The Islamic University of Gaza
Deanship of Research and Graduate Studies
Faculty of Information Technology
Master of Information Technology

الجامعـــــــــــة الإســــلاميــة بغـزة
عمادة البحث العلمي والدراسات العليا
كـليـــــة تكنولوجـــــيا المعلـــومـــات
ماجســــــــــتير تكنـولوجيا المعلومات

Detecting and Masking Arabic Hate Speech Texts Using Deep Learning Techniques

اكتشاف وإخفاء النصوص العربية التي تتضمن خطاب كراهية باستخدام تقنيات التعلم العميق

By

Salam Thabet Doghmash

Supervised by

Dr. Motaz Saad

Assistant Professor of Computer Science

**A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of information technology in the Islamic University of Gaza**

June/2022

إقـــــرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

Detecting and Masking Arabic Hate Speech Texts Using Deep Learning Techniques

اكتشاف وإخفاء النصوص العربية التي تتضمن خطاب كراهية باستخدام تقنيات التعلم العميق

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، وأن هذه الرسالة ككل أو أي جزء منها لم يقدم من قبل الاخرين لنيل درجة أو لقب علمي أو بحثي لدى أي مؤسسة تعليمية أو بحثية أخرى.

Declaration

I understand the nature of plagiarism, and I am aware of the University’s policy on this.

The work provided in this thesis, unless otherwise referenced, is the researcher's own work, and has not been submitted by others elsewhere for any other degree or qualification.

Student's name:	سلام ثابت دغمش	اسم الطالب:
Signature:	Salam.Th.D	التوقيع:
Date:	2022/4/6	التاريخ:

نتيجة الحكم على أطروحة الماجستير

Abstract

The world is going through a difficult period in terms of the spread of hate speech. The boost of social media and the increased migration wave recently helped to spread hate speech widely. The need for appropriate methods to address hate speech detection is indisputable. This can be done by detecting hate speech to help content moderators to moderate content.

In this research, we address two problems: The first one is to detect hate speech in Arabic text, and the second is to clean a given text from hate speech. The meaning of cleaning here is replacing each bad word with stars based on the number of letters for each word.

In the hate speech detection problem, we conduct several experiments using deep learning and transformers to determine the best model in terms of the F1 score. Regarding the text cleaning problem, we consider the problem of cleaning dirty text as a machine translation problem, where the input is a sentence containing dirty text and the output is the same sentence with masking the dirty text.

The presented methods achieve the best model in hate speech detection with a 92% Macro F1 score and 95% accuracy. These results outperform the best results that are published in the SemEval 2020 shared task. Regarding the text cleaning experiment, the best result in the hate speech masking model reached 0.3 in BLEU score with 1-gram, which is a good result compared with the state of the art machine translation systems.

Keywords: NLP; Deep learning; Hate speech detection; Hate speech masking; Transformer; Machine translation.

ملخص الدراسة

يمر العالم بفترة صعبة من حيث انتشار خطاب الكراهية. ساعد انتشار وسائل التواصل الاجتماعي وموجة الهجرة المتزايدة مؤخرًا على انتشار خطاب الكراهية على نطاق واسع. لا جدال في الحاجة إلى أساليب مناسبة للتصدي والكشف عن الكلام الذي يحض على الكراهية. يمكن القيام بذلك لمساعدة مشرفي المحتوى على تعديل المحتوى.

نتناول في هذا البحث مشكلتين: الأولى هي الكشف عن خطاب الكراهية في النص العربي، والثانية هي تنظيف نص معين من خطاب الكراهية. معنى التنظيف هنا هو استبدال كل كلمة سيئة بنجوم بناءً على عدد الحروف لكل كلمة.

في مشكلة اكتشاف الكلام الذي يحض على الكراهية، نجري العديد من التجارب باستخدام الشبكات والمحولات العصبية العميقة لتحديد أفضل نموذج من حيث Macro-F1. فيما يتعلق بمشكلة تنظيف النص، فإننا نعتبر مشكلة تنظيف النص الغير مناسب كمشكلة ترجمة آلية، حيث يكون المدخل جملة تتضمن نصًا سيئًا ويكون الناتج نفس الجملة مع عمل إخفاء للنصوص السئية.

الأساليب المقترحة تحقق أفضل نموذج في اكتشاف الكلام الذي يحض على الكراهية حيث وصلت نسبة Macro F1-score 92% و دقة النموذج وصلت الى 95٪. تتفوق هذه النتائج على أفضل النتائج التي تم نشرها في مهمة SemEval 2020 المشتركة. أما فيما يتعلق بتجربة إخفاء الكلام الذي يحض على الكراهية، وصلت أفضل نتيجة 0.3 في درجة BLEU 1-gram، وهي نتيجة جيدة مقارنة بأحدث أنظمة الترجمة الآلية.

الكلمات المفتاحية: معالجة اللغة الطبيعية؛ التعلم عميق؛ الكشف عن الكلام الذي يحض على الكراهية؛ إخفاء الكلام الذي يحض على الكراهية؛ المحولات؛ الترجمة الآلية.

Acknowledgment

Thanks to **Allah** for giving me the ability to finish this thesis.

Thanks to my supervisor, **Dr Motaz Saad**, for his support, assistance and invaluable advice in completing my thesis.

My special thanks to my first supporters in all stages of my life to my **Father**, and my **Mother**.

Thanks to my **brothers**, **sisters**, and **friends** for supporting me.

Special thanks to **Haneen Al shurafaa** for her efforts in hate speech masking.

Salam Th. Doghmash

Table of Contents

List of Tables

List of Figures

List of Abbreviations

BERT	Bidirectional Encoder Representation from Transformers
BLEU	BiLingual Evaluation Understudy
CNN	Convolutional Neural Network
DL	Deep learning
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
LSTM	Long-term short-term memory
ML	Machine learning
MT	Machine translation
NLP	Natural Language Processing
NB	Naive Bayes
RNN	Recurrent Neural Network
SVM	Support Vector Machines

Chapter 1
Introduction

Chapter 1
Introduction

Hate speech is a crime that has been on the rise in recent years, not just in reality but also on online platforms (Fortuna and Nunes 2018). Several factors contribute to this, increasing the number of people who use the internet, social media platforms that have helped it spread dramatically, and migration and growing conflicts around the world, which encourage people to express their opinions online, contributing to the spread of hate speech, thus contributing to the propagation of hate speech as well. Hate speech is defined as any expression that degrades an individual or group in terms of certain factors like a racial religious or national appearance that's a gender related sexual identity or other identities (Fortuna & Nunes, 2018) (Basile et al. 2019). The challenge of identifying hate speech and cleaning content has become increasingly important in recent years when hate speech has become the norm.

1.1 Background and context

The work on Arabic offensive language detection is relatively nascent with the high attention focused on English (Weber et al., 2019). Online communities, social media platforms, and technology companies have been investing heavily in ways to cope with offensive language to prevent abusive behavior in social media. In this thesis, we address two problems: the first one is Arabic hate speech detection by using different neural networks architectures, which applies it to a dataset from shared task SemEval-2020 for Arabic offensive language detection (Zampieri et al., 2020).

To the best of our knowledge, works in literature focus on hate speech detection, and there is no research which addresses the problem of cleaning such texts.

In the second problem in this thesis, we address the problem of cleaning offensive/hate speech texts. The idea is to consider the problem of cleaning dirty text as a machine translation problem, instead of providing the text in the source language to the MT system to produce the text in the target language, the input is dirty text that defined as the text that contains a feature of hate speech, sexual harassment, offensive, verbal abuse for disability or disease, gender bias, race bias, national origin, religion, and the output is clean text. In order to accomplish so, we need a parallel corpus. A parallel corpus is a collection of texts, each of which is translated into one or more other languages than the original (Teubert 1996). To build such corpus, we mainly need human workers to mask the dirty word from sentences by replacing each word with stars based on the number of letters for it.

1.2 statement of the problem

The richness of the Arabic morphology and the limited available resources makes hate speech detection tasks challenging. Thus, new experiments on detecting hate or dirty text are required.

In addition, the problem of Hate Speech cleaning has not been addressed in the literature before, so we pose this problem in this research and we propose a solution for it. The problem stems from the fact that existing methods focus on hate speech or dirty text detection only but they tend to handle detected texts by deleting them, There is a need for a new method to clean the sentences that contain dirty text, through mask the offensive and bad words to avoid hate speech.

1.3 Objectives

In this section, we present the main and specific objective that explain the main idea of our model.

1.3.1 Main objective

The main objective of this work is to choose the best deep learning model that helps to detect, and clean Arabic content from dirty text. This is done by giving the model a sentence that includes hate speech and the result is a sentence free of hate content.

1.3.2 Specific objectives

- Hate Speech Corpus acquisition
- Train and find the best parameters for existing models to detect hate speech and classify sentences into hate speech labels or not hate speech.
- Fine-Tune classification models to achieve the maximum model performance by selecting appropriate “hyperparameters”

- Evaluate hate speech detection model using F1-score
- Build a parallel corpus for using it with the machine translation model.
- Build a model by using machine translation for masking hate speech content from content.
- Tune the MT model to achieve the maximum model performance by selecting appropriate “values of hyperparameters”.
- Evaluate masking hate speech model using BLEU score (machine translation metric).

1.4 Importance of the research

Our models that detect and clean the content can be use in many applications such as:

1. Text editors that can suggest clean words when try write dirty text
2. Comments in social media
3. Chats
4. Remove or replace the bad word from movie Subtitles
5. Email content
6. Browsers extension that filter bad words when viewing a web page
7. Assist content moderators in quickly identifying potentially harmful content.

1.5 Scope and limitations

Scope of the work

- The model trained on specific data that is used in shared tasks for hate speech detection

Limitations of the work

- We use existing pre-processing methods to perform the proposed model without creating or custom new methods.
- We use existing pre-trained models for word embedding such as word2vec.
- Evaluation metrics considere in hate speech detection (Macro F1-score) as similar to those used in the literature for the Shared Task.

- If a sentence contains words associated with swearing, insults, or profanity, it very certainly be categorized as toxic, regardless of the writer's intent.

1.6 Overview of Thesis

The rest of the thesis is organized as follows. Chapter 2 gives an extensive overview of the main concepts in the work including deep learning, recurrent neural network, convolutional neural networks, natural language processing, machine translation, word embedding, pre-trained models, natural language inference. Chapter 3 shows related research as follows: works that construct new datasets for hate speech detection tasks, works that focus on generating hate/obscene/offensive terms automatically, works that use ML, DL, and transformer learning to detect hate speech, systematic/ analytical studies, and researches that are related to paraphrasing hate speech content are presented. Chapter 4 describes the methodology carried out to hate speech detection includes data preprocessing, different DL and transformer models. Also, display and discuss our experiments and evaluation for each model. Chapter 5 describes the methodology that is use for masking hate speech from content. Starting from the parallel corpus that we build and use in this experiment, then preprocessing steps, after that describe the model that is use for implementing hate speech masking. Finally, presents evaluation metrics, results and discussion. Finally, the conclusions of the research and future works are given in Chapter 6.

Chapter 2
Background

Background about the terminology of the problem and approaches to the solution

This chapter provides background information on several important related topics that used to detect and paraphrase hate speech content.

2.1 Hate speech

First, the word "hate" is understood to mean "extremely negative emotions and beliefs", group of individuals, or a specific member of that group, based on race, ethnicity, religion, gender, or sexual orientation. The term “hate speech” is used to describe all forms of racial hatred, xenophobia, anti-Semitism, or aggressive nationalism and ethnocentrism. It is intended to include any other form of hatred based on intolerance, including explicit intolerance (RING, 2013).

2.2 Artificial Intelligence

Artificial Intelligence (AI) is a branch of computer science concerned with automating activities associated with human thinking and activities such as decision-making, problem-solving, and learning (Russell & Norvig, 2003).

2.3 Deep Learning (DL)

DL involves an AI concept that helps in emulation the learning approaches that humans use to gain different types of knowledge (Ian Goodfellow, Yoshua Bengio, 2017). The method used to generate text includes DNN and is applied on different applications such as chatbots, text translation, etc. In our technique, we describe DL and how to use it. DL are a kind of neural network that's more complex and operates in additional than two layers. They use complex mathematical modeling in processing complex data (Simon Haykin, McMaster University, Hamilton, Ontario, 2005).

