Directional Pairwise Class Confusion Bias and Its Mitigation

Sudhashree Sayenju
Kennesaw State University
Kennesaw, GA USA
ssayenju@students.kennesaw.edu

Ramazan Aygun, PhD
Kennesaw State University
Kennesaw, GA USA
raygun@kennesaw.edu

Jonathan Boardman
Kennesaw State University
Kennesaw, GA USA
jboardma@students.kennesaw.edu

Duleep Prasanna Rathgamage Don
Kennesaw State University
Kennesaw, GA USA
drathgam@students.kennesaw.edu

Yifan Zhang, PhD
Kennesaw State University
Kennesaw, GA USA
yzhang60@kennesaw.edu

Bill Franks
Kennesaw State University
Kennesaw, GA USA
wfranks3@kennesaw.edu

Sereres Johnston, PhD
The Travelers Companies, Inc.
Hartford, CT USA
scjohnst@travelers.com

George Lee
The Travelers Companies, Inc.
Hartford, CT USA
glee2@travelers.com

Dan Sullivan
The Travelers Companies, Inc.
Hartford, CT USA
dsulliv9@travelers.com

Girish Modgil, PhD
The Travelers Companies, Inc.
Hartford, CT USA
gmodgil@travelers.com

2022 IEEE 16th International Conference on Semantic Computing (ICSC) | 978-1-6654-3418-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICSC52841.2022.00017

***Abstract*—Recent advances in Natural Language Processing have led to powerful and sophisticated models like BERT (Bidirectional Encoder Representations from Transformers) that have bias. These models are mostly trained on text corpora that deviate in important ways from the text encountered by a chatbot in a problem-specific context. While a lot of research in the past has focused on measuring and mitigating bias with respect to protected attributes (stereotyping like gender, race, ethnicity, etc.), there is lack of research in model bias with respect to classification labels. We investigate whether a classification model hugely favors one class with respect to another. We introduce a bias evaluation method called *directional pairwise class confusion bias* that highlights the chatbot intent classification model's bias on pairs of classes. Finally, we also present two strategies to mitigate this bias using example biased pairs.**

***Index Terms*—Natural Language Processing, Chatbots, Intent classification, Directional Pairwise Class Confusion Bias, Bias mitigation**

I. Introduction

Conversational chatbots are commonly used by businesses to help end users or customers with their concerns or problems to provide immediate assistance during anytime of the week. With the help of new methods in Artificial Intelligence (AI) and Natural Language Processing (NLP), chatbots aim to provide better customer service. Using chatbots, companies save time and financial resources by utilizing their human resources for more complicated tasks. Additionally, chatbots are also convenient for customers since they do not have to read through large FAQ pages or be in the waiting list until a customer support employee is available.

Chatbots first aim to find the intent behind human text utterances. After recognizing the intent, chatbots can provide appropriate information or guide the end users in the correct direction. Although chatbots have evolved over time they still have some limitations [1], [2]. Chatbots are mostly trained on a specific domain. Therefore, if a customer asks regarding a slightly different topic, it might not know how to respond like a human. Chatbots are generally incapable of recognizing grammatical errors or misspellings. In cases where customers come from different backgrounds, chatbots might not be able to understand their accents or lingo. This leads to poor conversation understanding and runs the risk of incorrect intent classification. Such biases in intent classification could cause chatbots to give replies that sound robotic, give ambiguous answers, lead customers in the wrong direction or even frustrate the customer [3].

In general machine learning models are vulnerable to bias. As a result of this, their decision could be undesirable or unfair. To understand how bias occurs in machine learning models, it is important to recognize the intimate relationship between bias in data and bias in algorithms. Since most machine learning models are data-driven, it is possible for these models to learn the bias in data during training and reflect it in predictions. Also, algorithms can change the level of bias in data or display a bias that is not present in data. The outcome of such biased models is then introduced to real-world systems such as chatbots and subsequently affects human decisions. Then it may produce even more biased data for future model training [4].

The bias in the data or a predictive model mostly refers to stereotyping which can be identified through structured or semi-structured data in the form of protected attributes [5]–[8]. Popular tools like AI Fairness 360 (AIF360) [9] and AWS Sagemaker Clarify [10] address bias related to protected attributes such as age, gender, race, ethnicity and more. These

978-1-6654-3418-8/22/$31.00 ©2022 IEEE
DOI 10.1109/ICSC52841.2022.00017

biases found in protected attributes are part of the input features. In unstructured data like text, stereotyping bias can also culminate into semantic biases. Additionally, deviations from standard text corpora like grammatical errors, spelling mistakes, accents and regional lingo might be embedded in the semantics of the text. The accumulation of all these biases is capable of influencing model behavior and classification label preferences. While there is research that solely addresses the class imbalance problem [11]–[14], not much research has been conducted in model's classification bias due to various anomalies enclosed in the text semantics. In our application, class refers to classification labels of the chatbot such as Document_Related or Payment_Related. A model could favor a certain class more than another in a task like intent classification. Therefore, defining and quantifying such class-specific bias can help find the cause of the bias and eventually find procedures to mitigate it.

The research in this paper focuses on class level bias of a NLP classification model. The results presented in this paper shows the bias of a BERT [15] model used in a chatbot for intent classification. We propose a measure called *directional pairwise class confusion bias*. The aim of this measure is to find whether the trained model makes mispredictions in the favor of one class compared to another class. The directional pairwise class confusion bias is visualized to reveal the most critical bias cases. Such biases in the model might arise due to class imbalance in the training data, or other semantic biases encapsulated through accents, misspelling or chatbot's limited domain knowledge. Additionally, this paper also proposes two strategies to mitigate bias.

