Offensive Content Identification in Arabizi Tweets

Majd Al Aawar, Ramzi Haddad, Yazan Hajj Diab, Ali Ismail

***Abstract*—Due to having versatile functionality, social media platforms have become more popular. With the increasing traffic on social media platforms comes an increase in negative and destructive behavior by users across the global multilingual landscape. Removing these actions or posts has been one of the main problems companies such Facebook, Twitter, and Instagram are trying to tackle. Due to the global nature of these platforms, detecting hostile content in every language has not been accomplished yet. In this paper, we focus on detecting offensive posts on Twitter written in Arabizi by classifying each tweet into one of the three categories: Normal, Abuse or Hate. Arabizi is a new and popular Arabic writing system, mainly used as an easier form of communication on the internet, that is a combination of Latin alphabets and digits. Since there are no available datasets on Arabizi, we transliterated the Arabic L-HSAB dataset into Arabizi using a rule-based converter of our design. Naive Bayes, Bidirectional Long Short-Term Memory (BiLSTM) network, convolutional network/LSTM combination (CNN-LSTM) and Bidirectional Encoder Representations from Transformers (BERT) models were developed and tested for both the Arabic and Arabizi data. Results were compared with the state-of-the-art for Arabic, Arabic results were also compared to those of Arabizi. In Arabic, BERT achieved state-of-the-art performance while in Arabizi the CNN-LSTM architecture achieved the best performance with a slight edge over BERT in terms of recall.**

# Introduction

Social media has seen its influence increase tremendously in the 21st century. Although social media is mainly used to connect people together, platforms such as Twitter, Facebook, and Instagram have been involved in elections [1], personal data scandals [2] and have become a prominent source of news [3] and a powerful marketing and advertising tool [4].

Unfortunately, the ability to communicate with people instantaneously is not always beneficial. According to a poll by UNICEF and the UN Special Representative of the Secretary-General on Violence against Children, 33% of children in 30 countries have been the target of offensive posts on social media platforms. This has led 20% of these children to deliberately absent from school as they are being cyberbullied [5]. In addition, social media allows people endorsing hate speech to reach an audience and recruit each other to form racist, sexist, or xenophobic groups that aim to hurt other groups of people [6].

Due to the harmful effects of these posts, social media platforms are implementing measures to block them [7]. However, it is challenging for these platforms to moderate posts in various languages. Facebook finds it difficult to keep up with the flood of new languages being used on its platform [8]. The difficulty lies in the training process typically being language specific, meaning that for each language to be processed, there is a need to collect a separate, large set of data which is comparable to solving the problem from scratch [9]. Even though Facebook is available in 111 languages, its community standards, which prevent users from posting offensive content, are only available in 41 languages. This impedes the job of Facebook in making it a friendly social media platform.

Arabic is one of the most commonly used languages on social media especially in the MENA region. It is also the fastest growing language on Twitter [10]. Arabic is often regarded by the natural language processing (NLP) society as a low-resource language offering its own set of challenges due to its complex morphology and the large variety of arabic dialects used which introduce ambiguity in word meanings since the same word can have different meanings in different dialects (for example “صلب” (“*solb*”) means “solid” in Egyptian but in Sudanese it has a profane meaning) [11][12]. Furthermore, Arabizi has emerged as a way to type arabic with latin characters and numbers. It is widely adopted by the young generation because it’s easy to type on keyboards [13].

Offensive post detection in social media has been the object of a plethora of works due to its problematic nature and the challenges it presents. Several have undertaken the task of detecting whether a social media post is offensive or not [14][15]. Many have focused on distinguishing between specific types of offense such as hate/general offense [16][17][18], racism/sexism [19] and offensive posts targeting specific people or groups [20]. Because of the lack of uniformity in depicting offensive social media content (hate speech, cyberbullying, abusive language, toxicity, racism, sexism…), the Offensive Language Identification Dataset (OLID) [21] was compiled from Twitter. Each tweet is annotated in a hierarchical manner: **Offensive** or **Not**, **Targeted** insult or **Untargeted**, and targeting an **Individual** or a **Group**. The release of this dataset has spurred a large number of works addressing this task using different approaches [22 - 29]. There is also a developing interest in accomplishing offensive content identification in different languages such as German, Hindi, Marathi, Italian and Indonesian [11][31][32][33].

Since social media has a prominent role in the Arab World, Arabic has attracted the attention of researchers in the field of NLP for offensive content detection [11][12][34][35]. Yet the ability to do this in Arabizi text has not been explored at the time of writing. In fact, many works consider Arabizi text as noise and therefore remove it [36]. A few papers have targeted the Arabizi language for other tasks such as sentiment analysis and yet many of them transliterate it into Arabic in the process [37][38]. Therefore, a clear gap in the processing of Arabic text written in Arabizi is identified and needs to be addressed as this writing system is becoming a norm among arab-speaking users.

In this work, we aim to detect offensive content in Arabizi tweets. We will then compare the classification results on Arabizi to classification on regular Arabic text. This will inform on which writing system is more easily learned by machine learning algorithms and whether it is a better practice to work with Arabizi writing or transliterate to standard Arabic as others have previously done. Fig. 1 shows the inputs and outputs of the intended model.

