

# AlexU-AIC at Arabic Hate Speech 2022: Contrast to Classify

**Ahmad Shapiro, Ayman Khalafallah, Marwan Torki**

Computer and Systems Engineering Department

Alexandria University

Alexandria, Egypt

{ahmad.shapiro, ayman.khalafallah, mtorki}@alexu.edu.eg

## Abstract

Online presence on social media platforms such as Facebook and Twitter has become a daily habit for internet users. Despite the vast amount of services the platforms offer for their users, users suffer from cyber-bullying, which further leads to mental abuse and may escalate to cause physical harm to individuals or targeted groups. In this paper, we present our submission to the Arabic Hate Speech 2022 Shared Task Workshop (OSACT5 2022) using the associated Arabic Twitter dataset. The shared task consists of 3 sub-tasks, sub-task A focuses on detecting whether the tweet is offensive or not. Then, For offensive Tweets, sub-task B focuses on detecting whether the tweet is hate speech or not. Finally, For hate speech Tweets, sub-task C focuses on detecting the fine-grained type of hate speech among six different classes. Transformer models proved their efficiency in classification tasks, but with the problem of over-fitting when fine-tuned on a small or an imbalanced dataset. We overcome this limitation by investigating multiple training paradigms such as Contrastive learning and Multi-task learning along with Classification fine-tuning and an ensemble of our top 5 performers. Our proposed solution achieved 0.841, 0.817, and 0.476 macro F1-average in sub-tasks A, B, and C respectively.

**Keywords:** Offensive Language Detection, Contrastive Learning, Multi-task Learning

## 1. Introduction

The Internet has revolutionized the way humans communicate, providing organizations and people with many features to promote and express themselves. Social media platforms (e.g. Facebook, Twitter, etc.) became a daily habit and even a source of income for many individuals. As of (2021<sup>1</sup>) Twitter had 206 million monetizable daily active users worldwide who can interact with each other and freely express their opinions. Unfortunately, without proper moderation and prevention, offensive language and hate speech may result in mental abuse to users or groups of individuals, as a matter of fact, social media can act as a propagation mechanism for violent crimes by enabling the spread of extreme viewpoints (Müller and Schwarz, 2020).

Research community has been focused on identifying the offensive language on social media in multiple languages (such as English, German, etc.), but offensive language detection is a challenge for Arabic, not only because it's a morphologically rich language, but because Arabic is considered as "macrolanguage" with many dialects. Arabic dialects differ in various ways from MSA "Modern Standard Arabic". These include phonological, morphological, lexical, and syntactic differences (Abdul-Mageed et al., 2018).

To address those challenges, hate speech datasets for multiple dialects have been collected such as L-HSAB (Mulki et al., 2019) for Levantine Dialects, T-HSAB (Haddad et al., 2019) for Tunisian Dialects. Also, previous shared tasks such as : OffensEval 2020 (Zampieri et al., 2020) that focused on identifying offensive language from Tweets in Arabic and other multiple lan-

guages, OSACT4 2020 (Mubarak et al., 2020) that focused on the detection of both offensive Language and hate Speech as its two sub-tasks respectively.

OSACT5 2022 presents a fine-grained detection of hate speech on Arabic Twitter shared task that consists of three sub-tasks. Sub-task A focuses on detecting whether the tweet is offensive or not. Then, for offensive Tweets, sub-task B focuses on detecting whether the tweet is hate speech or not. Finally, for hate speech tweets, sub-task C focuses on detecting the fine-grained type of hate speech among six different classes.

We approach the problem by exploring pre-trained transformer models using Arabic corpus. Given the imbalanced small dataset of 8.8k labeled tweets, transformers models tend to over-fit easily under this setting. Hence, we explore different training strategies such as Contrastive learning with different losses and training paradigms. Also, we explore the Multi-task learning approach. We also do a comparative study to decide which training strategy succeed on each sub-task. Our proposed solution of an ensemble of our top five models for Sub-task A, and a Multi-task learner for both Sub-tasks B and C solution achieved 0.841, 0.817, and 0.476 macro F1-average in sub-tasks A, B, and C respectively. Our results show a significant improvement on the majority baselines of 0.394, 0.472, 0.135 macro F1-average.

The following abbreviations will be used throughout the paper : Offensive Language (OFF), Hate-Speech (HS), Hate-Speech Classes (HS-C), Multi-task Learning (MTL).

