W266: Natural Language Processing with Deep Learning
Social Bias and Offensiveness Detection

Hao Wu & Alan Zhang

Abstract

In an increasingly divisive world, many argue that social media platforms have a social responsibility to filter toxic contents and proactively identify and remove these content to promote a more harmonious platform. This practice has significant social and business implications. The social media platform that can successfully implement these editorial processes could improve brand image, as well as user engagement. The existing platforms have certainly made progress to identify more obvious hate speech, but identification of more subtle bias in posts remains a challenge. In this project, we explored the inclusion of users' demographic features in CNN model structure to test the effectiveness of users' demographics on their interpretation of offensiveness in an online comment. The significant improvement in evaluation metrics indicates that it is an effective feature to include.

1 Introduction

The objective of our project is to classify whether a sentence is offensive by combining the content of sentences and the demographic features of viewers. Essentially, we want to improve the model performance by including additional categorical variables about writers and viewers of the sentences. We believe we could achieve solid model performance if we focus solely on the sentences' content. However, there could be cases when the bias is not apparent or when the bias is only evident to specific groups of people. Therefore, we want to focus on this project to assess whether we could take it one step further to detect the more subtle bias in sentences using additional categorical variables about the writers and the viewers.

We hypothesize that basic demographic features, specifically annotator/viewer's race, gender and age for our research, could help improve the model performance. First, we built a baseline model with only the content of sentences from a smaller dataset to test our hypothesis. Next, we applied basic data processing in this baseline model, used Glove embeddings, and leveraged a base CNN model structure. Then, we compared the validation accuracy of the baseline model with the validation accuracy of the other iterations, which included the demographic features of viewers and some other variations of the data. We would like to see an improved evaluation metric in the form of validation accuracy after we include other pre-processing techniques and demographic features.

2 Methodology Overview

2.1 Dataset

We explored the Social Bias Frames dataset from Hugging Face (https://huggingface.co/datasets/social_bias_frames), which contains over 150k structured annotations of social media posts, spanning over 34k implications about a thousand demographic groups. Each instance contains a post that may contain an offensive statement and annotated information concerning the nature of the offensive implication as well as the demographics of the annotator and origin of the post. The source data for these posts came from various social media platforms,

such as Reddit and Twitter, and each post is annotated by multiple annotators to label the data. Ideally, we wanted to test demographic features of both parties, but only the data about the viewers, or the annotators, are available in our dataset.

We used the *offensiveYN* variable as the data label, which has three labels: “yes”, “maybe”, and “no”. One nuance about the labeling is that multiple annotators will label the same post, so there could be multiple labels for one post. The multiple labels might also be conflicting each other, creating another complication for data processing. We generated data for the baseline model by aggregating the labels for the same post using a voting scheme utilizing mode. After further investigation, we also identified a subset of posts with multiple modes and reclassified these posts to “maybe”. We will discuss the result of this iteration in a later section.

Because the voting scheme requires a certain level of data aggregation by features, the dataset used for iterations of training and exploration will vary. We will discuss how we isolated the effect of implementing demographic features from the increased data size in below sections.

We used the *post* variable as the main data input into our models. In the posts, there is a combination of texts and emojis, which adds another layer of complexity. In the baseline model, we removed all emojis and focused solely on the texts. In other iterations, we used pre-existing packages such as emot to convert emojis into texts and tested the effect of this specific data pre-processing. Another pre-processing variation we tried is to remove the stopwords in the posts.

2.2 Approach Comparison

The unique data structure indicates volatility in the labels for the same post when it is annotated by multiple annotators. Also, we wanted to extract and leverage information about the demographic feature of the annotators in our classification problem. The intuition behind this approach is that subtle and less obvious bias would be more sensitive to certain groups of annotators, thus adding the demographic features of the annotators could improve model performance by identifying those biases.

Many researchers have looked into the classification problems for social bias, or toxicity in languages. The most common approach is to collect a large number of toxic social media posts and hire annotators to label these social media posts as “ground truth”. The research has primarily focused on the text data of the posts, but not that many have considered the biases of the “ground truth” itself. The Social Bias Framework dataset differentiates itself because the group of annotators is larger and more diverse, which gives us an unique perspective to discover how understanding of the same content can be drastically different based on the audience’s cultural/demographic background. Based on these we would like to see the demographic feature could affect a prediction model performance.

