Should John Be More Likely A Physician Than Lisa: Bias-Performance Trade-Off for Gendered Pronoun Resolution

Shivank Goel, Jialu Li, Heather Zheng
{sg2359, jl3855, sz488}@cornell.edu

ABSTRACT

Pronoun resolution is an important task for natural language understanding and is useful for a range of applications, but still remains a longstanding challenge. Pronoun resolution can be considered as one of the subtasks during coreference resolution but existing corpora for the coreference resolution tasks do not capture ambiguous pronouns in sufficient volume or diversity. Furthermore, there exists a bias in the existing corpora which favor the masculine entities. To address this Webster et. al.[1] recently introduced GAP, a gender-balanced labeled corpus of 8,908 ambiguous pronoun-name pairs sampled to provide diverse coverage of challenges posed by real-world text. In this paper we identify and explore the methods to tackle the two sources of gender bias among the state-of-the-art coreference resolvers.

INTRODUCTION

The task of *Coreference Resolution* aims to identify the phrases (mentions) referring to the same entity. Although, significant advances have been made in the coreference detecting models, there has been the evidence that these models carry the risk of capturing societal stereotypes present in training data, such as the gender bias, that can significantly impact the performance of these models for some demographic groups, i.e., the female gender in our case.

NLP systems are designed to generalize from the observations and can inadvertently learn to make predictions based on the stereotyped associations. These systems can not only mimic such bias but they also bring forth a potential risk to amplify such stereotypes in the society and exacerbate the existing social problems. Such implicit human bias is quite common in the imbalanced datasets. When learning models on such datasets, it is possible that the under-represented samples in the data are neglected, but it does not influence the overall accuracy as much. Hence these models has to follow various kind of data specific augmentation approaches to balance the representation of the various classes in the dataset.

Majorly there exists two sources of gender bias in the co-reference systems. Firstly, in some cases female entities are significantly underrepresented in the training corpus. To reduce the impact of such bias in the dataset one can generate an auxiliary dataset by swapping the male and the female entities. Secondly, there is a possibility of bias in the resources used for training the NLP task. For example, many recent methods have emerged that try to tackle the problem by removing the bias from the fixed resources such as word embeddings. We discuss these two cases in detail in the following subsections.

Training Data Bias

This bias is caused when the imbalance in the training data corpus itself enforces the co-reference systems to have severe gender imbalance. For example in the *BBN Pronoun Coreference and Entity Type Corpus*[1], entities that have a mention headed by gendered pronouns (e.g.he, she) are over 83% male.

One can use the Gender Swapping technique to remove such a bias, by swapping all the male entities with the female entities and vice-versa. The models can then be trained on both the original and swapped corpora. For example, John went to his house can be accurately swapped to E1 went to her house.

Resource Bias

Word Embeddings are widely used in NLP applications. Recent work has shown that they can be severely bi-

[1]https://catalog.ldc.upenn.edu/LDC2005T33

ased, for example, Bolukbasi et. al.[2] shows that man tends to be closer to programmer than woman in the word embeddings. Caliskan et. al.[3] observed that the female names are more associated with family than career words, compared with the male names. Hence, the co-reference systems build on these word embeddings have the risk of inheriting this bias. Bolukbasi et al.[2] Also proposed a hard and a soft debiasing approach to obtain the embeddings with debiased vectors. There are other ways also in which the underlying resources used for the NLP tasks can carry such a gender bias. It can be possible that the feature rich and the rule-based approaches which rely on the corpus based gender statistics that are mined from the external resources, may carry forward such a bias with them.

In this work we note that using debiasing approaches such as swapping or debiased word embeddings often lead us to lose some contextual information contained in the original data source. Hence these approaches provide us with a slightly poor performance. On the other hand, these approaches help the models to perform better for the female examples and hence help in mitigating the gender bias. We call this observation as the *Bias Performance Trade-off*. We will discuss this in more detail in the subsequent sections.

RELATED WORK

Studies showing the gender bias

Several studies have demonstrated instances of the bias problems in the NLP setting. Bolukbasi et al.[2] and Caliskan et al.[3] show that the word embeddings can encode the sexist stereotypes.

