Team Members:

Aysha Mohamed
Mostafa Hasona
Youssef Muhammed Abd El-alim

Emojis as Anchors to Detect Arabic Offensive Language and Hate Speech

1. Abstract

This paper 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.

2. Introduction

This project focuses on detecting hate speech in Arabic tweets using machine learning models. The primary goals include:

- Preprocessing Arabic text data.
- Utilizing pre-trained models like BERT and fastText for classification.
- Evaluating model performance.
- Exploring adversarial robustness by perturbing input data.

Key libraries and their roles:

- **PyTorch**: A deep learning framework used for loading and running the BERT model.
- **Transformers**: A library by Hugging Face for accessing state-of-the-art pre-trained models like BERT.
- **Pandas and NumPy**: Essential for data manipulation and numerical computations.
- **NLTK**: Provides tools for tokenization and stopwords.
- **BeautifulSoup**: Used for parsing and cleaning HTML content from tweets.
- **fastText**: Used for obtaining word embeddings.

- **FarasaStemmer**: Specifically designed for stemming Arabic words.
- **emoji**: For handling and processing emojis present in the text.

3. Related Work

Previous studies have primarily focused on English-language datasets for hate speech detection, with fewer resources available for Arabic. Notable work includes the use of traditional machine learning methods and more recent deep learning approaches like BERT. However, the robustness of these models against adversarial attacks remains underexplored. Our research extends this literature by specifically targeting Arabic language challenges and assessing model resilience to adversarial perturbations, a gap not fully addressed in existing studies

4. Methods

Files and URLs:

- **Tweets File**: OSACT2022-sharedTask-test-tweets.txt containing tweet texts.
- **Labels File**: OSACT2022-sharedTask-test-taskB-gold-labels.txt containing the corresponding hate speech labels.
- **Arabic Stop Words**: Arabic_stop_words.txt lists common Arabic stop words.
- **Pre-trained Model Files**: qarib pretrained model and fasttext-ar-vectors-150.bin provide the necessary pre-trained models.

Data Preparation and Pre-processing

```
Train data:  "https://alt.qcri.org/resources/OSACT2022/OSACT2022-sharedTask-train.txt"
```

```
Dev data:  "https://alt.qcri.org/resources/OSACT2022/OSACT2022-sharedTask-dev.txt"
Test data: "https://alt.qcri.org/resources/OSACT2022/OSACT2022-sharedTask-test-tweets.txt"
labels for test: "https://alt.qcri.org/resources1/OSACT2022/OSACT2022-sharedTask-test-taskB-gold-labels.txt"
```

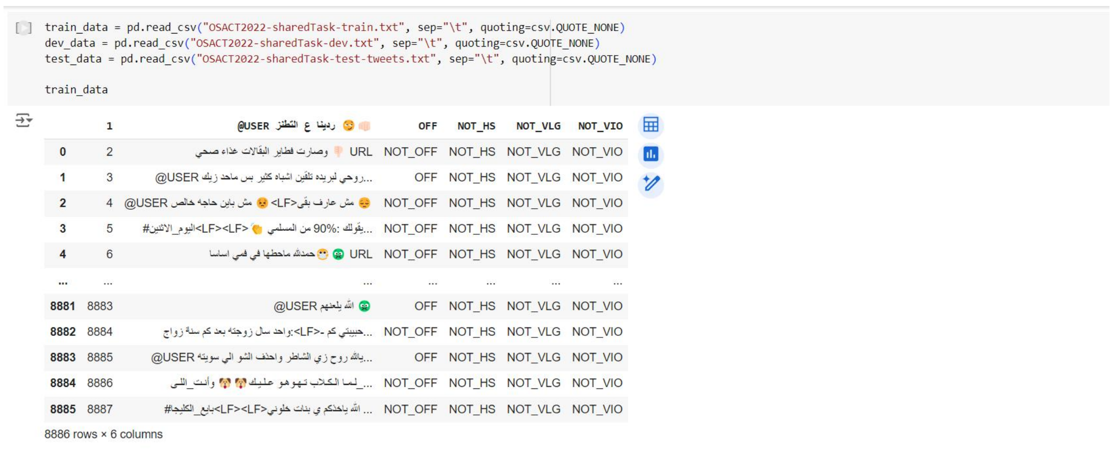
1- Data Loading

The tweet data and labels are loaded into a pandas DataFrame

Columns are renamed for consistency and clarity.