<sup>1</sup><https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/>

## 2. Related Work

Hugely influenced by (Aldjanabi et al., 2021) work, we were able to explore many previous approaches to Arabic (HS) and (OFF) detection using (MTL). The first Arabic Religious (HS) Twitter dataset was collected by (Albadi et al., 2018). Their model encoded the tweets using GRUs trained on AraVec embeddings (Ashi et al., 2018) and then the features are passed to SVM classifier. They achieved the best performance with 79% accuracy.

(Haddad et al., 2019) collected 6k tweets for (HS) and abusive language for Tunisian Dialect (T-HSAB). They used Term Frequency weighting to extract n-grams features from tweets. Features are then used to train Naive Bayes and SVM classifiers. Their proposed method achieved 0.836 F1-score.

Related work from OSACT2020 (Mubarak et al., 2020) submissions that incorporates (MTL) are (Djandji et al., 2020; Abu Farha and Magdy, 2020; Hassan et al., 2020).

(Djandji et al., 2020) fine-tuned AraBERT (Antoun et al., 2020) with (MTL). They obtained a great results with the small imbalanced dataset setting. Their proposed method achieved 0.9 macro-averaged F1-score. (Hassan et al., 2020) Experimented with multiple Classical Machine learning and Deep learning approaches. They used CNN-BiLSTM, SVM and M-BERT for the (HS) sub-task. Their stacked SVMs achieved 0.806 F1-Score.

(Abu Farha and Magdy, 2020) trained CNN-BiLSTM with (MTL) on the two sub-tasks, in addition to Mazajak Arabic Sentiment Analysis dataset (Abu Farha and Magdy, 2019), detecting the sentiment of the text. We can deduce a correlation between negative sentiment and the tweet being (HS) or (OFF). Their proposed model achieved 0.904, 0.737 F1-score in the (OFF) and (HS) sub-tasks respectively.

Moving from OSACT2020 submissions, (Aldjanabi et al., 2021) explores (MTL) more widely. They use dataset from OSACT2020 (HS) and (OFF), T-HSAB (Haddad et al., 2019), and (L-HSAB) (Mulki et al., 2019). They experimented with both AraBERT (Antoun et al., 2020) and MarBERT (Abdul-Mageed et al., 2021) models. They train 6 different (MTL) models using OSAT2020 two sub-tasks (HS) and (OFF) as the main sub-tasks, in addition two (L-HSAB) or (T-HSAB) or both, on both MarBERT and AraBERT. They report their best results on both (OFF) and (HS) sub-tasks using MarBERT model trained on the (HS) sub-task, (OFF) sub-task, and (L-HSAB) which is 3 class classification (Abusive, HS, Normal). Their score was 0.9234, 0.8873 F1-Scores in (OFF), (HS) respectively.

In our work we focus on exploring different training paradigms using pre-trained Arabic Transformer models due to their efficiency in Natural Language Understanding (NLU) tasks instead of classical machine

learning models. We use a different (MTL) approach by only considering the main 3 sub-tasks with under-sampled version of dataset, and balanced version of dataset using another datasets of the same tasks.

## 3. Approach

We follow a pragmatic study in model selection and training strategy selection for each sub-task.

We based our approaches on Encoder-Based Transformers models because of their efficiency on (NLU) tasks, but their only flaw is over-fitting on small and imbalanced data-sets. We overcome this problem by exploring multiple training paradigms such as :

- Classification Fine-tuning
- Contrastive Learning
- Multi-task Learning

Also, we use regularization techniques such as Dropout and Early-Stopping.

All of our models were developed using HuggingFace Library (Wolf et al., 2019), Sentence-Transformers Library (Reimers and Gurevych, 2019).

We had two choices of models that showed promising results on previous Arabic shared tasks, AraBERT (Antoun et al., 2020) and MarBERT (Abdul-Mageed et al., 2021). We loaded their latest checkpoints from hugging face.

In Section 3.4 we show how we chose only the best of those models - based on current task performance and pre-training data of the model - to use it as our main encoder that will run for the rest of the experiments.

### 3.1. Exploratory Data Analysis

The dataset (Mubarak et al., 2022) for the three sub-tasks is the same, containing 12.7K tweets that were annotated for :

- Sub-task A : OFF and NOT\_OFF
- Sub-task B : HS and NOT\_HS
- Sub-task C : NOT\_HS, HS1 (Race), HS2 (Religion), HS3 (Ideology), HS4 (Disability), HS5 (Social Class), and HS6 (Gender).

