Hate Speech Detection in Arabic Text

Report (2)

Name: Mohamed Ahmed Mohamed Eissa
ID: 20100303

Abstract

This research investigates various models for detecting hate speech in Arabic text. The study evaluates classical machine learning algorithms, advanced deep learning techniques, and Transformer models. The objective is to develop a robust system for hate speech detection that addresses the challenges posed by the Arabic language's complex morphology, extensive vocabulary, and diverse dialects.

---

1. Introduction

1.1 Problem Statement

The proliferation of social media has led to a significant increase in harmful online content, making hate speech detection crucial for maintaining respectful digital environments. Detecting hate speech in Arabic is especially challenging due to the language's intricate morphology, rich vocabulary, and multiple dialects.

1.2 Objective

The objective of this research is to develop a robust system for detecting hate speech in Arabic text by utilizing classical machine learning, advanced deep learning, and Transformer models.

1.3 Foundation

This research is based on the GitHub repository Hate-Speech-Detection_OSACT4-Workshop.

---

2. Methodology

2.1 Dataset

Data Preprocessing:
Initial preprocessing techniques were applied to clean the input text. The preprocessing functions used include:

- Removing diacritics
- Normalizing Arabic text
- Removing punctuation
- Eliminating repeating characters
- Removing English words and numbers
- Cleaning extra spaces

Data Augmentation:
To enhance the diversity and size of the training dataset, various data augmentation techniques were employed:

- Synonym Replacement: Substituting words with their synonyms.
- Random Insertion: Adding random words at various positions.
- Random Oversampling: Increasing the number of instances in the minority class by duplicating existing examples.

2.2 Data Preparation

Modifications were made to the classification model to enable it to classify text as either hate speech or not hate speech and also as offensive or not offensive.

---

3. Model Classification

3.1 Classical Machine Learning Algorithms with TF-IDF

Three classical machine learning algorithms were evaluated:

-

Support Vector Machine (SVM):
Accuracy = 95%, Recall = 50%
Finds the hyperplane that best separates data into different classes.

Random Forest:
Accuracy = 96%, Recall = 43%
Uses an ensemble of decision trees for making predictions.

Logistic Regression:
Accuracy = 95%, Recall = 52%
A linear model used for binary classification.

3.2 Deep Learning Models with AraVec Word Embeddings

Three deep learning models were tested:

Long Short-Term Memory (LSTM):
Accuracy = 96%, Recall = 55%
Captures long-term dependencies in sequential data.

Gated Recurrent Unit (GRU):
Accuracy = 95%, Recall = 55%
A variant of LSTM with a simpler architecture and fewer parameters.

Convolutional Neural Network (CNN):
Accuracy = 97%, Recall = 57%
Typically used for image data but applicable to text by treating it as a sequence of characters or words.

3.3 Transformer Models with Word Embeddings

Two Transformer models were evaluated:

AraBERT:
Accuracy = 92%, Recall = 80%
A bidirectional model designed to capture contextual information from both directions in a sentence.

MARBERT:
Accuracy = 93%, Recall = 85%
A more advanced Transformer model that offers improved context understanding.

---

4. New Completed Tasks

4.1 Data Preprocessing Enhancements

A new function to handle emojis has been added, removing emoji from text

4.2 Model Expansion

All previously planned models have been successfully integrated and evaluated:

Classical Machine Learning:

- **Decision Trees:** Added. Uses a flowchart-like structure for decision-making based on attribute tests.

Accuracy = 87%, Recall = 50%

- **Gradient Boosting:** Added. An ensemble method that sequentially builds models to correct errors made by previous ones.

Accuracy = 96%, Recall = 60%

Deep Learning Models:

- **Bidirectional LSTM (BLSTM):** Added. Extends LSTM networks by processing data in both forward and backward directions, capturing context from both directions.

Accuracy = 95%, Recall = 61%

Transformer Models:

- **ARAT5:** Added. A Transformer-based model designed for a range of NLP tasks by converting them into a text-to-text format.

Accuracy = 91%, Recall = 75%

- **ARAGPT2:** Added. A Transformer-based model trained to predict the next word in a sequence.

Accuracy = 94%, Recall = 85%

---

5. Deployment

Models such as AraBERT, MARBERT, and ARAT5 have been deployed successfully. These models will improve the system's performance in real-world applications, providing more accurate and context-aware hate speech detection.

---

6. GUI System for Hate Speech Classification

6.1 Overview

A GUI has been developed to enable users to classify Arabic text as either inoffensive, offensive but not hate speech, or offensive and hate speech. The GUI allows users to select from multiple pre-trained models, including MARBERT, BERT, DistilBERT, GPT-2, and a "Best of All Models" option that uses a majority voting system.

6.2 Features

Model Selection:

- Users can choose between MARBERT, BERT, T5, GPT-2, and the "Best of All Models" option.
- The "Best of All Models" utilizes predictions from all models and selects the outcome with the most votes.

User Interface:

- The GUI features a text entry field, model selection dropdown, classify button, and a display for showing the prediction.
- The interface is user-friendly with a clean design and a professional color scheme.

---

7. Conclusion

This research successfully developed a robust system for detecting hate speech in Arabic text by integrating classical machine learning algorithms, advanced deep learning models, and state-of-the-art Transformer models. The deployment of these models and the accompanying GUI provides a comprehensive and reliable tool for hate speech detection across various dialects and contexts in the Arabic language.