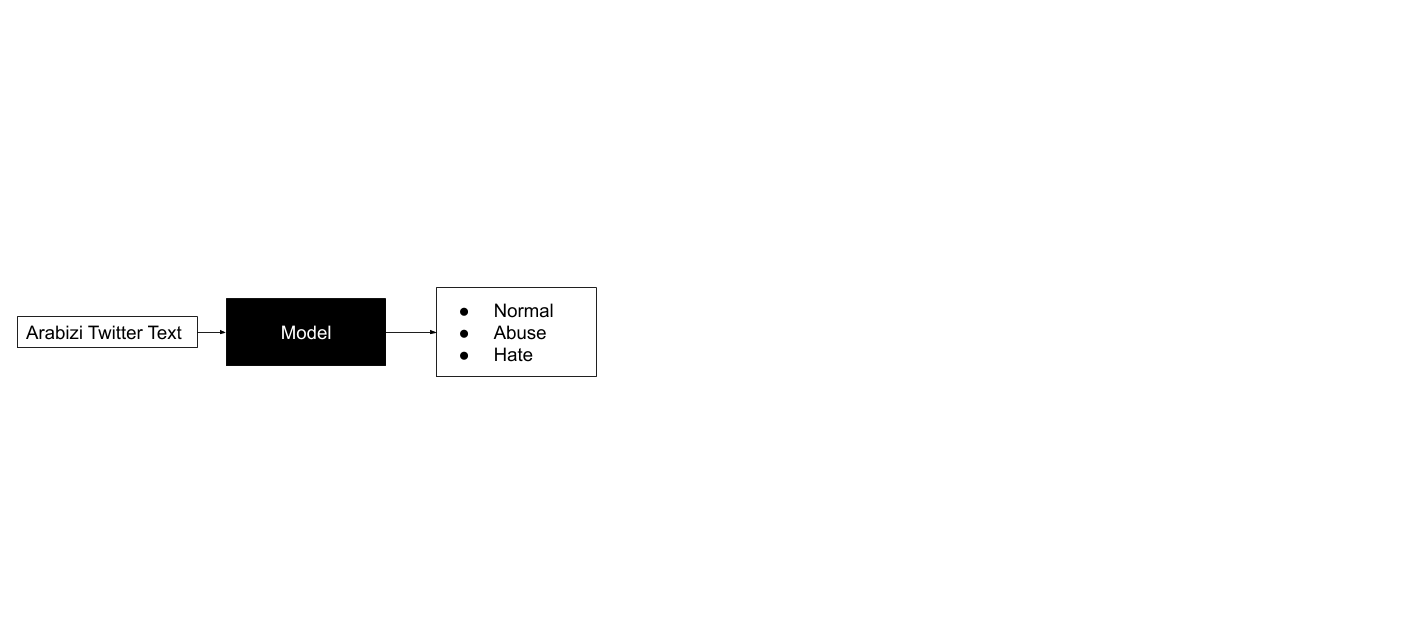


Fig. 1. Expected model inputs and outputs

The L-HSAB dataset[[1]](#footnote-0) will be used in this work. It consists of 5846 Arabic tweets annotated as one of the three classes: **Normal**, **Abuse** or **hate** (targeted offense). It is the most similar description of offensive content to that of the OLID in the Arabic language. This dataset will be transliterated into Arabizi using a set of rule-based conversions (since at the time of writing there is no native Arabizi dataset) and used to train and test the Arabizi model. It will also be used in its original form to compare classifications in Arabic to Arabizi.

# Related Work

There have been several works done on offensive text identification in various languages. This section will cover a comprehensive summary of previous approaches and more detailed descriptions on papers that were found to be seminal and state-of-the-art.

## Offensive Language Identification

Most of the works that have tackled the challenge of offensive text identification did so in English.

Noteworthy works done on English offensive language identification include Davidson et al. (2017) work on attempting to separate hate speech from other offensive language [16]. In this work, a crowdsourced dataset of tweets categorized as hate speech, offensive language, and neither was used. Using basic models like Logistic Regression and Linear SVM, it was found that: Lexical methods are effective ways to identify potentially offensive terms but are inaccurate at identifying hate speech; Certain terms are particularly useful for distinguishing between hate speech and offensive language; People tended to misclassify some of the data.

Although they intended to address the problem of distinguishing between offensive language and hate speech content, their results yielded a 40% misclassification rate of the hate speech content. This limitation can be expected when working with labels which have a somewhat overlapping definition.

The various works submitted by competitors for SemEval’s OffensEval 2019 task are shown by Zampieri et al. (2019) [22]. Results showed that among the top-10 teams, seven used BERT while the top non-BERT model (ranked sixth) used an ensemble of CNN and BLSTM+BGRU, together with Twitter Word2vec embeddings and token/hashtag normalization.

For a comprehensive review of other works done in this field the reader is referred to the survey by Al-Hassan and Al-Dossari [11]. This survey reveals that most of the work done in the field of offensive content detection is concentrated in English and that there is a lack of appropriate tools to perform this task in low-resource languages like Arabic. It also shows the non-uniformity in attributing offensive content across different works (abuse, bullying, racism...) which has been recently addressed by Zampieri et al. [21].

## Offensive Language Identification in Arabic

Arabic is considered to be a challenging language because it has many linguistic properties that set it apart from Indo-European languages. For instance, Arabic is written and read from right to left and has no capitalization. Also, short vowels are expressed using special punctuation marks called diacritics which are often omitted for ease of typing. Three variations of the Arabic language are recognized [12]: Traditional arabic used in ancient Arabic literature and in religious scripts. Modern Standard Arabic (MSA) which is the formal language officially taught in schools and used in books and formal documents. It is known to all arabic speakers. And Arabic dialects which are commonly practiced between people and vary between countries and even groups of people within a single country. A survey of the literature reveals several publications that have worked on the identification of offensive posts in arabic social media.

Haidar et al. (2017) [12] were the first to tackle this problem in Arabic. First, they provide a detailed background on cyberbullying detection in other languages and on other work done on Arabic such as sentiment analysis and named entity recognition. To develop their model, they scraped posts from Twitter and Facebook while restricting their search to the Middle East and Gulf area. They preprocessed and manually labelled the samples as “bullying” or “not bullying”. They experimented with the Naive Bayes classifier and with a support vector machine (SVM). For the SVM, they adapted TweetToSentiStrengthFeatureVector for Arabic which assigns sentiment weights to tweets. The SVM model outperformed NB but their results in general suffered from class imbalance.

Haidar et al. (2018) [39] continue their previous work by updating the dataset they collected by removing URLs, emoticons and all non-arabic characters. They use a Feed Forward Neural network. Their experiments centered around tuning hyperparameters such as number of epochs and number of hidden layers. They show that their final results are an improvement over their previous work.

In a more recent publication, Haidar et al. (2019) [40] introduce ensemble methods to further improve their system. They use the same dataset as before and experiment with different models (KNN, random forest, SVM, logistic regression) and different ensemble strategies (stacking, bagging and boosting). The optimal results were obtained with the stacking model and they are an improvement over their previous results.

Mohaouchane et al. [41] have worked on binary offensive classification of Youtube comments on popular Arab celebrities. They preprocess the data by removing non-Arabic characters, diatritics and punctuation. They oversample the offensive class (by duplication) to balance classes. The AraVec model is used and fine tuned to generate word embeddings. Four architectures are trained and compared: CNN, BiLSTM, BiLSTM with attention unit and CNN-LSTM. The CNN model achieves the highest overall performance. However, combining CNN and LSTM and including an attention unit with the BiLSTM increase the recall of the model but at the cost of precision. The authors argue that, in this context, it is better to have a higher recall since it means that offensive samples are more likely to be classified accurately.

To classify Youtube and Twitter comments as cyberbullying or not, Mouheb et al. [42] rely on a lexicon based approach as opposed to the more prevalent machine learning approaches. They manually compile a list of bullying words from a sample of comments and each word is assigned a weight ranging from slight to strong bullying. Comments are classified as bullying if they contain one word from the lexicon. Each comment is also assigned a degree of strength (mild, medium or strong) based on a function that takes into account the weight, number and repetition of offensive words. This approach is not context aware as it cannot detect offense unless explicitly expressed with certain vocabulary. Also, no validation criteria is proposed.

Overall, and despite many of the mentioned works achieving efficient solutions to the problem of offensive text in Arabic, the issue of different dialects is often overlooked even though expressions and meanings can vary greatly between dialects. Some works have datasets with mixed MSA and dialects while others try to reduce the variation in dialects by restricting their data geographically. It hasn’t been made clear whether it is better to develop machine learning models on very large datasets incorporating all the dialects or to develop one model per dialect.