Zhao et. al.[4] shows that the coreference resolution systems exhibit the gender bias by solving tasks that require linking gendered pronouns to either male or female stereotypical occupations. They created a dataset centered on people referred by their occupations from a vocabulary of 40 occupations. They observed the associated occupation statistics to determine what constitutes gender stereotypical roles. For example, they observed that 90% of nurses are women. None of the total $3,160$ examples in their dataset can be disambiguated by the gender of the pronoun. But gender of the pronoun can potentially distract the coreference model. Gender of the pronoun tend to link the pronouns to occupations dominated by that gender (pro-stereotyped condition) more accurately than occupations not dominated by the specific gender (anti-stereotyped condition).

Existing approaches for coreference resolution

Machine learning methods have a long history in coreference resolution. Vincent Ng[5] shows a detailed survey of how the coreference resolution approaches have exhibited a shift from the heuristic based approaches to machine learning approaches over a decade.

Rule based approach

Raghunathan et. al.[6] proposed an approach which uses hand engineered systems built on top of automatically produced parse trees. All components in this approach were build using only deterministic models. He proposed an unsupervised sieve like architecture that applies tiers of deterministic coreference models one at a time, by building on the entity clusters constructed by the previous models in the sieve.

Feature based approach

Durrett and Klein[7] built a learning-based, mention-synchronous coreference system that uses a set of features to tackle the various aspects of coreference resolution. Peng et. al.[8] proposed a predicate schemas representation which can be automatically compiled into constraints given a context.

Neural network based approach

Wiseman et al.[9] proposed a recurrent neural networks based architecture to learn latent, global representations of entity clusters directly from their mentions. They argue that these representations can be incorporated into end-to-end neural coreference systems. Clark and Manning[10] present a neural network based system that produces high-dimensional vector representations for pairs of coreference clusters. Lee et al.[11] introduces the first end-to-end neural coreference model. This model considers all spans in a document as a potential mention and learn distributions of potential antecedents for each. Their model is similar to the mention ranking, but they reason over a larger space by jointly detecting mentions and predicting coreference.

Our model is motivated by Lee et. al.[11], but in our case, since mention is already given we do not need to detect the possible mentions. Hence instead of finding

Datasets	Male			Female		
	He	His/Him	Total	She	Her	Total
GAP						
Training	373	627	1000	428	572	1000
Test	348	652	1000	396	604	1000
Development	93	134	227	87	140	227
BNN						
Training	1430	960	2390	107	93	200
Development	40	22	62	42	20	62
Test	71	49	120	60	60	120

Table 1. Datasets Summary

all the spans we simply calculate the span representations of the given mentions. We will discuss our model in more detail in the subsequent sections. Webster et. al.[1] provides baseline performance of the various neural network based approaches discussed above on the *GAP* dataset. We show that our model performs significantly better over these baselines, both in obtaining a better F1-score and reducing the gender bias.

TASK

The *GAP*[1] dataset formulates the problem of pronoun resolution as a classification task between three given choices. The problem aims to identify the target of a given pronoun within a text passage. Given the pronoun and two candidate names from the input text, the machine learning model needs to identify whether pronoun refers to the first name, the second name, or neither. Google also floated this problem as a Kaggle Competition. [2]

Specifically, our goal is to analyze that how much the two sources of bias, i.e., the *Training Data Bias* and the *Resource Bias* impacts the performance of the coreference models. For measuring the performance we use two metrics, i.e., the average *Weighted F1-Score* and the *Gender Bias*. *Weighted F1-Score* is calculated by obtaining the weighted average of the F1-Score for each class, i.e., the Male and Female classes in our case, where weights are proportional to the number of data points of that class in the dataset. We use the same metric used by Webster et. al.[1] for evaluating the *Gender Bias*. They define the *Gender Bias* to be the ratio of the F1-Scores obtained for the Female and the Male classes.