2.3.1 Recurrent neural network

A recurrent neural network (RNN) is any network that contains a cycle inside its network connections (Jurafsky et al. 2019) as shown in Figure 2.1. That is any network where the value of a unit is directly, or indirectly, dependent on previous outputs as an input. Such networks are difficult to reason about and train, despite their power. However, within the general class of recurrent networks, there are constrained architectures that have proven to be extremely effective when applied to written and spoken language (Jurafsky et al. 2019)

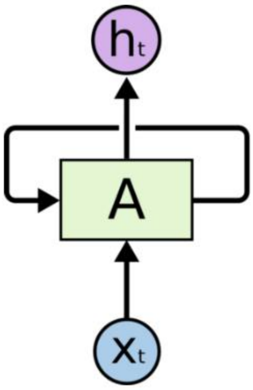

Figure (2.1): A recurrent neural network structure

RNN has two various architectures: LSTM and the GRU. The LSTM network has a similar architecture to the RNN, but it has more repeating modules and hence requires more operations (Aggarwal, 2018). The advantage of LSTM is that it remembers long-term dependence. Some of the operations that make the LSTM network remember more include; forget and update gate operation. The GRU does not have a cell state and the unit is and the unit is made up of only two gates, the update and reset gate, rather than three (Chernyatin & Ostroukhov, 1968).

RNNs work effectively in applications where sequential information is critical because the meaning could be misunderstood if sequential information is not used (Hettiarachchi & Ranasinghe, 2020). This model architecture contains four layers: an input layer (embedding layer), a two Bidirectional LSTM layer, and finally the output layer, as represented in Figure 2.2, the input tweets are fed into the embedding layer, which maps tweets tokens into a 300-dimensional real-valued vector. The embedding layer produces an output matrix, this output matrix is received by two parallel Bidirectional LSTM layers with 128, 64 units sequentially, the shape that is produced is passed to a dropout layer with a rate of 0.5 is used to reduce the overfitting problem. The final layer is a sigmoid layer that produces the final predictions.

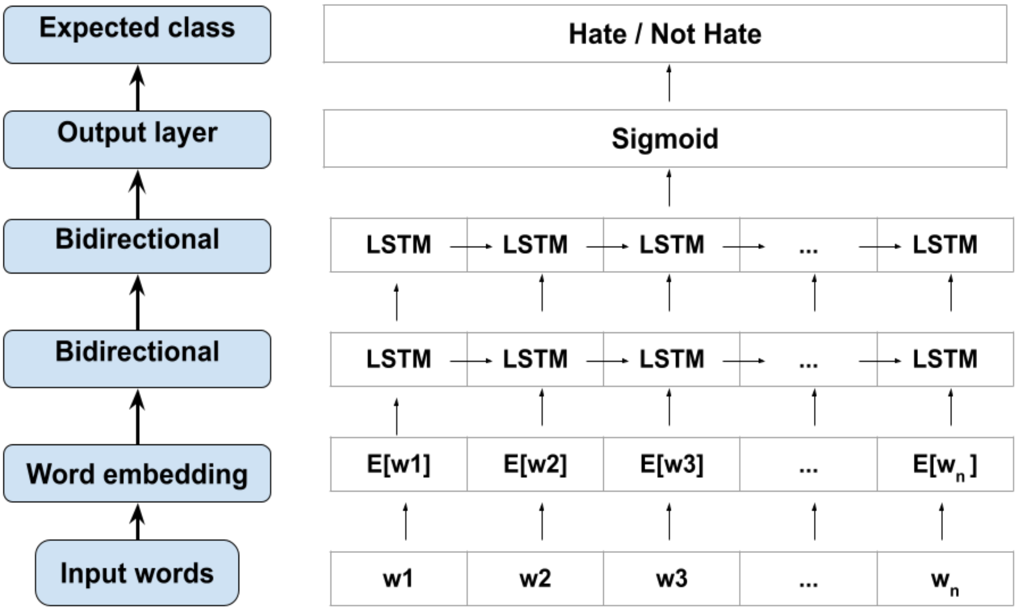

Figure (2.2): RNN architecture: Left represent RNN layers and right represent layers details

2.3.2 Convolutional neural network

A convolutional neural network (CNN) is a type of neural network that may use the internal structure of data, such as the 2D structure of image data, to improve its performance (Johnson & Zhang, 2015). CNN models, which were originally developed for computer vision, have now been proven to be useful for NLP, with outstanding results in semantic parsing, search query retrieval, sentence modeling, and other standard NLP tasks (Kim, 2014).

2.4 Natural Language Processing

The NLP refers to the automatic computational processing of human languages (Jurafsky and Martin 2019). This includes both algorithms that recive human-produced text as input, and algorithms that generate natural-looking text as outputs. For more than a decade, core NLP techniques were dominated by linear modeling approaches to supervised learning, centered around algorithms such as perceptron, linear vector support, and logistic regression, trained over very high dimensional yet very sparse feature vectors (Jurafsky et al. 2019).

Around 2014, the sector began to see some success in switching from such linear models over sparse inputs to nonlinear neural network models over dense inputs. A number of the neural network techniques are simple generalizations of the linear models and might be used as almost drop-in replacements for the linear classifiers. Others are more advanced, require a change of mindset, and present new modeling opportunities. specifically, a family of approaches supported by recurrent neural networks (RNNs) alleviates the reliance on the Markov Assumption that was prevalent in sequence models, allowing conditions on arbitrarily long sequences and producing effective feature extractors (Brownlee 2017). These advances result in breakthroughs in language modeling, automatic machine translations and other applications.

2.5 Machine translation

Machine translation is defined as translating sentences from one language into another automatically. The main resources used to train modern translation systems are bitexts known as parallel texts or bitexts. These are large text collections consisting of pairs of sentences from different languages that are translations of one another. Traditionally in Machine translation, the text being translated is referred to as the source and the translation output is called the target (Jurafsky et al. 2019)

2.6 Word embedding

A word embedding is a way to transform words in a text to numerical vectors so that standard machine learning algorithms that require vectors as numerical input can analyze them (Brownlee,2019). It represents words in a coordinate system in which related phrases are placed closer together, based on a corpus of relationships. Word embedding has two strategies: Static word embedding which always represent each word into exact representation regardless of the context where it occurs such as: Word2Vec, and contextualized language representation which aims to capture word semantics in different contexts to address the issue of the context-dependent nature of words (Popov et al., 2018).

2.6.1 Word2Vec

Word2Vec for Word Representation was released by Google in 2013. A neural network that processes text data. Word2Vec is not a single algorithm, but it does contain two learning models: the Continuous Bag of Words (CBOW) and Skipgram. CBOW predicts words based on their context, while Skipgram predicts context based on words (Ma & Zhang, 2015).

2.6.2 Wordpiece

Wordpiece embedding is designed for Google's speech recognition system for Asian languages such as Korean and Japanese. These languages have a large inventory of letters, homonyms, and little or no space between words. No or less space means that the text needs to be segmented. However, segmentation creates many out-of-vocabulary words (OOVs) in the model. Therefore, the WordPiece representation was created to automatically learn word-by-word from large amounts of data and not generate OOV. This technique for handling OOV is used in BERT. OOV is ignored in word2vec and GloVe, but the letter ngram representation of a word in FastText corrects OOV. WordPiece tokenization splits a word into different tokens. The most important words are retained and the other words are subdivided. To represent the continuity of the token, if the token is part of a priority token, two pounds (##) are prepended to the token. In this example, the word "playing" is split into "play" and "## ing". This means that when the WordPiece expression was trained, that word occurred less than any other word. Separated tokens are called subwords. Two special tokens

[CLS] and [SEP] have been added to represent the start of the data instance and the end of the statement (Alyafeai et al., 2021).

2.7 Pre-trained Models

A pre-trained model is a model that has already been trained to address a similar problem. Rather than creating a new model to address a similar problem, you start with a model that has already been trained on other problems (Gupta, 2017). In addition, there are pre-trained models by Research Laboratories that have great potential, where they trained on very large data so that the training process took a lot of time, and these Research Laboratories launched their trained models so that other researchers can benefit from them (Gluonnl, 2019)

2.8 Transformer

The Transformer model is a novel neural network for sequence modeling, The transformer was initially created to solve language translation problem (Vaswani et al., 2017). Since then, it has become the dominant architecture for natural language processing problems including text classification, summarization, language understanding, and other. Transformer model contains a stack of 6 encoder and decoder layers respectively (Vaswani et al., 2017). The encoder consiste of encoding layers that process the input iteratively one layer after another. The function of each encoder layer is to generate encodings that contain information about which parts of the inputs are relevant to each other and it passes its encodings to the next encoder layer as inputs (Vaswani et al., 2017). See Figure 2.2

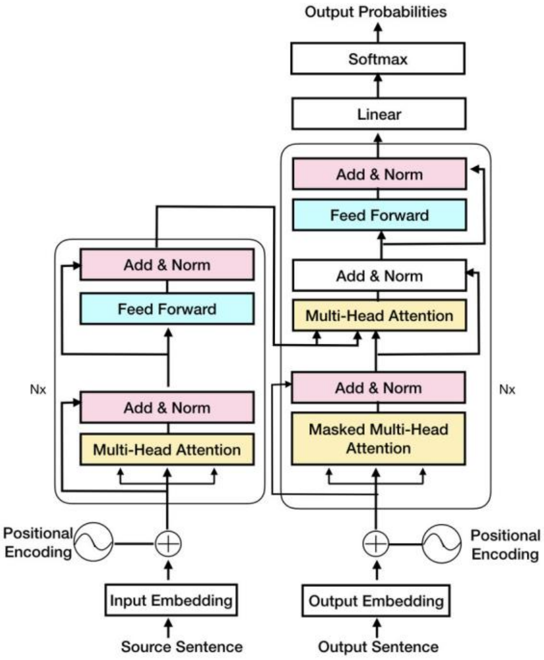

Figure (2.3): Transformer model architcture (Vaswani et al., 2017)

The decoder includes decoding layers that do the opposite of the encoder. Each decoder layer takes the encoder's output and uses their incorporated contextual information to generate an output sequence (Vaswani et al., 2017). An attention mechanism is used by both the encoder and the decoder. Attention is looking at the entire input and deciding at each step which part of the sentence should get more important. This is done by a set of weights that are learned during training (Vaswani et al., 2017).

2.8.1 QARiB Model[1]

QCRI Arabic and Dialectal BERT (QARiB) model is an Arabic BERT model that is developed by Qatar Computing Research Institute (Abdelali et al., 2021). The QARiB is implemented based on the BERT model and trained on large datasets, including both formal and informal text. Formal text data combines the following Corpus: Arabic Gigaword Fourth Edition[2], Abu El-Khair Corpus (El-khair, 2016), and OPUS[3]. The Informal text is Arabic tweets collected using Twitter API. In Total the QARiB model

[1] https://huggingface.co/qarib/bert-base-qarib
[2] https://doi.org/10.35111/v9fm-zn61
[3] https://opus.nlpl.eu/

trained on 420 million tweets and 180 million sentences of text. The model is available on GitHub.

2.8.2 MARBERT Model[4]

MARBERT are pre-trained Arabic transformer language models model developed by (Abdul-mageed et al., 2021), and they are available on GitHub. These models are trained on large to massive datasets covering different text domains and genres, including social media, to adapt natural language processing techniques to serve wider and more diverse communities. The MARBERT uses BERT-base architecture but without next sentence prediction (NSP) because it is trained on short tweets. The MARBERT is trained on a large Twitter dataset of 128 GB (15.6B tokens) that includes different Arabic dialects and Modern Standard Arabic (MSA).