The data includes:

- tweet: The tweet text.
- class: The label indicating whether the tweet is hate speech (HS) or not (NOT_HS).

- 

```
train_data = pd.read_csv("OSACT2022-sharedTask-train.txt", sep="\t", quoting=csv.QUOTE_NONE)
dev_data = pd.read_csv("OSACT2022-sharedTask-dev.txt", sep="\t", quoting=csv.QUOTE_NONE)
test_data = pd.read_csv("OSACT2022-sharedTask-test-tweets.txt", sep="\t", quoting=csv.QUOTE_NONE)

train_data
```

	1	@USER ردينا ع التطنز 😏 👊🏻	OFF	NOT_HS	NOT_VLG	NOT_VIO
0	2	وصارت فطاير البقالات غذاء صحي 👎 URL	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
1	3	روحي لبريده تلقين اشباه كثير بس ماحد زيك @USER...	OFF	NOT_HS	NOT_VLG	NOT_VIO
2	4	😔 مش عارف بقى<LF> 😡 مش باين حاجه خالص @USER	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
3	5	يقولك :90% من المسلمي 👏 <LF><LF>اليوم_الاثنين#...	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
4	6	حمدلله ماحطها في فمي اساسا 😷 🤮 URL	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
...	...	...	...	...	...	...
8881	8883	🤮 الله يلعنهم @USER	OFF	NOT_HS	NOT_VLG	NOT_VIO
8882	8884	حبيبتي كم -<LF>:واحد سال زوجته بعد كم سنة زواج...	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
8883	8885	يالله روح زي الشاطر واحذف الشو الي سويته @USER...	OFF	NOT_HS	NOT_VLG	NOT_VIO
8884	8886	لما الكلاب تهوهو عليك 🐶 🐶 وأنت_اللي...	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO
8885	8887	الله ياخذكم ي بنات خلوني<LF><LF>بايع_الكليجا# ...	NOT_OFF	NOT_HS	NOT_VLG	NOT_VIO

8886 rows × 6 columns

2- Text Cleaning and Normalization

Several functions are defined to preprocess the tweets:

1.Normalization:

- Replace variations of certain Arabic characters with a standard form.
- Remove Arabic diacritics which can change the meaning of words.

2.Tokenization and Stop Words Removal:

- Tokenize the text using NLTK's WordPunctTokenizer.
- Remove English and Arabic stop words using lists defined in NLTK and Arabic_stop_words

3.Additional Cleaning:

- Remove repeating characters (e.g., "هههه" becomes "هه").
- Remove punctuation marks using a custom list of Arabic and English punctuation.

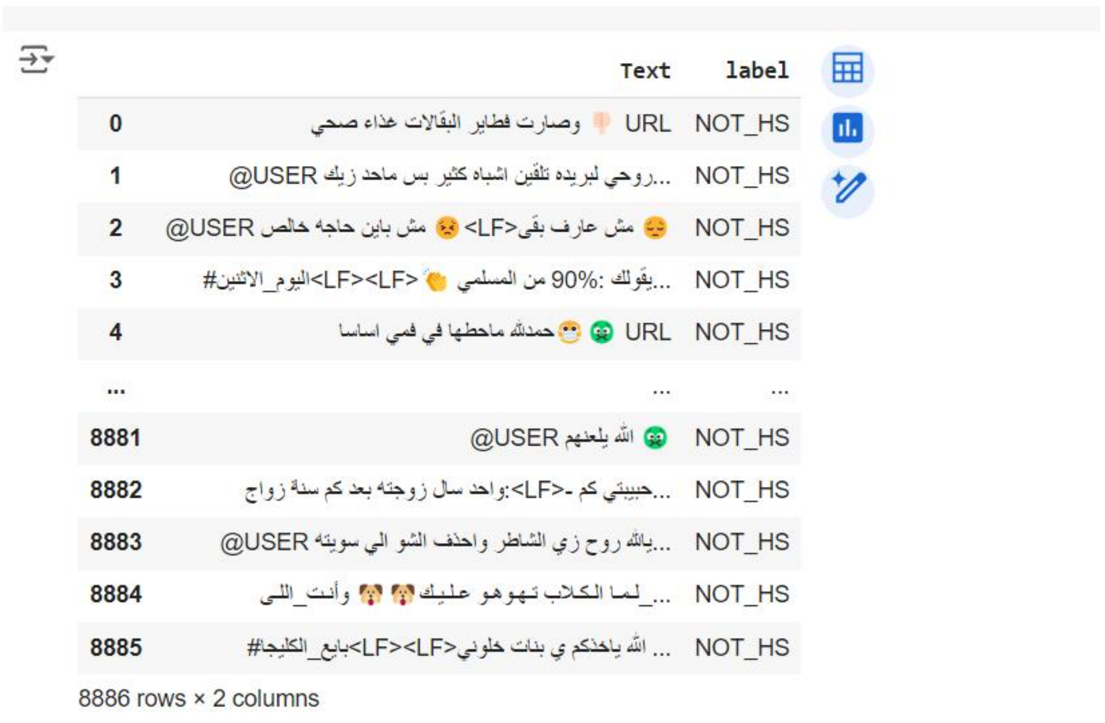
4.HTML and Non-Arabic Content Removal:

- Clean HTML tags using BeautifulSoup.
- Remove non-Arabic content and URLs.

