MIA Experiment Report

1 Introduction

Offensive language is a widespread issue on social media. A potential solution to tackle this problem is the automatic detection of offensive language (Zampieri et al., 2019). The report describes the MIA implementation for a hate-speech detection task, including the experiment setup and analysis of the results.

2 Experiment setup

In this experiment, to train the binary detection model, I use English samples from two datasets, namely the Offensive Language Identification Dataset (OLID) (Zampieri et al., 2019) and Hate Speech and Offensive Content Identification in Indo-European Languages (HASOC) (Mandl et al., 2019). All datasets were processed to either have the label "0" for non-offensive language or hate speech, and "1" for offensive language/hate speech. In terms of differences, the HASOC dataset contains texts from Facebook and Twitter, while OLID contains Twitter posts only.

I choose BERT-base-cased (Devlin, 2018) as the foundation model of our target and shadow models because the cased version can better capture strong emotions conveyed through capital letters. The main motivation for using this setup is that both datasets were all released after BERT, suggesting that the training data for BERT likely did not include these datasets.

The MIA experiment is conducted through the following steps:

1. Create a target model: fine-tune a bert-base-cased model on 13k OLID data samples with 1 epoch training.

2. Create a shadow model: fine-tune a bert-base-cased model on 5k OLID and 3k HASOC data samples

3. Generate data for the attack model: I take the 8k membership samples in step 2 with ˜8k non-memberships. Figure 1 shows the label distribution of the training data portion. I further append the offensive label to the beginning of each sample demarcated with a special character.

4. Process data for the attack model training: I use a sentence transformer to transform the input texts (with offensive label appended) into 384 dimension vectors. Then I concatenate each vector with the output positive probability generated by the shadow model.

5. Train the attack model with 5-fold: I then feed all features into a logistic regression model and train it with membership labels 0 or 1.

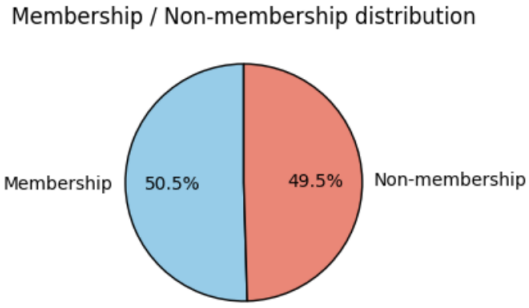

Figure 1: Membership label distribution in ˜16k training samples for the attack model

3 Experiment results

I perform 5-fold cross-validation and compute the AUC scores for each fold. The scores for each fold are 0.660, 0.665, 0.648, 0.653 and 0.643. The attack model achieves an average AUC score of 0.65 (See Figure 3), indicating performance slightly better than random guessing.

Regarding classification evaluation, the attack model achieves an overall accuracy of 0.61 with

a standard deviation of 0.005 across the five folds, showing consistent and stable performance. The classification report is shown in Figure 2.

	precision	recall	f1-score	support
0	0.61	0.62	0.62	7852
1	0.62	0.60	0.61	8000
accuracy			0.61	15852
macro avg	0.61	0.61	0.61	15852
weighted avg	0.61	0.61	0.61	15852

Figure 2: Classification report of the attack model on 5-fold aggregated test sets.

4 Discussion

4.1 Label-aware attack model

When transforming texts, ADePT also ingests intent labels alongside the utterances. I adopt this approach in attack model training by appending the offensive label to the text, marked with a special character and a tab. I also test on the offensive-label-unaware setting, which yields an average AOC score of 0.63 and an accuracy of 0.59 – slightly lower than the label-aware setting. I assume the MIA performance on the target model is also lower without the offensive label.

4.2 Feature dimension concern

To convert the text input into feature representations, I use a sentence transformer that generates a 364-d vector for each sample, instead of using a 768-d larger model. This choice was made to focus on the probability feature as higher-dimensional vectors might dilute its impact.

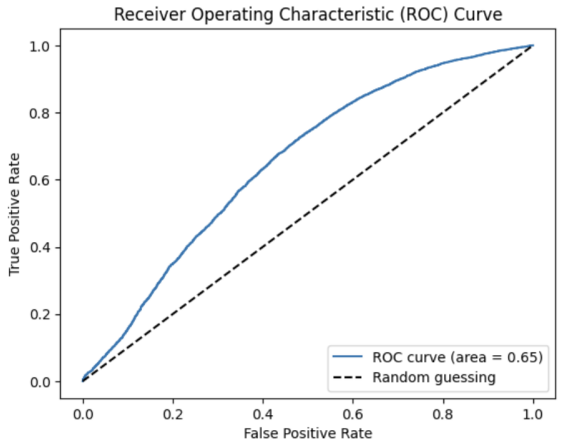

Figure 3: ROC curve of the attack model on 5-fold aggregated test sets.

4.3 Reproducibility

To ensure the reproducibility of the experiment, I set seed values for the target, shadow, and attack model training. In stratified-5-fold, a random state is also set.

4.4 Limitations

In this experiment, both the target and shadow models are trained for only one epoch. Future experiments could explore the impact of training with different epoch settings. I infer that the performance of the MIA attack model would improve with additional training epochs, as this would help the model to memorize the data (Duan et al., 2024).

5 Summary

This report describes a basic experiment implementing MIA for a hate speech detection task. The final attack model has a mediocre capability in distinguishing between membership and non-membership.

References

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. 2024. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*.

Thomas Mandl, Sandip Modha, Prasenjit Majumder, Daksh Patel, Mohana Dave, Chintak Mandlia, and Aditya Patel. 2019. Overview of the hasoc track at fire 2019: Hate speech and offensive content identification in indo-european languages. In *Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation*, pages 14–17.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Predicting the type and target of offensive posts in social media. *arXiv preprint arXiv:1902.09666*.