DATASET

We primarily use the *Gendered Ambiguous Pronoun* (GAP) dataset for testing our hypothesis. GAP is a gender-balanced dataset containing 8,908 coreference-labeled pairs of ambiguous pronouns and potential antecedent names. It is sampled from Wikipedia and released by Google AI Language. [3]. The pronoun-antecedent pairs in the dataset are sampled to provide diverse coverage of challenges posed by the real-world text. For comparison purposes, we also test our results on the pronoun coreference dataset obtained from the *BBN Pronoun Coreference and Entity Type Corpus*.[4] *BBN* contains the pronoun annotations extracted from the *Wall Street Journal* corpus. The complete *BBN* corpus contains the data on several pronouns and their antecedents, such as, 'they', 'it' etc. However, for the purpose of our task we filter out the pronouns which are not gender specific. This filtering leaves us with the 1149 female pronoun-antecedent pairs and 6727 male pronoun-antecedent pairs from the BBN dataset. We further filter out the *BBN* dataset to normalize the data in the same format as of the *GAP* dataset. Specifically, the *GAP* dataset offers the names of the two candidates as the potential antecedents for each pronoun. Hence, we select examples where the gendered pronoun is referring to some proper noun phrase and the example also contains another proper noun phrase apart from the correct answer which can serve as the

[2]https://www.kaggle.com/c/gendered-pronoun-resolution
[3]https://github.com/google-research-datasets/gap-coreference
[4]https://catalog.ldc.upenn.edu/LDC2005T33

other candidate. Finally the filtered dataset contains 2572 male examples and only 382 female examples. We further split this data into the training, development, and test sets. A summary statistics of the *GAP* dataset and the curated *BBN* dataset has been provided in the Table 1.

MODELS

Feature Extraction

We create the feature representations for each pronoun and its two possible mentions. These feature representations are similar to the ones introduced by Clark and Manning.[10] Specifically, for each entity, i.e., the pronoun and its two possible mentions, the model identifies and extracts the various groups of words using the parse tree of the sentence. It then encode these groups of words into a feature representation using the word embeddings. We call this feature representation as the *Embedding Features*. Besides *Embedding Features*, the model also extracts features using the relative distance of the two possible mention words within the sentence. Instead of taking the absolute values of these distances, these relative distances are then put into one of the ten categories which we call as buckets. [5] Each bucket is represented as a one-hot encoded vector. Hence, we obtain an encoded vector for each of the relative distances. Since these distances capture the positional information, we call them as the *Position Features*. In the following subsections, we describe the two kinds of features in more detail.

Embedding Features

To obtain the embedding features for a given entity word, i.e., the pronoun or one of its two possible mentions, we create the following word embeddings :

1. Word embedding of the entity word.
2. Word embedding of the parent of the entity word in the parse tree of the sentence containing the entity word.
3. Word embedding of the first word of the sentence.
4. Word embedding of the last word of the sentence.
5. Word embeddings of the two preceding words to the entity word in the paragraph.
6. Word embeddings of the two following words to the entity word in the paragraph.
7. Average embedding of the five preceding words to the entiy word in the paragraph.
8. Average embedding of the five following words to the entity word in the paragraph.
9. Average embedding of all the words in the sentence containing the entity word.

Hence, in total we include eleven word embeddings in the *Embedding Features* for a given entity word. We hope that including the embedding of the parent word in the parse tree of the sentence can also capture some of the information that is derived from the structure of the parse tree. An interesting future work would be to look at the changes in the results by including more information from the parsed tree.

Position Features

We obtain the position features using each of the two possible mention names in the example. For obtaining the position features we calculate the following distances:

1. Relative distance between the two mention words
2. Distance of the mentions from sentence beginning
3. Distance of the mentions from sentence end

Hence, in total we obtain five distances which we use to create the position features. All of the five distances are categorized into one of the buckets and encoded into a one-hot vector as mentioned previously. We call the five one-hot vectors obtained as the *Position Features*.

