plateaus for $n \geq 500$, suggesting that a robust transformation can be learned with relatively little data. Use of GloVe vs. FastText vectors makes no difference as $n \to 2000$.

eraging difference vectors across all word pairs, as defined in (1). $\vec{b}$ is also used to estimate matrices for the orthogonal and linear transformations (see (2) and (3)). We then create a set of word pairs for each analogy category: e.g., {*(Berlin, Germany), (Paris, France)*, ... } for *country-capital*. Each type of transformation – translative, orthogonal, and linear – is evaluated by how accurately it maps source words to target words in this set of word pairs. We use pre-trained GloVe vectors (Pennington et al., 2014) and $n = 2000$ for our main results in Table 1 and repeat our experiments with FastText vectors (Bojanowski et al., 2017) in Figure 1. We source our analogies from Mikolov et al. (2013a), as it contains a diverse set of categories.

4.2 Results

As seen in Table 1, orthogonal transformations are almost as accurate as geometric translations on our word analogy task: the average accuracy is 0.761 and 0.782 respectively. Linear transformations are more accurate than both (0.798). We would expect linear transformations to outperform orthogonal ones, given that the set of possible linear transformations is a superset of the set of possible orthogonal transformations. However, it is surprising that linear maps also outperform geometric translations, given that they are ultimately learned using translation vectors.

In the right half of Table 1, we list the average cosine similarity between the transformed source vector and the actual target vector (e.g, $\cos(R(\vec{king}), \vec{queen})$). This is to mitigate concerns that because the transformed source vector is mapped to the closest word vector, orthogonal transformations are only accurate due to the sparsity of the word space. As seen in Table 1, such concerns would be unfounded: the average cosine similarity for orthogonal transformations across all categories is 0.743, almost as high as the 0.768 for translations. This suggests that even if we considered a larger portion of the vocabulary as candidate answers, or if the word space were denser, orthogonal transformations would still be almost as accurate as translations on our task.

Our only hyperparameter is n, the number of randomly sampled words used to generate X and Y. As seen in Figure 1, the accuracy plateaus for $n \geq 250$ and the average cosine similarity with the target vector plateaus for $n \geq 500$. This suggests that it is possible to learn orthogonal and linear transformations representing word relationships with relatively little data. As $n \to 2000$, differences in performance between GloVe and FastText disappear, though linear transformations are more accurate than orthogonal ones for all n. This also highlights why the translation vector $\vec{b}$ is used to learn the transformations instead of a subset of the actual word pairs: for most analogy categories, there are fewer than 250 pairs in the dataset, and learning with so few word pairs would lead to poor accuracy.

4.3 Implications

Evaluating Embeddings The literature has often evaluated the quality of word embeddings

by testing their ability to solve word analogies arithmetically (Mikolov et al., 2013b; Pennington et al., 2014). If word relationships were *exclusively* geometric translations, this would be reasonable. However, given that word relationships can also be orthogonal or linear transformations, the usefulness of these tests as a measure of embedding quality should be reconsidered. Other arguments, both theoretical and empirical, have been made in the past against the use of analogies for evaluation (Schluter, 2018; Drozd et al., 2016; Rogers et al., 2017).

Model Architecture Given that bias terms are not necessarily needed to learn word relationships, the architecture of downstream models trained on word embeddings can be modified accordingly. Transformers (Vaswani et al., 2017) already make extensive use of linear maps in multi-headed attention and appear to be justified in doing so. Moreover, recent work has found that certain attention heads are sensitive to certain syntax, positional information, and other linguistic phenomena (Voita et al., 2019; Clark et al., 2019). For example, Clark et al. (2019) identified heads that attend to the direct objects of verbs, noun determiners, and coreference mentions with surprisingly high accuracy. However, these studies have not examined whether semantic word relationships – such as gender – also correspond to certain attention heads. Given that our findings suggest that individual attention heads have the capacity to learn such relationships, this is a promising direction for future work.

Bias in Word Embeddings The most common method for removing gender bias in word embeddings involves defining a bias subspace in the embedding space and then subtracting from each word vector its projection on this subspace (Bolukbasi et al., 2016). Under certain conditions, this method can provably debias skipgram and GloVe word embeddings (Ethayarajh et al., 2019b), but in practice, these conditions are typically not satisfied and gender associations can still be recovered from the embedding space (Gonen and Goldberg, 2019). Our findings in this paper suggest another way in which downstream models may be learning such biases, by representing gender as an orthogonal or linear transformation. This, in turn, may help explain the existence of such bias in contextualized word representations as well (Zhao et al., 2019). Given that these transformations are another way in which social biases can manifest, exploring more diverse strategies for debiasing – or alternatively, understanding the limits of debiasing strategies – is another direction for future work.

5 Conclusion

Word relationships in embedding space are generally thought of as simple geometric translations or complex non-linear transformations. However, we found that there are parsimonious representations of relationships between these two extremes, namely orthogonal and linear transformations. In addition, we found that it is possible to easily learn an orthogonal or linear transformation for a word relationship given its mean translation vector. Analogical reasoning done using linear transformations is in fact more accurate than using geometric translations. This finding offers novel insight into how downstream NLP models may be inferring word relationships. For example, it suggests that a single attention head in a Transformer has sufficient capacity to represent a semantic or syntactical word relationship, concurring with recent findings that certain attention heads have syntax- and position-specific behavior.

Acknowledgments

We thank the anonymous reviewers for their insightful comments. We thank the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial support.

References

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155.

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

Tolga Bolukbasi, Kai-Wei Chang, James Y Zou, Venkatesh Saligrama, and Adam T Kalai. 2016. Man is to computer programmer as woman is to homemaker? debiasing word embeddings. In *Advances in Neural Information Processing Systems*, pages 4349–4357.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look

at? an analysis of bert's attention. *arXiv preprint arXiv:1906.04341*.

Alexis Conneau, Guillaume Lample, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. 2018. Word translation without parallel data. In *International Conference on Learning Representations*.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king-man+ woman= queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019a. Towards understanding linear word analogies. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3253–3262, Florence, Italy. Association for Computational Linguistics.

Kawin Ethayarajh, David Duvenaud, and Graeme Hirst. 2019b. Understanding undesirable word embedding associations. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1696–1705.

Hila Gonen and Yoav Goldberg. 2019. Lipstick on a pig: Debiasing methods cover up systematic gender biases in word embeddings but do not remove them. *arXiv preprint arXiv:1903.03862*.

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

Omer Levy and Yoav Goldberg. 2014a. Linguistic regularities in sparse and explicit word representations. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning*, pages 171–180.

Omer Levy and Yoav Goldberg. 2014b. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.

Tomas Mikolov, Kai Chen, Greg S Corrado, and Jeff Dean. 2013a. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.

W James Murdoch, Peter J Liu, and Bin Yu. 2018. Beyond word importance: Contextual decomposition to extract interactions from LSTMs. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*.

Sungjoon Park, JinYeong Bak, and Alice Oh. 2017. Rotated word vector representations and their interpretability. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 401–411.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (too many) problems of analogical reasoning with word vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 135–148.

Sascha Rothe, Sebastian Ebert, and Hinrich Schütze. 2016. Ultradense word embeddings by orthogonal transformation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 767–777.

Natalie Schluter. 2018. The word analogy testing caveat. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 242–246.

Peter H Schönemann. 1966. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31(1):1–10.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Jieyu Zhao, Tianlu Wang, Mark Yatskar, Ryan Cotterell, Vicente Ordonez, and Kai-Wei Chang. 2019. Gender bias in contextualized word embeddings. 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)*, pages 629–634.