5.Function Integration:

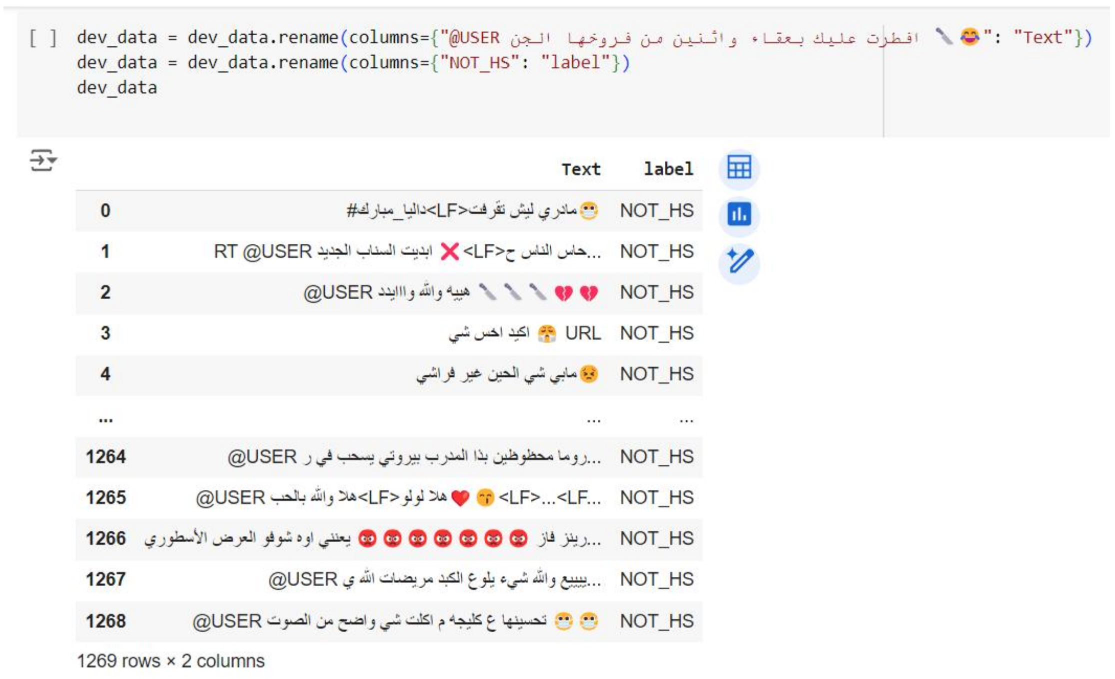
- The clean_str function integrates all these steps, providing a final cleaned version of the tweet text.

Train_data after preprocessing:



```
                                                        Text    label
0                 URL 👎 وصارت فطاير البقَالات غذاء صحي    NOT_HS
1           ...روحي لبريده تلقين اشباه كثير بس ماحد زيك @USER    NOT_HS
2  😔 مش عارف بقى<LF> 😣 مش باين حاجه خالص @USER    NOT_HS
3    ...يقولك :90% من المسلمي 👏 <LF><LF>اليوم_الاثنين#    NOT_HS
4                  URL 🤮 😷حمدلله ماحطها في فمي اساسا    NOT_HS
...                                                       ...      ...
8881                                   🤮 الله يلعنهم @USER    NOT_HS
8882       ...حبيبتي كم -<LF>:واحد سال زوجته بعد كم سنة زواج    NOT_HS
8883      ...يالله روح زي الشاطر واحذف الشو الي سويته @USER    NOT_HS
8884            ..._لما الكلاب تهوهو عليك 🐶 🐶 وأنت_اللى    NOT_HS
8885     ... الله ياخذكم ي بنات خلوني<LF><LF>بايع_الكليجا#    NOT_HS

8886 rows × 2 columns
```

