DeCogTeller - Towards Telling De-biased Stories

Nishtha Madaan[1], Yatin Gupta[1], Sameep Mehta[1]
[1]IBM Research-INDIA
{nishthamadaan, sameepmehta}@in.ibm.com , {yatingupta12}@gmail.com

Abstract

Written creative texts display different biases like gender, religious, societal etc. There is a strong need to remove such biases so that the *correct* version of the stories can be told. We focus on removing gender bias in this work. We present our initial system DeCogTeller to remove gender bias from stories. The system and associated cognitive algorithms can help create a gender bias free corpus which can then be used for learning generative models for creative story writing algorithms. Movies are a reflection of the society. They mirror (with creative liberties) the problems, issues, thinking & perception of the contemporary society. Therefore, we believe movies could act as a proxy to understand how prevalent gender bias and stereotypes are in any society. DeCogTeller is able to extract gender biased graphs from unstructured piece of text in stories from movies and de-bias these graphs to generate plausible unbiased stories. This information is further presented in an interactive information exploration user interface.

1 Introduction

Gender bias in text has been studied in different contexts in the literature (Holmes and Meyerhoff, 2008). Among the most important information pieces, we take text from movie plots for building our initial tool. Text de-biasing operation requires a concrete set of facts to detect bias from unbiased data. Our work is based on the observation that movies are a reflection of the society. They mirror (with creative liberties) the problems, issues, thinking & perception of the contemporary society. Therefore, we believe movies could act as the proxy to understand how prevalent gender bias and stereotypes are in any society. So we consider movie plots data set as a surrogate data for indicating bias.

As an instance, we take an excerpt from the plot of a blockbuster movie.

”Rohit is an aspiring singer who works as a salesman in a car showroom, run by Malik (Dalip Tahil). One day he meets Sonia Saxena (Ameesha Patel), daughter of Mr. Saxena (Anupam Kher), when he goes to deliver a car to her home as her birthday present.” This piece of text is taken from the *plot* of Bollywood movie *Kaho Na Pyaar Hai*. This simple two line plot showcases the issue in following fashion:

1. Male (Rohit) is portrayed with a profession & an aspiration
2. Male (Malik) is a business owner

In contrast, the female role is introduced with no profession or aspiration. The introduction, itself, is dependent upon another male character *”daughter of”*!

In this work, we present a tool DeCogTeller which takes in stories or text snippets like the one above and de-biases the stories while interchanging genders and further checking for plausibility. In the above case, if gender for Rohit and Sonia is interchanged, the plot still makes sense. We will consider how we de-bias this text snippet using our de-biasing tool DeCogTeller.

2 System Description

DeCogTeller enables the user to enter some biased text and generate unbiased version of that text snippet. For this task, we take a news articles data set and train word embedding using Google *word2vec* (Mikolov et al., 2013). This data acts as