2.8.3 Multi dialect Arabic BERT model[5]

Multi dialect Arabic BERT mode is based on a Bidirectional Encoder Representation from Transformers (BERT) architecture. But instead of training from scratch (Talafha et al., 2020), they initialized the weights of the model using Arabic-BERT[6], which is a publicly released BERT model trained on around 93 GB of Arabic content crawled from around the internet (Talafha et al., 2020). Then they improved the results by further pre-training AraBERT on the 10M tweets released by the Nuanced Arabic Dialect Identification (NADI[7]) organizers, for 3 epochs (Talafha et al., 2020).

2.8.4 dehatebert-mono-arabic model[8]

dehatebert-mono-arabic model is used to detect hate speech in Arabic. The term "mono" refers to training this model only on Arabic language data. It's been fine-tuned using the Bert multilingual model. With various learning rates, the model achieves the best validation score of 0.877609 for a learning rate of 2e-5

[4] https://huggingface.co/UBC-NLP/MARBERT
[5] https://huggingface.co/bashar-talafha/multi-dialect-bert-base-arabic
[6] https://github.com/alisafaya/Arabic-BERT
[7] https://sites.google.com/view/nadi-shared-task
[8] https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-arabic

2.8.5 xlm-r-large-arabic-toxic model[9]

xlm-r-large-arabic-toxic model used to classify toxically or hate speech in the Arabic language. The model is trained using Arabic comments by fine-tuning the XLM-Roberta-Large model[10].

2.8.6 Xlm-roberta-large-xnli model[11]

Xlm-roberta-large-xnli model is designed for zero-shot text classification, particularly in languages other than English. And it fine-tuned xlm-roberta-large, a transformer-based multilingual masked language model that has been pre-trained on a collection of 100 languages (Ruder et al., 2019), using a mix of Natural Language Inference (NLI) data in 15 languages. It has been fine-tuned using XNLI, a multilingual NLI dataset.

2.8.7 XLM-RoBERTa-large-XNLI-ANLI model[12]

XLM-RoBERTa-large-XNLI-ANLI model is used for zero-shot text classification. It fine-tunes the Xlm-roberta-large model on the XNLI dataset and ANLI dataset.

2.8.8 Roberta-large-mnli[13]

RoBERTa Is A zero-shot text classification, which is a BERT Pre-Training approach that is Robustly Optimized (Puranik et al., 2021). It was trained for 1000 times longer than BERT, with larger batches and 1000% more data. The Next Sentence Prediction (NSP) task employed in BERT's pre-training was eliminated and dynamic masking during training was introduced. It's also been trained on a 76 GB large new dataset (Puranik et al., 2021).

[9] https://huggingface.co/akhooli/xlm-r-large-arabic-toxic
[10] https://huggingface.co/xlm-roberta-large
[11] https://huggingface.co/joeddav/xlm-roberta-large-xnli
[12] https://huggingface.co/vicgalle/xlm-roberta-large-xnli-anli
[13] https://huggingface.co/typeform/roberta-large-mnli

Chapter 3
Related works

This chapter reviews the recent and relevant research in the field of hate speech detection and masking. We review the previous works as follows: First, there are works that construct new datasets for hate speech detection tasks. Second, there are works that focus on generating hate/obscene/offensive terms automatically. Third, there are works that use ML, DL, and transformer learning to detect hate speech. Fourth, there are systematic/ analytical studies. Finally, research works that are related to paraphrasing hate speech content are presented.

3.1 Construction datasets for hate speech detection tasks

(Alakrot et al. 2018) Provide an Arabic dataset from YouTube comments, designed for use in the detection of hate language in a machine learning scenario. In addition, they record the process of labeling they have carried out, taking into consideration the variation in the Arab dialects and the variety of understanding of offensive language throughout the Arab world. Additionally, they prepare and offered the dataset for use as a training dataset.

(Mulki et al., 2019) Gathered dataset from Twitter for identifying hate speech and abusive language in Arabic text, and it is available as a benchmark dataset called L-HSAB. There were 5,846 tweets in this dataset, which were divided into three categories: hate, normal, and abusive. They used ML classification using NB and SVM classifiers in experiments on their dataset. Using the NB classifier, the results were: 90.3%, 89.0%, 90.5%, and 89.6% in accuracy, recall, precision, and f1-measure.

(Albadi et al., 2018) Investigate the challenge of religious hate speech detection in Arabic Twitter which is the first effort in this field. By creating and publishing a dataset of 6,136 tweets, approximately 1000 for each of the six religious groups labeled as hate or not hate (Muslims, Jews, Christians, Atheists, Sunnis, and Shia). Then they produced and published three lexicons of religious hate phrases that can be used for a variety of purposes, including sampling microposts for religious hate speech. Finally, they looked into three methods for detecting religious hate speech: lexicon-based, n-gram-based, and deep learning-based methods. With 0.79 accuracy and 0.84 AUROC, the GRU-based RNN with pre-trained word embeddings had the best results.

(Mubarak et al. 2017) Provides an automated method for creating and extending a list of obscene terms. First, they collect 175 million tweets in the Arabic language as an initial data set, by searching for some patterns that are usually used in offensive communication, The words that appeared following these patterns were then gathered and personally analyzed to see if they were obscene or not, resulting in a final list that included 415 words after adding hashtags that are used to screen pornographic pages, second, from the same initial set they classified twitter users to two groups: the clean group who authored tweets that did not include a single obscene word, and obscene group who used at least one of the obscene words, the Log Odds Ratio (LOR) was then calculated for each word unigram and bigram that appeared at least ten times. The tweets written by clean tweeps work as a background corpus, whereas the tweets written by obscene tweeps work as a foreground corpus. Additional 3,430-word unigrams and bigrams were formed by the unigrams and bigrams that yielded a LOR equals infinity, which signifies they only appeared in the foreground corpus (obscene) but not in the background corpus (clean). The authors collected other datasets from a popular Arabic news site by capturing user comments that were deleted from the website then they violated the site's rules

Table 3.1 summarized the datasets that are mentioned in related works and the detail for each one.

Table 0.1):Details about datasets that we mentioned it in related works

Dataset name	Authors	Size	Labels	Size for each label
Obscene words[14]	(Mubarak et al., 2017)	288 words, 127 hashtags		
Aljazeera Deleted Comments[15]		32K comments	obscene, offensive, clean.	2% obscene, 79% offensive, and 19% clean.
TweetClassification[16]		100 original tweets plus 1,000 comment/reply tweets – **1,100 tweets all together**	obscene, offensive, clean.	19.1% obscene, 40.3% offensive, and 40.6% clean
CommentsFromYouTube[17]	(Alakrot et al., 2018)	This dataset conatins 167,549 YouTube comments from 84,354 users along with 87,388 replies from 24,039 users, from 150 YouTube videos.	offensive or not	
L-HSAB[18]	(Mulki et al., 2019)	5846	normal, abusive or hate	Normal 3650 Abusive 1727 Hate 469
Religious Hate Speech dataset[19]	(Albadi et al., 2018)	6,136	Hate or not hate	3058 not hate 2512 hate

[14] http://alt.qcri.org/~hmubarak/offensive/ObsceneWords.txt

[15] http://alt.qcri.org/~hmubarak/offensive/AJCommentsClassification-CF.xlsx

[16] http://alt.qcri.org/~hmubarak/offensive/TweetClassification-Summary.xlsx

[17] https://onedrive.live.com/?authkey=!ACDXj_ZNcZPqzy0&id=6EF6951FBF8217F9!105&cid=6EF6951FBF8217F9

[18] https://github.com/Hala-Mulki/L-HSAB-First-Arabic-Levantine-HateSpeech-Dataset

[19] https://github.com/nuhaalbadi/Arabic_hatespeech

3.2 Detection of hate speech using ML, DL, Transformer techniques

(Saksesi et al. 2018) Propose using a deep learning method with the Recurrent Neural Network (RNN) to determine whether or not the text contains hate speech. The total twitter dataset that uses it is 1235 records and there are 652 records classified as hate speech and 583 records are not hate speech. They had made the test with several techniques such as Recurrent Neural Network, Data Partition, Epoch, Learning Rate, and Batch Size. The testing results reached 91% precision, 90% recall, and 91% accuracy on average.

(Biere & Bhulai, 2018) propose using a deep learning model namely Convolutional Neural Network for classification, this classifier assigns each tweet to one of the classes of a twitter dataset: hate, offensive, or neither. The accuracy, precision, recall, and F-score of this model have all been used to assess its performance. The final model has a 91% accuracy, 91% precision, 90% recall and an F-measure of 90%. It should be noted that it is also suggested to further analyze the predictions and errors, to realize more insight on the misclassification.

(Alami et al., 2020) Proposed using AraBERT for the identification of offensive language from Arabic content. Starting from preprocessing tweets by dealing with emojis and replacing them with their meanings in Arabic. Then, in both the fine-tuning and inference steps, they replaced any emojis with the token [MASK]. The AraBERT concept was then applied to tweet tokens. Finally, they pass them into a sigmoid function to determine if a tweet is offensive or not. This approach achieved the best macro F1 score equal to 90.17% on the Arabic task in OffensEval 2020.

(Hassan et al., 2020) Performed many experiments for offensive language identification in Arabic with SVMs, DNNs, and Multilingual-BERT. The best results were obtained by the aforementioned models using an ensemble approach based on majority vote. With a macro F1 score of 90.16%, this model came in second in the official rankings.

(Wang et al., 2020) Provided a multilingual method using pre-trained language models ERNIE and XLM-R, their technique has two phases, starting with pre-training using large scale multilingual unsupervised texts, which results in a unified pre-training model that learns all language representation at the same time. Then used labelled data to fine-tune the pre-trained model. This technique obtained an F1 macro score of 0.89 on Arabic.

(Safaya et al., 2020) Proposed an approach using Convolutional Neural Networks with a pre-trained BERT model for the offensive language identification task from SemEval 2020 (Zampieri et al., 2020). They prove that combining BERT with CNN outperforms using BERT alone, and they highlight the necessity of employing pre-trained language models for downstream tasks. This approach acquired a macro averaged F1-Score of 0.897 in Arabic, which ranked fourth among participating teams for the Arabic language in the scope of the OffensEval 2020.

(Keleg et al., 2020) Presents different models for offensive language detection. These models are the TF-IDF and logistic regression, CNN using word embeddings from Aravec, Bi-directional LSTM using word embeddings from Aravec, fine-tuning multilingual BERT, and fine-tuning AraBERT. They've also created a list of obscenity words and utilized simple augmentation rules to construct the many variants of each. The AraBERT-based model, which outperformed the cased multilingual BERT model, was their best model. This system ranked fifth in the official rankings, with a macro F1 score of 89.6%.

(Mohaouchane et al., 2019) Offers many various neural networks models for detecting offensive language on Arabic social media. These models are CNN, bidirectional LSTM, and merged CNN-LSTM. These models are evaluated on an Arabic YouTube comments dataset that include 15,050 comments extracted from famous and contentious YouTube videos with Arab celebrities. They employ Arabic word embeddings to represent the comments and train this dataset through a set of pre processes. The combined CNN-LSTM network has the best recall of 83.46% while the CNN has the best accuracy of 87.84 and precision of 86.10.

(Husain & Uzuner, 2021) Proposed applying transfer learning to several Arabic offensive language datasets separately and testing it with other datasets separately, as well as investigating the impacts of concatenating all datasets to be utilized for fine-tuning and training the BERT model. These datasets involve Aljazeera.net Deleted Comments (Mubarak et al., 2017), YouTube dataset (Alakrot et al., 2018), Levantine Twitter Dataset for Hate Speech and Abusive Language (L-HSAB) (Mulki et al., 2019), and OSACT offensive and not offensive classification samples (Hassan et al., 2020). They totally depend on binary classes; offensive or non-offensive, and they change various types of offensive languages like abusive or hate to the offensive class. The highest recorded scores are shown for the OSACT dataset when used in training and testing. Their findings show that Arabic monolingual BERT models outperform BERT multilingual models, and that transfer learning across datasets from multiple sources and topics, such as YouTube comments from musicians channels and Aljazeera News comments from political articles, performs poorly. When comparing individual datasets, combining from multiple datasets at the same time has no effect on performance; however, it affects the performance of the highly dialectic dataset, L-HSAB, by 3% in macro F1 score.