## The Arabizi Language

Arabizi is a non-standardized writing system for the Arabic language using Latin characters. Digits are also used to express letters that are not present in Latin based languages (for example “5” is used to represent the voiceless uvular fricative “خ”). This way of writing is popular in informal situations and among the youth. The literature search performed during this work as well as the survey provided by Guellil et al. [43] show a few works targeting Arabizi.

Duwairi et al. [37] were the first to perform sentiment analysis on tweets written in Arabizi and classify them as positive, negative or neutral. The dataset for this paper was extracted using Twitter API and labelled using a crowdsourcing tool. As a preprocessing step, they tokenized every tweet into words, transformed every emoticon into its correlated word, removed stopwords, weighted every token using the Binary Model, and converted the tweet from Arabizi to Arabic using their built-in converter, which was one of the main contributions of this paper. Finally two classifiers were used, which were NB and SVM. The predictions, recall, and precision scores were better using SVM. Also, removing the Neutral class from the dataset improved the performance and the precision values in predicting the Positive and Negative classes. In this paper, their tool, which translates Arabizi to Arabic, wasn’t very efficient because usually filtering would increase the accuracy significantly while here the accuracy increased slightly when filtering was applied.

Duwari et al. [38] collected data using a PHP script and crowdsourcing. In Rapidminer, they built their own dictionary for stopwords and added it for topical classification. Also, they added their own packages to convert emoticons to their corresponding words, removed repeated letters, detect negation, map dialect words to MSA, remove links and mentions, and convert the Arabizi text to Arabic. The only used classifiers were the ones built in Rapidminer, which were KNN, NB and SVM. Since Rapidminer has memory issues the dataset used was small (1000 tweets) which makes this paper slightly reliable, but needs further investigation with more data.

At the time of writing, no works have been found to address the problem of offensive content detection in Arabizi text. In general, works targeting Arabizi are still few and interest in this field is emerging. Many of them in fact transliterate Arabizi into Arabic before performing their given task which opens the question, as Guellil et al. [43] point out, of whether it is really necessary to transliterate Arabizi and whether there are any downsides to performing the task on Arabizi text.

## Seminal and State-of-the-Art Work

In this section, we introduce works that are considered seminal and state-of-the-art to this work. The seminal paper is considered to be a previous work that introduced a significant change in the field of offensive content detection. The state-of-the-art paper is the work that is most similar to this work and to which we can compare our results.

Although SemEval releases tasks frequently, and in many languages, to address the issue of offensive text identification, they recently turned their focus to English and used the Offensive Language Identification Dataset (OLID). Zampieri et al. compiled this three-level hierarchical annotated dataset from a large dataset of English tweets so that they could satisfy the general need for a typology to characterize the different types of offenses [21]. This is considered the new seminal work in this task since their scheme became thereafter the gold standard for annotation of offensive language and it was carried to different languages and used in competitions. The data was collected from Twitter by searching for keywords related to offensive messages. A key challenge which was not resolved was obtaining a well-balanced dataset [21]. They also performed experiments with different machine learning models for each level so that they could set important baselines for future works. Models included a linear SVM trained on word unigrams, BiLSTM, and CNN (an input embedding layer (both using pre-trained FastText as well as updatable embeddings). Results were compared using macro-averaged F1-score to account for the data imbalance. It was found that for level A, B and C, the CNN had the highest macro-F1 score of 0.80, 0.69, and 0.47 respectively.

Mulki et al. [34] focus on the identification of hate speech specifically in the levantine dialect (Lebanese and Syrian). We consider this paper the most relevant state-of-the-art work since it is one of the more recent works on the problem of offensive text in Arabic and because we are using their dataset. They collected the **L**evantine **H**ate **S**peech and **AB**usive dataset (L-HSAB) from the Twitter timelines of politicians, activists and news sources and by querying certain keywords. They manually filtered symbols, digits, non-arabic characters and non-levantine tweets. They labelled the tweets according to the following scheme:

* **Normal**: no offensive content of any type.
* **Abusive**: tweets with offensive, aggressive or profane content.
* **Hate**: tweets that are **Abusive** and also direct that offense towards a person or a group of people based on identity (race, religion, nationality…).

The annotators were made aware of the context and background of these tweets (names of ethnic groups, political parties…). Samples that were labeled differently by each of the three annotators were discarded. With the dataset ready, they did some data exploration to discover the most frequent words occurring under each class and which words are significant to each class (by comparing them to their occurrence in the Normal class). They also evaluated the quality of their labelling using statistical annotator agreement measures (PRAM, Cohen’s kappa and Krippendorff’s alpha). It is shown that more disagreement is encountered in the **Hate** class since it relies on the annotator’s background and prior knowledge. They performed two classification experiments: a binary classification (**Abusive** and **Hate** merged into one class) and a multiclass classification. They removed stopwords and experimented with several n-gram schemes and used term frequency (TF) weighing. The classifiers used are SVM and NB. For the binary case, NB achieved an F1-score of 89.6% and the SVM achieved a score of 82%. In the multiclass case, the scores were 74.4% and 66.8%.

## Conclusion and Comparison Against Previous Work

In summary, the literature is rich in works that have attempted to classify offensive text in social media. For arabic, many works have tackled this task targeting different arabic dialects using different annotation layouts and classification methodologies. However, none have done so for text in Arabizi. In fact, research in Arabizi is emerging and many works rely on converting the text to Arabic rather than processing the text in its Arabizi form. Table I shows a comparison of the most relevant works to the work being done in this paper.

Table I. Comparison with related work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Ref** | **Problem** | **Target classes** | **Language** | **Models** |
| 21 | Offensive content detection | OFF/NOT,  TIN/UNT,  IND/GRP/OTH | English | SVM,  BiLSTM,  CNN |
| 34 | Offensive content detection | Normal, abuse, hate | Levantine Arabic | SVM, NB |
| 37 | Sentiment analysis | Positive/negative/neutral | Arabizi transliterated to Arabic | SVM, NB |
| **This work** | **Offensive content detection** | **Normal, abuse, hate** | **Arabizi** | **BERT, RNN-CNN** |

# Methodology

The aim of this work is to perform offensive content identification on Arabizi tweets by classifying them into either **Normal**, **Abuse** or **Hate**. We therefore propose a methodology for preparing the data and developing and testing models to perform this classification.

## Overview

As a first step, the L-HSAB dataset was preprocessed and cleaned in order to remove noise and normalize the text. Next, the dataset was transliterated into Arabizi using rule-based mappings.

Then, the following models were developed for classification: NB, BiLSTM, CNN-LSTM and BERT. For NB, text is represented as Term Frequency-Inverse Document Frequency (TF-IDF) vectors. For the BiLSTM, CNN-LSTM and BERT models, text is represented as embeddings given by an embedding layer. The output representations of the deep models are fed into dense softmax layers to obtain the classification output.