With two extra labels expressing tweet being vulgar : NOT\_VLG, VLG, and being violent : NOT\_VIO, VIO. Dataset was split into 70% (8887 tweets) for training, 10% (1270 tweets) for development, and 20% (2541 tweets) for testing. Sub-task C (HS Classes) labels distribution was very imbalanced; with 89.2%, 2.9%, 0.3%, 1.6%, 0%, 0.8%, 5.13% for classes from NOT\_HS, HS1 to HS6 respectively for training. We can see that HS4 (Religion) wasn't present in the training dataset. Development dataset follows a similar distribution but with only an extra example for HS4. Followed by sub-task B (HS) with only 10.8% (HS) labels

in training and 8.5% in development. And, finally sub-task A (OFF) with 35.7% (OFF) labels in training and 31.8% in development.

We discovered that only 2 out of the 8887 train tweets and 1270 development tweets combined didn't have emoji(s). This helped us in narrowing the search for Transformer models candidates to be used.

Our first candidate model was MarBERTv2 (Abdul-Mageed et al., 2021) for two reasons :

1. It was trained using 1B Arabic tweets which matches the text distribution of our dataset.
2. Emojis weren't filtered from the training dataset, as MarBERTv2 Vocabulary has 567 emoji.

Our second candidate was AraBERT (Antoun et al., 2020) due to its performance in our task as discussed in Section 2.

After submission, details about the dataset has been made public. We discovered that emojis were treated as anchors to build the dataset itself according to (Mubarak et al., 2022).

### 3.2. Data Pre-processing

Data pre-processing is an important step in classification tasks, many unnecessary tokens may not help in the given task, as a matter of fact, they may have bad influence on the final results. We ran the data-set through the following pre-processing steps :

- Arabic Letter Normalisation : We unify the Alef {ٱ} letter that may appear in different forms as following {ٱ̇ ٱ̈ ٱ̉} to {ٱ} .
- Punctuation Normalisation : We replace {؟} to {?}, {،} to {,}, {؛} to {;}.
- Digit Normalisation : We replace {...٠ ١٩٨٧٦٥٤٣٢١} to {1,2,3,4,5,6,7,8,9,10 ....}
- Hashtag segmentation : #هاش تاج to تاج\_هاش
- Diacritic removal except shaddah.
- Removal of symbols such as : {—, /, #, [, ], {, }, -, \_ , \*, @, USER, LF }
- Removal of repeated characters or emojis more than two times.

While (Djandji et al., 2020; Haddad et al., 2020) removed emojis as pre-processing step, and (Husain, 2020) replaced emojis with their description in Arabic. Normalisation of digits and punctuation, and removal of symbols and repeated characters, emojis were done to reduce scarcity of the representations. We decided to keep emojis without any cleaning or pre-processing because they are an important data feature as discussed

in Section 3.1.

We fine-tuned our models with and without pre-processing. We found that pre-processing improved the results as will be shown in Section 4.

### 3.3. Data Balancing and Additional Data Resources

We made a balanced version of our dataset (BALANCED) using dataset associated with OSACT2020 and OffenseEval2020 and (Chowdhury et al., 2020; Ousidhoum et al., 2019; Alakrot et al., 2018).

For Sub-task A , we used all samples from (Chowdhury et al., 2020; Alakrot et al., 2018; Ousidhoum et al., 2019) along with data associated with OSACT2020 and OffenseEval2020. We first took all OFF samples from OSACT2020 and then random under-sample all other data-sets, resulting in 19906 balanced data-set instead of the original 8887 associated with the task.

For Sub-task B, we used all samples from (Chowdhury et al., 2020; Ousidhoum et al., 2019) along with data associated with OSACT2020 and OffenseEval2020. We first took all HS samples from OSACT2020 and then random under-sampled all other data-sets. Which resulted in 4800 balanced data-set instead of the original 8887 associated with the task that had only 959 (HS) samples. We used (BALANCED) data only in Multi-task learning.

While other approaches (Ibrahim et al., 2020; Ibrahim et al., 2018) used data augmentation to tackle class imbalance. We chose to use extra data resources collected by different methods to study the effect of distribution mismatch.

### 3.4. Model Selection

As discussed in Section 3.1, we consider only two models in our Experiments : AraBERT (Antoun et al., 2020) and MarBERT (Abdul-Mageed et al., 2021). The introduction of Bidirectional Encoder Representation from Transformers (BERT) (Devlin et al., 2018) led to a revolution in the NLP world, as BERT-based models achieved state-of-the-art results in many tasks.

In the proposed architecture, we utilize a pre-trained language model and fine tune it for a specific task.