Part-of-speech Tag

Our best performing model also uses the POS Tags for each of the words mentioned in the *Embedding Features* section. Since our dataset only contains the mentions which are proper noun phrases, hence including the POS information can help the model to focus more on

[5] The ten buckets which we use are : [0,1,2,3,4,5-7,8-15,16-31,32-63,64+]

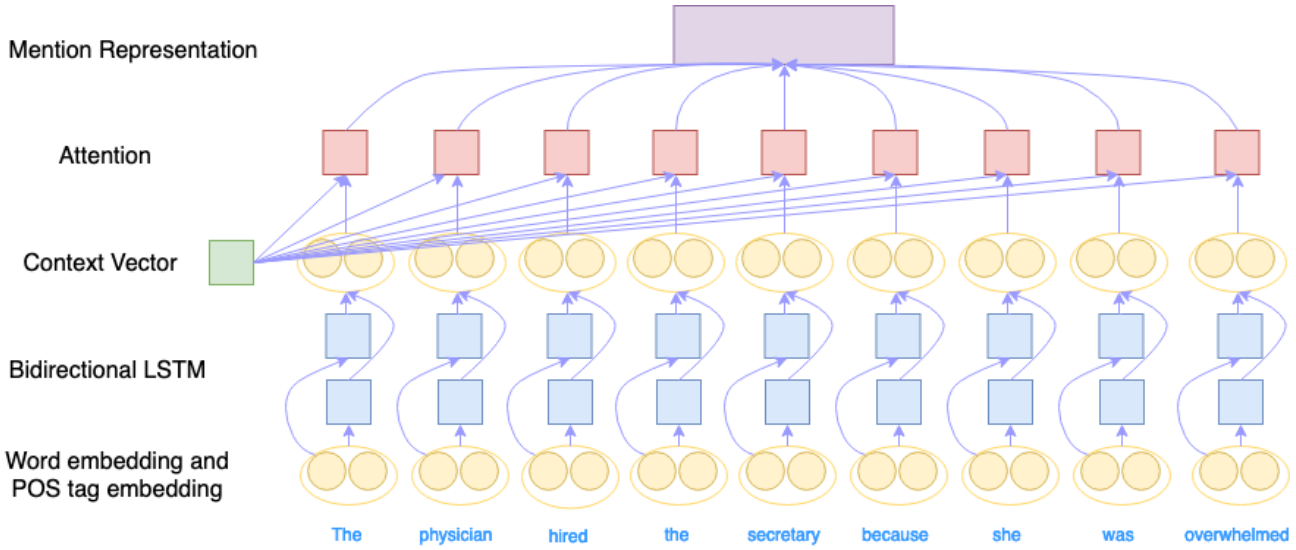

Figure 1. Computing embedding representations for the entities, i.e., the pronoun and the two possible mentions.

the potential candidates for a given pronoun. To generate these POS Tage we are using *Spacy* library which classifies each word into one of the 19 possible POS categories. In general for any pronoun resolution task, including the POS information can be helpful, since the possible POS Tags to which a pronoun can refer to can be very limited.

Baseline Models

For both the baseline models and our best performing model we use both the Embedding Features and the Position features as discussed previously.

Multi Layer Perceptron (MLP)
We first calculate the embedding features for the pronoun and its two mentions separately. We also find the position features using the two mentions. Then each of the embedding features and the position features are passed through a fully connected layer separately. The outputs of the fully connected layer are then concatenated and batch normalized to get a hidden representation. This hidden representation is then passed through two more fully connected layers. Finally, a softmax function is applied on the top of the last fully connected layer to get the probabilities for the three categories.

Multi Channel Convolutional Neural Network (CNN)
Similar to the Multi Layer Perceptron, we first calculate the embedding features for the pronoun and its two mentions separately. Then each of the embedding features are passed through two separate convolutional layers with max pooling. These two convolutional layers use two different window sizes for the convolution, and they can be seen as the two different channels. Different window sizes help us to ensemble information obtained from the different input spans. For encoding the position features, we used the same fully connected layer as in the Multi Layer Perceptron model. We then concatenated the encodings of each of the embedding features and the position features to get a hidden representation. This hidden representation is then again passed through the same fully connected layers as in MLP model, which gives us the probabilities for the three categories using a softmax function.