Model development and tuning was done for the Arabic data first where different word embedding strategies were implemented. Once the Arabizi dataset was finalized, models were also developed to perform the classification task on Arabizi. However, since no pre-trained embedding models for Arabizi are available, embedding weights were trained from scratch. Fig. 2 shows the high level architecture of the proposed system.

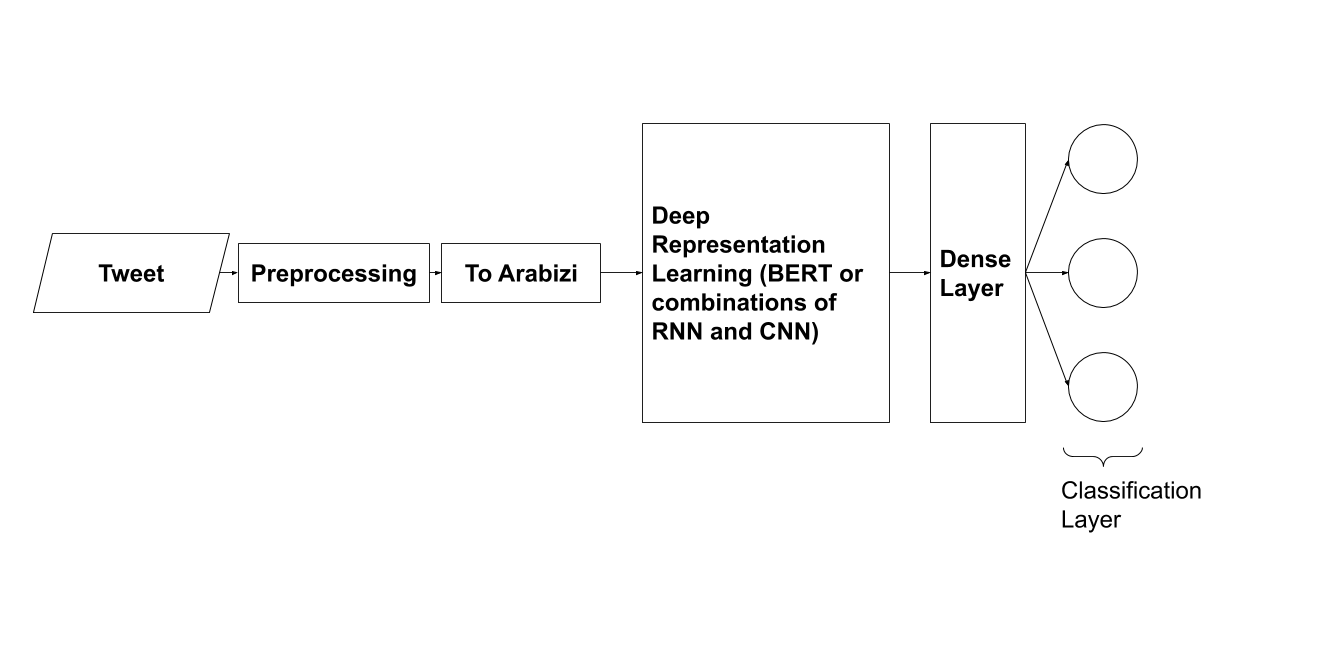


Fig. 2. System architecture of the proposed solution

Model performance is evaluated based on the F1-score, precision and recall. Accuracy was also computed but due to the uneven distribution of the classes, this metric would not reflect the actual performance of a model.

## Arabizi Transliteration

In order to effectively transliterate the given Arabic dataset, the first step which needs to be taken is to clean any noise in the text before so that the transliteration can be more effective. After doing so, the cleaned data can be transliterated into Arabizi by creating a set of mapping rules from Arabic letters into their corresponding latin letters or digits. Finally, in order to improve the quality of the data we iterated through the most frequently used words in the dataset and corrected any inconsistencies.

## Classification Models

In this section, the models used in the experiments are described.

### Naive Bayes

The Naive Bayes classifier has seen a lot of success in the field of text classification. In this work, this model is considered because it was the best performing model in the state-of-the-art [34] and therefore constitutes the baseline model. For this implementation, text was represented as TF-IDF vectors where each word of a tweet is given a TF-IDF score. TF is the frequency of occurrence of a word in a given tweet (document) and IDF is the inverse of the frequency of a word across different tweets which tries to capture common words across documents. In this case for example, certain inflammatory words may be common in the **Abuse** class.

### BiLSTM

Recurrent neural networks are a natural choice for working with text due to its inherent sequential nature. As more words from a sentence are read its meaning and purpose becomes clearer. It is also very useful to be able to use information from subsequent words as this enhances the understanding of the preceding content which calls for bidirectional networks. For this reason, bidirectional GRUs (gated recurrent units) and LSTMs were experimented with. LSTM networks were found to have superior performance and therefore the GRUs were dropped.

### CNN-LSTM

This combination of convolutional layers and LSTM layers is shown to improve recall when compared to other models in detecting offensive language on Arabic social media [41]. Input text is first fed to an embedding layer and its output is represented as a grid of dimensions 100 × 280 (Embedding size × Max comment length). This grid is then fed into the convolutional layer, where the network is able to capture local/short-term dependencies and positional relations between neighboring words. The low level feature outputs of the convolutional layers are then fed into the LSTM layers which learn global/long-term dependencies and outputs the high-level features of the sentences which are then used for classification.

### BERT

BERT (**B**i-directional **E**ncoder **R**epresentations from **T**ransformers) is a state-of-the-art NLP technique developed by Google in 2017 [44]. BERT uses Transformer, which is an encoder-decoder mechanism that contains self-attention. However, BERT drops the decoder portion of Transformers, only using the encoder, as this is the only part needed to generate a language model. The input to the encoder is the sum of the sentences’ token, segment, and position embeddings which designate the words or tokens in a sentence via a numerical vector, the sentence that the token belongs to, and the position of a token with that sentence respectively. The token embeddings are done using BERT tokenizer, which is pretrained on multiple languages, including Arabic. BERT tokenizer uses the WordPiece model, which splits words outside its vocabulary to known subsections or tokens. The encoder consists of a multi-head self-attention layer, to extract strong contextual meaning from the words, a layer-wise normalization module, and a position-wise feed forward layer with ReLU activations, with residual blocks being used between them. During training, BERT adopts a Masked Language Model (MLM) approach which allows deeper context to be extracted. MLM randomly hides 15% of the tokens and tries to predict them by concatenating a softmax layer to Transformer, outputting a probability for each word available in the dictionary. 80 % of these masked tokens will be substituted with a [MASK] token, 10 % will be substituted by a random token, while the rest of the tokens remain unchanged. In addition, BERT utilizes Next Sentence Prediction (NSP) during training, where two sentences are classified as either coming after one another or not to better understand the relationship between sentences. Similar to MLM, this is done by adding a softmax layer to the encoder.