This paper is organized as follows. The next section presents literature review of research that has been done in the field of bias detection and mitigation. Section III describes the directional pairwise confusion class bias and its mitigation. Section IV demonstrates the results of our experiments on chatbot data and discussion of the results. Finally, the last section concludes our paper.

II. Related Work

There are numerous cognitive biases that can be identified based on domains like social, behavioral and more. Stereotyping is type of a cognitive bias when assumptions are made or discrimination takes place on the basis of national, ethnic or gender groups [16]. On the other hand, model bias looks for whether a model has preference for certain classes or data groups.

When collecting structured data from humans, stereotyping bias is present in protected attributes such as age, gender, race, ethnicity, religion, profession, etc. The performance of a model can vary for different values of the same protected attribute. For example, if the protected attribute was gender, the performance of a model for male instances might be better than for female instances. Other than measuring bias only in protected attributes, bias in NLP has also been measured in their numerical representation: word embedding vectors. Popular word embeddings like Word2Vec [17] and Glove [18] have found to inherit gender, race and religion bias from the corpus they were trained on [19]–[22]. Apart from word embeddings, models like BERT [23]–[26] and GPT-3 [27] have been found to have stereotype bias too.

Bias can be introduced at several points in the machine learning pipeline, and Suresh et al. [28] provides a useful taxonomy of the corresponding biases. Shah et al. [29] mention four situations in the supervised NLP pipeline, specifically where bias can occur. They can be listed as label bias, selection bias, representation bias, and over-amplification. Label bias occurs in annotating training labels. Selection bias takes place in sampling observations. Representation bias occurs when a model incorrectly compares two situations. Finally, over-amplification is considered a bias that is associated with the machine learning hypothesis. Dixon et al. [30] introduce a method to measure and mitigate unintended bias in text classification models. They contrast unintended bias with fairness which is a measure of potentially negative impact on society. According to Dixon et al. [30], unintended bias is caused by the disproportional representation of demographic identity terms in training data.

For any machine learning model that makes decisions involving humans, inspecting the model's bias and fairness becomes very crucial. Detecting as well as mitigating bias is important. AI Fairness 360 (AIF360) [9] is an open source Python toolkit that provides various bias metrics and algorithms to mitigate bias in structured datasets and models. AIF360 includes over 71 bias detection metrics, 9 bias mitigation algorithms. Additionally, it also includes a unique extensible metric explanations facility to help consumers of the system understand the meaning of bias detection results. Although AIF360 is a very comprehensive tool, its bias detection and mitigation only works for structured data that contain protected attributes. Alternatively, Amazon Web Services (AWS) clients can make use of Sagemaker's Clarify. Clarify offers explainability, bias detection and bias mitigation. Clarify can schedule recurring jobs to monitor bias drifts and give explanations. The bias monitor includes 21 bias detection metrics and 4 bias mitigation algorithms. Although both AIF360 and AWS Sagemaker Clarify offer bias detection and bias mitigation techniques, their bias metrics and mitigation algorithms are designed for protected attributes included in the features dataset. However, they do not highlight class-level bias for the trained model.

In this paper, our focus will be on the class-level bias by the trained model. To the best of our knowledge, class-level bias or a class favored compared to another class has not been formally analyzed and quantified. Rather than just a statement on the presence of such bias, its quantification is important to be able to decide whether it can be mitigated or not.

III. Methods

In this section, we describe the directional pairwise class confusion bias and its mitigation. We analyze this bias based on an intent classification model for clarifying concepts.

epoch	Training Loss	Valid. Loss	Valid. Accur.	Training Time	Validation Time
1	0.67	0.51	0.85	0:42:37	0:01:39
2	0.43	0.48	0.85	0:42:38	0:01:39
3	0.35	0.50	0.85	0:42:37	0:01:39
4	0.28	0.52	0.86	0:43:26	0:01:40

Fig. 1. Results of all 4 epochs in the training phase.

A. Data

The dataset and label sets used in our experiments are provided by Travelers Indemnity Company. This dataset consists of customer (human) utterances with intent class. In total there were 128,201 user utterances, each belonging to one of 21 classes. A subset of the data was separated for modelling into training and testing. The training set had 96,150 utterances and 18 classes. The test set had 20,031 utterances with 15 labels.

B. Building Intent Classification Model using Transfer Learning with BERT model

Intent classification with more than two classes is a complex task requiring a highly developed model. In order to have high predictive power in our model and save time with the limited computing resources, transfer learning was applied on a pre-trained BERT (Bidirectional Encoder Representations from Transformers [15]) model namely, *bert-base-uncased*. The fundamental concept of transfer learning is to reuse a machine learning model originally developed for one task in a different task with limited dataset.

The pre-trained model *bert-base-uncased* was trained on BooksCorpus and English Wikipedia (excluding lists, tables and headers). As the name suggests *bert-base-uncased* was trained on lower-cased English text. The model consists of 12 transformer blocks, 768-hidden layers, 12 self attention-heads and a total of 110 million parameters. Training of *bert-base-uncased* required total of 16 TPUs. Transfer learning was done on *bert-base-uncased* for the variant that does single sentence classification task. The input for training was our chatbot data (human utterances) and their corresponding class labels were used in the softmax layer. The 4 epochs were run giving validation accuracy of 86% in epoch 4 (Fig.1).