(Alshalan & Al-Khalifa, 2020) Provides dataset collected from Twitter to detect hate speech, this dataset contains 9316 tweets labeled as hateful, abusive, and normal. Then they evaluated different Deep neural network models based on CNN and RNN, these models are CNN, GRU, CNN + GRU, and BERT. The results appear that CNN outperformed other models, with an F1-score of 0.79 and an AUROC of 0.89, whereas BERT failed to increase performance, which might be due to the fact that BERT was trained on Wikipedia, which is a different kind of dataset.

(Faris et al., 2020) Collected dataset from Twitter that included hate expressions on a variety of topics in the Arabic language. This data was gathered using various terms such as racism, sport, and Islam, and then categorised as Hate or Normal. The authors proposed using a deep learning approach for the automatic detection of cyberhate speech. This approach combines a convolutional neural network (CNN) and a long short-term memory (LSTM) network with the Word2Vec and AraVec word embedding techniques to extract a set of words features that can take the hidden relations of words in the dataset. The proposed method performed well in identifying tweets as Hate or Normal, with the best one scoring 66.564%, 79.768%, 68.965%, and 71.688% for the accuracy, recall, precision, and F1 measure.

3.3 Systematic/ analytical studies

(Fortuna & Nunes, 2018) Provides a literature review to clarify the current state of the art and potential in the subject of automatic hate speech detection. They see it's still in its beginnings. Furthermore, the most of papers discovered have a low amount of citations. In the practical side, the most of researchs regard this as a binary classification issue, however, a small number of people have utilized a multiclass method. Twitter is the most popular social media platform, and English the most common language. They arrived at the conclusion that authors do not use public datasets and do not publish their own. This makes comparing results and conclusions extremely challenging. In addition, comparative research and surveys are limited in this field.

(Al-Hassan & Al-Dossari, 2019) Examines the topic of hate speech, particularly "cyber hate," which is expressed through social media and the internet. Furthermore, they compared several anti-social behaviors, such as (hate speech, abusive and offensive language, radicalization and cyberbullying). After then, a detailed analysis of how text mining can be employed in social networks was presented. They then looked into certain challenges that could serve as a roadmap for the Arabic hate speech detection model.

3.4 Paraphrasing hate speech content

(Qiang & Wu, 2019) Provides the first unsupervised text simplification system based on a phrase-based machine translation system, which takes advantage of careful phrase tables initialization and language modeling. Their model only uses the ordinary English Wikipedia as a knowledge base without any simple corpus. They use the WikiLarge, WikiSmall, and Newsela datasets to test the suggested model. The results of their experiments suggest that their model improves significantly.

The related works that addressed the detection of hate speech using ML, DL, Transformer techniques are represented as follows: (Saksesi et al. 2018), (Alshalan & Al-Khalifa, 2020) used RNN, (Biere & Bhulai, 2018), (Alshalan & Al-Khalifa, 2020), (Keleg et al., 2020), (Mohaouchane et al., 2019) used CNN, (Husain & Uzuner, 2021), (Alshalan & Al-Khalifa, 2020), (Keleg et al., 2020), (Alami et al., 2020), (Hassan et al., 2020), (Wang et al., 2020) utilized pre-trained model like BERT, AraBert, Multilingual-Bert, ERNIE and XLM-R, (Faris et al., 2020), (Keleg et al., 2020), (Mohaouchane et al., 2019) applied LSTM and bidirectional LSTM, and (Safaya et al., 2020) Combined BERT with CNN. From our reviews, it's clear from Arabic-related works in the area of hate speech detection that there is limited work using Deep learning approaches and Transformer pre-trained models for this task. Therefore, more investigation and research are required. In this research, we try to bridge this gap in detecting hate speech in Arabic research by using existing modern techniques to classify hate speech. These techniques depend on understanding the contextual meaning of the text, such as the RNN, and pre-trained transformers.

Chapter 4
Hate Speech Detection

This chapter describes the methodology that we use for hate speech detection. The methodology steps including dataset description, preprocessing techniques, text features, and the models that we use for hate speech detection. Finally, the chapter presents experimental setups, evaluation metrics, results and discussion.

4.1 Methodology

This section describes the methodology that we follow in this research. Figure 4.1 presents a brief overview of the methodology phases including data acquisition, data preprocessing, training models, and evaluation metrics.

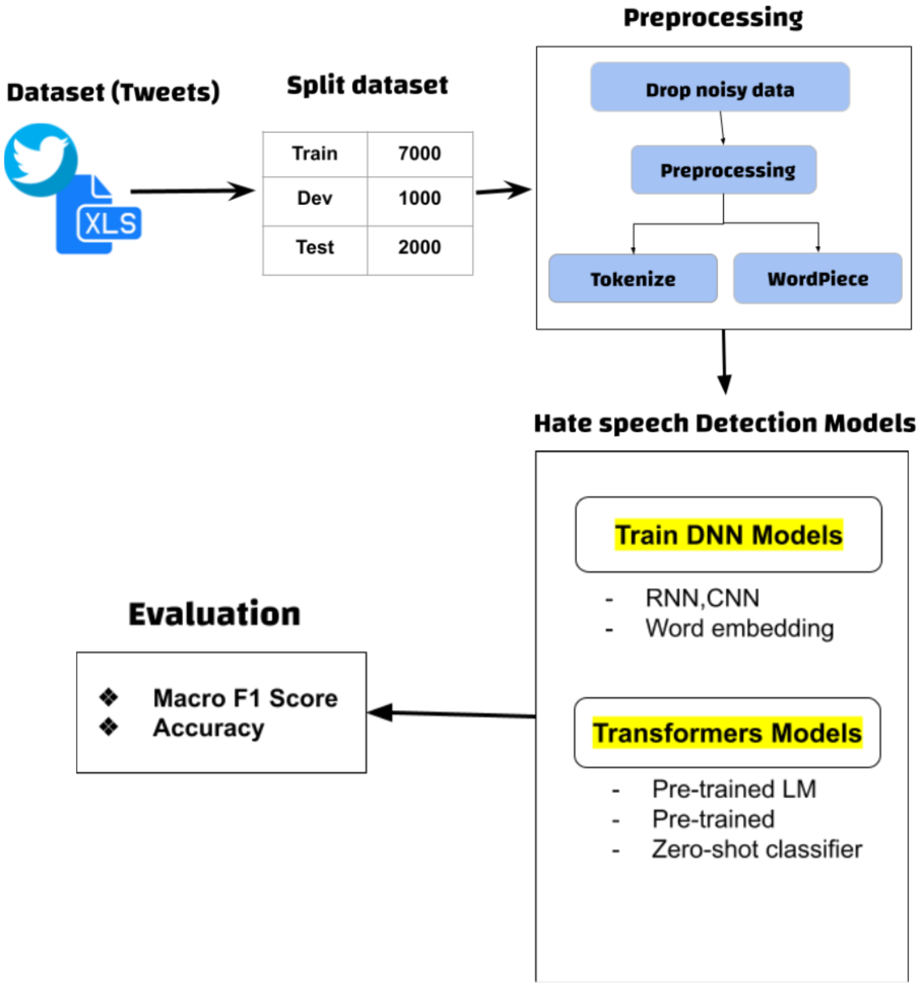

Figure 0.1): The Brief of Hate speech detection Methodology

4.1.1 Dataset

We use the dataset that is published in the shared task SemEval-2020 (Zampieri et al., 2020) for Arabic offensive language detection. This dataset was collected from Twitter and contains 10,000 tweets labeled either for offensive or not offensive. The dataset is partitioned into 7000 tweets for training, 1000 tweets for development, and 2000 for testing, just like the SemEval competition. Table 4.1 shows the details of the dataset parts. As shown in the Table 4.1, class distribution is imbalanced, i.e there are only 1,991 offensive tweets vs 8000 not offensive.

Table (4.1): The dataset distribution

	Training set	Development set	Test set
Offensive	1410	179	402
Not offensive	5590	821	1598
Total	7000	1000	2000

4.1.2 Data Preprocessing

This step is important for cleaning data from unnecessary content and transforming it into a consistent format that can be simply processed and analyzed. In our work, we used classical text preprocessing steps as following:

1. Letter normalization: which means the process of transforming letters that appear in different forms into a single form. The normalization step includes: replace (أ ، إ ، آ ، ا) with (ا), (ة) with (ه), and (ى) with (ي), the purpose of this step is to reduce the orthographic differences that can be seen in tweets.
2. Remove punctuations and diacritics: We exclude question marks and exclamation marks from this step.
3. Remove repeating characters.
4. Remove all words that contain non-Arabic characters.

4.2.2.1 Tokenize and AraBert WordPiece

In the transformer models, the text is passed to a transformer model divided by the text with WordPiece that we mentioned in the background.

4.2.2.2 Text features

In the three basic deep learning models (RNN, CNN, RNN-CNN), the features are represented as word embeddings through Word2Vec. We use word2vec model to load Twt-CBOW[20] that is collected from 66,900,000 Arabic tweets and contains 1,259,756 vocabs (Soliman et al., 2017).

[20] https://bakrianoo.ewr1.vultrobjects.com/aravec/full_uni_cbow_300_twitter.zip

4.2.3 Hate speech detection models

In this section we present models that are use for hate speech detection. We investigate different neural networks architectures for detection, these models include Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and Transformers.

In our hate speech detection experiments we applied two strategies, the first strategy is building and training deep learning models DL from scratch, and the second one uses pre-trained and transformer models.

4.2.3.1 Building RNN Model

We selected the RNN architecture for this challenge because sequential information is important in detecting hate speech sentences. We applied a model from TensorFlow that uses RNN for text classification[21]. In this experiment we use the following parameters: batch size 1024, RNN sequence length 25, number epochs 30, Learning Rate is 1e-3, and Adam optimizer.

4.2.3.2 Building CNN Model

Figure 4.2 illustrates our CNN architecture, which includes five layers: an input layer (embedding layer), a convolution layer, a pooling layer, a hidden dense layer, and finally the output layer. Here Similar to RNN architecture we refer to that all tweets were mapped into 300-dimensional real-valued vectors by the embedding layer, the embedding layer then passes an input feature matrix to a dropout layer with a rate of 0.5. The dropout layer's primary objective is to help prevent overfitting issues. Then, the output is received by the convolution layer that has 128 filters with the same kernel sizes: 7 and a rectified linear unit (ReLU) function for activation. After that, these convolution features are fed as input to a max-pooling layer (global) for downsampling, then concatenated and passed as input to a fully connected dense layer containing 128 neurons followed by a dropout layer with a rate of 0.5. The output is then fed to the output layer with sigmoid activation to produce the final predictions.

[21] https://www.tensorflow.org/text/tutorials/text_classification_rnn

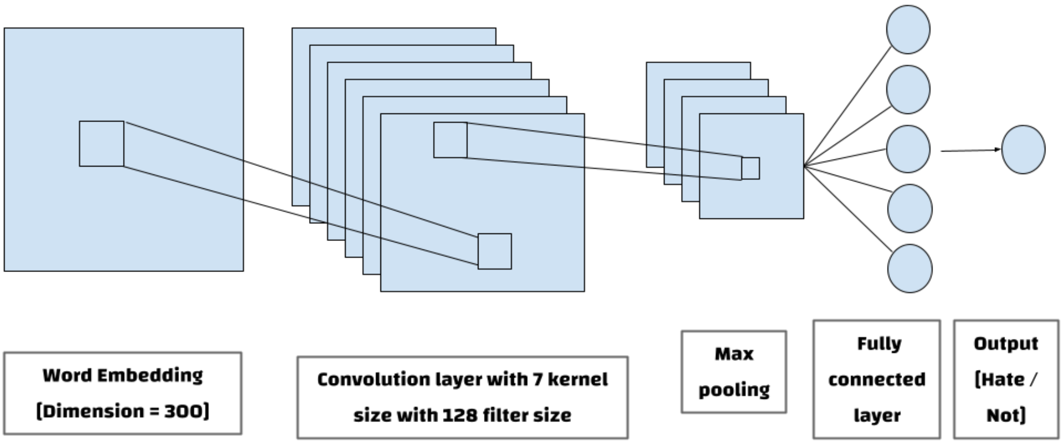

Figure (4.2):CNN architecture for text classification. CNN layers include word embedding as input layer, Conv1D, max pooling, fully connected, and output layer

4.2.3.3 Building CNN-RNN Model

In this model, we make a combination of both CNN and RNN as shown in Figure 4.3. The CNN and RNN combination architecture contain six layers: an input layer (embedding layer), a convolution layer with 128 filters and a kernel size of 7, a max-pooling layer, a Bidirectional LSTM layer, another LSTM layer, and finally the output layer. The embedding layer starts by mapping the tweets into a 300-dimensional vector space, producing a tweets matrix. This matrix is then passed to a dropout layer with a rate of 0.2 to avoid the overfitting problem. Then, the output of the dropout layer is fed into the convolution layer, which has 128 filters with kernel sizes of 7. The rectified linear unit (ReLO) function is used for activation. then passed into a max-pooling layer with a pool size of 2 and a dropout layer with a rate of 0.2. This produces vector output, which can be considered as extracted features. These extracted features are then passed to the RNN (Bidirectional LSTM) layer with 128 units, followed by the LSTM layer also with 128 units, then followed by a dropout layer with a rate of 0.2. Finally, the output of the dropout layer is then fed into the output layer with sigmoid activation to produce the final predictions.

