CS60075: Natural Language Processing

Term Project

Explainable Detection of Online Sexism (EDOS)

Group 1

Submitted by:

19EC10035 : Kistamgari Sri Harika Reddy

19EC10086 : Shruti Shreyasi

19EC39007 : Bbiswabasu Roy

19EC39041 : Anik Mandal

19EC39044 : Ujwal Nitin Nayak

Problem Statement:

The task contains three hierarchical subtasks:

- Binary Sexism Detection: a two-class (or binary) classification where systems have to predict whether a post is sexist or not sexist
- Category of Sexism: for posts which are sexist, a four-class classification where systems have to predict one of four categories:
    - threats
    - derogation
    - animosity
    - prejudiced discussion
- Fine-grained Vector of Sexism: for posts which are sexist, an 11-class classification where systems have to predict one of 11 fine-grained vectors

Dataset:

Link to the dataset: https://www.kaggle.com/datasets/himarusti/project6-grp1-dataset

Description of data:

Below is the description of the hierarchical classes of the sentences in the dataset:

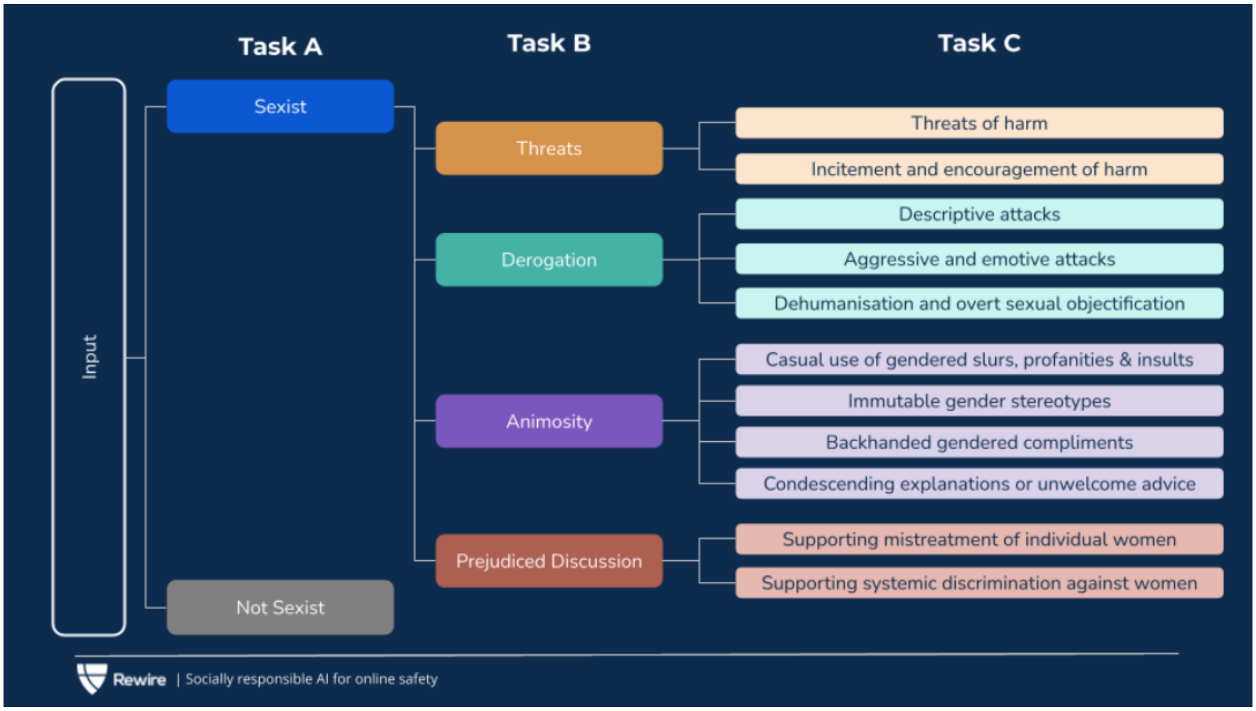

Task A
Task B
Task C
Input
Sexist
Not Sexist
Threats
Derogation
Animosity
Prejudiced Discussion
Threats of harm
Incitement and encouragement of harm
Descriptive attacks
Aggressive and emotive attacks
Dehumanisation and overt sexual objectification
Casual use of gendered slurs, profanities & insults
Immutable gender stereotypes
Backhanded gendered compliments
Condescending explanations or unwelcome advice
Supporting mistreatment of individual women
Supporting systemic discrimination against women
Rewire | Socially responsible AI for online safety

FIG: Classes for the sentences

Preprocessing:

After analyzing the **text** column of the training dataset, following preprocessing were applied:

- Since the dataset was found to contain emojis, we removed all the non-ASCII characters from the dataset
- Next, all square brackets were removed
- It was found that when non-alphanumeric characters were removed, the results improved. Hence, we removed them and converted all the remaining alphabets to lowercase
- Next, for each column we converted all the string classifiers such as “sexist”, “not sexist” to numerical values using label encoders.
- A new dataset was built from the given dataset by considering only those rows for which it was labeled as “sexist” so that a model can be trained to classify the sentences for the last level of the hierarchy
- All sentences were tokenized using RobertaTokenizer

Model:

We experimented with various model architectures and the following was found to yield best results:

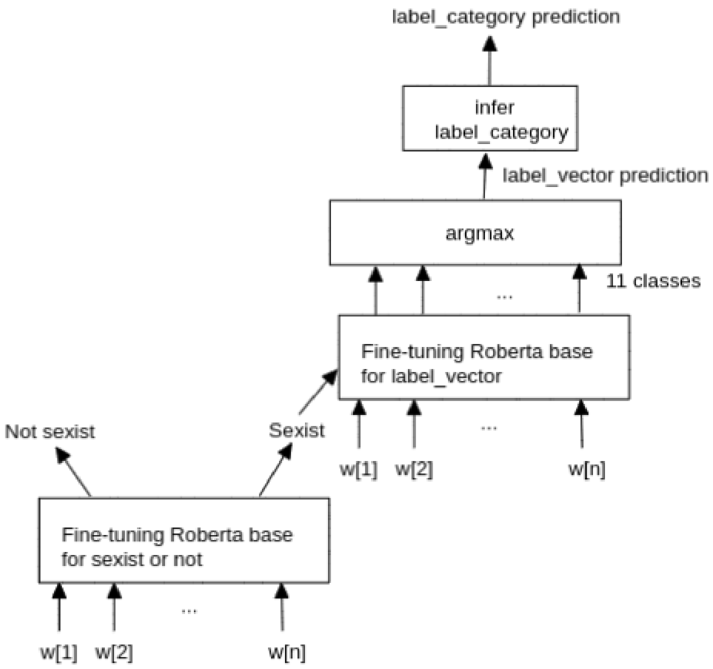

label_category prediction
infer label_category
label_vector prediction
argmax
11 classes
...
Fine-tuning Roberta base for label_vector
w[1]
w[2]
...
w[n]
Not sexist
Sexist
Fine-tuning Roberta base for sexist or not
w[1]
w[2]
...
w[n]

FIG: Model Architecture

The description of the model is as follows:

- At the first layer, we feed the encoded sentences to fine-tune Roberta so that it learns to predict whether the sentence is sexist or not. If it is not sexist, we can stop there.
- Else, we feed the encoded sentences to fine tune another Roberta which learns to predict the 11 class classification on label_vector
- Once the prediction on label_vector was obtained, we infer the result for label_category from there itself. For example, if it predicts label **3.1** on label_vector, we infer label **3** on label_category

Hyperparameters:

Based on some experimentation, following hyperparameters were found to give good results:

For sexist prediction model:

- Number of epochs = 4
- Sequence length = 68
- Train batch size = 32
- Test batch size = 32

For label_vector prediction model:

- Number of epochs = 20
- Sequence length = 128
- Train batch size = 128
- Test batch size = 64

Link to trained models and evaluation script:
https://drive.google.com/drive/folders/1-C7KIwsMikFHukKt3D-SVSS9FrO8GAAG?usp=sharing

Results:

Below are the F1 scores and accuracies obtained for the different classification models:

Binary Sexism Detection:

Below are the observations noted for binary classification on the label: 'label_sexist'. The train dataset was used to fine tune ELECTRA, XLNet and RoBERTa. These models were chosen because they have been pretrained on a very large corpus.

(Model having best performance has been highlighted)

Model	**F1 macro**
electra base discriminator	0.6932
xlnet base cased	0.7152
roberta base	0.8095

Categorical Sexism Detection:

Label_category (hierarchical training) inferred from Label_vector

Model	**F1 macro**
electra base discriminator	0.5041
xlnet base cased	0.5627
roberta base	0.5849

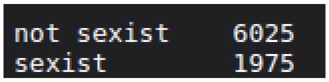
Fine-grained vector of sexism:

Label_vector determined by directly classifying all sentences labelled as sexist

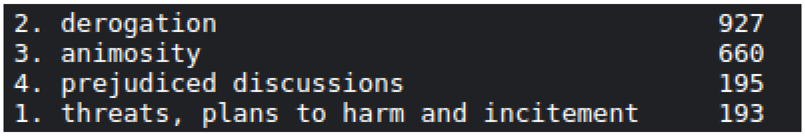
Model	**F1 macro**
electra base discriminator	0.3284
xlnet base cased	0.3715
roberta base	0.3962

Discussion:

- One possible approach to handle this classification problem is to train a classifier model from scratch. This approach is not feasible in our case because the amount of data is less. Hence we have directly started with fine tuning a pretrained model.
- All the label columns in the dataset were found to have high class imbalance. For example, the distribution for label_sexist and label_category were as follows:



```
not sexist     6025
sexist         1975
```



```
2. derogation                                 927
3. animosity                                  660
4. prejudiced discussions                     195
1. threats, plans to harm and incitement      193
```

We experimented by undersampling and oversampling the dataset to tackle class imbalance problem but that didn't give improvement over the validation set.

- It was found that with the given dataset, the F1 score on label_category was close to 0.5 which if included in the prediction hierarchy would propagate a lot of error while making predictions on label_vector. So, we tried to train a model to predict label_vector and based on the prediction, we inferred its classification on label_category. From the results, it can be seen that this technique yielded much better results on both label_category as well as label_vector.
- Just for experimenting, when we trained the classifier for label_category on the combined training dataset of 3 groups, we found its F1 score to be 0.9 which was a huge improvement from what we got with only the dataset given to our group. However, with only the dataset given to us, it could not be improved beyond 0.53 when directly trained on label_category.
- The reason why we have considered ELECTRA is because it is trained using two transformer models - a generator and a discriminator and the discriminator's task is to identify words which have been replaced by the masked language model. Hence, ELECTRA is more capable than BERT to identify sentences which might be sexist, as BERT might not give much attention to these words.
- We have also explored RoBERTa as an extension to BERT with a different pretraining procedure. XLNet is also explored because since it has more parameters it is expected to perform better but might be slightly computationally inefficient.

References:

- Hierarchical classification: https://arxiv.org/pdf/1709.08267.pdf
- Initial code: https://mccormickml.com/2019/07/22/BERT-fine-tuning/
- Pre-trained models: https://huggingface.co/docs/transformers/model_doc/roberta
- XLNet: https://arxiv.org/abs/1906.08237
- ELECTRA: https://arxiv.org/abs/2003.10555
- RoBERTa: https://arxiv.org/abs/1907.11692v1