3 Model Methodology

3.1 Baseline Model

For our baseline model, we applied basic data processing, used GloVe embeddings, and leveraged a base CNN model structure. On the data end, we used an aggregated dataset with about 35k posts. As we explained in the **2.1 Dataset** section, each post could have multiple labels, and we used the mode of the labels as the model labels. In the posts, we removed all emojis in the baseline model. For embeddings, we applied the GloVe embeddings in the baseline model. The benefit of using GloVe is that it

incorporates both the local and global statistics to reflect word co-occurrence to obtain word vectors. Last but not least, we leveraged the CNN model structure in the baseline model as illustrated in the diagram below. We chose the CNN model for the following reasons: (1) Most of the posts are comparatively short sentences which is suitable for CNN learning to extract text features. (2) CNN models have relatively simple overhead compared to other types of models which allow us to implement various input data representation and capture the training results in short turnaround time.(3) CNN is less "data hungry" compared with methods like transformers and our aggregated dataset is still limited in size.

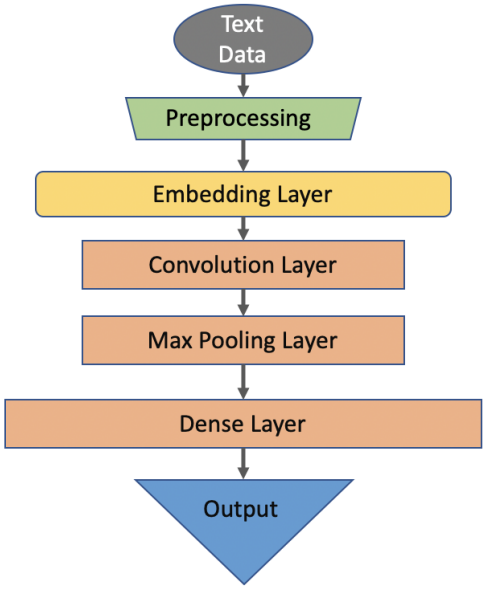

Diagram 1. Baseline CNN Model Structure

3.2 Model with Demographic Features

The final iteration we tested is to include the demographic feature of viewers of potentially offensive posts, represented by the demographic feature of annotators. We implemented the both categorical and numerical variables about the annotators in the CNN model structure using two different methods as explained below.

3.2.1 Concatenation: In this model structure, we added the race and age of the annotators into the CNN model. The categorical variable (race) is added by concatenation with text data of social media posts, and the numerical variable (age) is added as a separate concatenation layer, combined with the flatten layer with information about text and categorical data. Information from all three data types will then enter the last dense layer to generate our model output.

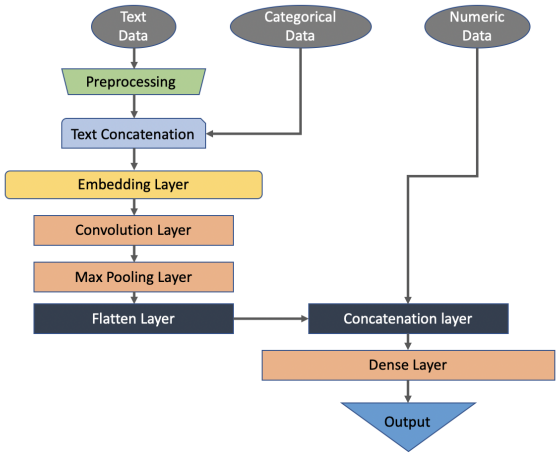

Diagram 2. CNN Model Structure with Demographic Feature as Concatenation

3.2.2 Separate Embedding Layer: The second model structure we tested is to include a separate categorical embedding layer. The additional embedding layer will feed into the categorical flatten layer. Then, the text flatten layer, categorical flatten layer and numerical data will be added to the concatenation layer, as illustrated in the model structure diagram below.