Once training was complete, the model was evaluated on an unseen test set. The test accuracy of the model was 74.4%. Considering there are 18 classes, the test accuracy is fairly good and performs much better than a random guess ($\frac{1}{\text{number of classes}} = \frac{1}{18} = 5.5\%$). However, note that the goal of our research is not to improve the performance of the model but to investigate model's bias at class level. This model will be used to analyze bias.

C. Directional pairwise class confusion bias

The most common measure to evaluate a machine learning model is *accuracy*. However, *accuracy* cannot provide insight about the model's performance if the class distribution is unbalanced. Hence, measures such as *precision, recall, sensitivity,* and *specificity* could be used for evaluating the model at the class level. Still these measures do not provide where the mispredictions originate from at the class level. For example, the sensitivity measure may not reveal the misclassifications that happen with respect to a specific class. Hence, the model could be biased towards one class when the actual instances belong to another class.

We begin our bias analysis by plotting the confusion matrix generated by the fine-tuned BERT model on the test dataset. Fig.2 shows the confusion matrix plot with true labels on the rows and model predicted labels in the columns. The values in each cell represent the number of samples that were predicted as the column label for the correct row label. Since there are a lot of classes, looking at the confusion matrix with a naked eye might not highlight the prominent values. Fig. 3 shows a heatmap of the confusion matrix. The largest (dark blue) values are found in the diagonal. This confirms the model being mostly accurate (74.4%).

In Fig. 2 there are cells above and below the diagonal which have values greater than 0. Those cells show bias at class level and are of interest for this research. Since classes were not distributed evenly, it is difficult to observe any biases directly from the confusion matrix. If there is any bias, quantifying the bias is essential to prioritize bias mitigation.

In order to visualize the biases more clearly, the confusion matrix was modified to highlight the bias between a pair of classes. Since typically the classes are unbalanced, the confusion matrix needs to be normalized. Each cell in the confusion matrix (Fig. 2) was divided by the maximum of its column.

$$C'(i,j) = \frac{C(i,j)}{\max\limits_{k=1,\cdots n} C(k,j)} \tag{1}$$

where C represents the confusion matrix, $C(i,j)$ represents the number of classifications predicted class c_j but whose ground truth was c_i, and C' indicates the normalized confusion matrix. Doing this operation converts all the values in the matrix between 0 and 1. Normalization could be done with respect to rows (actual labels) rather than columns (predictions). Since the user of a machine learning model observes the predictions, it makes more sense to normalize with respect to the predictions. The normalized results are visualized in Fig. 4. As the original model is highly accurate (74.4%), the diagonal elements have the highest values in their respective columns. Due to this, most of the diagonal elements have the largest value 1.

The diagonal elements in Fig. 4 are accurate predictions and do not show bias. Hence, C' matrix is updated as $C'(i,i) = 0$ for every value in its diagonal. The cells that have some degree of blue color above and below the diagonal show cases where bias is present. In order to visualize these cases more clearly, the values in the diagonal were muted by setting them to be 0. The result showing the bias pairs are then visualized in Fig. 5. Equation 1 is updated as follows:

	Account_Related	Billing_Related	Cancel_Related	Claim_Related	Coverage_Related	Discount_Related	Document_Related	Escalation	EverythingElse	Payment_Related	Policy_Related	Premium_Related	Quote_Related	SmallTalk	deny
Account_Related	182	12	1	0	0	1	15	9	12	8	23	0	0	3	0
Billing_Related	1	320	14	2	0	8	10	3	2	68	5	14	1	1	0
Cancel_Related	2	14	1627	5	0	1	11	5	1	14	36	0	3	4	0
Claim_Related	0	3	1	376	9	1	22	6	6	4	3	3	8	0	0
Coverage_Related	0	2	6	25	234	2	127	3	4	3	30	4	46	0	0
Discount_Related	2	2	1	0	0	498	10	13	2	1	4	4	3	1	0
Document_Related	13	5	15	38	42	13	1964	25	32	32	111	23	42	7	0
Escalation	6	9	14	5	0	25	31	1611	50	30	27	8	6	40	0
EverythingElse	9	4	6	11	4	8	35	65	422	11	50	2	11	44	0
Payment_Related	5	51	12	2	3	7	37	32	14	2617	23	55	8	4	0
Policy_Related	21	2	45	3	30	3	116	20	32	28	2625	3	71	7	0
Premium_Related	1	15	7	3	9	7	16	2	1	57	10	423	23	0	0
Quote_Related	2	1	3	1	34	6	75	3	7	7	75	13	1083	5	0
SmallTalk	4	1	2	4	3	3	11	44	58	3	13	2	5	924	0
deny	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Fig. 2. Class Confusion matrix