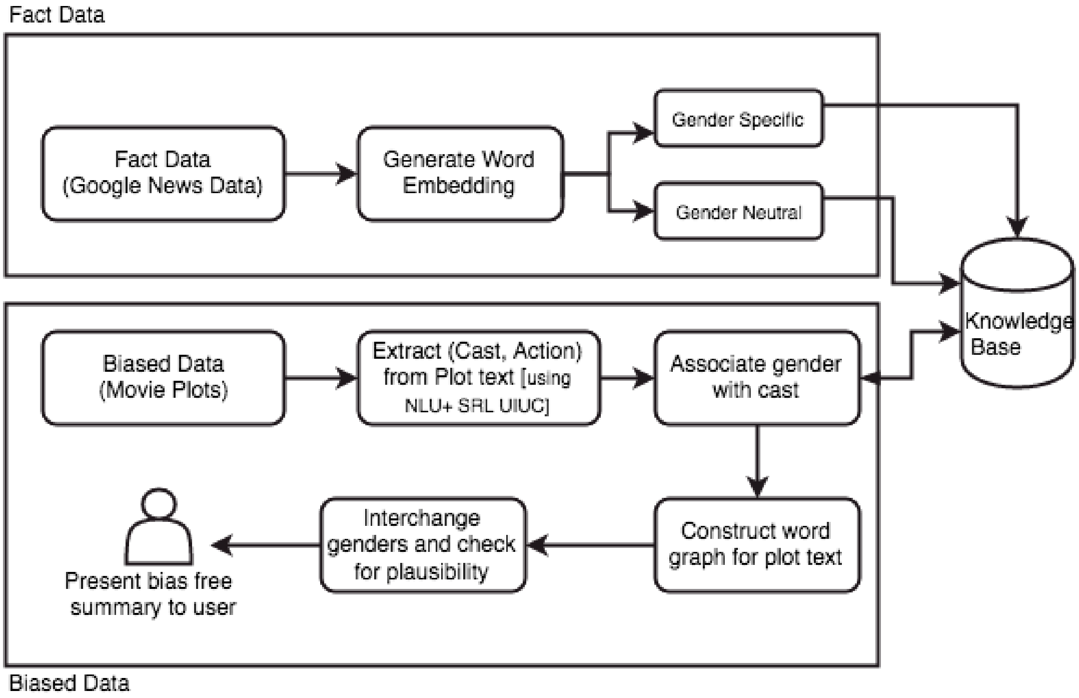

Fact Data
Fact Data (Google News Data)
Generate Word Embedding
Gender Specific
Gender Neutral
Knowledge Base
Biased Data (Movie Plots)
Extract (Cast, Action) from Plot text [using NLU+ SRL UIUC]
Associate gender with cast
Construct word graph for plot text
Interchange genders and check for plausibility
Present bias free summary to user
Biased Data

Figure 1: DeCogTeller- System Design

a *fact data* which is used later to check for gender specificity of a particular action as per the facts. Apart from interchanging the actions, we have developed a specialized module to handle occupations. Very often, gender bias shows in assigned occupation { (Male, Doctor), (Female, Nurse)} or { (Male, Boss), (Female, Assistant)}.

In Figure 1 we give a holistic view of our system which is described in a detailed manner as follows

I) **Data Pre-processing** - We first perform data pre-processing of the words in fact data and do the following operations -

(a) used Wordnet to look-up if the word present in fact data is present in Wordnet(Miller, 1995) or not. If it was not present in Wordnet, the word was simply removed.

(b) used Stanford stemmer to stem the words so that the words like *modern, modernized etc.* don't form different vectors.

II) **Generating word vectors** - After we have the pre-processed list of words from fact data, we train Google word2vec and generate word embedding from this data. We do a similar operation on biased data which in our case is movies data from Bollywood.

III) **Extraction of analogical pairs** - The next task is to find analogical pairs from fact data which are analogous to the $(man, woman)$ pair. As an instance, if we take an analogical word pair (x, y) and we associate a vector $P(x, y)$ to the pair , then the task is to find
$P(x, y) = (vec[man] - vec[woman]) - (vec[x] - vec[y])$
Here, in the above equation we replace man and woman vectors by *he* and *she*, respectively. The above equation becomes
$P(x, y) = (vec[he] - vec[she]) - (vec[x] - vec[y])$
The main intent of this operation is to capture word pairs such as doctor or nurse where in most of the data, doctor is close to he and nurse is closer to she. Therefore for $(x, y) = (doctor, nurse)$, $P(doctor, nurse)$ is given by $(vec[he] - vec[she]) - (vec[doctor] - vec[nurse])$. Another example of (x, y) found in our data is $(king, queen)$. We generate all such (x, y) pairs and store them in our knowledge base. To have refined pairs, we used a scoring mechanism to filter important pairs. If

$$\|\mathbf{P}(x, y)\| \leq \tau$$

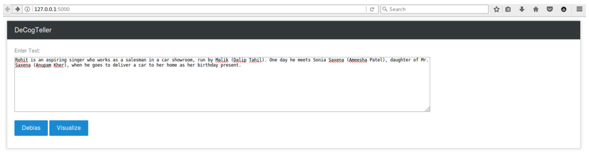
where τ is the threshold parameter, then add the word pair to knowledge base otherwise ignore. Equivalently, after normalizing $(vector[he] - vector[she])$ and $(vec[x] - vec[y])$, we calculated cosine distance as $cosine(vec[he] - vec[she], vec[x] - vec[y])$ which is algebraically equivalent to the above inequality.