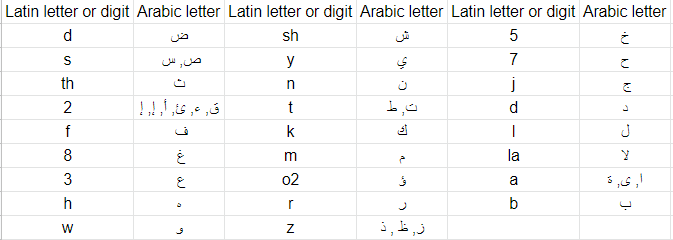
# Experiments and Results

## Dataset

The L-HSAB dataset was collected by Mulki et al. [34] from Twitter. The search was carried between March 2018 and february 2019 and was geographically constrained to the Levant area. After cleaning (removing symbols and non-arabic characters) and filtering irrelevant samples (duplicates, non-textual or promoted tweets) they achieved a dataset of 5846 tweets which were labeled by three annotators as either **Normal**, **Abusive** or **Hateful**.

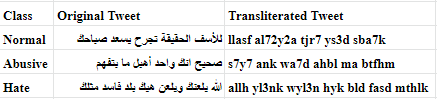
The first step taken in handling this dataset is to clean any noise in the text. As such, we removed all punctuation, emoji, interjections “ههههههه” (*laughter*), gibberish and normalized elongated words (“حمااااار” into “حمار”). Next, the cleaned data was transliterated into Arabizi which is the main target language of this work. The transliteration algorithm developed relies on a set of mapping rules from Arabic letters into their corresponding latin letters or digits as shown in Table II.

Table II. Arabizi Mapping Rules



Then, we went through the most frequently used words and corrected the output where our mapping algorithm did not perform well. For example, the word “الوزير” (*minister*), which was used a lot in the dataset, was being transliterated to “alozyr” instead of “alwazer”. Examples of the achieved data are shown in Table III.

Table III. Transliterated examples



Exploring the class distribution in the dataset reveals that the classes are unbalanced and that the class frequency in descending order is: **Normal**, **Abusive** then **Hate**. Fig. 3 shows the exact number of occurrences for each class.

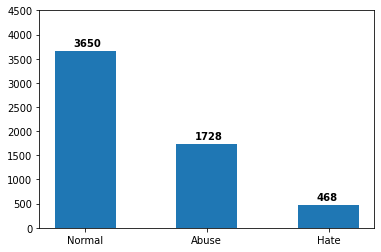


Fig. 3. Class distribution in L-HSAB

The labels were converted to a one-hot encoded presentation. For testing, 20% of the data was set aside. The split was performed in a way to maintain the same class distribution in both sets.

## Experimental Setup

Experiments were performed on both the Arabic data and the created Arabizi data using Google Colab GPU-accelerated virtual machines and several Python libraries (regular expressions, gensim, Scikit-learn, Keras, Pytorch and Simple Transformers). Class weights were incorporated into the learning objectives in all experiments as it was found to be an adequate strategy to account for class imbalance. These were calculated using Equation (1).

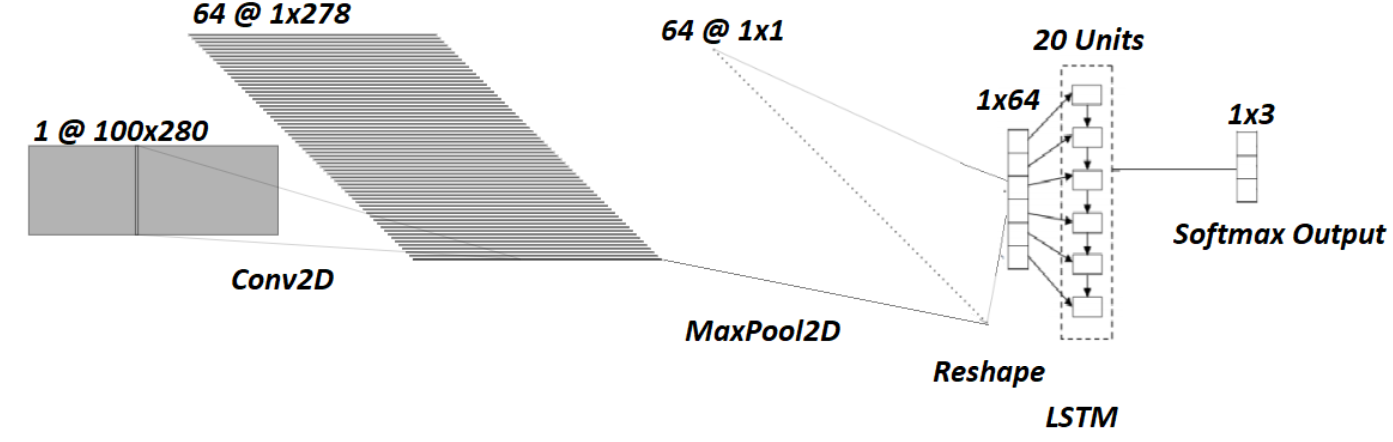
(1)

For training the NB, TF-IDF vectors of the data were used as features. It should be noted that our NB implementation does not fully reproduce the results of Mulki et al. [34] because they used a different train/test split: they had different class proportions in their splits and since the process is randomized it could not be replicated.

For Arabic, training from scratch of embedding weights or transfer learning from AraVec [48] (which is a Word2Vec-like network trained on Arabic tweets) and fine-tuning were implemented for the BiLSTM and CNN-LSTM models. In the case of Arabizi, the embedding layers were trained from scratch as no pre-trained Arabizi embeddings were found at the time of writing. For the BERT implementation, the base multilingual model was loaded and fine-tuned for the current task in both Arabic and Arabizi cases.

To find the optimal BiLSTM architecture, a randomized search and cross-validation was performed over the following hyperparameters: number of training epochs, mini-batch size, number of LSTM layers, number of LSTM units, number of dense layers, number of dense neurons and dropout rate. For Arabic, the selected architecture consisted of an embedding layer followed by two BiLSTM layers and a dense softmax layer. Dropout regularization with a rate of 0.8 was applied to the entire network. This model was trained with a mini-batch size of 100 for 60 epochs using the Adam optimizer. For Arabizi, a similar hyperparameter search was carried to find a similar optimal configuration as before but with a dropout rate of 0.5 and fewer training epochs (40).

In order to find the best layer configuration for the CNN-LSTM, the following methodology was used. Starting from the same architecture of the model used by Mohaouchane et al. [41], we experimented with various combinations of convolutional layers and LSTM layers until finally converging to an architecture, similar to that of Mohaouchane et al. [41], as seen in Fig. 4.

Fig. 4. CNN-LSTM Architecture Used

The architecture used consists of an embedding layer, a convolutional layer with filters of size 100×3, followed by a max pooling layer and the first dropout layer (dropout rate = 0.1). This is then reshaped and fed into an LSTM layer, with 20 units, followed by the second dropout layer (dropout rate = 0.7) and finally into the dense softmax output layer. All the values of the hyperparameters mentioned above were obtained through a randomized search and cross-validation. The Adam optimizer with a categorical cross-entropy loss function was used for training.