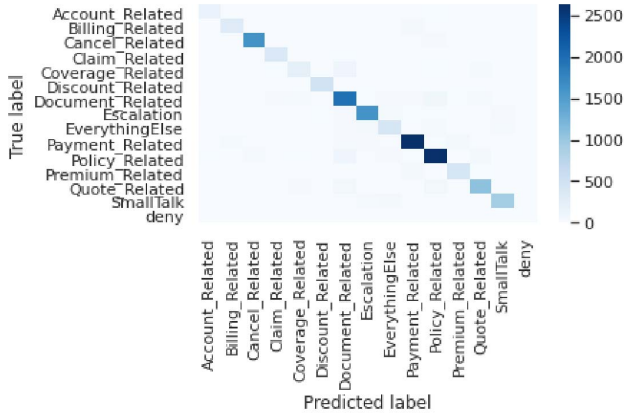

Fig. 3. Heatmap of Class Confusion matrix

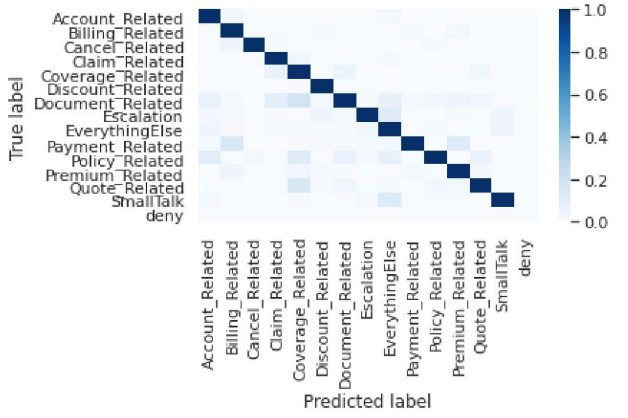

Fig. 4. Dividing each cell in the confusion matrix by the maximum of its column.

$$identity(i,j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{2}$$

$$C'(i,j) = \frac{C(i,j) * (1 - identity(i,j))}{\max\limits_{k=1,\cdots n} C(k,j)} \tag{3}$$

We coined the term *directional pairwise class confusion bias* for evaluating the bias. This indicates the likelihood of classifying an instance in one specific class into another class. Thus, there is a direction of misclassification.

Although Fig. 5 gives a clear view of directional pairwise class confusion bias, we are only interested in cases where the bias is strongly present. To reflect on the cases where bias is strong, the directional pairwise class confusion bias matrix was further pruned by setting a threshold. The threshold filter will return only those rows and columns where one of their values is above the threshold. Fig. 6 is the pruned matrix reflecting bias cases above threshold of 0.15. The plot clearly shows a strong bias for cases which are Coverage_Related but were classified by our BERT model as Document_Related. Now, we may formally define directional pairwise class confusion bias.

Definition: Directional pairwise class confusion bias. $c_i \xrightarrow{b} c_j$ represents a directional pairwise bias from class c_i to c_j for a machine learning model that indicates that there is a likelihood of a sample belonging to class c_i being classified as c_j by the trained model. This bias is quantified as $\beta(c_i \xrightarrow{b} c_j) = C'(i,j)$ and this bias is considered to be significant if $\beta(c_i \xrightarrow{b} c_j) > \theta_b$ where θ_b is a threshold for significance of bias and determined by an expert. The antecedent is called as the *source bias class* whereas the consequent is called as the *destination bias class*.

Directional pairwise class does not have the *identity property*. In other words, $\beta(c_i \xrightarrow{b} c_i) = 0$. The *symmetry* property may not always hold. Thus, if $c_i \xrightarrow{b} c_j$ is true, we cannot infer that $c_j \xrightarrow{b} c_i$. Similarly, we cannot claim the *anti-symmetry* property, if $c_i \xrightarrow{b} c_j$ is true, there is a likelihood of $c_j \xrightarrow{b} c_i$. The *transitive* property is unlikely to hold, since $c_i \xrightarrow{b} c_j$ and $c_j \xrightarrow{b} c_k$, there is no guarantee that $c_i \xrightarrow{b} c_k$ is true.

Fig. 6 shows the strongest pairwise class confusion bias between pair ($Coverage\_Related$, $Document\_Related$) represented as $c_{coverage} \xrightarrow{b} c_{document}$. Note that other pairs like $c_{billing} \xrightarrow{b} c_{payment}$, $c_{coverage} \xrightarrow{b} c_{quote}$, $c_{everythingElse} \xrightarrow{b} c_{escalation}$, $c_{everythingElse} \xrightarrow{b} c_{document}$ and $c_{quote} \xrightarrow{b} c_{document}$ also show significant bias.

D. Bias Mitigation Process

Here, we introduce two approaches for mitigating: *pairwise bias mitigation* and *boosted bias mitigator*. Here, we illustrate the mitigation process using class pair $c_s \xrightarrow{b} c_d$. Fig. 7 shows the entire mitigation process in a flow chart starting from left and ending to the right. Bias arises if for a huge proportion of cases the model predicts to be a different class. In this case,