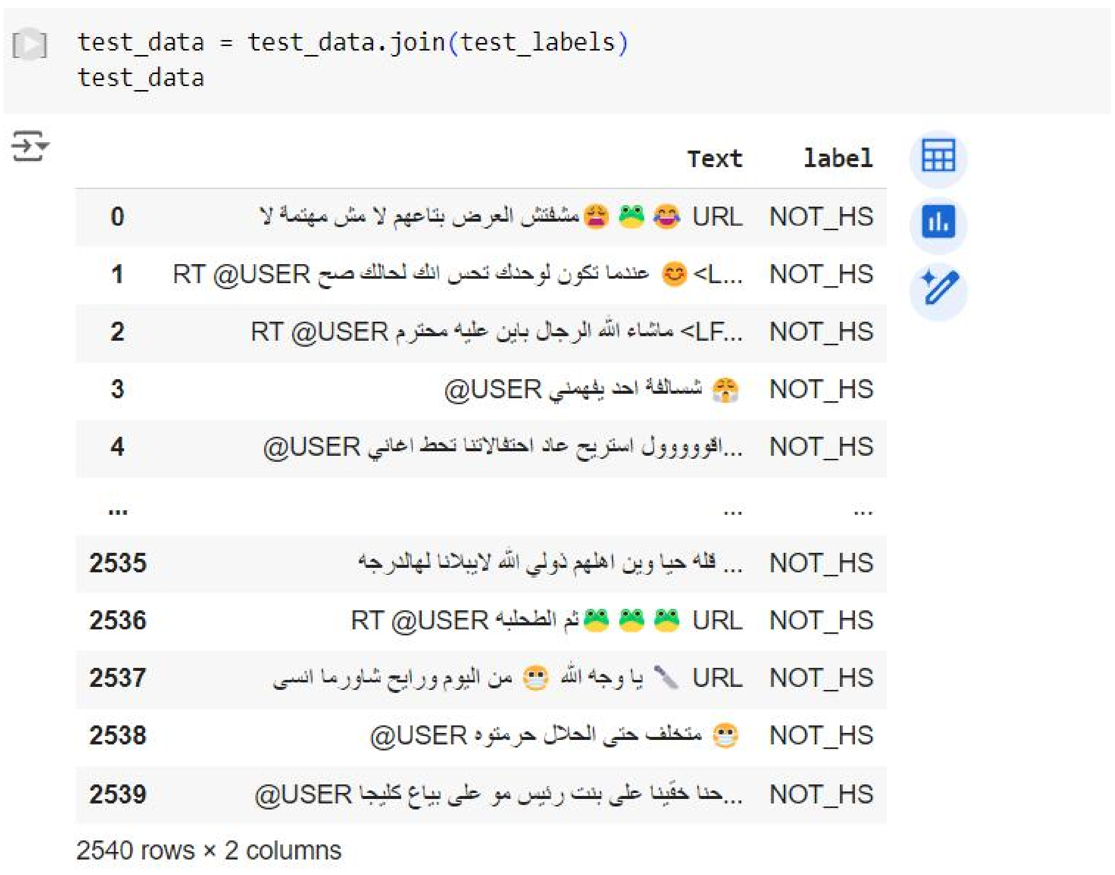
Dev_data after preprocessing:



```
[ ] dev_data = dev_data.rename(columns={"@USER الـجن  فـروخـهـا مـن واثـنـيـن بـعـقـاء عـلـيـك افـطـرت 🔪 😂": "Text"})
    dev_data = dev_data.rename(columns={"NOT_HS": "label"})
    dev_data
```

```
                                                      Text    label
0                      😷مادري ليش تقرفت<LF>داليا_مبارك#    NOT_HS
1          ...حاس الناس ح<LF> ❌ ابديت السناب الجديد RT @USER    NOT_HS
2                 💔 💔 🔪 🔪 🔪 هيبه والله واااايد @USER    NOT_HS
3                                 URL 😤 اكيد احس شي    NOT_HS
4                          😣مابي شي الحين غير فراشي    NOT_HS
...                                                    ...      ...
1264         ...روما محظوظين بذا المدرب بيروتي يسحب في ر @USER    NOT_HS
1265      ...<LF>...<LF 😚 ❤️ هلا لولو<LF>هلا والله بالحب @USER    NOT_HS
1266  ...ريتز فاز 😡 😡 😡 😡 😡 😡 😡 يعني اوه شوفو العرض الأسطوري    NOT_HS
1267          ...ييييع والله شيء يلوع الكبد مريضات الله ي @USER    NOT_HS
1268    😷 😷 تحسينها ع كليجه م اكلت شي واضح من الصوت @USER    NOT_HS

1269 rows × 2 columns
```

Test_data after preprocessing:



```
test_data = test_data.join(test_labels)
test_data
```