IV) **Classifying word pairs** - After we identify analogical pairs, we observe that the degree of bias is still not known in each pair. So, we need to classify word pairs as specific to a gender or neutral to the gender. For example, Consider a word pair *(doctor, nurse)*, we know that whether male or female anyone can be a doctor or a nurse. Hence we call such a pair as gender neutral. On the contrary, if we consider a word pair *(king, queen)*, we know that king is associated with a male while queen is associated from a female. We call such word pairs as gender specific. Now, the task is to first find out which pairs extracted in the above step correspond to gender neutral and which ones correspond to gender specific. To do this, we first extract the words from knowledge base extracted from biased data and find how close they are to different genders. For a word w, we calculate cosine score of w with *he* as $cos(w, he)$. If w is very close to *he*, then it is specific to a man. Similarly for a word w', we do the similar operation for *she*. And if w' is very close to *she*, then it is specific to a woman. If a word w'' is almost equidistant from *he* and *she*, then it is labelled as gender neutral.

V) **Action Extraction from Biased Movie Data** - After we have gender specific and gender neutral words from the fact data, we work on the biased data to extract actions associated with movie cast. We extract gender for movie cast by crawling the corresponding Wikipedia pages for actors and actresses. After we have the corresponding gender for each cast in the movie, we perform co-referencing on the movie plot using *Stanford OpenIE* (Fader et al., 2011). Next, we collate actions corresponding to each cast using *IBM NLU API* (Machines, 2017) and *Semantic Role Labeler by UIUC* (Punyakanok et al., 2008).

VI) **Bias detection using Actions** - At this point we have the actions extracted from biased data corresponding to each gender. We can now use this data against fact data to check for bias. We will describe in the following system walk-through section how we use it on-the-fly to check for bias.

VII) **Bias Removal** - We construct a knowledge graph for each cast using relations from *Stanford dependency parser*. We use this graph to calculate the between-ness centrality for each cast and store these centrality scores in a knowledge base. We use the between-ness centrality score to interchange genders after we detect the bias.



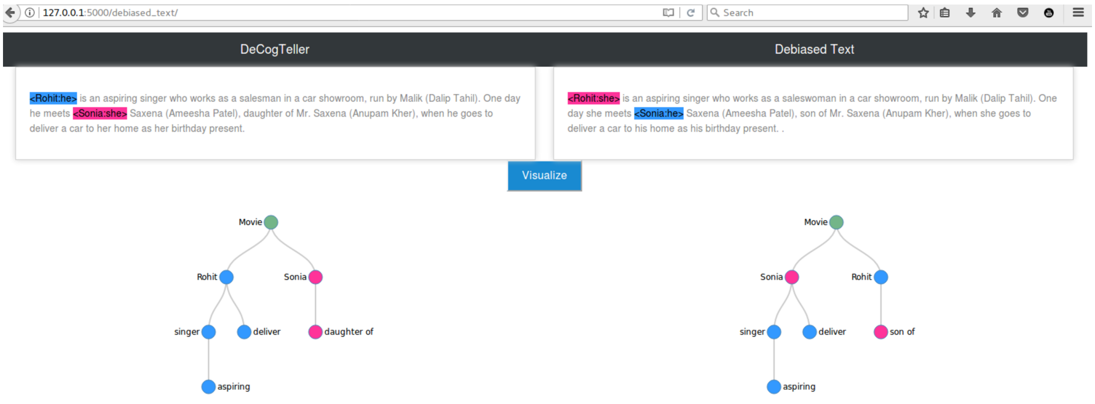
Rohit is an aspiring singer who works as a salesman in a car showroom, run by Malik (Dalip Tahil). One day he meets Sonia Saxena (Ameesha Patel), daughter of Mr. Saxena (Anupam Kher), when he goes to deliver a car to her home as her birthday present.