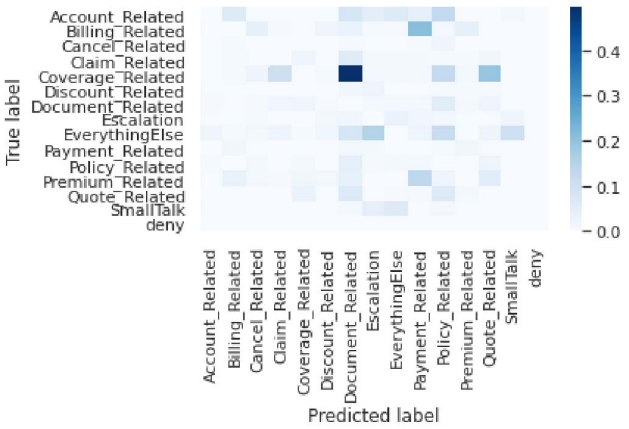

Fig. 5. Directional Pairwise Class Confusion Bias

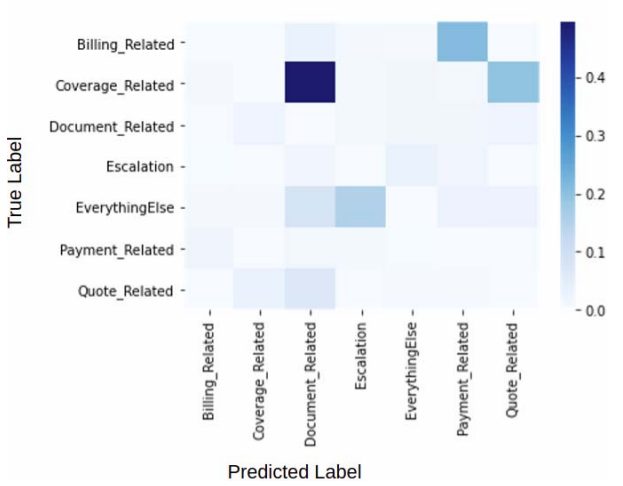

Fig. 6. Pruned Directional Pairwise Class Confusion Bias matrix after threshold was set to 0.15.

the model predicts c_d when the ground truth was c_s for a large proportion of the predicted class cases. The mitigation techniques *pairwise bias mitigation* and *boosted bias mitigator* are very similar. The only difference is that they use a different secondary model (random forest classifier) which we call the bias mitigator model in Fig. 7.

1) Pairwise Bias Mitigator: In this technique firstly, the results of the original classification model are analyzed. If the model predicts an instance as c_d, there is a significant likelihood that instance of c_s could be classified into c_d. The original classification aims to develop a global model that can separate each class from another class. However, coming up with an accurate model for a large number of classes may overlook pairwise misclassifications. If the presence of a directional pairwise class bias is detected, a binary classifier could be trained to distinguish the source bias class from the destination bias class.

Once c_d predicted instances are separated, we build a bias mitigator model. In this technique, bias mitigator model was trained on the original training set instances for which the ground truth class was either c_d or c_s. Since this classifier only distinguishes between the biased pair, we call it *pairwise bias mitigator*. Hence, our *pairwise bias mitigator* is a binary classifier. Then, the reclassified (now either c_d or c_s) instances are merged with all the other test instances that were predicted not to be c_d by the original model. In the end, the results of the original model and the results of the bias mitigated instances are compared (Section IV-B).

2) Boosted bias mitigator: Like the *pairwise bias mitigator*, the *boosted bias mitigator* also first considers the original test set results. Similarly, the test instances that were predicted to be c_d by the original model are separated. Instead of training the secondary classifier on the original training set instances, this method trains the secondary classifier on the test instances predicted as c_d but with their ground truth. Therefore, we name the secondary classifier *boosted pairwise mitigator*. We propose this technique not to repeat the same bias in the original model inherited from the training set. It should be noted that the secondary classifier is not necessarily binary but may contain as many classes as the ground truth of the original model predicted c_d instances. The results of the bias mitigated instances and remaining instances are merged and evaluated. These are compared with the original model results. Section IV-C presents results of *boosted pairwise classifier*.

IV. Experiments

In this section, we explain the results of the mitigation process. We use the BERT model that was fine-tuned for intent classification which was followed up with a bias mitigator model to mitigate the bias. Here, we firstly illustrate the mitigation process for $c_{coverage} \xrightarrow{b} c_{document}$ using *pairwise bias mitigator*. Then we provide the mitigation for both $c_{coverage} \xrightarrow{b} c_{document}$ and $c_{billing} \xrightarrow{b} c_{payment}$ using the *boosted bias mitigator*. To evaluate the bias before and after mitigation we use precision, recall and F1-score.

A. Bias in original BERT model for chatbot's intent classification

Fig. 8 presents the precision, recall, F1-score and support of the original BERT model built for intent classification. In the figure, the classes for directional bias $c_{coverage} \xrightarrow{b} c_{document}$ are highlighted. The F1-score for Coverage_Related class (0.55) is much lower than for Document_Related (0.81). It can be seen that both precision (0.79) and recall (0.83) for Document_Related are higher than the precision (0.63) and recall (0.49) of Coverage_Related. Ideally, if bias is mitigated, improvement is expected in the precision and recall values as well as the F1-score for both classes.

B. Pairwise Bias Mitigator

We firstly focus on $c_{coverage} \xrightarrow{b} c_{document}$. In this technique, if a data instance is classified as Document_Related, the data instance is reclassified using the bias mitigator model. In this case, we have used random forest classifier as a binary