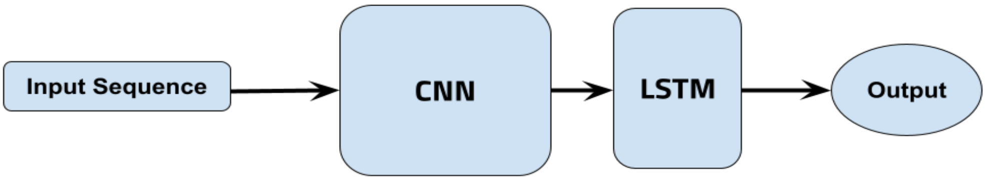

Figure (4.3): A combination between CNN with RNN: CNN block represents all CNN layers in Figure 5 except for the output

4.2.3.4 Using Transformer pre-trained models

In this section, we describe the transformer models that are use in our experiment. Starting from Arabic Pre-trained language models QARiB (QCRI, 2020), Marbert (Abdul-Mageed et al., 2020), and Multi-dialect-bert-base-arabic. Then, pretrained hate speech models xlm-r-large-arabic-toxic and dehatebert-mono-arabic, finally, that Zero-shot classifier models Xlm-roberta-large-xnli, XLM-RoBERTa-large-XNLI-ANLI, and Roberta-large-mnli.

Notes: All transformer models that we use are available through the HuggingFace [22] Transformers library.

[22] https://huggingface.co/models

4.2.3.4.1 Arabic Pre-trained language models

In our experiments use three Arabic pre-train language models QARiB, MARBERT, and Multi dialect Arabic BERT using hugging-face API; Table 4.4 shows the model names used and their details. We use the same Arabert implementation that available in the Github repository[23] to load models, and the parameters that we used are highlighted in Table 4.2. Then, we train and evaluate these models on the dataset that was describe in section 4.1.

Table (4.2): Parameters value that we used in pretrained models

Parameter	Value
Epsilon (Adam optimizer)	1e-8
Learning Rate	5e-5
Batch Size	16
#Epochs	8

4.2.3.4.2 Hate Speech Pretrained models

Hate Speech pre-trained models designe to classify hate speech and detect toxic content in the Arabic language. The first model we use it in our experements is dehatebert-mono-arabic[24], and the other model is xlm-r-large-arabic-toxic[25]. These model details are shown in Table 4.4.

The pre-trained models are used in two ways, the first is training from scratch, and the second way is transfer learning which means freeze some layers, and do additional training to the rest of layers to tune the model for the new domain/task/data. The parameters that we used are highlighted in Table 4.3.

[23] https://github.com/aub-mind/arabert/tree/master/arabert

[24] https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-arabic

[25] https://huggingface.co/akhooli/xlm-r-large-arabic-toxic

Table (4.3): Parameters value that we used in fine tuning hate speech pretrained models

Parameter	Value
Learning Rate	3e-5
Batch Size	7
#Epochs	3

4.2.3.4.3 Zero shot classifier models

The zero-shot text classification model makes a big difference in technology because it can classify any text into any category without prior data. To perform and inference(predict) with the zero-shot classifier, we need to pass the text and the candidate labels. We try many candidate labels and see which labels produce the best results. The Zero-Shot classifiers are xlm-roberta-large-xnli, roberta-large-mnli, and xlm-roberta-large-xnli-anli.

Table (4.3):summarized the models that are built and used in our work

Model Type	Training / pre-trained	NN Architecture	Trained on	Reference
Building DL model	Training	RNN	Semeval 2020 shared task dataset	-
		CNN	Semeval 2020 shared task dataset	-
		CNN-RNN	Semeval 2020 shared task dataset	-
Transformer models	Pre-trained transformer (Language Model)	QARiB Model	420 Million tweets and 180 Million sentences of text	(QCRI, 2020) https://huggingface.co/qarib/bert-base-qarib
		MARBERT Model	Random samples from 1B Arabic tweets	(Abdul-Mageed et al., 2020) https://huggingface.co/UBC-NLP/MARBERT
		Multi dialect Arabic BERT model	10M tweets released from the Nuanced Arabic Dialect Identification (NADI) shared task	https://huggingface.co/bashar-talafha/multi-dialect-bert-base-arabic
Hate Speech Pretrained models	Pre-trained transformer	dehatebert-mono-arabic model	16 datasets from 9 different languages.	(Aluru et al., 2020) https://huggingface.co/Hate-speech-CNERG/dehatebert-mono-arabic
		xlm-r-large-arabic-toxic model	Arabic comments by fine-tuning XLM-Roberta-Large	https://huggingface.co/akhooli/xlm-r-large-arabic-toxic
Hate Speech Roberta models	Zero shot classifier models	Xlm-roberta-large-xnli model	Combination of NLI data in 15 languages.	(Ruder et al., 2019) https://huggingface.co/joeddav/xlm-roberta-large-xnli
		XLM-RoBERTa-large-XNLI-ANLI model	several NLI datasets	https://huggingface.co/vicgalle/xlm-roberta-large-xnli-anli
		Roberta-large-mnli		https://huggingface.co/typeform/roberta-large-mnli

4.3.2 Evaluation metrics

Evaluation metrics considered in this study (Macro F1-score) as similar to those used in the literature for the Shared Task, also we recorded precision, recall, f1 score, and accuracy for each model. The definition of the metrics used is as follows:

Precision computes the proportion of instances predicted as positives that were correctly evaluated. Precision is calculated mathematically as follows:

$$\text{Precision} = \frac{TP}{TP+FP}$$

The recall counts the proportion of positive instances that were correctly evaluated, and its calculated mathematically as follows:

$$\text{Recall} = \frac{TP}{TP+FN}$$

F1-score is the harmonic mean of precision and recall combining both values in a single number, and it's calculated as follows:

$$\text{F} = 2.\frac{precision.recall}{precision+recall}$$

Accuracy defined as a proportion of correctly predicted cases among the total number of cases tested.

$$\text{Accuracy} = \frac{tp+tn}{tp+tn+fp+fn}$$

We will show the following definitions for better understanding:

True Positive (TP): is an instance correctly classified as offensive.

False Positive (FP): is a not offensive instance wrong classified as offensive.

True Negative (TN): is an instance correctly classified as not offensive.

False Negative (FN): is an offensive instance miss classified as not offensive.

4.3 Experiments

We perform a set of experiments to evaluate the proposed models for Arabic hate speech detection tasks using deep learning models listed in Table 4.2. In all experiments, we used a binary classification task in which tweets were classified to one of these classes (offensive or not-offensive, hate or not-hate, or toxic or not toxic). In this section, we describe our experimental setup, evaluation metrics, text features, model parameters for DL, transformer models, and experimental results.

4.3.1 Experimental tools

To implement our experiments, we use the Keras library with TensorFlow as a backend for all neural network models training[26], which is python framework for deep machine learning, and the HuggingFace library for all transformer models[27]. We implemented the experiment on Google Collaboratory[28], which provides a free Jupyter notebook environment that requires no setup, free access to GPUs, runs entirely on the cloud, and easy sharing.

[26] https://github.com/huggingface/transformers

[27] https://huggingface.co/models

[28] https://colab.research.google.com/notebooks/intro.ipynb#recent=true

4.3.3 Experimental results

This section describes the results of experiments of all models described in the previous section.

4.3.3.1 DL models

In DL models experiments, the results are very close to each other, and we notice models that use CNN achieved the best results, where the Macro F1 score is 51% and accuracy is 80%, as shown in Table 4.5. It can be noted from the table that training DL models from scratch obtained a good result, but not as good as the results that are recorded in the shared task. DL models alone can not achieve the best results alone, it should be combined with other pre-trained language models to improve the results as you will see in the next experiments.

Table (4.4): DL models results

Model	Offensive			Not offensive			Macro average			Accuracy
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
RNN | more LSTM layers | AraVec	0.50	0.05	0.10	0.81	0.99	0.89	0.65	0.52	0.49	0.80
CNN | AraVec	0.50	0.07	0.13	0.81	0.98	0.89	0.65	0.53	**0.51**	**0.80**
LSTM with CNN | AraVec	0.38	0.09	0.14	0.81	0.96	0.88	0.59	0.53	0.51	0.79

Generally, these results are very low compared to the rest of the experiments, and the models stop early after a number of epochs as Figure 4.4 shows, and there is no overfitting. In addition, we notice from Table 4.5 that the recall ratio is very low. In our opinion, these Bad results appear due to an imbalance in data that we talked about it in dataset section 4.1.1.

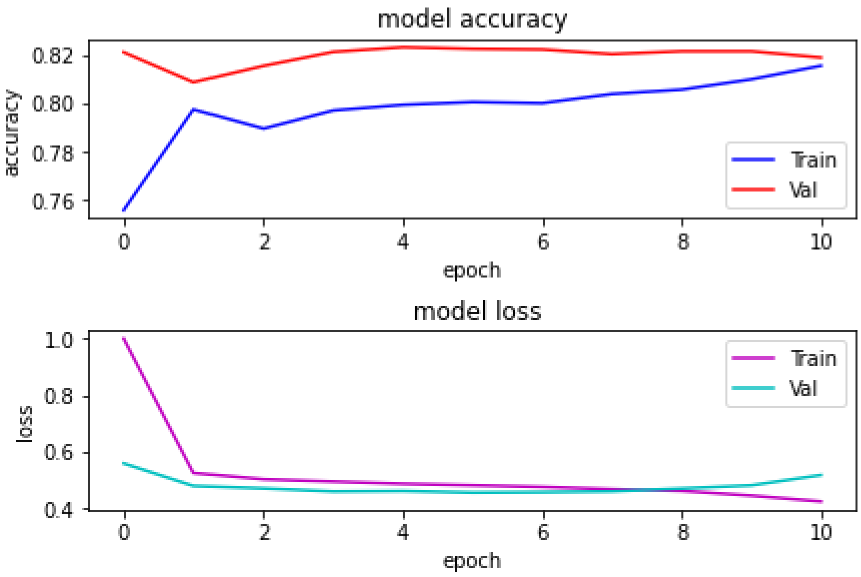

Figure (4.4): Monitor the performance of training CNN+RNN model

4.3.3.2 Transformer models

In this section, we describe the results for transformer models experiments

4.3.3.2.1 Arabic Pre-trained language models

In pre-trained language models experiments, the highest result is from the QARiB model with the AraBert preprocessing experiment, where the macro F1-score is 92% and accuracy is 95% as shown in Table 4.6, and it's the best result we achieved on all experiments. Also in the other experiments with other language models (MARBERT, multi-dialect-bert-base-arabic) we obtained interesting Macro F1 scores and accuracy compared with our other experiments. It can be noted that pre-trained language models improve the classification accuracy. That is one of the advantages of transformer language models that can be adopted for any NLP task.