	Text	label
0	مشفتش العرض بتاعهم لا مش مهتمة لا 😫 🐸 😂 URL	NOT_HS
1	RT @USER عندما تكون لوحدك تحس انك لحالك صح 😊 <L...	NOT_HS
2	RT @USER ماشاء الله الرجال باين عليه محترم <LF...	NOT_HS
3	@USER شسالفة احد يفهمني 😤	NOT_HS
4	@USER اقووووول استريح عاد احتفالاتنا تحط اغاني...	NOT_HS
...	...	...
2535	قله حيا وين اهلهم ذولي الله لايبلانا لهالدرجه ...	NOT_HS
2536	RT @USER ثم الطحلبه 🐸 🐸 🐸 URL	NOT_HS
2537	يا وجه الله من اليوم ورايح شاورما انسى 😷 🔪 URL	NOT_HS
2538	@USER متخلف حتى الحلال حرمتوه 😷	NOT_HS
2539	@USER حنا حقَينا على بنت رئيس مو على بياع كليجا...	NOT_HS

2540 rows × 2 columns

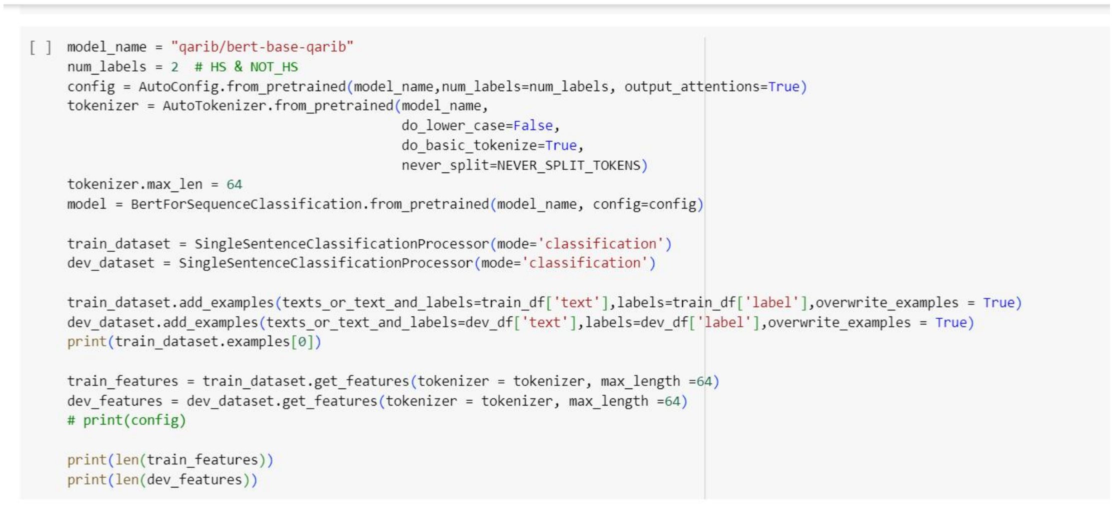
3. Data Augmentation

- This involves altering the input tweets to create new examples. The goal is to test the robustness of the model by checking if small changes in the input can change the classification.
- The process includes replacing words with similar ones using fastText embeddings and checking if the classification changes.