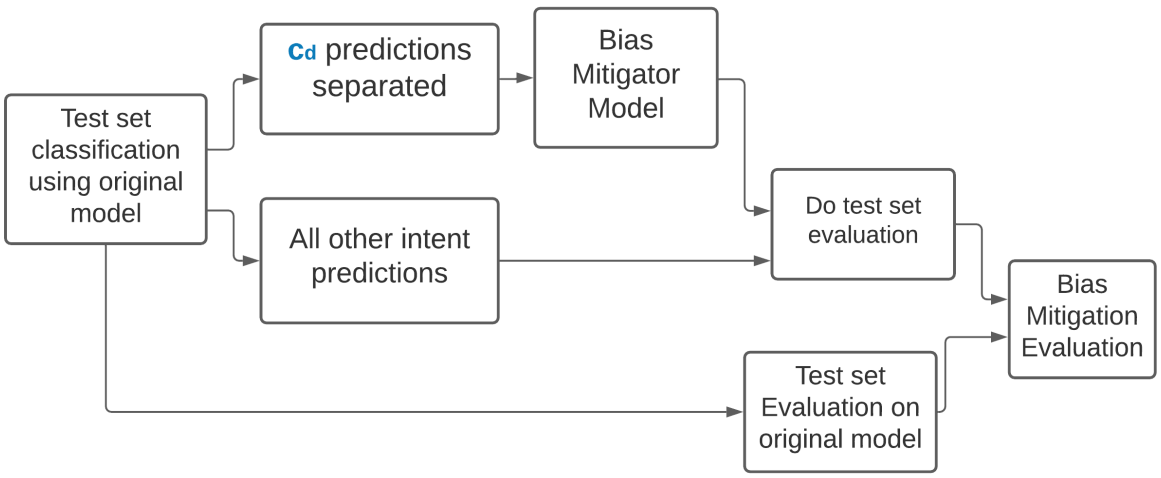

Fig. 7. Bias mitigation process

	precision	recall	f1-score	support
Account_Related	0.73	0.68	0.71	266
Billing_Related	0.72	0.71	0.71	449
Cancel_Related	0.93	0.94	0.94	1723
Claim_Related	0.78	0.85	0.81	442
Coverage_Related	0.63	0.49	0.55	475
Discount_Related	0.85	0.92	0.89	541
Document_Related	0.79	0.83	0.81	2363
Escalation	0.87	0.87	0.87	1862
EverythingElse	0.65	0.62	0.64	682
Payment_Related	0.90	0.91	0.90	2870
Policy_Related	0.85	0.87	0.86	3017
Premium_Related	0.75	0.74	0.74	574
Quote_Related	0.82	0.82	0.82	1315
SmallTalk	0.88	0.86	0.87	1077
deny	0.00	0.00	0.00	1
nan	0.00	0.00	0.00	2374
accuracy			0.74	20031
macro avg	0.66	0.65	0.65	20031
weighted avg	0.74	0.74	0.74	20031

Fig. 8. Test set evaluation on original intent classification BERT model without mitigating bias.

	precision	recall	f1-score	support
Account_Related	0.73	0.68	0.71	266
Billing_Related	0.72	0.71	0.71	449
Cancel_Related	0.93	0.94	0.94	1723
Claim_Related	0.78	0.85	0.81	442
Coverage_Related	0.59	0.53	0.56	475
Discount_Related	0.85	0.92	0.89	541
Document_Related	0.79	0.82	0.80	2363
Escalation	0.87	0.87	0.87	1862
EverythingElse	0.65	0.62	0.64	682
Payment_Related	0.90	0.91	0.90	2870
Policy_Related	0.85	0.87	0.86	3017
Premium_Related	0.75	0.74	0.74	574
Quote_Related	0.82	0.82	0.82	1315
SmallTalk	0.88	0.86	0.87	1077
deny	0.00	0.00	0.00	1
nan	0.00	0.00	0.00	2374
accuracy			0.74	20031
macro avg	0.65	0.66	0.65	20031
weighted avg	0.74	0.74	0.74	20031

Fig. 9. Results after Pairwise Bias Mitigator on pair $c_{coverage} \xrightarrow{b} c_{document}$

classifier. The training set is a subset of the original training set where only the instances whose ground truth are $c_{document}$ or $c_{coverage}$ are included. Fig. 9 shows the results of pairwise bias mitigator. Compared to the results in Fig. 8 this technique shows slight difference. For Coverage_Related, the F1-score slightly improves from 0.55 to 0.56 but this happens as a result of decrease in precision (0.63 to 0.59) and increase in recall (0.49 to 0.55). This technique has barely improved the F1-score. The performance for Document_Related class declined slightly. The precision remained the same whereas the F1-score reduced from 0.81 to 0.80. Although the results of pairwise mitigation did not show any significant improvement, there is potential for larger change if more biased pairs are mitigated using this technique.

C. Boosted Bias Mitigator

After pairwise bias mitigation, we analyze how boosted bias mitigator performs. The results of boosted bias mitigator are shown in Figures 10 and 11.

1) $c_{coverage} \xrightarrow{b} c_{document}$*:* In comparison to the results of original BERT model (Fig.8), the performance of both Coverage_Related and Document_Related classes improved (Fig. 10). While the precision of Coverage_Related remained 0.63, the recall improved from 0.49 to 0.55. As a result, the F1-score improved to from 0.55 to 0.57. For Document_Related class, the precision improved from 0.79 to 0.83. Although the

	precision	recall	f1-score	support
Account_Related	0.73	0.68	0.71	266
Billing_Related	0.72	0.71	0.71	449
Cancel_Related	0.93	0.94	0.94	1723
Claim_Related	0.77	0.87	0.82	442
Coverage_Related	0.63	0.53	0.57	475
Discount_Related	0.82	0.93	0.87	541
Document_Related	0.83	0.82	0.83	2363
Escalation	0.87	0.87	0.87	1862
EverythingElse	0.65	0.63	0.64	682
Payment_Related	0.89	0.92	0.91	2870
Policy_Related	0.85	0.88	0.86	3017
Premium_Related	0.75	0.74	0.75	574
Quote_Related	0.82	0.83	0.83	1315
SmallTalk	0.88	0.86	0.87	1077
deny	0.00	0.00	0.00	1
nan	0.00	0.00	0.00	2374
accuracy			0.75	20031
macro avg	0.66	0.66	0.66	20031
weighted avg	0.74	0.75	0.74	20031