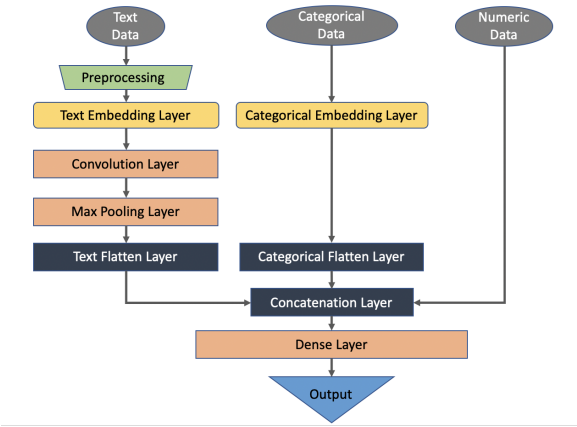

Diagram 3. CNN Model Structure with Demographic Feature as Separate Embedding Layer

3.2.3 Comparison with baseline: As we introduced new demographic features into the model and consolidated model labels with the aggregated data, both the available data size and the number of input features available increased. To isolate the impact of adding new input features from the increased data size, we randomly sampled a sub-dataset equivalent to the baseline dataset. We retrained the model to measure performance improvement.

3.3 Extended Tuning and Training

In addition to the iterations of data pre-processing and categorical features, we also iterated our CNN model's extended tuning and training. We tested different numbers of batch sizes, epochs, and tokenization top words, as well as modifications to CNN model structures. We could not achieve better results than our previous CNN model with demographic feature structure 2. In our first round of research, we used epoch size of 5, batch size of 50, and GloVe tokenization top words of 5000. With an epoch size of 5, the validation accuracy was still increasing at epoch 5, which indicates the potential for further performance improvement if we run additional epochs. We later extended the epoch size to 10 to further explore this pattern. However, the increasing pattern quickly stopped after epoch five, and we didn't achieve better results. In the testing for batch size, we noticed that the learning effect is not as strong when we increase the batch size.

3.4 Additional Model Iterations

As explained below, we implemented additional diagnostics and testings around different data pre-processing methods. However, these iterations didn't improve our model performance, so they are not included in our final model.

3.4.1 Stop Word Handling: We implemented the first variation to remove the stop words in the original posts. We decided to remove the stop words because they do not provide information in our classification problem. The hypothesis is that model performance would improve after removing stop words because fewer and only significant tokens are left.

3.4.2 Emoji Handling: In social media posts, emojis are often included in the content. The emojis in posts serve different purposes. In some cases, they are used as expressions of emotions aligned with the intended meaning. In other cases, they could have implicit meanings different from the original meaning. Regardless of the user cases, the emojis carry information that could be additive to our classification objective.

In the baseline model, we deleted the emojis in the posts to simplify the problem. In this model iteration, we explored keeping the tailed emojis and the conversion of emojis to texts using a python package called emot.

3.4.3 Data Label Normalization: As discussed in previous sections, there are different labels for the same post because

multiple annotators will label the posts. We normalized the labels using mode in the baseline model. In further exploration, we identified a nuance in the normalization method that about 9% of the posts have multiple modes for labels. The default setting to calculate mode in the NumPy package is to use the first mode with multiple modes. Under this setting, the posts with multiple modes would be labeled as "no." We overwrote these labels to "maybe" in this iteration. The reasoning behind this is that we believe the label "maybe" better represents these posts because annotators cannot reach a consensus when they evaluate these posts independently.

4 Results

4.1 Accuracy for All Models

Model Iterations	Section Num	Best Train Acc	Best Val Acc
Baseline + Voting Scheme 1	**3.1**	0.8753	0.7349
Text + Demographic Feature			
Concatenation	**3.2.1**	0.9129	0.8066
Separate Embedding Layer	**3.2.2**	0.8986	0.8123
Separate Embedding Layer: Same data size as Baseline	**3.2.3**	0.9241	0.7887
Model Diagnostics			
Baseline + Voting Scheme 2	**3.4.3**	0.8574	0.7013
Baseline + Tail Emoji	**3.4.2**	0.8912	0.7270
Baseline + Emoji transformation to words	**3.4.2**	0.8761	0.7315
Regularization	**3.3**	0.8839	0.8073
Max Token Size	**3.3**	0.8888	0.8120
Stopword Removal	**3.4.1**	0.8828	0.8053