4. Model Setup and Loading

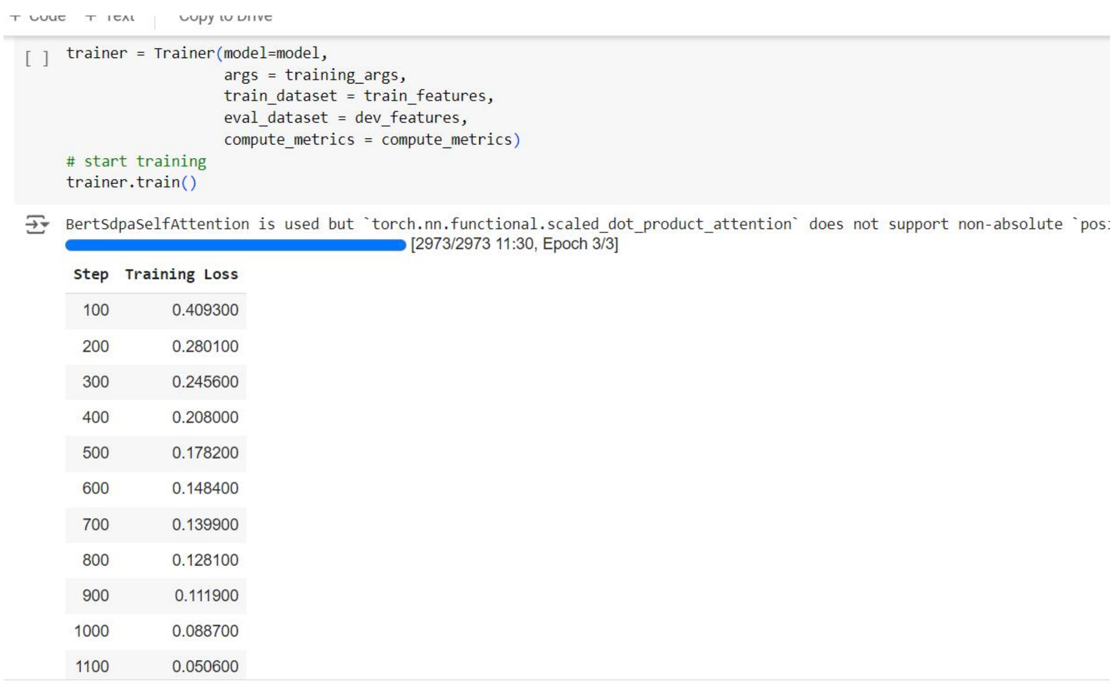
1.BERT-based Qarib Model

- **Model Loading**: The BERT model is loaded using the transformers library
- **Tokenizer**: The tokenizer is responsible for converting text into a format that the model can understand. It handles tasks like tokenization and padding.
- **State Dictionary**: Pre-trained weights are loaded to initialize the model with learned parameters.



```
[ ] model_name = "qarib/bert-base-qarib"
    num_labels = 2  # HS & NOT_HS
    config = AutoConfig.from_pretrained(model_name,num_labels=num_labels, output_attentions=True)
    tokenizer = AutoTokenizer.from_pretrained(model_name,
                                          do_lower_case=False,
                                          do_basic_tokenize=True,
                                          never_split=NEVER_SPLIT_TOKENS)
    tokenizer.max_len = 64
    model = BertForSequenceClassification.from_pretrained(model_name, config=config)

    train_dataset = SingleSentenceClassificationProcessor(mode='classification')
    dev_dataset = SingleSentenceClassificationProcessor(mode='classification')

    train_dataset.add_examples(texts_or_text_and_labels=train_df['text'],labels=train_df['label'],overwrite_examples = True)
    dev_dataset.add_examples(texts_or_text_and_labels=dev_df['text'],labels=dev_df['label'],overwrite_examples = True)
    print(train_dataset.examples[0])

    train_features = train_dataset.get_features(tokenizer = tokenizer, max_length =64)
    dev_features = dev_dataset.get_features(tokenizer = tokenizer, max_length =64)
    # print(config)

    print(len(train_features))
    print(len(dev_features))
```

Training the model:



```
[ ] trainer = Trainer(model=model,
                      args = training_args,
                      train_dataset = train_features,
                      eval_dataset = dev_features,
                      compute_metrics = compute_metrics)
    # start training
    trainer.train()
```

```
BertSdpaSelfAttention is used but `torch.nn.functional.scaled_dot_product_attention` does not support non-absolute `pos:
[2973/2973 11:30, Epoch 3/3]

Step  Training Loss
 100  0.409300
 200  0.280100
 300  0.245600
 400  0.208000
 500  0.178200
 600  0.148400
 700  0.139900
 800  0.128100
 900  0.111900
1000  0.088700
1100  0.050600
```

2.fastText Model

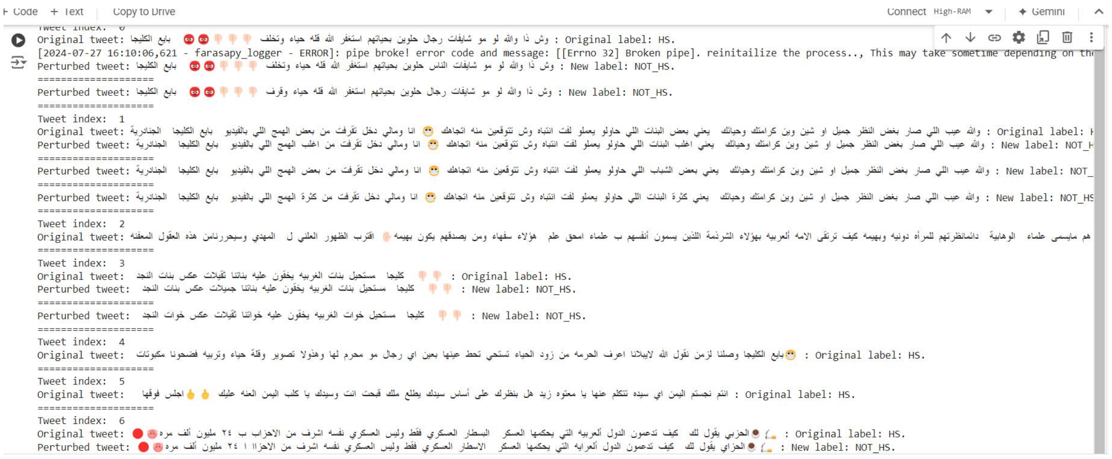
- **Loading**: The fastText model is loaded for word embeddings, crucial for semantic similarity tasks