For tuning BERT to this task, a softmax layer is added on top of the encoder that outputs three different probabilities regarding whether a tweet is **Normal**, **Abusive** or **Hateful** as shown in Fig. 5.

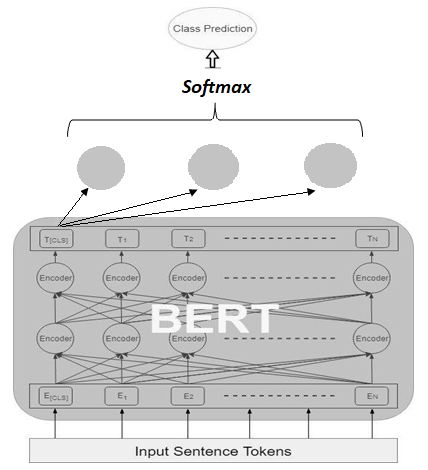


Fig. 5. BERT Architecture (diagram adapted from [45])

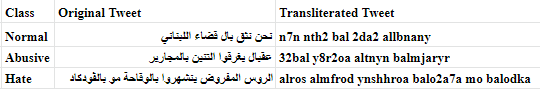
Since BERT’s multilingual model has been trained on Arabic, the pretrained model is loaded and fine-tuned to classify the Arabic tweets. The BERT-base version is used which contains 12 Transformer blocks, 12 attention heads, 768 hidden layer size, and 108M parameters to learn. It was trained for 10 epochs, using 80/20 training-validation split. However, for Arabizi, a pretrained BERT model is not available. Therefore, given the labels and using 10 epochs, BERT was trained on the new language.

## *Results and Discussion*

### Transliteration Results

After achieving the Arabizi dataset, we looked through a sample of our dataset (200 tweets) and noticed that roughly 35% of our dataset is in MSA. This explains why our mapping algorithm did not perform perfectly on that portion of our dataset. For example in words spelled with “ز” and “ذ” would usually sound like we are always using “ز” and not “ذ” in the spoken Arabic language. More examples are shown in Table IV. Also, to perfect the transliteration between Arabic and Arabizi we would also have to rely on diacritics which would make the Arabizi much more understandable [46]. For example, “يلعنك” (*damn you*) would be mapped to “yl3nk”, but if diacritics where used it would be transliterated to “yel3anak” which is more understandable than the former one. We tried several available diacritizers (Farasa, Shakkala) to add the diacritics on the tweets in the dataset, but it performed poorly on the tweets that are in the Levantine colloquial dialect, which are almost 65% of our dataset.

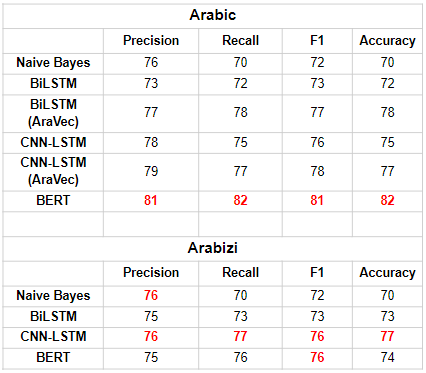
Table IV. Examples of flawed transliterations



### Classification Results

After model training and tuning , each model was tested on the 20% unseen data. The evaluation criteria are the precision, recall and F1-score. Since the classes are unbalanced, the class-weighted averages of these scores are considered. These are shown in Table V.

Table V. Classification test scores



The first thing to note is that for Arabic, our results outperform the state-of-the-art [34] with BERT achieving clear superiority in performance. The other thing to note in Arabic is that pre-trained AraVec embeddings improve results in both BiLSTM and CNN-LSTM models. This is expected since we found AraVec embeddings to capture the vocabulary of this dataset in the right context. For example the embedding for the word “خنزير” (*pig*) which is used as an insult is similar to other insults such as “نجس” (*filthy*) and “بغل” (*mule*) which are both insults instead of being similar to vocabulary of animals.

In Arabizi, the highest scores are achieved by the CNN-LSTM closely followed by BERT. The reason why BERT did not have an edge here is because it was pre-trained on multilingual Wikipedia articles which would not have had Arabizi text. Pires et al. [47] also demonstrate that BERT multilingual does not transfer well to transliterated languages in another case study (Hindi transliterated to Latin script). Therefore in this case, the CNN-LSTM is the best model especially with its higher recall which means that it’s more likely to catch offensive samples.

### Further Discussion

A noteworthy observation which can be made from the results obtained from the Arabic models, is that when the AraVec embeddings were used, the models were more likely to misclassify **Hate** as **Abusive** rather than **Normal** as seen in the confusion matrices (Fig. 6).

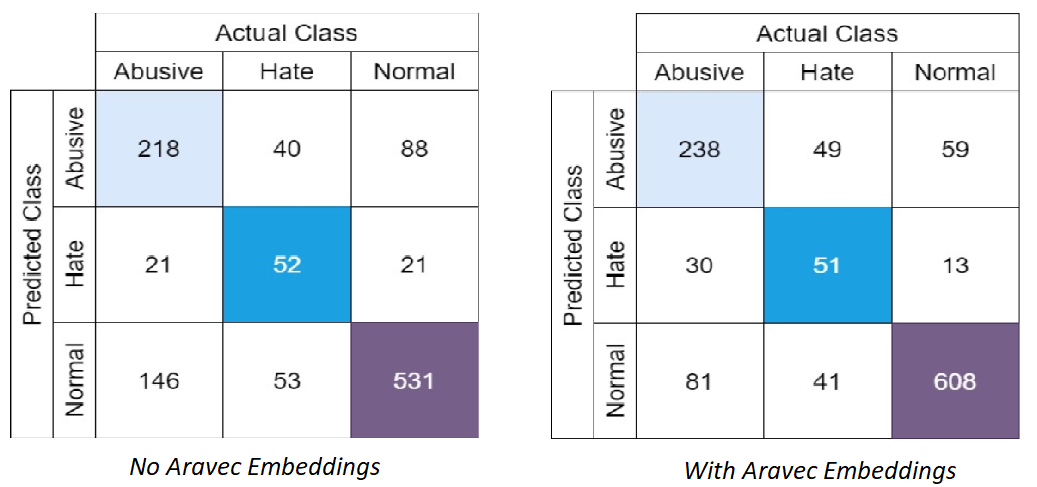
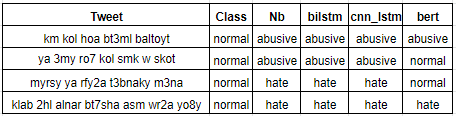


Fig. 6. Confusion matrices of CNN-LSTM for Arabic

This is due to the AraVec embeddings helping the model understand the similarity between the words being used, an observation also made by Davidson et al. [16]. Therefore we can attribute some of this misclassification as being due to the use of similar words in the **Hate** and **Abusive** examples which can be explained further by the ambiguity in the labeling that was done. In the case where the embedding layers were trained, rather than using AraVec, we notice that the misclassifications were more random. As a result we can expect to achieve higher recall for our Arabizi models if pretrained and well defined Arabizi embeddings were available.

