CS340 Final Project: Depth to Fairness

Overview

In this project, you will open your imagination to a variety of methods to mitigate the bias from a benchmark model. Different from Assignment 2 and Assignment 3, where the used dataset is numerical, in this task, the dataset is a mixture of textual and numerical types.Besides, the models used for evaluation also extend from machine learning models to neural networks. Basically, this task concerns the model's detection of whether the sentences commented by a diverse range of ethical groups are toxic. Suppose a conversation AI builds from such toxicity comments, the models may incorrectly learn to associate the names of frequently attacked groups with toxicity. Models predicted a high likelihood of toxicity for comments containing those identities (such as "gay"), even when those comments were not actually toxic (such as "I am a gay woman"). **Under this context, you're challenged to build a model to recognize toxicity while minimizing this type of unintended bias with respect to mentions of groups.**

To this end, you will be given a dataset and a benchmark model to mitigate the aforementioned bias using a hybrid of unrestricted methods. The anatomy of this task is generalized as follows:

- **Benchmark:** Run the benchmark model, identify and quantify types of bias. **(5 points)**
- **Experiment and report:** Modify the pre-processing phase, the model structure, the training phase, etc., to obtain a bias-mitigated model and results. **(15 points)**
- **Presentation:** Compare the result of the new model with the benchmark model and explain how they are different. **(10 points)**

As before, detailed steps will be given later, and you will then present your interesting and valuable insights by writing a report. **The assignment requires a submission of the *source code* and the *report*, and takes up a total of 30 points in your final grading.**

Submission Guide

Please submit a zip file named <*StudentID-Name-FinalProject.zip*> to Blackboard, where the submitted zip includes two parts:
- The project folder (.zip), which includes all the source code and other relevant files necessary for running the project.
- A written report (.pdf), which is a complete and coherent description on how you evaluate bias and your insight on the results.

How to get help

If you encounter any problems, please do not hesitate to reach out to the TA team by either sending posts on the Blackboard discussion board or making in-person appointments via QQ. We are here to help. Our ultimate goal is for all of you to acquire knowledge through proper training instead of overwhelming you :)

Dataset

At the end of 2017, the *Civil Comments* platform shut down and released its ~2 million public comments in a lasting open archive. This project utilizes a meticulously annotated dataset **<*train.csv*>**/ **<text.*csv*>** based on these comments. It includes more than 925,000 instances with 45 features.

The text of the individual comment is found in the comment_text column. Each comment in the dataset is associated with a **<*target*>** attribute showing the toxicity of this comment, which is the **Y** variable for model training and prediction. This attribute, as well as all the other attributes, represents the fraction of human raters who believed the attribute applied to the given comment. Dataset instances with *target >= 0.5* should be considered to be toxic.

A subset of comments have been labeled with a variety of identity attributes such as **<*gender*>** (male-female-trans), **<*religion*>** (christian-jewish-muslim), **<*race*>** (black-while -asian-latino), representing the identities that are mentioned in the comment. The columns corresponding to identity attributes are listed below.

Columns	Description
id	comment id
target	toxicity
comment_text	text of comment
severe_toxicity, obscene, identity_attack, sexual_explicit, insult, threat	toxicity type
female, male, bisexual, homosexual_gay_or_lesbian, heterosexual, transgender, other_gender, other_sexual_orientation	gender
white, black, asian , latino, hindu, other_race_or_ethnicity	race
atheist, buddhist, christian, jewish, muslim, other_religion	religion
physical_disability, psychiatric_or_mental_illness, intellectual_or_learning_disability, other_disability	disability
identity_annotator_count, toxicity_annotator_count	annotation metadata
funny, wow, sad, likes, disagree	civil's attitude
created_date, publication_id, parent_id, article_id, rating	civil comment's metadata

Table 1: Description of dataset

Now follow the steps

1. Run the benchmark model and analyze the result (5 points)

The benchmark model is given in the file **<*benchmark.ipynb*>**, containing complete code for data processing, model training, and toxicity prediction. Since the object of this task is text, a specific pre-processing step is to use the built-in tokenizer of Keras to map each word in the text into a vector and then pad all word vectors in the sentence into a matrix composed of vector sequences of equal length. Then, the model is a 4-layer CNN with **Y** as the **<*target*>** attribute and **X** as the other attributes. Notice that the other attributes include meaningless ones such as <*id*>, date time <*created_date*>, text <*rating*>, and empty attributes, which should be carefully treated.

In order to facilitate the comparison of the performance of the models from all of you and the benchmark model at the same level, we also set a benchmark metric. Code can be found in the <*benchmark.ipynb*>.

$$score = w_0 AUC_{overall} + \sum_{a=1}^{A} w_a M_p(m_{s,a}) \qquad M_p(m_s) = \left(\frac{1}{N} \sum_{s=1}^{N} m_s^p \right)^{\frac{1}{p}}$$

where *A* is the number of submetrics, $m_{s,a}$ is the bias metric for identity subgroup s using submetric a, w_a is a weighting for the relative importance of each submetric; all four *w* values set to 0.25. M_p is the *p*-th power-mean function, m_s is the bias metric *m* calulated for subgroup *s*, *N* is number of identity subgroups.