5. 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



```
Tweet index:  0
Original tweet: وش ذا والله لو مو شايفات رجال حلوين بحياتهم استغفر الله قله حياء وتخلف 👎👎👎😡😡 بايع الكليجا : Original label: HS.
[2024-07-27 16:10:06,621 - farasapy_logger - ERROR]: pipe broke! error code and message: [[Errno 32] Broken pipe]. reinitailize the process.., This may take sometime depending on the
Perturbed tweet: وش ذا والله لو مو شايفات الناس حلوين بحياتهم استغفر الله قله حياء وتخلف 👎👎👎😡😡 بايع الكليجا : New label: NOT_HS.
====================
Perturbed tweet: وش ذا والله لو مو شايفات رجال حلوين بحياتهم استغفر الله قله حياء وقرف 👎👎👎😡😡 بايع الكليجا : New label: NOT_HS.
====================
Tweet index:  1
Original tweet: والله عيب اللي صار بغض النظر جميل او شين وين كرامتك وحيائك  يعني بعض البنات اللي حاولو يعملو لفت انتباه وش تتوقعين منه اتجاهك 😷 انا ومالي دخل تقرفت من بعض الهمج اللي بالفيديو  بايع الكليجا  الجنادرية : Original label: H
Perturbed tweet: والله عيب اللي صار بغض النظر جميل او شين وين كرامتك وحيائك  يعني اغلب البنات اللي حاولو يعملو لفت انتباه وش تتوقعين منه اتجاهك 😷 انا ومالي دخل تقرفت من اغلب الهمج اللي بالفيديو  بايع الكليجا  الجنادرية : New label: NOT_H
====================
Perturbed tweet: والله عيب اللي صار بغض النظر جميل او شين وين كرامتك وحيائك  يعني بعض الشباب اللي حاولو يعملو لفت انتباه وش تتوقعين منه اتجاهك 😷 انا ومالي دخل تقرفت من بعض الهمج اللي بالفيديو  بايع الكليجا  الجنادرية : New label: NOT_
====================
Perturbed tweet: والله عيب اللي صار بغض النظر جميل او شين وين كرامتك وحيائك  يعني كثرة البنات اللي حاولو يعملو لفت انتباه وش تتوقعين منه اتجاهك 😷 انا ومالي دخل تقرفت من كثرة الهمج اللي بالفيديو  بايع الكليجا  الجنادرية : New label: NOT_HS
====================
Tweet index:  2
Original tweet: هم مايسمى علماء  الوهابية  دائمانظرتهم للمرأه دونيه وبهيمه كيف ترتقى الامه ألعربيه بهؤلاء الشرذمة اللذين يسمون أنفسهم ب علماء امحق علم  هؤلاء سفهاء ومن يصدقهم يكون بهيمه ✋ اقترب الظهور العلني ل  المهدي وسيحررناس هذه العقول المعفنه
====================
Tweet index:  3
Original tweet:  👎👎  كليجا  مستحيل بنات الغربيه يخقون عليه بناتنا ثقيلات عكس بنات النجد : Original label: HS.
Perturbed tweet:  👎👎  كليجا  مستحيل بنات الغربيه يخقون عليه بناتنا جميلات عكس بنات النجد : New label: NOT_HS.
====================
Perturbed tweet:  👎👎  كليجا  مستحيل خوات الغربيه يخقون عليه خواتنا ثقيلات عكس خوات النجد : New label: NOT_HS.
====================
Tweet index:  4
Original tweet:  😷بايع الكليجا وصلنا لزمن نقول الله لايبلانا اعرف الحرمه من زود الحياء تستحي تحط عينها بعين اي رجال مو محرم لها وهذولا تصوير وقلة حياء وتربيه فضحونا مكبوتات : Original label: HS.
====================
Tweet index:  5
Original tweet:  انتم نجستم اليمن اي سيده تتكلم عنها يا معتوه زيد هل بنظرك على أساس سيدك يطلع ملك قبحت انت وسيدك يا كلب اليمن العنه عليك  🖕 🖕اجلس فوقها : Original label: HS.
====================
Tweet index:  6
Original tweet: 🚬 🟤الحزبي يقول لك  كيف تدعمون الدول ألعربيه التي يحكمها العسكر  البسطار العسكري فقط وليس العسكري نفسه اشرف من الاحزاب ب ٢٤ مليون ألف مره 🐷🔴 : Original label: HS.
Perturbed tweet: 🚬 🟤الحزاي يقول لك  كيف تدعمون الدول ألعرايه التي يحكمها العسكر  الاسطار العسكري فقط وليس العسكري نفسه اشرف من الاحزاا ا ٢٤ مليون ألف مره 🐷🔴 : New label: NOT_HS.
```

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.