Fig. 10. Results of Boosted Bias Mitigator on pair $c_{coverage} \xrightarrow{b} c_{document}$

	precision	recall	f1-score	support
Account_Related	0.73	0.73	0.73	266
Billing_Related	0.78	0.71	0.74	449
Cancel_Related	0.93	0.95	0.94	1723
Claim_Related	0.77	0.87	0.82	442
Coverage_Related	0.63	0.53	0.57	475
Discount_Related	0.85	0.93	0.89	541
Document_Related	0.83	0.82	0.83	2363
Escalation	0.87	0.87	0.87	1862
EverythingElse	0.65	0.64	0.64	682
Payment_Related	0.89	0.92	0.91	2870
Policy_Related	0.85	0.88	0.86	3017
Premium_Related	0.75	0.76	0.75	574
Quote_Related	0.82	0.83	0.83	1315
SmallTalk	0.88	0.86	0.87	1077
deny	0.00	0.00	0.00	1
nan	0.43	0.00	0.00	2374
accuracy			0.75	20031
macro avg	0.69	0.66	0.66	20031
weighted avg	0.79	0.75	0.75	20031

Fig. 11. Results of Boosted Bias Mitigator on pairs $c_{coverage} \xrightarrow{b} c_{document}$ and $c_{billing} \xrightarrow{b} c_{payment}$

recall for Document_Related decreased slightly by 0.01 (from 0.83 to 0.82), the F1-score still increases (from 0.81 to 0.83) as a result of increase in precision by a larger margin.

2) $c_{billing} \xrightarrow{b} c_{payment}$*:* Since boosted bias mitigator worked well for $c_{coverage} \xrightarrow{b} c_{document}$, we also used this technique on top of it for one more pair namely $c_{billing} \xrightarrow{b} c_{payment}$. In other words, the bias was mitigated sequentially, first for $c_{coverage} \xrightarrow{b} c_{document}$, then the results of it were used as from the original model when mitigating for $c_{billing} \xrightarrow{b} c_{payment}$. Fig.11 highlights the F1-score for both pairs. The performance of Billing_Related and Payment_Related classes improve compared to the original BERT model in Fig.9. The F1-score for Billing_Related increases from 0.71 to 0.74 whereas for Payment_Related it increases from 0.90 to 0.91.

D. Discussion

Since there is lack of research on class related model bias, this paper quantifies class related model bias called *directional pairwise class confusion bias*. We also presented two strategies to mitigate the bias.

Pairwise bias mitigator uses a binary random forest classifier as a secondary classifier. This method only had a slight increase on the F1-score for each class (Section IV-B). A possible reason why this method's performance was limited is that the model was trained on the original subset of instances coming from each class. Although some instances could easily be correctly classified, they were already used in training of the original model. In other words, the mitigator did not focus on the learning space where the original model fails.

Boosted bias mitigator follows a similar process except that it solely focuses on the mispredicted test results. This mitigator model tries to correct the bias by learning from the test instances of the original model that makes mistakes for the destination bias class. This technique increased F1-scores for both classes of the biased pair $c_{coverage} \xrightarrow{b} c_{document}$ (Section IV-C1). When this technique was used for two biased pairs $c_{coverage} \xrightarrow{b} c_{document}$ and $c_{billing} \xrightarrow{b} c_{payment}$, the results improved for all 4 classes (Section IV-C2). When repeated for other biased pairs, the boosted bias mitigator model can potentially mitigate directional pairwise class confusion bias significantly as well as improve the overall performance of the classifier.

The overall mitigation could be improved by choosing alternate classifiers for the bias mitigator models. Here, we have chosen a simple random forest classifier. After trying a variety of classifiers if the system's performance does not improve, it may be a good idea to investigate the labeling of data. For example, in this scenario, if $c_{coverage}$ is closely related to $c_{document}$ because of a subclass relationship, an alternate labeling or hierarchical classification may be considered to distinguish these classes.

V. Conclusion

Due to reasons like class imbalance and noise, NLP classification models might favor one class more than the other. While a lot of studies have focused on stereotyping bias of humans, little work has been done on a model's class related bias. This paper introduced directional pairwise class confusion bias to indicate a model's favoring of a class compared to another class. We quantified and visualized this bias, revealing biased pairs. Furthermore, we also presented two strategies to mitigate the bias. Both techniques make use of a secondary classifier that corrects the biased outputs. Pairwise bias mitigator showed slight improvement only. The boosted bias mitigator showed better results after bias mitigation. We anticipate more progress if this mitigation is done for all other major biased class pairs. These results show quantification of directional class confusion bias and its mitigation. Even for cases where mitigation is limited, directional class confusion bias still gives

insights about the cases that are hindering the performance of the model.

ACKNOWLEDGMENT

The work was supported primarily by the Travelers Indemnity Company. The opinions, findings and conclusions or recommendations expressed in this material only reflect those of the authors in their individual capacities.

REFERENCES

[1] M. Nuruzzaman and O. K. Hussain, "A survey on chatbot implementation in customer service industry through deep neural networks," in *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*, pp. 54–61, 2018.

[2] S. A. Thorat and V. Jadhav, "A review on implementation issues of rule-based chatbot systems," in *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020.

[3] A. Følstad, C. B. Nordheim, and C. A. Bjørkli, "What makes users trust a chatbot for customer service? an exploratory interview study," in *Internet Science* (S. S. Bodrunova, ed.), (Cham), pp. 194–208, Springer International Publishing, 2018.

