Software Engineering Report

Fine-Grained Hate Speech Detection in Arabic Tweets Using Adversarial Machine Learning

Team Members:

Aysha Mohamed
Mostafa Hasona
Youssef Muhammed Abd El-alim

Abstract:

This report introduces an innovative method for detecting offensive language and hate speech in Arabic social media posts by utilizing emojis as extralinguistic anchors. The research addresses the challenge of identifying such content across the diverse linguistic landscape of the Arabic language, which includes numerous dialects and cultural expressions. By focusing on emojis, the study circumvents the limitations of keyword-based methods that often fail to capture the full spectrum of offensive language. The collected dataset, the largest of its kind, was annotated and categorized into offensive, hate, vulgar, and violent speech. Various machine learning models were benchmarked, demonstrating superior performance compared to traditional methods. The findings underscore the importance of cultural context in detecting offensive content and provide a valuable resource for further research.

Introduction:

Background: The project involves detecting hate speech in Arabic tweets using a fine-tuned BERT model, combined with an evaluation of the model's robustness through adversarial attacks. The project also includes integrating this system into the Serbot platform to enable interactive features, such as responding to user input and detecting hateful content.

Objectives:

1. Pre-process Arabic text data.
2. Utilize pre-trained models like BERT and fast Text for classification.
3. Evaluate model performance.
4. Explore adversarial robustness by perturbing input data.

Methdology:

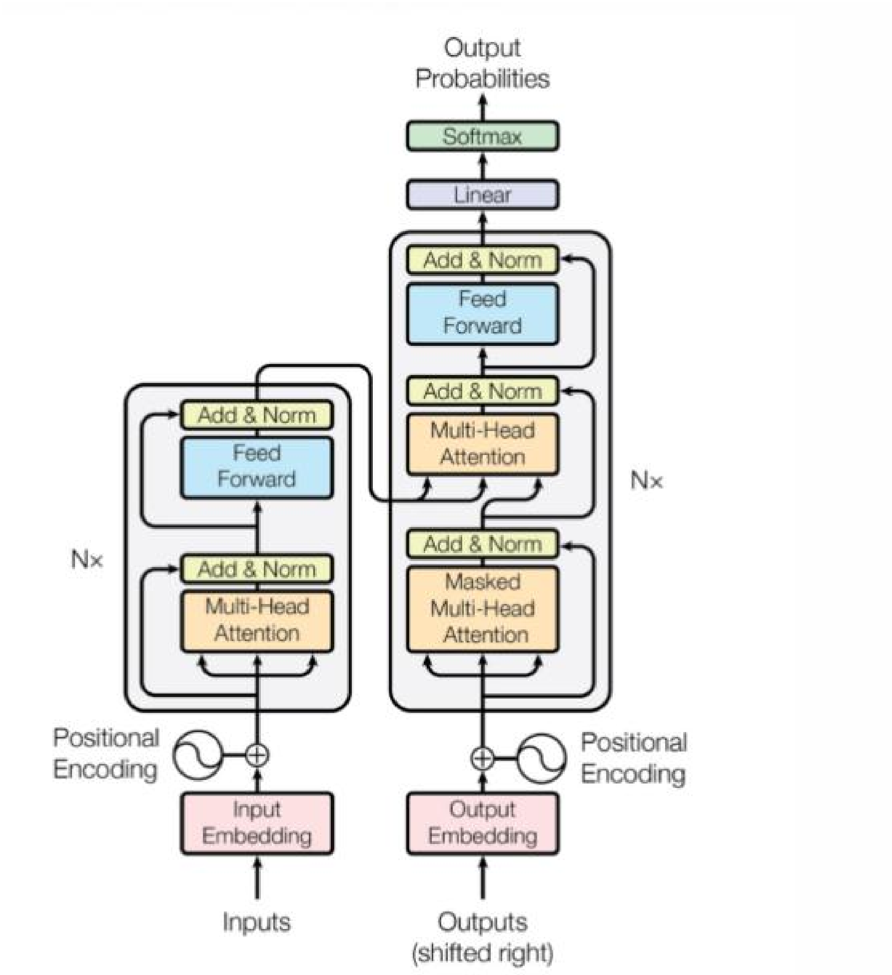
. Model Setup and Loading

1. BERT-based Qarib Model
2. fastText Model

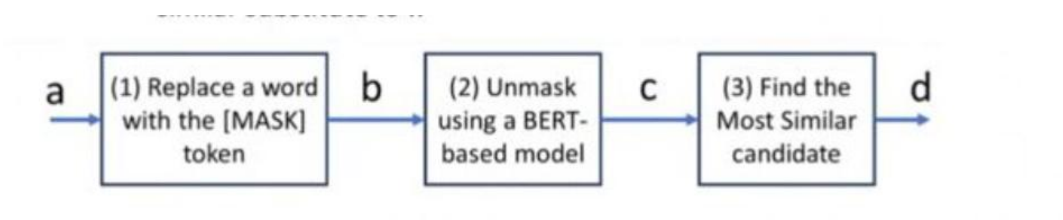
Adversarial Attack Method:

The adversarial attack involves perturbing tweets by replacing words with synonyms or similar terms, thereby attempting to mislead the text classification model. The script includes error handling to manage potential interruptions during execution.
We present a black-box greedy approach consisting of
1- Replacing the word with the [MASK] token
2- Applying Bert to find substitute words
3- Utilizing a pre-trained word vector model to find the most similar substitute

Model Architecture:



Adversarial attack:



Result and Evaluation:

Prediction and Evaluation
- **Classification**: The cleaned tweets are passed through the model, which outputs logits. The class with the highest logit is chosen as the prediction.
- **Accuracy Calculation**: Accuracy is calculated specifically for the hate speech

class, providing insights into the model's performance on this critical task.