#### 3.4.1. MarBERT

MarBERTv1 is a large-scale pre-trained masked language model focused on both Dialectal Arabic (DA) and Modern Standard Arabic (MSA). It was trained on 1B Arabic tweets (15.6B tokens), (Abdul-Mageed et al., 2021) using a BERT-base architecture but without the Next Sentence Prediction (NSP) objective since tweets length are naturally short.

MarBERTv2 differs from v1 in the training dataset only. They add multiple data-sets and train the model for 40 epochs, readers can refer to the original paper (Abdul-Mageed et al., 2021) for more details.

### 3.4.2. AraBERT

AraBERT differs from MarBERTv1 and v2 in the training data. Most of its training data is MSA instead of DA as in MarBERT. They also use Farasa (Darwish and Mubarak, 2016) Arabic morphological segmentation in the text pre-processing.

As discussed earlier in Section 2, AraBERT showed a good performance in (MTL).

### 3.5. Classification Fine-tuning

We use Huggingface library (Wolf et al., 2019) to fine-tune our BERT-Based Models on a binary classification task for Sub-task A and B. We use ADAM optimizer and fine-tune for 100 epochs with early stopping patience of 10 epochs and report the best checkpoint.

We fine-tune AraBERTv2, MarBERTv1, MarBERTv2 on Sub-task A data with and without pre-processing to choose the model we will proceed with, and whether we will pre-process our data or not. Results are reported in Section 4.

### 3.6. Contrastive Learning

Instead of classification fine-tuning which adds a linear layer after the BERT encoder to leverage the pooled BERT representation in classification. And then back propagate the cross entropy loss to fine tune both the linear layer, and the BERT encoder parameters for classification objective. We explore another training objective, contrastive learning. It's main objective is minimizing the distance of pooled BERT representations between similar sentence pairs, and maximizing distance between dissimilar pairs.

There are many distance metrics, such as Cosine Similarity, Euclidean Distance, and Manhattan Distance. All of our experiments uses Cosine Similarity as distance metric with 0.7 margin between positive pairs and negative pairs.

We use Sentence-Transformers Library (Reimers and Gurevych, 2019) for training which is built over HuggingFace (Wolf et al., 2019) library.

The main reason we chose contrastive learning is data imbalance. The construction of data-set for contrastive learning eliminates any imbalance and increase the dataset by order of  $n^2$  as shown in Section 3.6.4. But it's very sensitive to annotation errors and differences in data distribution.

We experiment with different variants of contrastive loss and we use only the original data not the (BALANCED) 3.3.

#### 3.6.1. Contrastive Loss

Contrastive loss (Hadsell et al., 2006) expects as input two texts and a label of either 0 or 1.

If the label = 1 (Positive/Similar Examples), then the distance between the two embeddings is minimized.

If the label = 0 (Negative/dissimilar Examples), then the distance between the embeddings is maximized. Loss is calculated for all examples in each batch.

#### 3.6.2. Online Contrastive loss

Online Contrastive loss is similar to Contrastive Loss 3.6.1, but it selects hard positive (positives that are far apart) and hard negative pairs (negatives that are close) and computes the loss only for these pairs.

#### 3.6.3. Batch All Triplet Loss

Batch All Triplet Loss (Hermans et al., 2017) takes a batch with (label, sentence) pairs and computes the loss for all possible, valid triplets, i.e., anchor and positive must have the same label, anchor and negative a different label.

#### 3.6.4. Contrastive Data Creation

We limit our experiments to only sub-task A (OFF). Positive examples are pair of sentences with the same label (OFF, OFF) and (NOT\_OFF, NOT\_OFF). Negative samples are pair of sentences with different labels (OFF, NOT\_OFF). Let the number of examples with (OFF) label =  $n$ , (NOT\_OFF) =  $m$ .

We make three pools of examples :

1. Negative Examples : product of set (OFF) and (NOT\_OFF), resulting in size =  $n * m$
2. Positive Examples (OFF) : product of set (OFF) with itself, resulting in size =  $n^2$
3. Positive Examples (NOT\_OFF) : product of set (NOT\_OFF) with itself, resulting in size =  $m^2$

We experiment with different data sizes. Let our selected data *size* be 20K examples. To ensure balance between data, we sample 10K examples from the Negative Examples Pool, 5K from Positive Examples (OFF), Positive Examples (OFF) each respectively.

Generally,  $size/2$  from Negative examples,  $size/4$  from Positive Examples of (OFF) and (NOT\_OFF) respectively.