[4] N. Mehrabi, F. Morstatter, N. Saxena, K. Lerman, and A. Galstyan, "A survey on bias and fairness in machine learning," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–35, 2021.

[5] A. Field, S. L. Blodgett, Z. Waseem, and Y. Tsvetkov, "A survey of race, racism, and anti-racism in NLP," *CoRR*, vol. abs/2106.11410, 2021.

[6] I. Garrido-Muñoz , A. Montejo-Ráez , F. Martínez-Santiago , and L. A. Ureña-López , "A survey on bias in deep nlp," *Applied Sciences*, vol. 11, no. 7, 2021.

[7] S. L. Blodgett, S. Barocas, H. D. III, and H. M. Wallach, "Language (technology) is power: A critical survey of "bias" in NLP," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020* (D. Jurafsky, J. Chai, N. Schluter, and J. R. Tetreault, eds.), pp. 5454–5476, Association for Computational Linguistics, 2020.

[8] A. Field, S. L. Blodgett, Z. Waseem, and Y. Tsvetkov, "A survey of race, racism, and anti-racism in NLP," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021* (C. Zong, F. Xia, W. Li, and R. Navigli, eds.), pp. 1905–1925, Association for Computational Linguistics, 2021.

[9] R. K. E. Bellamy, K. Dey, M. Hind, S. C. Hoffman, S. Houde, K. Kannan, P. Lohia, J. Martino, S. Mehta, A. Mojsilovic, S. Nagar, K. N. Ramamurthy, J. T. Richards, D. Saha, P. Sattigeri, M. Singh, K. R. Varshney, and Y. Zhang, "AI fairness 360: An extensible toolkit for detecting, understanding, and mitigating unwanted algorithmic bias," *CoRR*, vol. abs/1810.01943, 2018.

[10] "AWS (Amazon Web Services) Sagemaker Clarify." https://aws.amazon.com/sagemaker/clarify/, December 2020. Last accessed on 2021-08-30.

[11] H. He and Y. Ma, *Imbalanced learning: foundations, algorithms, and applications*. Wiley-IEEE Press, 2013.

[12] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[13] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *Journal of Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[14] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[15] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 4171–4186, Association for Computational Linguistics, 2019.

[16] M. G. Haselton, D. Nettle, and D. R. Murray, "The evolution of cognitive bias," *The handbook of evolutionary psychology*, pp. 1–20, 2015.

[17] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2013.

[18] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[19] T. Bolukbasi, K. 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 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain* (D. D. Lee, M. Sugiyama, U. von Luxburg, I. Guyon, and R. Garnett, eds.), pp. 4349–4357, 2016.

[20] J. Zhao, T. Wang, M. Yatskar, V. Ordonez, and K. Chang, "Gender bias in coreference resolution: Evaluation and debiasing methods," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)* (M. A. Walker, H. Ji, and A. Stent, eds.), pp. 15–20, Association for Computational Linguistics, 2018.

[21] J. Zhao, Y. Zhou, Z. Li, W. Wang, and K. Chang, "Learning gender-neutral word embeddings," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018* (E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, eds.), pp. 4847–4853, Association for Computational Linguistics, 2018.

[22] T. Manzini, Y. C. Lim, A. W. Black, and Y. Tsvetkov, "Black is to criminal as caucasian is to police: Detecting and removing multiclass bias in word embeddings," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)* (J. Burstein, C. Doran, and T. Solorio, eds.), pp. 615–621, Association for Computational Linguistics, 2019.

[23] M. Babaeianjelodar, S. Lorenz, J. Gordon, J. N. Matthews, and E. Freitag, "Quantifying gender bias in different corpora," in *Companion of The 2020 Web Conference 2020, Taipei, Taiwan, April 20-24, 2020* (A. E. F. Seghrouchni, G. Sukthankar, T. Liu, and M. van Steen, eds.), pp. 752–759, ACM / IW3C2, 2020.

[24] B. Hutchinson, V. Prabhakaran, E. Denton, K. Webster, Y. Zhong, and S. Denuyl, "Social biases in NLP models as barriers for persons with disabilities," *CoRR*, vol. abs/2005.00813, 2020.

[25] R. Bhardwaj, N. Majumder, and S. Poria, "Investigating gender bias in BERT," *Cogn. Comput.*, vol. 13, no. 4, pp. 1008–1018, 2021.

[26] J. Vig, "A multiscale visualization of attention in the transformer model," in *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations* (M. R. Costa-jussà and E. Alfonseca, eds.), pp. 37–42, Association for Computational Linguistics, 2019.

[27] L. Floridi and M. Chiriatti, "GPT-3: its nature, scope, limits, and consequences," *Minds Mach.*, vol. 30, no. 4, pp. 681–694, 2020.

[28] H. Suresh and J. V. Guttag, "A framework for understanding unintended consequences of machine learning," *CoRR*, vol. abs/1901.10002, 2019.

[29] D. Shah, H. A. Schwartz, and D. Hovy, "Predictive biases in natural language processing models: A conceptual framework and overview," *arXiv preprint arXiv:1912.11078*, 2019.

[30] L. Dixon, J. Li, J. Sorensen, N. Thain, and L. Vasserman, "Measuring and mitigating unintended bias in text classification," in *Proceedings of the 2018 AAAI/ACM Conference on AI, Ethics, and Society*, pp. 67–73, 2018.