There are also several notable cases in Arabizi where the models were misled by the presence of certain vocabulary. Table VI illustrates a few examples of such misclassifications.

Table VI. Misclassifications due to misleading vocabulary



We can see from the examples above that our models all tended to misclassify some **Normal** tweets as **Abusive** or **Hate**. This is because these tweets have words that are highly occurring in these classes such as “ya” and “kol” which often precede curse words. This pattern also shows itself in many other examples, such as in the misclassification of the **Normal** tweet “*ah ya sho2yry ant*” as **Abusive** since the words ‘ant’ and ‘ya’ are the two most common words in the **Abusive** class. The first three samples in Table VI present similar cases. Some tweets also contain offensive vocabulary but do not really convey an offensive message. For example, the last tweet in Table VI has the curse word “klab” (*dogs*) but the commenter is actually sarcastically making fun of the terminology and shows no offensive intentions.

# Conclusion

In this paper, we tackled the problem of transliterating Arabic tweets, from social media, into Arabizi and identifying them as **Normal**, **Hateful**, or **Abusive** comments. Arabizi is widely used among the Arab youth and there is thus a need to efficiently regulate Arabizi social media content in order to protect users from any harmful posts.

It was found that proper diacritization is needed in order to achieve more realistic results from the transliteration. We also compared the performance of four different models in this task. These models were the Naive Bayes, BiLSTM, CNN-LSTM, and BERT. We found that BERT achieved state-of-the-art performance in Arabic but not in Arabizi because multilingual pretraining does not transfer well to languages not pre-trained on. We also found that the CNN-LSTM had the best results, and recall, for Arabizi since it is able to effectively capture both the short term and long term dependencies from the word n-grams. Furthermore, we did not find any disadvantages or handicaps in doing classification on Arabizi compared to Arabic since the Arabizi models performed reasonably well.

Future work which will help increase the accuracy of these models include collecting a large Arabizi dataset directly from social media, in order to ensure realistic results. Developing and making use of embedding models specific to Arabizi will also lead to more accurate results in the future.