End to End Neural Coreference Resolution Model

The best performing model which we propose for the task of pronoun resolution is motivated by Lee et. al., 2017.[11]. In this model instead of just using the word embedding w_i of each word we also concatenate its POS tag embedding pos_i. We pass the concatenated embedding of each word into a bi-directional LSTM network to obtain the final embedding of each

word. Specifically, the hidden embedding of each word is obtained by concatenating the outputs of the forward hidden units h_{i+} and backward hidden units h_{i-}. We denote the hidden embedding of a word as h_i. The final embedding of the word is then obtained as, $u_i = tanh(W_w h_i + b_w)$, i.e., by passing the hidden embedding through a fully connected layer with *tanh* activation. Following the attention mechanism proposed by Yang et. al., 2016,[12] we finally calculate the representations m_t, for the entity u_t, i.e., either the pronoun or its two mentions by using the following equations:

$$u_i = tanh(W_w h_i + b_w) \tag{1}$$

$$\alpha_i = \frac{exp(u_i^T u_t)}{\sum_i exp(u_i^T u_t)} \tag{2}$$

$$m_t = \sum_i \alpha_i u_i \tag{3}$$

Explicitly using the output embedding of the pronoun or its two mentions, i.e., u_t as the context vector for the attention mechanism enable us to encode more information into their final representations. This approach works better than directly using the output of the bi-directional LSTM network as the entity representations. After generating these representations, we calculate the mention score for the pronoun and its two possible mentions using a feed forward network.

$$s_t = w_s FFNN(m_t) \tag{4}$$

Then similarly to the baseline models, we also obtain the representations for the position features, p_t, by passing them through a fully connected layer. We concatenate the output p_t with the representations of the pronoun s_{t0} and the two mentions s_{t1}, s_{t2} to obtain the final representation, i.e., $[s_{t0}, s_{t1}, s_{t2}, p_t]$. We then pass this final representation through two fully connected layers and apply a softmax function to get the final probabilities for the three categories.

GENDER SWAPPING & ANONYMIZATION

Following the work of Zhao et. al.,[4] to balance the number of male and female pronouns in our datasets we construct another dataset where all the male pronouns are switched to the female pronouns and vice-versa. This is especially important for the *BBN* dataset, since it is very skewed in the proportion of the male and the female pronouns. We note that this swapping significantly improves the overall performance of the model when we train the model only on the *BBN* dataset. As shown in Table 3 the overall F1-Score increases from 45.9 to 50.3 and gender bias decreases from 0.91 to 0.96, when we train on the *BBN* dataset and test on the *GAP* test set. Most of the conversions between the male to female pronouns are one-to-one, such as, 'he' can only be converted into 'she' and vice versa. However, we have one exception, i.e., 'her' can be either converted into 'his' or into 'him'. For this case we use the POS Tag of the pronoun to resolve this ambiguity. Further, to rule out the possibility of any gendered information present in the possible mention names, we anonymize the mention words. For example, "Gary P. Smaby said" can be anonymized to "E1 said" in the datasets we use.

DEBIASED EMBEDDINGS

Previous research shows that the word embeddings used for the NLP tasks inherit implicit gender bias with them from the data corpus they are trained on. Bolukbasi et. al.,[2] shows that the relative gap between a *man* and a *woman* is similar to the gap between a *programmer* and a *homemaker*, i.e., $\vec{man} - \vec{woman} \approx \vec{programmer} - \vec{homemaker}$. Bolukbasi et. al.,[2] further proposes a *hard-debiasing* method to decrease this gender stereotype in the word embeddings. Hard debiasing builds upon a pre-trained word embedding corpus, and it tries to equalize the distance of the gender neutral words from the male and the female gendered words. For our case, we are using the pre-trained *Glove*[6] (these cotain $840B$ tokens and are trained on the Common Crawl corpus) embeddings to get the hard-debiased embeddings. To obtain these hard-debiased embeddings, first they manually identified ten pairs of gendered opposite words, such as, 'he' and 'she'. Then they created a male and a female gendered set of words by comparing them with these ten pairs. Finally they used these sets to make the neutral words equidistant to all the words in each of the male set and the female set.

However, the hard-debiasing method needs a classifier to identify the male and the female sets. However this classifier is prone to error. Hence, Zhao et.

