

Our implementation utilizes parts of the code from [1]. For generating topic proportion distributions, we use the code from [2].

1. `main.py`: It is used to run all the deep learning based methods including the baselines used in the submission.
2. `neural_approaches.py`: It involves training, prediction, training data creation and transformation, loss function assignment, class imbalance correction, and more.
3. `dlModels.py`: It consists of the deep learning architectures for the proposed and baseline methods
4. `loadPreProc.py`: It contains functions for data loading, preprocessing, and more.
5. `sent_enc_embed.py` : It generates sentence embeddings from BERT, USE, etc.
6. `word_embed.py`: It generates word embeddings from ELMo, GloVe, etc.
7. `gen_batch_keras.py`: It contains functions to generate batches of inputs for training and testing.
8. `Kmeans.py`: It is used to compute k-means clustering information.
9. `neg_data_gen.py`: It is used to filter in negatively labeled posts for the sexism detection auxiliary task from Blog Authorship Corpus.
10. `rand_sample.py`: It is used to generate data for sexism detection auxiliary task.
11. `arranging.py`: It involves data loading for some auxiliary tasks and more.
12. `topic_rep.py`: It involves saving the topic proportion distributions into a file.
13. `corr_stats.py`: It computes the label correlation statistics used in the proposed loss functions.
14. `evalMeasures.py`: It comprises functions used for multi-label classification evaluation.
15. `rand_approach.py`: It performs random label selection in accordance with the normalized label frequencies of labels in the training data.
16. `prepare_unlabdata.py`: It generates augmented data for the self-training baseline.
17. `analysis_labels_per_post.py`: It is used to A) get results for the test data segments created based on the number of labels per post and B) help select the samples and the associated information in Table 3.
18. `chart_train_ratio.py`: It generates a graph showing performance variation across different percentages of the training data used.
19. `chart_class_wise_performance.py`: It is used to plot class-wise F score.
20. `gen_co_mat_heatmap.py`: It is used to visualize conditional label co-occurrence probabilities.
21. `TraditionalML_LP.py`: It implements traditional machine learning methods.
22. `lda2vec`: This folder contains all the files used to generate the topic proportion distributions.

23. Config\_deep\_learning.txt: It is a configuration file for the proposed methods and deep learning baselines.
24. Config\_traditional\_ML.txt: It is a configuration file for the traditional ML methods.

## References

1. P. Parikh, H. Abburi, P. Badjatiya, R. Krishnan, N. Chhaya, M. Gupta, and V. Varma, "Multi-label categorization of accounts of sexism using a neural framework," in Proceedings of EMNLP-IJCNLP, 2019, pp. 1642–1652.
2. Moody, Christopher E. "Mixing dirichlet topic models and word embeddings to make lda2vec." *arXiv preprint arXiv:1605.02019* (2016).