### 3.7. Multitask Learning

We focused in contrastive learning to increase the amount of data we have and solve the data imbalance by changing the training objective to contrast instead of classifying without any additional examples.

Hugely influenced with the results of (MTL) as discussed in Section 2. We experiment with Multi-task learning with our two versions of data (BALANCED) and the original task data as will be discussed in Section 4.

Rather than training the model on a single task, multi-task learning enables the model to benefit from multiple tasks at the same time. Given the existence of relatedness between tasks, an inductive transfer of knowledge will take place in the process of multitask learning (Djandji et al., 2020). We can see that the 3 main sub-tasks are an extensions of each other. Not offensive (sub-task A) tweets are always not hate speech (sub-task B), and the class of hate speech (sub-task C) is an explicit extension of hate speech detection (sub-task B).

The tasks share the same encoder, but there’s a task specific dense layer for prediction. We limit (MTL) tasks to Sub-tasks A, B, and C, as illustrated in Figure 1.

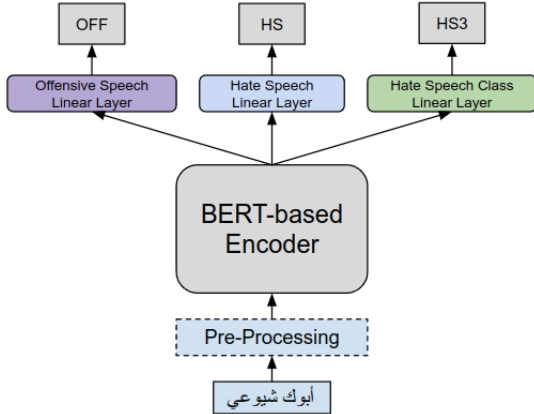


Figure 1: Multi-task Model Architecture.

## 4. Experimental Evaluation

In this section we report our results on both development set and test set for different approaches we used. We notice a quick over-fit while fine-tuning our models. All of the models achieve the best development set F1-score in the first 3 epochs.

### 4.1. Encoder Selection

We fine-tuned AraBERTv2, MarBERTv1, and MarBERTv2 on classification for Sub-task A, without pre-processing and a Dropout probability = 0.1 (Hugging-face default) and learning rate =  $2^{-5}$ .

As we can see in Table 1, both Versions of MarBERT performed better than AraBERT, so we moved forward with MarBERTv2.

Model	Dev F1
AraBERTv2	0.694
MarBERTv1	0.783
MarBERTv2	<b>0.841</b>

Table 1: Encoder Selection

### 4.2. Text Pre-processing

To test the effect of the pre-processing approach 3.2 we used. We fine-tuned both versions of MarBERT with and without pre-processing on Sub-task A.

We evaluated them on the development set, F1-Score is reported in Table 2.

As we can see that our pre-processing approach improved the results for both models.

Concluding this comparative study, we decided to move forward with MarBERTv2 as our encoder for the rest of the experiments, and with our text pre-processing approach.

Model	w/o PP	w/ PP
MarBERTv1	0.783	0.801
MarBERTv2	0.841	<b>0.850</b>

Table 2: Text Pre-Processing Effect

We tuned the dropout probability and found the best results with the default probability of 0.1 .

### 4.3. Contrastive Learning

As discussed in Earlier in Section 3.6.1, we fine-tuned our model using multiple contrastive objectives. Following the contrastive fine-tuning phase, we trained a linear layer and froze MarBERT parameters using original data. All contrastive fine-tuning was done on Sub-task A, results are shown in Table 3.

Model	Data Size	Dev F1
Online Contrastive	50K	0.851
Online Contrastive	1M	0.849
Contrastive	50K	0.847
Contrastive	250K	0.833
Batch All	*	0.847

Table 3: Contrastive Training Results

We plot separation between classes using PCA as shown in Figures 2 through 4.

They key difference between contrastive, online contrastive and batch all triplet is the combination of losses and the creation of data. For both contrastive loss and online contrastive loss, data is created manually as discussed in Section 3.6.4. Therefore, at a given iteration we have pair of sentences as single example, and a label that corresponds of whether the two sentences are similar or not. If the label corresponds to the pair = 1 (positive), distance between two sentences are reduced, and minimized otherwise. Contrastive calculates the loss for all pairs, but online contrastive calculates only hard examples (positive that are far apart, and negative that are close).

In contrast, Batch All uses the original from of data : single sentence with one label (OFF = 1, NOT\_OFF = 0). It creates valid triplets (Positive, Anchor, Negative) in a given batch and calculates the loss for triplet as whole not as single similarity between pair of sentences.