Table (4.5): Transformer model [Language models] results

Model	Offensive			Not offensive			Macro average			Accuracy
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
QARiB with classical preprocessing, without emoji	0.88	0.84	0.86	0.96	0.97	0.97	0.92	0.90	0.91	0.94
QARiB with AraBert preprocess	0.88	0.87	0.87	0.97	0.97	0.97	0.92	0.92	**0.92**	**0.95**
MARBERT	0.88	0.76	0.82	0.94	0.97	0.96	0.91	0.87	0.89	0.93
multi-dialect-bert-base-arabic with AraBert preprocess	0.83	0.80	0.81	0.96	0.96	0.96	0.89	0.88	0.89	0.93
multi-dialect-bert-base-arabic	0.86	0.82	0.84	0.95	0.97	0.96	0.91	0.89	0.90	0.94

4.3.3.2.2 Hate speech pre-trained models

In Hate speech Pre-trained models experiments, the best result is from the xlm-r-large-arabic-toxic model with AraBert preprocessing experiment, where the Macro F1 score reached 75% and the accuracy to 83% as shown in Table 4.7. Our experiments with fine-tuning pre-trained models got low results since the best result is 60% and the accuracy is 67% from the experiment of a fine-tuning dehatebert-mono-arabic model. It can be noted that the pre-trained model has some limitations to classify new data as it was trained on older data, and even with fine tuning, the performance is still not high, and there is a room for improvement.

Table (4.6):Transformer model [Pre-trained model] results

Model	**Offensive**			**Not offensive**			**Macro average**			**Accuracy**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
Fine-tuning (freeze one layer from two) | xlm-r-large-arabic-toxic	0.25	0.45	0.32	0.82	0.66	0.73	0.54	0.55	0.52	0.61
Xlm-r-large-arabic-toxic with common preprocess	0.47	0.75	0.57	0.92	0.78	0.85	0.69	0.77	0.71	0.78
xlm-r-large-arabic-toxic with AraBert preprocess	0.56	0.67	0.61	0.91	0.87	0.89	0.74	0.77	**0.75**	**0.83**
Fine-tuning (freeze one layer from two) | dehatebert-mono-arabic	0.33	0.64	0.44	0.88	0.67	0.76	0.61	0.66	**0.60**	**0.67**
dehatebert-mono-arabic	0.34	0.64	0.44	0.88	0.69	0.77	0.61	0.66	0.61	0.68

4.3.3.2.3 Zero-shot classifier models

In zero-shot classifier models experiments, the best one is xlm-roberta-large-xnli-anli without any preprocess and with toxic, not_toxic classes, where the Macro F1 score is 68%, and the accuracy is 80%, as shown in Table 4.8.

Table (4.7):Transformer model [Zero-Shot classifier model] results

Model	**Offensive**			**Not offensive**			**Macro average**			**Accuracy**
	P	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**	
xlm-roberta-large-xnli-anli | without preprocess | toxic&not_toxic	0.51	0.47	0.48	0.87	0.89	0.88	0.69	0.68	**0.68**	**0.80**
xlm-roberta-large-xnli-anli | with AraBert preprocess | toxic&not_toxic	0.54	0.36	0.43	0.85	0.92	0.89	0.69	0.64	0.66	0.81
xlm-roberta-large-xnli | with AraBert pre | hate&Not_hate	0.54	0.37	0.44	0.85	0.92	0.89	0.70	0.64	0.66	0.81
roberta-large-mnli | toxic&not_toxic	0.18	0.30	0.22	0.79	0.65	0.71	0.48	0.47	0.47	0.58

In general, comparing macro F1 score and the accuracy results in Tables 4.5, 4.6, 4.7, 4.8, the QARiB model with AraBERT preprocessor has achieved the best result, where the Macro F1 score is 92% and the accuracy is 95%. This result outperformed the best results that are published in the Semeval 2020 shared task, where the best one for the ALAMIHamza team (Alami et al., 2020) obtained 90.17% in Macro F1-score and 93.9% in accuracy. Which, by the way, we are using the same dataset splitting that used in the shared task 7000 tweets for training, 1000 tweets for development, and 2000 for testing. Table 4.9 displays some predicted samples from the best experiment.

Table (4.8): Some classified examples from the best model

Predicted Label	Actual Label	Tweets	#
OFF	OFF	اما انت تقعد طول عمرك لا مبدا ولا راي ثابت يا صعلوك اقسم بربي ماتجي حاجه يا داعشي يا عميل الانكليز راس مالك كلام بس عقاب جبان معمر سيد وحفتر راح يركب عليك	1
OFF	OFF	بتخاف نسوانك يزعلوا ولا ايه 😂 اه يا هلفوت يا بتاع الورد 😂	2
NOT_OFF	NOT_OFF	يا عـسـانـي نـبـقي يا عـمري حـبايـب وحـبنـا يكبر معانا 💛	3
OFF	OFF	باقي البيان وينو ما شفنه يا برهان ورينا يا برهان ورينا شو بيحصل في شله الكيزان سلم بشير العار للحاكم السجان وباقي الكل...	4
NOT_OFF	NOT_OFF	اللهم انت الشافي المعافي اشفيه وجميع مرضي المسلمين والمسلمات شفاء لا يغادر سقما واصبغ عليهم عافيتك يا شافي يا معافي يا ارحم الراحمين	5
NOT_OFF	NOT_OFF	اه يا غيث يا حبيبنا لو تعرف حجم الفراغ الي تركته فينا الله يرحمك و يغفرلك و يجمعنا بيك بعد العمر المديد في الفردوس...	6
NOT_OFF	NOT_OFF	يا حب يا دنيا جديده يا احلي يا احلي ابيات القصيده❤️	7
NOT_OFF	NOT_OFF	يا ملاك الحسن والفتنه يا جمال الضحكه دي اهل الصباح مع يونس	8
NOT_OFF	OFF	بس يا شريره يا اللي مش صاحبتي 😒✋😂	9
OFF	OFF	ينعل الذي رباش يا وصخه يا جيفه تف ع وجهك وعاده غاليه عليك يا صندل	10

Chapter 5
Hate Speech Masking

This chapter describes the methodology that is used for masking hate speech from content. At the first, we describe the parallel corpus that is built and used in our work. Then, it describes the methodology steps including the model that is used for hate speech masking. Finally, the chapter presents experimental setups, evaluation metrics, results and discussion.

As mentioned in the problem statement, to the best of our knowledge, this problem has not been addressed in the literature, so here we try to open the door for a new research direction to address this new task.

5.1 Methodology

In this section, we describe the methodology that we follow to implement the hate speech masking model. Figure 5.1 presents a brief overview of the methodology phases including parallel corpus preparation, data preprocessing, training model, and evaluation metrics.

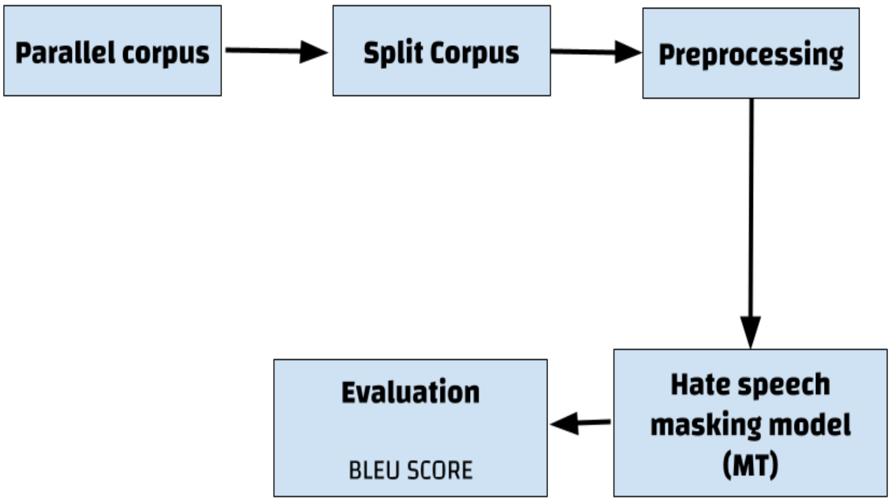

Figure (5.1): The Brief of Hate Speech Masking Methodology

5.1.1 Parallel corpus preparing

We build a parallel corpus that contains pairs of sentences as shown in Table 5.1, the first pair containing part from the dataset that is published in the shared task SemEval-2020 (Zampieri et al., 2020) for Arabic offensive language detection. The second pair contains the same sentences in the first pair with masking the bad words in the sentences, by replacing each bad word with stars based on the number of letters for each word, which requires human workers to mask the dirty word.

In this step, "Haneen Al-Shurafa" worked as a volunteer and prepared the second pair from the parallel corpus, where she masked bad words from sentences for the entire dataset as shown in Table 5.1.

Table (5.1): Parallel corpus samples

التغريدات بعد استبدال الكلمات البذيئة بنجم بحسب عدد الحروف	التغريدات الأصلية (الجمل التي تتضمن كلام عدائي، بذيء يحض على الكراهية)
يا **** يا *****امشي اشتري ليك حساب ما ممكن تكون فاتح ليك حساب قبل ١٥ يوم فكره انعل ابو مخك زاتو	يا مندس يا حيوان امشي اشتري ليك حساب ما ممكن تكون فاتح ليك حساب قبل ١٥ يوم فكره انعل ابو مخك زاتو
بس يا ***** يا ******* يا ***	بس يا متخلف يا مطبلاتي يا كلب
اه يا شوية **** يا ******... الشعوب وعيت خلاص #السودان	اه يا شوية سفلة يا ملاعين ... الشعوب وعيت خلاص #السودان
والله حريتكم خلوها تنفعكم والله انكم تتمنون الرجوع للوطن بس خلاص تورطتو يا ****** يا ***** المجتمعات يا ***** الشوارع صزقوني تكلمو الى يوم القيامه السعوديه مثل حبل طويق لن تنهز ما دام ال يسعود حكامنا تاج راسكم وراس الي عط	والله حريتكم خلوها تنفعكم والله انكم تتمنون الرجوع للوطن بس خلاص تورطتو يا مرتزقه يا حثاله المجتمعات يا زباله الشوارع صزقوني تكلمو الى يوم القيامه السعوديه مثل حبل طويق لن تنهز ما دام ال يسعود حكامنا تاج راسكم وراس الي عط
وانت يا **يا ****** لا تمثل المجتمع الكوردي	وانت يا لص يا متصهين لا تمثل المجتمع الكوردي
يا **** يا ****مجرد تنحي الرئيس لا يعني صلاح الاحوال ثورتهم فاشلة مغيرها وسترى.	يا دجال يا مجرم مجرد تنحي الرئيس لا يعني صلاح الاحوال ثورتهم فاشلة مغيرها وسترى.

5.1.2 Split parallel corpus

We have adopted two different sizes of data in our strategy, the first one contains 3183 pairs partitioned into 1992 pairs classified as hate speech and 1592 pairs classified as not hate speech. And the second one contains 4783 pairs partitioned into 1992 pairs classified as hate speech and 2791 pairs classified as not hate speech. Each group of datasets is divided for training set, development set, and testing set as shown in Table 5.2

Table (5.2):Parallel corpus groups Details

	Training set	Development set	Test set	Total
First group	2388	795	401	3183
Second group	3287	1095	401	4783

5.1.3 Dataset preprocessing

We are doing the same preprocessing step in classical preprocessing techniques that we mentioned in the previous chapter, which include:

1. Letter normalization: which means the process of transforming letters that appear in different forms into a single form. The normalization step includes: replace (أ ، إ ، آ ، ا) with (ا), (ة) with (ه), and (ى) with (ي), the purpose of this step is to reduce the orthographic differences that can be seen in tweets.
2. Remove punctuations and diacritics: We excluded question marks and exclamation marks from this step.
3. Remove repeating characters.