5.2 Perturbation Analysis

- **Process**: Each tweet is perturbed by replacing words with their semantically similar counterparts, as determined by the fastText embeddings.
- **Evaluation**: The impact of these perturbations is measured by the model's change in predictions, indicating the model's robustness.

BERT-based Qarib Model Training Evaluation:

```
trainer.evaluate()
```

```
[80/80 00:06]
(1269, 2)
(12, 1269, 12, 64, 64)
1269
              precision    recall  f1-score   support

           0       0.95      0.99      0.97      1160
           1       0.77      0.50      0.60       109

    accuracy                           0.94      1269
   macro avg       0.86      0.74      0.79      1269
weighted avg       0.94      0.94      0.94      1269

[[1144   16]
 [  55   54]]
{'eval_loss': 0.45425209403038025,
 'eval_macro_f1': 0.7866272281835168,
 'eval_macro_precision': 0.8627785058977719,
 'eval_macro_recall': 0.7408098702942107,
 'eval_accuracy': 0.9440504334121356,
 'eval_runtime': 12.8356,
 'eval_samples_per_second': 98.866,
 'eval_steps_per_second': 6.233,
 'epoch': 3.0}
```

BERT-based Qarib Model Testing Evaluation:

```
print(classification_report(test_df["label"].values, outputs, target_na
```

```
              precision    recall  f1-score   support

      NOT_HS       0.95      0.99      0.97      2269
          HS       0.82      0.56      0.67       271

    accuracy                           0.94      2540
   macro avg       0.89      0.77      0.82      2540
weighted avg       0.94      0.94      0.94      2540
```

Adverserial attack result:

Detailed Steps:

Emoji Check:

- Ignores words containing emoji, as they are not considered for replacement.

Word Masking:

- Masks the word to be replaced and generates candidate sequences using a masked language model (unmasker_MARBERT).

Semantic Similarity:

- Computes semantic similarity between the original and candidate words using FastText embeddings.

Selection Criteria:

- Filters out candidates with the same root as the original word or those that do not fit the sentence context.
- Selects the candidate with the highest semantic similarity that changes the tweet's classification.

Future Work and Improvements :

Model Enhancement: Future improvements could involve experimenting with more advanced models, such as transformers or ensemble methods, to enhance classification accuracy and robustness.

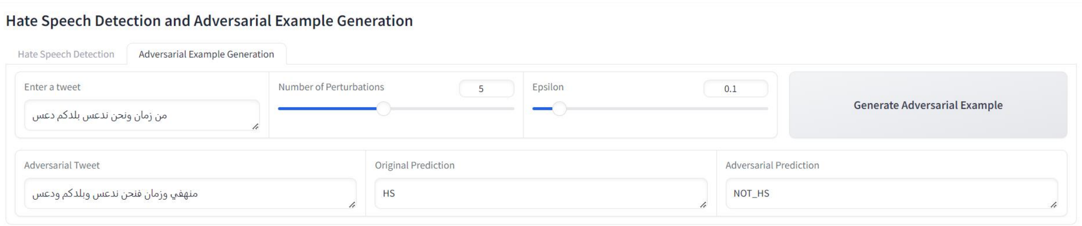
Data Augmentation: Incorporating additional data sources, including different dialects and contexts, can improve the model's understanding and generalization.

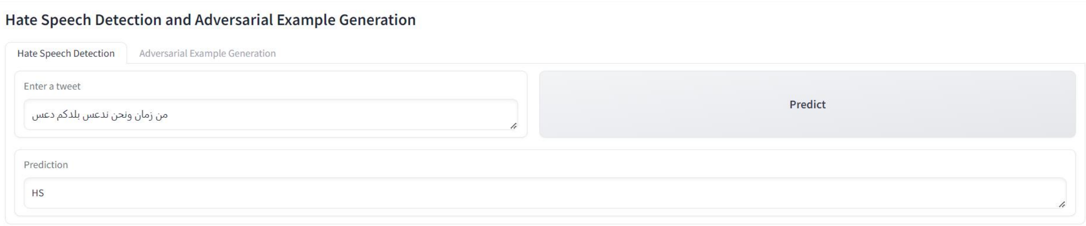
Adversarial Training: Implementing adversarial training techniques could help the model learn to resist perturbations, making it more robust to adversarial attacks

Conclusion:

This project showcases the application of state-of-the-art NLP techniques for detecting hate speech in Arabic tweets. The comprehensive workflow includes data preparation, model loading, prediction, and robustness testing through adversarial perturbation. The results highlight the model's capabilities and areas for potential improvement.

Output:


Hate Speech Detection and Adversarial Example Generation
Hate Speech Detection
Adversarial Example Generation
Enter a tweet
من زمان ونحن ندعس بلدكم دعس
Number of Perturbations
5
Epsilon
0.1
Generate Adversarial Example
Adversarial Tweet
منهفي وزمان فنحن ندعس وبلدكم ودعس
Original Prediction
HS
Adversarial Prediction
NOT_HS


Hate Speech Detection and Adversarial Example Generation
Hate Speech Detection
Adversarial Example Generation
Enter a tweet
من زمان ونحن ندعس بلدكم دعس
Predict
Prediction
HS