- **Effectiveness:** The perturbation mechanism is designed to exploit potential vulnerabilities in the text classification model. The effectiveness of perturbations depends on the model's robustness and the quality of word replacements.
- **Pre-trained Models:** Ensure that the pre-trained models and custom functions used for word replacement and classification are properly configured and accessible.

5. Evaluation and Results

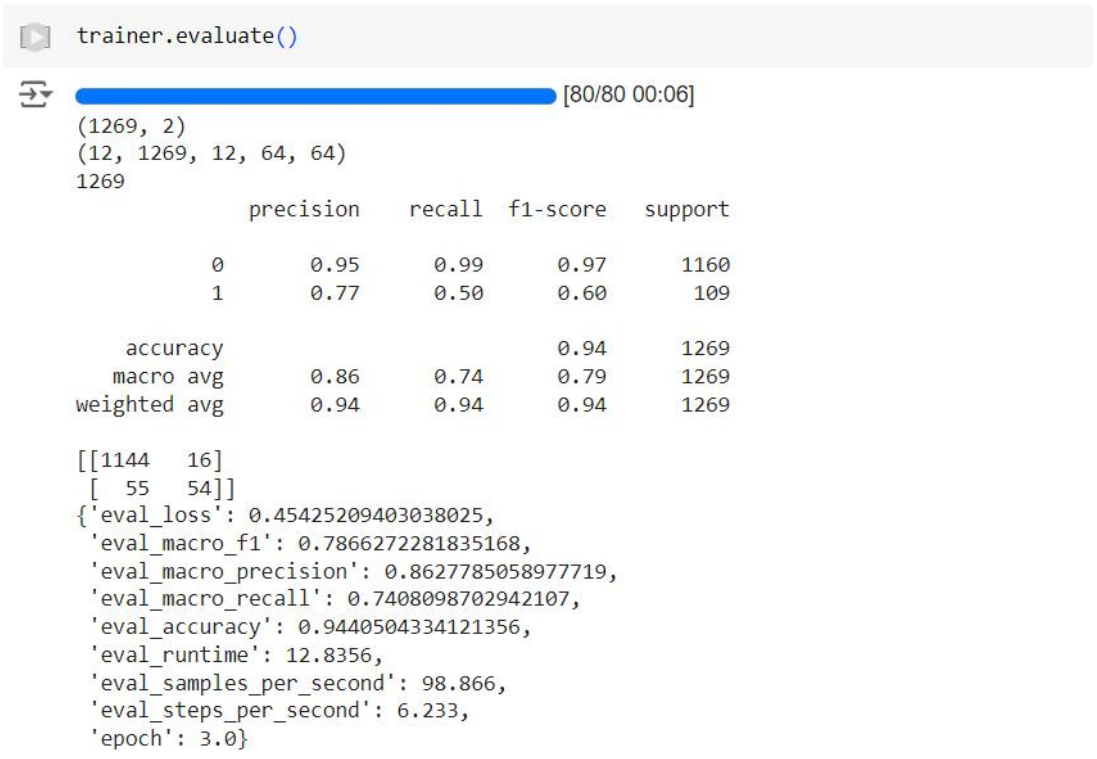
5.1 Prediction and Evaluation

- **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.

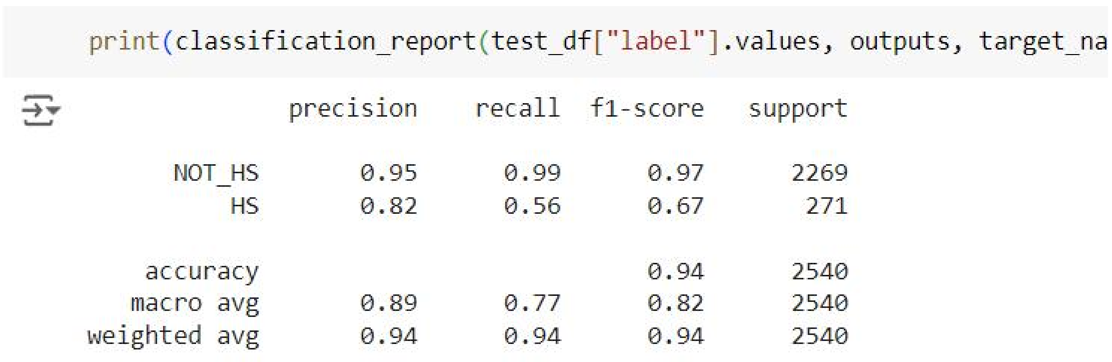
BERT-based Qarib Model Training Evaluation:



```
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
```

6. 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

7. 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.