##### References

1. PR Choice page", Npr.org, 2020. [Online]. Available: https://www.npr.org/2019/10/08/768319934/senate-report-russians-used-used-social-media-mostly-to-target-race-in-2016.
2. "Facebook Security Breach Exposes Accounts of 50 Million Users", Nytimes.com, 2020. [Online]. Available: https://www.nytimes.com/2018/09/28/technology/facebook-hack-data-breach.html.
3. "Social media outpaces print newspapers in the U.S. as a news source", Pew Research Center,2020. [Online]. Available: https://www.pewresearch.org/fact-tank/2018/12/10/social-media-outpaces-print-newspapers-in-the-u-s-as-a-news-source/.
4. "Is Social Media Replacing Traditional Advertising?", *Cobizmag.com*, 2020. [Online]. Available: <https://www.cobizmag.com/Business-Insights/Is-Social-Media-Replacing-Traditional-Advertising/>.
5. "Cyberbullying Statistics", 2019. [Online]. Available: https://enough.org/stats\_cyberbullying.
6. "Council on Foreign Relations", 2019. [Online]. Available: <https://www.cfr.org/backgrounder/hate-speech-social-media-global-comparisons>.
7. "Instagram is using AI to stop people from posting abusive comments", MIT Technology Review, 2020. [Online]. Available: https://www.technologyreview.com/f/613930/instagram-is-using-ai-to-stop-people-posting-abusive-comments/.
8. U.S. (2020). Facebook’s flood of languages leave it struggling to monitor content. [online] Available at: <https://www.reuters.com/article/us-facebook-languages-insight/facebooks-flood-of-languages-leave-it-struggling-to-monitor-content-idUSKCN1RZ0DW>
9. V. Stoyanov and N. Fazil Ayan, "Under the hood: Multilingual embeddings", Ai.facebook.com, 2018. [Online]. Available: <https://ai.facebook.com/blog/under-the-hood-multilingual-embeddings/>.
10. "The importance of the Arabic language in social media | Seven Media", Sevenmedia.ae. [Online]. Available: <https://sevenmedia.ae/the-importance-of-the-arabic-language-in-social-media/>.
11. A. Al-Hassan and H. Al-Dossari, “DETECTION OF HATE SPEECH IN SOCIAL NETWORKS: A SURVEY ON MULTILINGUAL CORPUS,” in Computer Science & Information Technology(CS & IT), 2019, pp. 83–100, doi: 10.5121/csit.2019.90208.
12. B. Haidar, M. Chamoun, and A. Serhrouchni, “A Multilingual System for Cyberbullying Detection: Arabic Content Detection using Machine Learning,” Adv. sci. technol. eng. syst. j., vol. 2, no. 6, pp. 275–284, Dec. 2017, doi: 10.25046/aj020634.
13. "Arabic in Latin Script: Lost in Arabic - Qantara.de", Qantara.de - Dialogue with the Islamic World. [Online]. Available: https://en.qantara.de/content/arabic-in-latin-script-lost-in-arabic.
14. H. Yenala, A. Jhanwar, M. K. Chinnakotla, and J. Goyal, “Deep learning for detecting inappropriate content in text,” Int J Data Sci Anal, vol. 6, no. 4, pp. 273–286, Dec. 2018, doi: 10.1007/s41060-017-0088-4.
15. G. Xiang, B. Fan, L. Wang, J. Hong and C. Rose, "Detecting offensive tweets via topical feature discovery over a large scale twitter corpus", Proceedings of the 21st ACM international conference on Information and knowledge management - CIKM '12, 2012. Available: 10.1145/2396761.2398556.
16. T. Davidson, D. Warmsley, M. Macy, and I. Weber, “Automated Hate Speech Detection and the Problem of Offensive Language,” arXiv:1703.04009 [cs], Mar. 2017.
17. Z. Zhang, D. Robinson, and J. Tepper, “Detecting Hate Speech on Twitter Using a Convolution-GRU Based Deep Neural Network,” in The Semantic Web, vol. 10843, A. Gangemi, R. Navigli, M.-E. Vidal, P. Hitzler, R. Troncy, L. Hollink, A. Tordai, and M. Alam, Eds. Cham: Springer International Publishing, 2018, pp. 745–760.
18. A. Gaydhani, V. Doma, S. Kendre, and L. Bhagwat, “Detecting Hate Speech and Offensive Language on Twitter using Machine Learning: An N-gram and TFIDF based Approach,” arXiv:1809.08651 [cs], Sep. 2018.
19. G. K. Pitsilis, H. Ramampiaro, and H. Langseth, “Detecting Offensive Language in Tweets Using Deep Learning,” Appl Int/242-y.
20. V. S. Chavan and Shylaja S S, “Machine learning approach for detection of cyber-aggressive comments by peers on social media network,” in 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 2015, pp. 2354–2358.
21. [1]M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, “Predicting the Type and Target of Offensive Posts in Social Media,” arXiv:1902.09666 [cs], Apr. 2019, Available: http://arxiv.org/abs/1902.09666.
22. M. Zampieri, S. Malmasi, P. Nakov, S. Rosenthal, N. Farra, and R. Kumar, “SemEval-2019 Task 6: Identifying and Categorizing Offensive Language in Social Media (OffensEval),” arXiv:1903.08983 [cs], Apr. 2019.
23. S. Sapora, B. Lazarescu, and C. Lolov, “Absit invidia verbo: Comparing Deep Learning methods for offensive language,” arXiv:1903.05929 [cs], Mar. 2019.
24. H. Uglow, M. Zlocha, and S. Zmyślony, “An Exploration of State-of-the-art Methods for Offensive Language Detection,” arXiv:1903.07445 [cs, stat], Mar. 2019.
25. A. Cambray and N. Podsadowski, “Bidirectional Recurrent Models for Offensive Tweet Classification,” arXiv:1903.08808 [cs], Mar. 2019.
26. N. Frisiani, A. Laignelet, and B. Güler, “Combination of multiple Deep Learning architectures for Offensive Language Detection in Tweets,” arXiv:1903.08734 [cs, stat], Mar. 2019.
27. R. Ong, “Offensive Language Analysis using Deep Learning Architecture,” arXiv:1903.05280 [cs], Mar. 2019.
28. A.-B. Puiu and A.-O. Brabete, “Towards NLP with Deep Learning: Convolutional Neural Networks and Recurrent Neural Networks for Offensive Language Identification in Social Media,” arXiv:1903.00665 [cs], Apr. 2019.
29. H. Bansal, D. Nagel, and A. Soloveva, “HAD-Tübingen at SemEval-2019 Task 6: Deep Learning Analysis of Offensive Language on Twitter: Identification and Categorization,” in Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, Minnesota, USA, 2019, pp. 622–627, doi: 10.18653/v1/S19-2111.
30. J. Pavlopoulos, N. Thain, L. Dixon, and I. Androutsopoulos, “ConvAI at SemEval-2019 Task 6: Offensive Language Identification and Categorization with Perspective and BERT,” in Proceedings of the 13th International Workshop on Semantic Evaluation, Minneapolis, Minnesota, USA, 2019, pp. 571–576, doi: 10.18653/v1/S19-2102.
31. R. Pawar and R. R. Raje, “Multilingual Cyberbullying Detection System,” in 2019 IEEE International Conference on Electro Information Technology (EIT), Brookings, SD, USA, 2019, pp. 040–044, doi: 10.1109/EIT.2019.8833846.
32. P. Saha, B. Mathew, P. Goyal, and A. Mukherjee, “HateMonitors: Language Agnostic Abuse Detection in Social Media,” arXiv:1909.12642 [cs], Sep. 2019.
33. T. Ranasinghe, M. Zampieri, and H. Hettiarachchi, “BRUMS at HASOC 2019: Deep Learning Models for Multilingual Hate Speech and Oﬀensive Language Identiﬁcation,” p. 9, 2019.
34. H. Mulki, H. Haddad, C. Bechikh Ali, and H. Alshabani, “L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language,” in Proceedings of the Third Workshop on Abusive Language Online, Florence, Italy, 2019, pp. 111–118, doi: 10.18653/v1/W19-3512.
35. H. Mubarak, K. Darwish, and W. Magdy, “Abusive Language Detection on Arabic Social Media,” in Proceedings of the First Workshop on Abusive Language Online, Vancouver, BC, Canada, 2017, pp. 52–56, doi: 10.18653/v1/W17-3008.
36. N. Ousidhoum, Z. Lin, H. Zhang, Y. Song, and D.-Y. Yeung, “Multilingual and Multi-Aspect Hate Speech Analysis,” arXiv:1908.11049 [cs], Aug. 2019.
37. R. M. Duwairi, M. Alfaqeh, M. Wardat, and A. Alrabadi, “Sentiment analysis for Arabizi text,” in 2016 7th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2016, pp. 127–132, doi: 10.1109/IACS.2016.7476098.
38. R. M. Duwairi, R. Marji, N. Sha’ban, and S. Rushaidat, “Sentiment Analysis in Arabic tweets,” in 2014 5th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2014, pp. 1–6, doi: 10.1109/IACS.2014.6841964.
39. B. Haidar, M. Chamoun, and A. Serhrouchni, “Arabic Cyberbullying Detection: Using Deep Learning,” in 2018 7th International Conference on Computer and Communication Engineering (ICCCE), Kuala Lumpur, 2018, pp. 284–289, doi: 10.1109/ICCCE.2018.8539303.
40. B. Haidar, M. Chamoun, and A. Serhrouchni, “Arabic Cyberbullying Detection: Enhancing Performance by Using Ensemble Machine Learning,” in 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, 2019, pp. 323–327, doi: 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00074.
41. H. Mohaouchane, A. Mourhir, and N. S. Nikolov, “Detecting Offensive Language on Arabic Social Media Using Deep Learning,” in 2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS), Granada, Spain, 2019, pp. 466–471, doi: 10.1109/SNAMS.2019.8931839.
42. D. Mouheb, R. Ismail, S. A. Qaraghuli, Z. A. Aghbari, and I. Kamel, “Detection of Offensive Messages in Arabic Social Media Communications,” in 2018 International Conference on Innovations in Information Technology (IIT), Al Ain, 2018, pp. 24–29, doi: 10.1109/INNOVATIONS.2018.8606030.
43. I. Guellil, H. Saâdane, F. Azouaou, B. Gueni, and D. Nouvel, “Arabic natural language processing: An overview,” Journal of King Saud University - Computer and Information Sciences, p. S1319157818310553, Feb. 2019, doi: 10.1016/j.jksuci.2019.02.006.
44. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. “Attention is all you need”. arXiv:1706.03762 [cs], 2017.
45. ElJundi, Obeida, et al. "hULMonA: The Universal Language Model in Arabic.". arXiv:2003.00104v2 [cs], 2019.
46. United Nations, "Romanization System from Arabic letters to Latinized letters", UNSTATS. New York, United Nations Publications, 2007. Available: <https://unstats.un.org/unsd/geoinfo/UNGEGN/docs/11th-uncsgn-docs/E_Conf.105_137_CRP.137_14_Romanization%20System%20from%20Arabic%20%20letters%20to%20Latinized%20%20%20letters%202007%20-%20ENGLISH.pdf>.
47. T. Pires, E. Schlinger, and D. Garrette, “How multilingual is Multilingual BERT?,” arXiv:1906.01502 [cs], Jun. 2019, [Online]. Available: <http://arxiv.org/abs/1906.01502>.
48. A. Soliman, K. Eissa and S. El-Beltagy, "AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP", Procedia Computer Science, vol. 117, pp. 256-265, 2017. Available: 10.1016/j.procs.2017.10.117.

1. https://github.com/Hala-Mulki/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset [↑](#footnote-ref-0)