We noticed that contrastive learning is very sensitive to annotation errors and different text distributions. We also noticed that contrastive loss performance degrades with the increase of data size, unlike online contrastive loss which doesn’t face the same rate of degradation. We noticed a similar results between online contrastive and batch all, this can be attributed to the fact that both losses doesn’t take all data samples in consideration.

We’ve tried multiple classical machine learning classification algorithms, using BERT encoder output as our

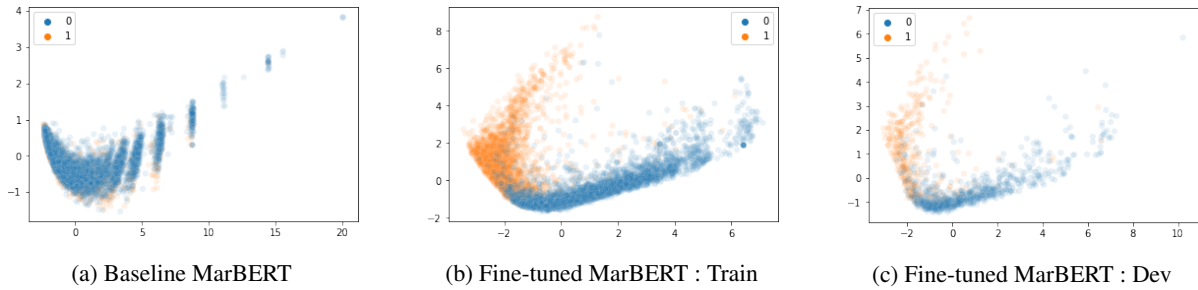


Figure 2: MarBERT vs Classification Fine-tuned MarBERT

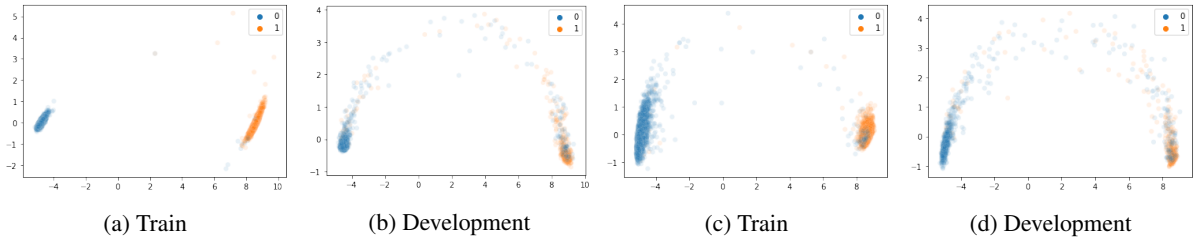


Figure 3: Left: Contrastive 50K. Right: Online Contrastive 50K

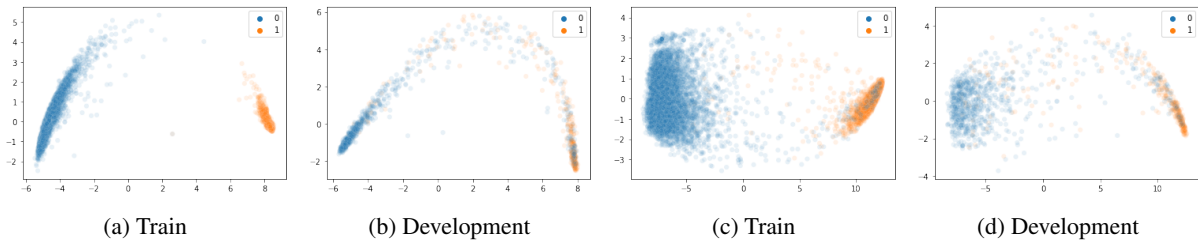


Figure 4: Left: Online Contrastive 1M. Right: Batch All Triplet Loss

features. We didn't see any improvement in the results. We tried SMOTE oversampling after applying dimensionality reduction using PCA to the encoder outputs, but we didn't see any improvement too.

#### 4.4. Multitask Learning

As discussed in Section 3.7, we fine-tuned MarBERT on multi-task learning objective using all 3 sub-tasks for 5 epochs and learning rate =  $2^{-5}$ .

We used the original data and the (BALANCED) data. Results on development dataset are shown in Table 4.