Figure 2: The screen where a user can enter the text

3 System Walk-through

The system takes in a text input from the user. The user starts entering a biased movie plot text for a movie, say, "Kaho na Pyar Hai" in Figure 2. This natural language text is submitted into the system in which, first, the text is co-referenced using *OpenIE*. Then, using *IBM NLU API* and *UIUC Semantic Role Labeller* actions pertaining to each cast are extracted and these are checked with gender specific and gender neutral lists. If for a corresponding cast_gender,action pair the corresponding vector is located in gender specific list then it can not be termed as a biased action. But on the other hand if a cast_gender,action pair occurring in the plot is not found in gender-specific but the opposite gender is found in gender-neutral list, then we tag the statement as a biased statement.

As an example text if the user enters - "Rohit is an aspiring singer who works as a salesman in a car showroom, run by Malik (Dalip Tahil). One day he meets Sonia Saxena (Ameesha Patel), daughter of Mr. Saxena (Anupam Kher), when he goes to deliver a car to her home as her birthday present." At the very fist step, co-referencing is done which coverts the above text to - "Rohit is an aspiring singer who works as a salesman in a car showroom, run by Malik (Dalip Tahil). One day Rohit meets Sonia Saxena (Ameesha Patel), daughter of Mr. Saxena (Anupam Kher), when Rohit goes to deliver a car to her home as her birthday present." After this step, we extract actions corresponding to each cast and then check for bias. Here corresponding to cast Rohit we have the following actions - {*singer, salesman, meets, deliver*}. The gender for Rohit is detected by using wiki page of Hritik Roshan and is labelled as "male". We find actions corresponding to cast Sonia and find the following actions- {*daughter-of*}. Then we run our gender-specific and gender neu-



<Rohit:he> is an aspiring singer who works as a salesman in a car showroom, run by Malik (Dalip Tahil). One day he meets <Sonia:she> Saxena (Ameesha Patel), daughter of Mr. Saxena (Anupam Kher), when he goes to deliver a car to her home as her birthday present.

<Rohit:she> is an aspiring singer who works as a saleswoman in a car showroom, run by Malik (Dalip Tahil). One day she meets <Sonia:he> Saxena (Ameesha Patel), son of Mr. Saxena (Anupam Kher), when she goes to deliver a car to his home as his birthday present. .

Figure 3: The screen where text is debiased and the knowledge graph can be visualized

tral checks and find that the actions are gender neutral. Hence there is a bias that exists. We do the similar thing for other cast members. Then, at the background, we extract highest centrality male and highest centrality female. And then switch their gender to generate de-biased plot. Figure 3 shows the de-biased plot. Also, there is an option given to the user to view the knowledge graphs for biased text and unbiased text to see how nodes in knowledge graph change.

4 Conclusion and Future Work

The demo of the presented tool : DeCogTeller will allow users to interact with biased text and explore how on switching genders the stories still make sense. One can play around with the tool to interchange the gender of high centrality male character with a high centrality female character and see how it leaves no change in the story but de-biases it completely. We will explain how the tool was developed using *Flask*. Following is a list of off-the-shelf tools used by our current pipeline - *Natural Language Understanding* tool from (Machines, 2017), Dependency Parser from Stanford(De Marneffe et al., 2006).

5 Ongoing Work

Our current work is in two directions.

1. In this work, we have only focused on role or occupation interchanging and **not** change any other part of the text. As an extension, we are examining if updating the story text by adding adverbs and adjectives makes gender bias removal more effective. *We would strive to add this piece in final demo.*

2. A long term goal is to add support for removing other forms of biases.

References

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*. Genoa Italy, volume 6, pages 449–454.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1535–1545.

Janet Holmes and Miriam Meyerhoff. 2008. *The handbook of language and gender*, volume 25. John Wiley & Sons.

International Business Machines. 2017. https://www.ibm.com/watson/developercloud/developer-tools.html .

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

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM* 38(11):39–41.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* 34(2):257–287.