The result of the benchmark model under this metric is **0.88**, which is the benchmark result, and your bias-mitigated model should be more advantageous than the benchmark model at this benchmark metric, which is requires to over **0.9**.

[TODO]
1. Train and run the benchmark model to retrieve the prediction result on comment toxicity.
2. Select some of the comments to give your decision on its toxicity, and compare them with the prediction result from the model to experience whether the results of the model align with human intuition. This line of prevalent research is called *value alignment*.
3. Using the result from prediction, evaluate and analyze the bias in this model. The bias can be speculated from different angles, such as gender, race, religion. Write up your observation. And the metrics of evaluation can be either the metrics we have learned from Assignment 2 or any other metrics you prefer.

2. Mitigate bias from a variety of methods. (15 points)

Bias mitigation techniques reduce the disparities among the groups and intersectional subgroups measured by the aforementioned metrics. Methods of mitigating model bias and their limitation is stated below:

- **Post-processing techniques:** aim to reduce fairness bias during model inference. Those approaches enforce model predictions to follow the same distribution observed during training. These techniques, however, require access to protected attributes during inference, which is not always available due to data scarcity or privacy reasons.
- **Dataset pre-processing techniques:** These include methods such as balancing the distribution of data labels, down sampling and sample re-weighing to alleviate modeling bias to a certain extent (as you work on Assignment 3). However, research shows that data pre-processing, and balancing datasets often have limited effect, compared with training inherently unbiased models. Apart from data balancing, one can also delete protected attributes from the training set or mask them. However, this is not sufficient since protected attributes are often correlated with other attributes in the data.
- **Train-time techniques:** These methods aim to combat fairness bias during model training. This can be accomplished using constraints based on adversarial loss, feature importance, decision boundary, or statistical dependence. Adversarial loss requires defining additional heads or constraints for a specific protected attribute. It maximizes the primary objective of a specific task while minimizing the model's ability to predict specific protected attributes. Constraints based on feature importance, on the other hand, heavily rely on the feature contribution score, which is not always reliable.

Some references that may help to consolidate your project:

- **[1] Fairlearn API**
(https://fairlearn.org/v0.10/user_guide/mitigation/)
This package includes metrics for assessing model bias, and algorithms for mitigating unfairness issues. It also provides an excellent and comprehensive tutorial for bias assessment and mitigation pipeline. You

are encouraged to use these ready-made packages to assist with your homework.

- [2] Ethics and AI : how to prevent bias on ML ?
(https://www.kaggle.com/code/nathanlauga/ethics-and-ai-how-to-prevent-bias-on-ml)
This is an amazing tutorial on bias identification and mitigation. This will be a great example of how you outline your report, as this tutorial is quite similar to what you need to do in this project, i.e., bias analysis - bias mitigation - compare between models. Notice that our project does not require such a workload, but we need to learn its clear workflow.

For demonstration, we provide an example of how you can mitigate bias, where codes and detailed explanations are provided in file ***<simple-lstm-pytorch-version.ipynb>***. We make the following:

1. Change the CNN model to LSTM: As CNN is usually used for image tasks, while our task is NLP. Therefore, a natural idea is to replace the current CNN model with a sequential model that is more suitable for the semantic task scenario.

2. We also provide an example of adjusting the loss function in the ***<loss_example.py>*** file. You can customize a loss function to replace the loss in your neural network. Unique loss functions designed according to the task tend to perform better.

[TODO]
1. Read the reference materials to gain a comprehensive understanding of existing bias mitigation efforts. Read the example code to see how you can do it.
2. Design your method and model to combat bias in the comment toxicity classification task. You can take any approach or a combination of them without limiting to the methods we introduced above. For example, you can take the data re-weighting method together with a loss function constraint method. However, it is ***not allowed*** to copy the sample code, but you can use the method in the example as a baseline and make adjustments. For example, modify the network structure and adjust the loss function. (**Note:** We do not admit that you only modified the hyper parameters in the example.)
3. Evaluate and analyze bias in the new model. It needs to be analyzed at the same dimension as the benchmark model. Compare the results from your analysis of the two models. Analyze the tradeoff between accuracy and fairness. Can they be achieved at the same time? Or does achieving better fairness come at the expense for model accuracy?
4. Write the report. It should clearly state your design idea, implementation details, rationale, evaluation result for your model.

Note: the scoring policy for this part are as follows:

-The experiment code and your report **(10 points)**.

-The performance of your method under the benchmark metric **(5 points)**. Your method under the benchmark metric should ≥0.90 (the benchmark is 0.88), and is accordingly divided into five levels:

Lv. 1	0.90⩽yours＜0.91	1 points
Lv. 2	0.91⩽yours＜0.92	2 points
Lv. 3	0.92⩽yours＜0.93	3 points
Lv. 4	0.93⩽yours＜0.94	4 points
Lv. 5	0.94⩽yours＜0.95	5 points

3. Presentation (10 points)

Make a slideshow showing your entire project implementation. It should contain details on the experimental plan, design, results, and the advantages and disadvantages of your method.