[6] https://nlp.stanford.edu/projects/glove/

Models	M	F	B	O	M	F	B	O
	GAP Test Set				BBN Test Set			
MLP Model								
GAP	65.7	64.3	**0.98**	65.0	77.3	77.4	**1.00**	77.4
BBN	54.6	51.8	0.95	53.2	98.4	97.5	0.99	97.9
Both	67.4	65.3	0.97	66.4	98.4	96.7	0.98	97.5
CNN Model								
GAP	64.3	65.8	**1.02**	65.1	73.0	67.0	0.92	70.0
BBN	54.4	50.2	0.92	52.3	98.3	96.7	0.98	97.5
Both	68.2	64.1	0.94	66.1	98.3	96.7	0.98	97.5
RNN + Attention Model								
GAP	**71.4**	**67.1**	0.94	**69.2**	86.7	78.3	0.91	83.4
BBN	48.1	43.8	0.91	45.9	**100**	**100**	**1.00**	**100**
Both	67.8	66.0	0.97	67.0	96.7	92.6	0.96	94.6

Table 2. Performance of the pronoun resolution models after training on the *BBN*, *GAP* and *Both* the datasets. In this table, rows refers to the corpus on which we train our model while column refers to the corpus which we use for testing. The column *M* shows the F1-Score for the Male class, *F* shows the F1-Score for the Female class, *B* refers to the ratio of the F1-Scores for the Female and the Male classes and *O* shows the overall weighted F1-Score. Other tables will be following similar pattern.

al.[13], proposes another debiasing method which separates the protected attributes, i.e., the gender information of the words from the other context specific information contained in the word embeddings, while training the embedding from the raw text. We call this approach as soft-debiasing. This model generates the word embeddings w such that they consist of two parts, i.e., $w = [w^{(a)}; w^{(g)}]$, where $w^{(a)}$ encodes the other contextual information apart from the gender and $w^{(g)}$ encodes the gender information of the words. Specifically, these models are trained to minimize: $J = J_G + \lambda_d J_D + \lambda_e J_E$, where J_G captures the relative proximity between the words, J_D maximizes the distance of the neutral words from the two gendered groups, and J_E tries to nullify the gender information, i.e., $w^{(g)}$ of the neutral words. For our models we did not observe any significant difference in the performance while using the hard-debiased or the soft-debiased embeddings. In this paper we will present our results using the soft-debiasing approach.

RESULTS

Training Data Bias

As mentioned previously, it is occured when the imbalance in the training data corpus itself enforces the coreference systems to have the gender imbalance. In this section, firstly we compare if the *BBN Dataset* contains more gender-bias then the *GAP Dataset*. We note that on training on the *BBN Dataset* does not provide any performance gains either in increasing the overall f-score of the model or in decreasing the bias. If we look at the performance of the model which is trained on the *BBN Dataset* and tested on the *GAP Dataset*, we note that the F1-Score decreases significantly. This can be the case because the model trained on *BBN* might not be able to generalize well on the *GAP*. However, if we assume that this depreciation in performance is same for the Male and the Female classes, then the gender bias should not be worse compared to the case when we train on the *GAP Dataset*. A higher gender-bias while using the *BBN Dataset* shows that the choice of training datasets can significantly impact the bias in the performance of the coreference resolution models. This also supports the claim of Webster et. al.[1] that trianing on the balanced datasets like *GAP* can significantly help in tackling the gender bias. These observations have been summarized in Table 2.