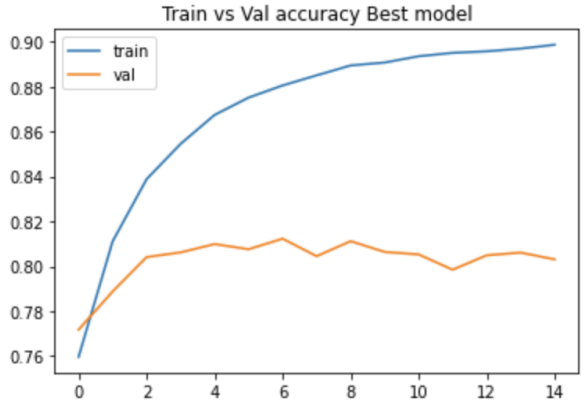

Chart 1: Text + Demographic Feature with Separate Embedding Layer (Section 3.2.2)

4.2 Discussion of Results

The baseline model showed strong validation accuracy at 0.7349 using only the text features of social media posts. However, most model iterations fail to achieve a better validation accuracy than the baseline model. The best performing model iteration is the model with the text feature, categorical demographic feature as a separate embedding layer, and numerical demographic feature. In this model, we achieved a validation accuracy of 0.8123, a nearly 8% increase in model performance.

Similarly, when comparing the baseline performance against that of the model with demographic features on comparable sized data, we see a model performance improvement of nearly 6% (0.7349 vs. 0.7887).

Both model improvements show an indication that annotators' background may affect their view of whether a post is offensive or not. In our context, this is a

measure of model performance improvement, but translating into a real-world scenario shows the degree of bias in viewing these online posts.

4.3 Potential Improvements

There are several potential improvements we can implement that might increase the accuracy of our model.

(1) The first and potentially the most promising improvement is to include more demographic features. Due to our dataset's limitations, we only have demographic data on the viewers. We could further improve the model if we could also include the demographic features of the writers of the social media posts.

(2) Another potential improvement we could make to expand on this analysis is to collect more data. Based on the validation performance curve we could see the model struggles to learn after reaching about 80% accuracy, and based on the diagnostics we have tried, a likely limitation that handicapped further learning of the model is the size of the data. We could collect more information to create a larger dataset and even train a transformer based on this larger dataset.

(3) Emoji representation can still be a part of the model improvement. Intuitively emojis are a big part of the social media dynamic and conveys a wide range of information, we can research more on how to scientifically embed emojis into our model to improve learning.

(4) Different model structures could be explored. Our focus on this research has been more towards how to implement existing features and preprocessing the text information, but it is worthwhile to construct other models such as LSTM or transformers or different CNN model structures. For example, we could implement deeper layers or better regularization in current layers.

5 Conclusion

Demographic features significantly impacted our prediction model, and we achieved around an 8% increase in the validation accuracy. However, the other iterations in data pre-processing, tuning, and training did not improve model performance. One fundamental limitation of our project is that we have limited demographic features of the annotators and limited aggregated data size in the NLP context. Given the performance improvement of our model with demographic features, it is intriguing to implement more demographic features in granular levels to assess whether the model performance can be further improved.

As we tried in our research to quantify the biases of people’s interpretation of an online post with a model performance increase, it is worth reflecting on how the labeling of these situations could have been executed in real life. For example, when we flag an online post or video as offensive, do we mark them as offensive when a single inspector feels uncomfortable reading/watching it? Or do we take a voting scheme with scientific reasoning? How do we select an inspector population to monitor these situations? Can we count on machines to do it?

These questions remain unanswered and await more researchers and policymakers to dig in.

References

1. Ganegedara, T. (2021, November 15). *Light on math ML: Intuitive Guide to Understanding Glove embeddings*. Medium. Retrieved December 4, 2021, from https://towardsdatascience.com/light-on-math-ml-intuitive-guide-to-understanding-glove-embeddings-b13b4f19c010.
2. Breitfeller, L., Ahn, E., Jurgens, D., & Tsvetkov, Y. (2019, November). *Finding microaggressions in the wild: A case for locating elusive phenomena in social media posts*. ACL Anthology. Retrieved December 4, 2021, from https://aclanthology.org/D19-1176/.
3. Sap, M., Gabriel, S., Qin, L., Jurafsky, D., Smith, N. A., & Choi, Y. (2020, April 23). Social bias frames: Reasoning about social and power implications of language. arXiv.org. Retrieved December 4, 2021, from https://arxiv.org/abs/1911.03891.
4. Maarten Sap, Swabha Swayamdipta, Laura Vianna, Xuhui Zhou, Yejin Choi & Noah A. Smith (2021). *Annotators with Attitudes: How Annotator Beliefs And Identities Bias Toxic Language Detection.* arXiv