Sub-task	Original Data	BALANCED
A (OFF)	<b>0.838</b>	0.830
B (HS)	0.810	<b>0.830</b>
C (HS-C)	<b>0.435</b>	0.431

Table 4: Multitask Learning Results

We noticed that training for extra epochs achieves better results in Sub-task C, but degrades the performance on Sub-task A, B respectively. This can be attributed to So we decided to use the checkpoint trained for 5 epochs. We tuned the learning rate and found that  $2^{-5}$  achieved the best results.

As we can see in Table 4 that using (BALANCED) data didn't achieve better results in all sub-tasks, and it wasn't tied with data imbalance. Sub-task C had much extreme case of data imbalance than B, but when we used (BALANCED) data for A and B, C's result degraded. We can assume that this is due to the difference in distribution in data-sets used to construct the (BALANCED) dataset. And also due to the extreme class imbalance in sub-task C.

#### 4.5. Our Submission

##### 4.5.1. Sub-Task A

We used an ensemble of the following MarBERT based models:

1. Classification Fine-tuned
2. Batch All Fine-tuned.
3. Online Contrastive Fine-tuned with 50K examples.
4. Online Contrastive Fine-tuned with 1M examples.
5. Contrastive Fine-tuned with 50K examples.

We tried two ensemble techniques :

1. Summing positive and negative logits of ensembled models and the maximum between summed positive and summed negative is the classification result.
2. Using positive and negative logits of ensembled models as features to multiple classification algorithm to achieve a weighted voting.

Both methods achieved the same results on development set in terms of F1-Score. We moved forward with the former. We achieved 0.86 F1-Score on the development set. We report our results on the test set in Table 5.

Model	Majority Baseline	Ours
F1	0.394	<b>0.841</b>
Precision	0.325	<b>0.842</b>
Recall	0.5	<b>0.839</b>
Accuracy	0.651	<b>0.856</b>

Table 5: Sub-Task A (OFF) Test Results

#### 4.5.2. Sub-Task B and C

We used multitask model trained with (BALANCED) data to submit our result. Sub-Task B, C results are shown in Tables 6 and 7 respectively.

Model	Majority Baseline	Ours
F1	0.472	<b>0.817</b>
Precision	0.447	<b>0.855</b>
Recall	0.5	<b>0.787</b>
Accuracy	0.893	<b>0.937</b>

Table 6: Sub-Task B (HS) Test Results

Model	Majority Baseline	Ours
F1	0.135	<b>0.476</b>
Precision	0.128	<b>0.49</b>
Recall	0.143	<b>0.47</b>
Accuracy	0.893	<b>0.923</b>

Table 7: Sub-Task C (HS Classes) Test Results

## 5. Conclusion and Future Work

In this paper, we experimented multiple approaches along with classification fine-tuning to approach the problems of offensive language detection, hate-speech detection, and fine-grained hate-speech classes classification.

We evaluated BERT-based models trained on Arabic corpus. We found that MarBERTv2 performed the best, and better with our pre-processing approach.

We found that contrastive learning achieved slightly better results than classification fine-tuning when data imbalance wasn't extreme, and an ensemble of models

trained with contrastive objective and classification objective achieved better results than each of them solely. We used multitask learning to tackle extreme data imbalance. We found that training for more epochs benefits tasks with extreme data imbalance, but degrades the performance for tasks with mild and slight data imbalance.

For future work, we plan to investigate contrastive learning for extreme cases of data imbalance, accompanied with curriculum learning and a care-full selection of contrastive samples.

We also plan to tackle data imbalance by using data from multiple languages for the same task, using a language agnostic encoder trained with contrastive objective as LaBSE (Feng et al., 2020).

## 6. Acknowledgements

The authors would like to thank the Applied Innovation Center of Egyptian MCIT for providing necessary resources to complete the project presented in this paper.

## 7. Bibliographical References

- Abdul-Mageed, M., Alhuzali, H., and Elaraby, M. (2018). You tweet what you speak: A city-level dataset of Arabic dialects. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, May. European Language Resources Association (ELRA).
- Abdul-Mageed, M., Elmadany, A. A., and Nagoudi, E. M. B. (2021). ARBERT & MARBERT: deep bidirectional transformers for arabic. *CoRR*, abs/2101.01785.
- Abu Farha, I. and Magdy, W. (2020). Multitask learning for Arabic offensive language and hate-speech detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 86–90, Marseille, France, May. European Language Resource Association.
- Aldjanabi, W., Dahou, A., Al-qaness, M. A. A., Elaziz, M. A., Helmi, A. M., and Damaševičius, R. (2021). Arabic offensive and hate speech detection using a cross-corpora multi-task learning model. *Informat-ics*, 8(4):69, October.
- Antoun, W., Baly, F., and Hajj, H. M. (2020). Arabert: Transformer-based model for arabic language understanding. *CoRR*, abs/2003.00104.
- Ashi, M. M., Siddiqui, M. A., and Nadeem, F. (2018). Pre-trained word embeddings for arabic aspect-based sentiment analysis of airline tweets. In *Advances in Intelligent Systems and Computing*, pages 241–251. Springer International Publishing, August.
- Darwish, K. and Mubarak, H. (2016). Farasa: A new fast and accurate Arabic word segmenter. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*,