Secondly, we measure the performance of our models on the *Gender Swapped and Anonymized* dataset which we created as explained previously. We observe that using only the anonymized dataset provide us with a slightly poor F1-Score but improves slightly on the Gender-Bias metric. A decrease in F1-Score can be the case because using the anonymized data makes us lose some contextual information contained in the original data-set

RNN + Attention Model	M	F	B	O
	GAP Test Set			
Original Trainset				
GAP	71.4	67.1	0.94	69.2
BBN	48.1	43.8	0.91	45.9
Both	67.8	66.0	0.97	67.0
Anonymized Trainset				
GAP	64.6	66.3	1.03	65.5
BBN	44.5	47.2	1.06	45.8
Both	55.7	56.5	**1.01**	56.1
Both Original & Anonymized				
GAP	**71.8**	**69.0**	0.96	**70.4**
BBN	51.4	49.2	0.96	50.3
Both	69.0	67.2	0.97	68.1

Table 3. Effect of using *Gender Swapping and Anonymization*

contained in the gendered-names but on the other hand helps to mitigate the Gender-Bias. We call this observation as the *Bias Performance Trade-off*. We however note that by augmenting the original dataset with the anonymized dataset helps us to improve both the F-1 Score and the Gender Bias. These results have been summarized in Table 3.

Resource Bias

Word Embeddings are widely used in NLP applications but they can be severely biased. We studied the effect of using the debiased word embeddings in tacking the Gender-Bias. We obtained these embeddings as mentioned previously in the *Debiased Embeddings Section*. We again observe the same *Bias Performance Trade-off*, as before. The debiased embeddings lose some contextual information which can be helpful for the pronoun resolution models to improve their overall performance, i.e., the F-1 Score. However, the difference in the performance is not very high for the two cases. Hence, our findings prescribe to use the approach based on the need of the practical application. These observations have been summarized in Table 4.

Comparison to the baselines from GAP paper

We compare our results with the baseline provided by Webster et. al.[1]. We note that the paper provides these baselines on the *GAP* development-set which contains 454 examples of pronoun resolution task. Since, we are using the development set for fine-tuning the hyper parameters, we report our results on the *GAP* test-set which contains 1000 examples. Our models significantly perform better than these baselines and obtain both the higher F1-Scores and a lower Gender-Bias. These results have been reported in Table 5.

Error Analysis for best model

To gain insights about the cases where our model is not performing well, we do the model error analysis for our best performing model. We majorly conclude three reasons for the errors in our model.

Firstly, our model only utilizes the sentences that includes either the pronoun or its two mentions when calculating the pronoun and the mention score. It does not consider the other sentences in the paragraph that may have important background information for deciding between the two mentions. Secondly, we find that the model gets fooled by the information like the addition of "Mrs" to a name. Since "Mrs" and "she" are highly correlated with each other, our model tends to associate the pronoun "she" with the name containing "Mrs", although it is not the right mention. Finally, we also find that the model is not good at capturing the gender information about the potential mentions. For example, in the sentence *"**Duncan** made a gift of the house he was renovating to **Anne** and **her** daughter"*, it's difficult to understand whether 'her' refers to Duncan or Anne. Only when we realize that the Duncan has been referred as a male person, we can conclude that her refers to Anne. However such failures demonstrate the impact of eliminating some of the gendered

RNN + Attention Model	M	F	B	O
	GAP Test Set			
Original Trainset				
GAP	71.4	67.1	0.94	69.2
BBN	48.1	43.8	0.91	45.9
Both	67.8	66.0	0.97	67.0
Original Trainset + Soft Debiased				
GAP	69.1	67.4	0.98	68.3
BBN	49.6	45.9	0.93	47.8
Both	69.4	65.4	0.94	67.4
Original Trainset + Hard Debiased				
GAP	69.0	66.6	0.96	67.8
BBN	46.0	45.0	0.98	45.5
Both	71.7	67.3	0.94	69.4
Both Original & Anonymized				
GAP	**71.8**	69.0	0.96	**70.4**
BBN	51.4	49.2	0.96	50.3
Both	69.0	67.2	0.97	68.1
Both Original & Anonymized + Soft Debiased				
GAP	69.5	**69.4**	**0.998**	69.5
BBN	47.0	45.9	0.97	46.5
Both	68.9	68.9	**1.00003**	68.9

Table 4. Effect of using debiased word embeddings

Models	M	F	B	O
GAP Paper Baselines				
Lee et al. (2013)	55.4	45.5	0.82	50.5
Clark and Manning	58.5	51.3	0.88	55.0
Wiseman et al.	68.4	59.9	0.88	64.2
Lee et al. (2017)	67.2	62.2	0.92	64.7
Our Model				
RNN + Attention + Anonymized	**71.8**	69.0	0.96	**70.4**
RNN + Attention + Anonymized + Soft Debiased	69.5	**69.4**	**0.998**	69.5

Table 5. Comparison with the GAP Baselines

information using anonymization and debiased word embedding. Detailed examples can be seen in Table 6.