5.1.4 Hate speech masking model

We build a hate speech masking model using a neural machine translation with a transformer model since we consider the problem as a machine translation problem. The model starts with parsing the data, each line contains a sentence that contains bad words and its corresponding same sentence that masking to bad words. The sentence that contains bad words is the source sequence and the same sentence with the bad word mask is the target sequence. We prepend the token "[start]" and we append the token "[end]" to the target sentence. Then, the model uses two instances of the TextVectorization layer to vectorize the text data, which it to turn the original strings into integer sequences where each integer represents the index of a word in a vocabulary. Then, building the architecture of the sequence-to-sequence transformer[29] which consists of a Transformer Encoder and a Transformer Decoder chained together.

[29] https://keras.io/examples/nlp/neural_machine_translation_with_transformer/

The source sequence will be passed to the transformer encoder, which will produce a new representation of it. This new representation will then be passed to the transformer decoder then will seek to predict the next words in the target sequence. The Transformer Decoder recieves the entire sequences at once, and thus we must make sure that it only uses information from target tokens 0 to N when predicting token N+1. After that, we training model, we used 64 batch size, embedding dimention 256, and 30 epochs. We used early stopping on the validation set with patience 2 to terminate training when the validation loss has stopped decreasing after two epochs with no improvement. And we add L2 regularization which is a technique to reduce the complexity of the model. It does so by adding a penalty term to the loss function.

5.1.5 Evaluation Metrics

In this section, we describe the evaluation metric that is used in our work. Since the methodology that we use considers the problem as a machine translation problem, we use BLEU Score for evaluation, which is defined as an algorithm for evaluating the quality of text which has been machine-translated from one natural language to another (Papineni, K., et al. 2002). The primary assumption behind BLEU is that the closer a machine translation is to a professional human translation, the better it is. BLEU was one of the first metrics to claim a high correlation with human judgements of quality (Coughlin 2003) and remains one of the most popular automated and inexpensive metrics. In our experiments, we use BLEU to compare the generated text with the reference test text sets. The BLEU score calculations allow you to specify the weighting of different n-grams in the calculation of the BLEU score. This gives you the flexibility to calculate different types of BLEU scores, such as individual and cumulative n-gram scores.

In our work, we use the same BLEU score implementation that is available by (Sharma et al., 2017) to evaluate our experiments with two data set sizes and with various vocabulary sizes.

5.2 Experiments

We executed a set of experiments to evaluate the proposed models for Arabic hate speech Masking using a machine translation model. In this section, we describe our experimental setup, model parameters for Machine translation, and experimental results.

5.2.1 Experimental setup

For the training machine translation model, Keras with TensorFlow backend deep Learning framework is used. We have used Google Colab[30] (an open source platform) for Training purposes.

5.2.2 Experimental results

This section describes the results of the experiment model that described in the previous section.

In the dataset that has 3183 sentences, which contain1592 from it are classified as not hate sentences, the best BLEU score we got is (0.29, 0.17, 0.10, 0.07) for (1-gram, bi-gram, 3-gram, 4-gram) sequentially, when the vocabulary size is 8000 words, as shown in Table 5.3. The Not-HS size column displays the number of sentences that are classified as not hate from each dataset, and the UNK column represents the number of unknown words for each experiment after prediction.

Table (5.3): BLEU score for five experiments trained on dataset size equal 3183 with various vocabulary sizes.

Dataset size	**Not-HS size (Not hate speech size)**	**vocab_size**	**UNK (unknown)**	**BLEU 1-Gram**	**BLEU 2-Gram**	**BLEU 3-Gram**	**BLEU 4-Gram**
3183	**1592**	15000	**1537**	0.25	0.14	0.08	0.05
		12000	1957	0.22	0.11	0.06	0.03
		10000	2853	0.27	0.14	0.08	0.05
		8000	1707	**0.29**	0.17	0.10	0.07
		6000	3246	0.28	0.16	0.10	0.06

[30] https://colab.research.google.com/notebooks/intro.ipynb#recent=true

In the dataset that has 4383 sentences, which contain 2791 from it are classified as not hate sentences, the best BLEU score we got is (0.30, 0.18, 0.12, 0.08) for (1-gram, bigram, 3-gram, 4-gram) sequentially, when the vocabulary size is 12000 words, as shown in Table 5.4.

Table (5.4): BLEU score for five experiments trained on dataset size equal 4383 with various vocabulary sizes.

Dataset size	**Not-HS size**	**vocab_size**	**UNK**	**BLEU 1-Gram**	**BLEU 2-Gram**	**BLEU 3-Gram**	**BLEU 4-Gram**
4383	**2791**	15000	2712	0.28	0.16	0.10	0.06
		12000	**1542**	**0.30**	**0.18**	**0.12**	**0.08**
		10000	2027	0.26	0.16	0.10	0.06
		8000	2402	0.29	0.17	0.11	0.07
		6000	2764	0.29	0.18	0.11	0.07

In general, comparing the BLEU score in Table 5.3,5.4, the model with dataset size 4383 and vocabulary size 12000 has achieved the best result, where BLEU score with 1-gram is 30%, which is a good result compared with the state of the art MT systems. We divide the test dataset results into good predictions where the BLEU score is greater than 25 as shown in Table 5.5, and bad predictions where the BLEU score is lower than 25 as shown in Table 5.6. It can be noted from the Table that this task is very challenging, but we open a new research direction with this new task.

Table (5.5):Good prediction samples

Input	Actual	Predicted	1-Gram	2-Gram	3-Gram	4-Gram
يا جماعة يا جدعان ياشباب يامحترمين يا أساتذة يا ولاد الناس يا ولاد الوسخة ميسي لايقارن	[start]يا جماعة يا جدعان ياشباب يامحترمين يا أساتذة يا ولاد الناس يا ولاد ****** ميسي لايقارن[end]	[start]يا [UNK]يا جدعان ياشباب يا[UNK] يا [UNK]يا ولاد الناس يا ولاد **** [UNK]على لا [end]	0.606	0.414	0.414	0.346
لو هفك انت بقي الشوق تبقي اطلعلي فوق 😂 و اصحي وصحصح يا نايم احنا الأول يا بهايم 😎😎	[start]لو هفك انت بقي الشوق تبقي اطلعلي فوق 😂 و اصحي وصحصح يا نايم احنا الأول يا ***** 😎😎 [end]	[start]لو [UNK]انت بقي عمري كله [UNK]فوق 😂 و **** على يا حلوه احنا على يا [UNK] ***** [end]	0.571	0.263	0.263	0.000
امشي يا سلمان يا ابن الوسخة انت خارب كل الدول م دايرين منك اي شي	[start]امشي يا سلمان يا ابن ****** انت خارب كل الدول م دايرين منك اي شي[end]	[start]امشي يا سلمان يا ابن ال**** انت [en]كل يا م م من اي شي [en] [en] [en]	0.550	0.383	0.383	0.136
لن تكون ارحم من المملكه وأبناءها وحكامها على قطر وأهل قطر يا مرتزق .. الله يأخذك وأشكالك ويفكنا شركم يا عبدة المال يا خونه انتم الفتنه قبحكم الله	[start]لن تكون ارحم من المملكه وأبناءها وحكامها على قطر وأهل قطر يا ***** .. الله يأخذك و****** ويفكنا شركم يا **** المال يا **** انتم ****** ***** الله[end]	[start]لن تكون ارحم من [UNK] [UNK] [UNK] [UNK]يا [UNK]قطر يا [UNK]الله [UNK] [UNK] [UNK] [UNK] يا[UNK]	0.341	0.209	0.209	0.160
يا ابن اللبوة يا معرص يا كلب السرايا الشعب هيفشخك ليك يووم يا ابن العاهره يا بتاع الخرابة طيزك حمرا يا ابن الوسخة و فيها شعر يا خول	[start]يا ابن ****** يا **** يا *** السرايا الشعب ****** ليك يووم يا ابن ******* يا بتاع الخرابة **** **** يا ابن ****** و فيها شعر يا *** [end]	[start]يا ابن ****** يا **** يا *** يا ابن الشعب [UNK]ليك على ويا ابن ويا بتاع **** [UNK] [UNK]	0.465	0.263	0.263	0.227

Table (5.6): Bad prediction samples

Input	Actual	Predicted	1-Gram	2-Gram	3-Gram	4-Gram
لعنة الله عليك وعلى والديك يا حمار يا اللي معك من اسمك نصيب حميري يعني حقي الحمير الشعب عرف انكم دجالين ولوطه ونهايتكم قربت واليمن محرم عليكم يا عيال موزه اتفوه على وجهك ووجه ابوك ياحق المسابح والورعان كرتكم	**** [start] الله عليك وعلى والديك يا **** يا اللي معك من اسمك نصيب ***** يعني حقي ****** الشعب عرف انكم ****** و**** ونهايتكم قربت واليمن محرم عليكم يا عيال موزه ***** على وجهك ووجه **** ياحق المسابح و******* كرتكم [end]	[start] [UNK] الله عليك [UNK] [UNK]يا ****يا [UNK] اللي الله [UNK] الفوز [en] الله من [UNK] [UNK] [en] [UNK]	0.157	0.055	0.055	0.000
بتهدو فى الكيان و بتتبها بفنلتة مش عارف اقول اية عليكم انتم مثل القاتل الذى يمشى فى جنازة القتيل اةة يا شوية مصلحجية اتفو عليكم وعلى الى مشغلكم يا واطين يا ولاد 60 فى 70	[start]بتهدو فى الكيان و بتتبها بفنلتة مش عارف اقول اية عليكم انتم مثل القاتل الذى يمشى فى جنازة القتيل اةة يا شوية ******* **** عليكم وعلى الى مشغلكم يا ***** يا ولاد 60 فى 70[end]	[start] [UNK] [UNK]يلا و [UNK] [UNK] مش عارف اقول اقول عليكم [UNK] [UNK] [UNK] [en] [UNK] [UNK] [UNK]الحب	0.200	0.070	0.070	0.000
#الزمالك_حسنية_اغادير استمرارهم في الكنفدراليه مصلحه جدا حيث الارهاق و الاصابات وضغط الماتشات ورينا بقي يا بيضه يا حلوه انتي هتتعملي معاهم ازاي بس ايه رايكم في الحكم كيوت مش كده اوعي الحلال يا عم الحج😂😂	# [start]الزمالك_حسنية_اغادير استمرارهم في الكنفدراليه مصلحه جدا حيث الارهاق و الاصابات وضغط الماتشات ورينا بقي يا بيضه يا حلوه انتي هتتعملي معاهم ازاي بس ايه رايكم في الحكم كيوت مش كده اوعي الحلال يا عم الحج[end] 😂😂	[start] [UNK] [UNK]في [UNK] [UNK] جدا قال [UNK] و [UNK] [UNK] [UNK] [UNK]بقي يا **** يا حلوه انتي[UNK]	0.200	0.070	0.070	0.000
#ضد_السناب_الامني_العنصري مجتمعكم كله عنصري .. يا عبد يا كويحه يا طعس يا طرش يا صفر سبعه يا لحجي يا مخلفات حجاج الله يرحم الملك فيصل يا حنس وغيرها .. واضح انه تجاوزهم بسبب عنصري والمضحك ترقيعته ان فيه صوت واحد معصب سبهان الله 😁 زود على اليتم عنصريه .. لازم يتحاسب	[start] #ضد_السناب_الامني_العنصري مجتمعكم كله ***** .. يا *** يا ***** يا *** يا *** يا *** سبعه يا **** يا ****** حجاج الله يرحم الملك فيصل يا حنس وغيرها .. واضح انه تجاوزهم بسبب ***** والمضحك ترقيعته ان فيه صوت واحد **** سبهان الله 😁 زود على اليتم ****** .. لازم يتحاسب[end]	[start] [UNK] [UNK]كله يا ****يا [UNK] يا [UNK] يا [UNK] [UNK] [UNK]يا ****يا [UNK] يا [en] [UNK] [UNK]	0.077	0.029	0.029	0.000

Chapter 6

Conclusion and Future Works

Conclusion

In this research we handle two problems: the first is Arabic hate speech detection using different neural networks architectures including RNN, CNN, and Transformers, and the second problem of cleaning offensive/hate speech texts. The dataset used is from the shared task SemEval-2020 (Zampieri et al., 2020) for Arabic offensive language detection.