- pages 1070–1074, Portorož, Slovenia, May. European Language Resources Association (ELRA).
- Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2018). BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.
- Djandji, M., Baly, F., Antoun, W., and Hajj, H. (2020). Multi-task learning using AraBert for offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 97–101, Marseille, France, May. European Language Resource Association.
- Feng, F., Yang, Y., Cer, D., Arivazhagan, N., and Wang, W. (2020). Language-agnostic BERT sentence embedding. *CoRR*, abs/2007.01852.
- Haddad, B., Orabe, Z., Al-Abood, A., and Ghneim, N. (2020). Arabic offensive language detection with attention-based deep neural networks. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 76–81, Marseille, France, May. European Language Resource Association.
- Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742.
- Hassan, S., Samih, Y., Mubarak, H., Abdelali, A., Rashed, A., and Chowdhury, S. A. (2020). ALT submission for OSACT shared task on offensive language detection. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 61–65, Marseille, France, May. European Language Resource Association.
- Hermans, A., Beyer, L., and Leibe, B. (2017). In defense of the triplet loss for person re-identification. *CoRR*, abs/1703.07737.
- Husain, F. (2020). OSACT4 shared task on offensive language detection: Intensive preprocessing-based approach. *CoRR*, abs/2005.07297.
- Ibrahim, M., Torki, M., and El-Makky, N. (2018). Imbalanced toxic comments classification using data augmentation and deep learning. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 875–878.
- Ibrahim, M., Torki, M., and El-Makky, N. (2020). AlexU-BackTranslation-TL at SemEval-2020 task 12: Improving offensive language detection using data augmentation and transfer learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1881–1890, Barcelona (online), December. International Committee for Computational Linguistics.
- Mubarak, H., Darwish, K., Magdy, W., Elsayed, T., and Al-Khalifa, H. (2020). Overview of OSACT4 Arabic offensive language detection shared task. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 48–52, Marseille, France, May. European Language Resource Association.
- Mubarak, H., Hassan, S., and Chowdhury, S. A. (2022). Emojis as anchors to detect arabic offensive language and hate speech. *arXiv preprint arXiv:2201.06723*.
- Müller, K. and Schwarz, C. (2020). Fanning the Flames of Hate: Social Media and Hate Crime. *Journal of the European Economic Association*, 19(4):2131–2167, 10.
- Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhev, G., Mubarak, H., Derczynski, L., Pitenis, Z., and Çöltekin, c. (2020). SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020). In *Proceedings of SemEval*.

## 8. Language Resource References

- Abu Farha, I. and Magdy, W. (2019). Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy, August. Association for Computational Linguistics.
- Alakrot, A., Murray, L., and Nikolov, N. S. (2018). Dataset construction for the detection of anti-social behaviour in online communication in arabic. *Procedia Computer Science*, 142:174–181. Arabic Computational Linguistics.
- Albadi, N., Kurdi, M., and Mishra, S. (2018). Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, ASONAM '18*, page 69–76. IEEE Press.
- Chowdhury, S. A., Mubarak, H., Abdelali, A., Jung, S.-g., Jansen, B. J., and Salminen, J. (2020). A multi-platform arabic news comment dataset for offensive language detection. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC'20)*.
- Haddad, H., Mulki, H., and Oueslati, A. (2019). T-hsab: A tunisian hate speech and abusive dataset. In *ICALP*.
- Mulki, H., Haddad, H., Bechikh Ali, C., and Alshabani, H. (2019). L-HSAB: A Levantine Twitter dataset for hate speech and abusive language. In *Proceedings of the Third Workshop on Abusive Language Online*, pages 111–118, Florence, Italy, August. Association for Computational Linguistics.
- Ousidhoum, N., Lin, Z., Zhang, H., Song, Y., and Yeung, D.-Y. (2019). Multilingual and multi-aspect hate speech analysis. In *Proceedings of EMNLP*. Association for Computational Linguistics.



- Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11.
- Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface’s transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.