CONCLUSION

State-of-the-art coreference resolvers face the problem of capturing the societal biases while learning from the unbalanced datasets. We explore how can we tackle the two sources of gender bias. We find that balanced datasets, such as *GAP*, can significantly perform better in tackling the Gender-Bias. We observe a significant improvement in the performance in our models by augmenting the original dataset by using the *Gender Swapping & Anonymizing* technique. Finally, we observe that there exists a *Bias Performance Trade-off* while using the debiased resources and approaches, for training these models.

REFERENCES

1. K. Webster, M. Recasens, V. Axelrod, and J. Baldridge, "Mind the gap: A balanced corpus of gendered ambiguou," in *Transactions of the ACL*, p. to appear, 2018.

Error Type	Example
Loss context information	Her third album Maba followed in 2004, and she is currently working on a new album. Nana is also a dancer, and famous for her heavily choreographed videos. In 2007, it was announced that **Nana** would be a judge on Idols West Africa, alongside Nigerian **Dede Mabiaku** and American Dan Foster; at twenty-seven, **she** was one of the youngest judges in Idols history.
Fooled by word "Mrs" or "Mr"	Her first mission as senior agent comes in season 3 episode 2, when she and new agent Steve Jinks are sent to snag Typhoid Mary 'ss knife. During the events of Warehouse 2's reactivation in season 2 episode 11, Mrs. **Frederic** explains to **Claudia** that everyone at the Warehouse has a purpose, and that **she** has to be prepared.
Not capture opposite semantic information	She met Duncan some months later, in Seacouver, and he ended up helping Anne through labor, when they were trapped in a subway station after an explosion. She gave birth to a baby girl, whom she named after Duncan's mother, Mary. **Duncan** made a gift of the house he was renovating to **Anne** and **her** daughter.

Table 6. Examples from the error analysis

2. T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai, "Man is to computer programmer as woman is to homemaker? debiasing word embeddings," in *Advances in neural information processing systems*, pp. 4349–4357, 2016.

3. A. Caliskan, J. J. Bryson, and A. Narayanan, "Semantics derived automatically from language corpora contain human-like biases," *Science*, vol. 356, no. 6334, pp. 183–186, 2017.

4. J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K.-W. Chang, "Gender bias in coreference resolution: Evaluation and debiasing methods," *arXiv preprint arXiv:1804.06876*, 2018.

5. V. Ng, "Supervised noun phrase coreference research: The first fifteen years," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, pp. 1396–1411, Association for Computational Linguistics, 2010.

6. K. Raghunathan, H. Lee, S. Rangarajan, N. Chambers, M. Surdeanu, D. Jurafsky, and C. Manning, "A multi-pass sieve for coreference resolution," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pp. 492–501, Association for Computational Linguistics, 2010.

7. G. Durrett and D. Klein, "Easy victories and uphill battles in coreference resolution," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pp. 1971–1982, 2013.

8. H. Peng, D. Khashabi, and D. Roth, "Solving hard coreference problems," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 809–819, 2015.

9. S. Wiseman, A. M. Rush, and S. M. Shieber, "Learning global features for coreference resolution," *arXiv preprint arXiv:1604.03035*, 2016.

10. K. Clark and C. D. Manning, "Improving coreference resolution by learning entity-level distributed representations," *arXiv preprint arXiv:1606.01323*, 2016.

11. K. Lee, L. He, M. Lewis, and L. Zettlemoyer, "End-to-end neural coreference resolution," *arXiv preprint arXiv:1707.07045*, 2017.

12. Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1480–1489, 2016.

13. J. Zhao, Y. Zhou, Z. Li, W. Wang, and K.-W. Chang, "Learning gender-neutral word embeddings," *arXiv preprint arXiv:1809.01496*, 2018.