In the hate speech detection task, we conduct several experiments to find the best model by checking the best macro F1 score and accuracy. The best Macro F1 score with 92% and accuracy of 95% was obtained by the QARiB model with the AraBERT preprocessor. And in cleaning hate speech texts, we use BLEU Score for evaluation, based on considering the problem of cleaning dirty text as a machine translation problem, and the best result achieved is 30% with 1-gram, which is achieved with dataset size 4383 and vocabulary size 12000 as explained in section 5.2.

As a summary of our work, the result of one of our experiments in hate speech detection outperformed the best results that are published in the Semeval 2020 shared task. And to the best of our knowledge, we worked on the first experiment in Arabic hate speech masking as a machine translation model, and it achieved a good result compared with the state of the art MT systems.

For future work, the parallel corpus that we use in the hate speech masking model will be increased because that will increase the BLEU score as we noted in our experiments, build web applications to deploy the hate speech detection and masking models, and publish the lexicon that we extracted when we built the parallel corpus, and which include hate/offensive words with the category for each of them. In additional, build a model for hate speech paraphrasing using a machine translation model.

References

Qiang, J., & Wu, X. (2019). Unsupervised Statistical Text Simplification. *IEEE Transactions on Knowledge and Data Engineering*, *PP*(8), 1–1. https://doi.org/10.1109/tkde.2019.2947679

Popov, M., Kulnitskiy, B., Perezhogin, I., Mordkovich, V., Ovsyannikov, D., Perfilov, S., Borisova, L., & Blank, V. (2018). Catalytic 3D polymerization of C 60. *Fullerenes Nanotubes and Carbon Nanostructures*, *26*(8), 465–470. https://doi.org/10.1080/1536383X.2018.1448388

Master Transfer learning by using Pre-trained Models in Deep Learning. (n.d.). Retrieved February 8, 2020, from https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/

Pre-trained Word Embeddings — gluonnlp 0.8.2 documentation. (n.d.). Retrieved February 8, 2020, from https://gluon-nlp.mxnet.io/examples/word_embedding/word_embedding.html

Brownlee Disclaimer, J. (2017). *Deep Learning for Natural Language Processing Develop Deep Learning Models for Natural Language in Python Acknowledgements Copyright Deep Learning for Natural Language Processing*.

artificial intelligence | Definition, Examples, and Applications | Britannica. (n.d.). Retrieved January 23, 2020, from https://www.britannica.com/technology/artificial-intelligence

Al-Hassan, A., & Al-Dossari, H. (2019). *Detection of Hate Speech in Social Networks: a Survey on Multilingual Corpus*. 83–100. https://doi.org/10.5121/csit.2019.90208

Casacuberta Nolla, F., & Peris Abril, Á. (2017). Neural Machine Translation. *Tradumàtica: Tecnologies de La Traducció*, *15*, 66. https://doi.org/10.5565/rev/tradumatica.203

Klabunde, R. (2002). Daniel Jurafsky/James H. Martin: Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition. In *Zeitschrift fur Sprachwissenschaft* (Vol. 21, Issue 1). https://doi.org/10.1515/zfsw.2002.21.1.134

Kwaik, K. A., Saad, M., Chatzikyriakidis, S., & Dobnika, S. (2018). A Lexical Distance Study of Arabic Dialects. *Procedia Computer Science*, *142*, 2–13. https://doi.org/10.1016/j.procs.2018.10.456

Fortuna, P. (2017). *Automatic detection of hate speech in text: an overview of the topic and dataset annotation with hierarchical classes*. 109. https://repositorio-aberto.up.pt/handle/10216/106028

RING, C. E. (2013). *hate speech in social media : an exploration of the problem and its proposed solutions by caitlin elizabeth ring b. A ., Clemson University , 2001 M . S ., University of Denver , 2004 A dissertation submitted to the Faculty of the Graduate School of the*.

Shtovba, S., Shtovba, O., & Petrychko, M. (2019). Detection of social network toxic comments with usage of syntactic dependencies in the sentences. *CEUR Workshop Proceedings*, *2353*, 313–323.

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, 1412–1421. https://doi.org/10.18653/v1/d15-1166

Saksesi, A. S., Nasrun, M., & Setianingsih, C. (2018). Analysis Text of Hate Speech Detection Using Recurrent Neural Network. *Proceedings - 2018 International Conference on Control, Electronics, Renewable Energy and Communications, ICCEREC 2018*, 242–248. https://doi.org/10.1109/ICCEREC.2018.8712104

Yadav, S. H., & Manwatkar, P. M. (2015). An approach for offensive text detection and prevention in Social Networks. *ICIIECS 2015 - 2015 IEEE*

International Conference on Innovations in Information, Embedded and Communication Systems, 3–6. https://doi.org/10.1109/ICIIECS.2015.7193018

Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Dataset Construction for the Detection of Anti-Social Behaviour in Online Communication in Arabic. *Procedia Computer Science, 142*, 174–181. https://doi.org/10.1016/j.procs.2018.10.473

i Orts, Ò. (2019). Multilingual Detection of Hate Speech Against Immigrants and Women in Twitter at {S}em{E}val-2019 Task 5: Frequency Analysis Interpolation for Hate in Speech Detection. *Proceedings of the 13th International Workshop on Semantic Evaluation, C*, 460–463. https://doi.org/10.18653/v1/S19-2081

Khou, K., & Narwal, N. (n.d.). *Detecting and Classifying Toxic Comments*.

Martins, R., Gomes, M., Almeida, J. J., Novais, P., & Henriques, P. (2018). Hate speech classification in social media using emotional analysis. *Proceedings - 2018 Brazilian Conference on Intelligent Systems, BRACIS 2018*, 61–66. https://doi.org/10.1109/BRACIS.2018.00019

Mubarak, H., Darwish, K., & Magdy, W. (2017). *Abusive Language Detection on Arabic Social Media*. 52–56. https://doi.org/10.18653/v1/w17-3008

Fortuna, P., & Nunes, S. (2018). A survey on automatic detection of hate speech in text. *ACM Computing Surveys*, *51*(4). https://doi.org/10.1145/3232676

Shanita Biere Supervisor dr Sandjai Bhulai, A. (2018). *Hate Speech Detection Using Natural Language Processing Techniques*.

Qiang, J., & Wu, X. (2019). Unsupervised Statistical Text Simplification. *IEEE Transactions on Knowledge and Data Engineering, PP*(8), 1–1. https://doi.org/10.1109/tkde.2019.2947679

Shanita Biere Supervisor dr Sandjai Bhulai, A. (2018). *Hate Speech Detection*

Using Natural Language Processing Techniques.

Qiang, J., & Wu, X. (2019). Unsupervised Statistical Text Simplification. *IEEE Transactions on Knowledge and Data Engineering*, *PP*(8), 1–1. https://doi.org/10.1109/tkde.2019.2947679

Shashirekha, H. and Nayel, H., (2019). DEEP at HASOC2019 : A Machine Learning Framework for Hate Speech and Offensive Language Detection.

Wiedemann, G., Ruppert, E., & Biemann, C. (2019). *UHH-LT at SemEval-2019 Task 6: Supervised vs. Unsupervised Transfer Learning for Offensive Language Detection*. 782–787. https://doi.org/10.18653/v1/s19-2137

Barbieri, F., Anke, L. E., & Camacho-Collados, J. (2021). *XLM-T: A Multilingual Language Model Toolkit for Twitter*. *2015*. http://arxiv.org/abs/2104.12250

Puranik, K., Hande, A., Priyadharshini, R., Thavareesan, S., & Chakravarthi, B. R. (2021). *IIITT@LT-EDI-EACL2021-Hope Speech Detection: There is always Hope in Transformers*. http://arxiv.org/abs/2104.09066

Ruder, S., Søgaard, A., & Vulic, I. (2019). Unsupervised cross-lingual representation learning. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Tutorial Abstracts*, 31–38. https://doi.org/10.18653/v1/p19-4007

Talafha, B., Ali, M., Za'ter, M. E., Seelawi, H., Tuffaha, I., Samir, M., Farhan, W., & Al-Natsheh, H. T. (2020). *Multi-Dialect Arabic BERT for Country-Level Dialect Identification*. http://arxiv.org/abs/2007.05612

Edunov, S., Baevski, A., & Auli, M. (2019). Pre-trained language model representations for language generation. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4052–4059. https://doi.org/10.18653/v1/n19-1409

Irie, K., Zeyer, A., Schlüter, R., & Ney, H. (2019). Language modeling with deep transformers. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *2019-Septe*, 3905–3909. https://doi.org/10.21437/Interspeech.2019-2225

Abdul-Mageed, M., Elmadany, A., & Nagoudi, E. M. B. (2020). *ARBERT & MARBERT: Deep Bidirectional Transformers for Arabic*. *i*. http://arxiv.org/abs/2101.01785

Antoun, W., Baly, F., & Hajj, H. (2020). *AraBERT: Transformer-based Model for Arabic Language Understanding*. http://arxiv.org/abs/2003.00104

Abu Farha, I., & Magdy, W. (2020). Multitask Learning for Arabic Offensive Language and Hate-Speech Detection. *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, *May*, 86–90. https://www.aclweb.org/anthology/2020.osact-1.14

Al-Yahya, M., Al-Khalifa, H., Al-Baity, H., Alsaeed, D., & Essam, A. (2021). Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches. *Complexity*, *2021*. https://doi.org/10.1155/2021/5516945

QCRI. (2020). *qcri/QARiB*. https://github.com/qcri/QARiB

Ben-Sghaier, M., Bakari, W., & Neji, M. (2020). Classification and analysis of Arabic natural language inference systems. *Procedia Computer Science*, *176*, 551–560. https://doi.org/10.1016/j.procs.2020.08.057

Zampieri, M., Nakov, P., Rosenthal, S., Atanasova, P., Karadzhov, G., Mubarak, H., Derczynski, L., Pitenis, Z., & Çöltekin, Ç. (2020). *SemEval-2020 Task 12: Multilingual Offensive Language Identification in Social Media (OffensEval 2020)*. *OffensEval*. http://arxiv.org/abs/2006.07235

Hettiarachchi, H., & Ranasinghe, T. (2020). *BRUMS at SemEval-2020 Task 12 : Transformer based Multilingual Offensive Language Identification in Social*

Media. https://www.researchgate.net/publication/351904204_BRUMS_at_SemEval-2020_Task_12_Transformer_based_Multilingual_Offensive_Language_Identification_in_Social_Media

Alami, H., Ouatik El Alaoui, S., Benlahbib, A., & En-nahnahi, N. (2020). LISAC FSDM-USMBA Team at SemEval-2020 Task 12: Overcoming AraBERT’s pretrain-finetune discrepancy for Arabic offensive language identification. Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2080–2085. https://www.aclweb.org/anthology/2020.semeval-1.275

Soliman, A. B., Eissa, K., & El-Beltagy, S. R. (2017). AraVec: A set of Arabic Word Embedding Models for use in Arabic NLP. Procedia Computer Science, 117(September), 256–265. https://doi.org/10.1016/j.procs.2017.10.117

Hassan, S., Samih, Y., Mubarak, H., & Abdelali, A. (2020). ALT at SemEval-2020 Task 12: Arabic and English Offensive Language Identification in Social Media. Proceedings of the Fourteenth Workshop on Semantic Evaluation, 2017, 1891–1897. https://www.aclweb.org/anthology/2020.semeval-1.249

Keleg, A., El-Beltagy, S. R., & Khalil, M. (2020). ASU OPTO at OSACT4-Offensive Language Detection for Arabic text. May, 11–16. https://github.com/LDNOOBW/List-of-Dirty-Naughty-

Mohaouchane, H., Mourhir, A., & Nikolov, N. S. (2019). Detecting Offensive Language on Arabic Social Media Using Deep Learning. 2019 6th International Conference on Social Networks Analysis, Management and Security, SNAMS 2019, December, 466–471. https://doi.org/10.1109/SNAMS.2019.8931839

Safaya, A., Abdullatif, M., & Yuret, D. (2020). KUISAIL at SemEval-2020 Task 12: BERT-CNN for Offensive Speech Identification in Social Media. Ml.

http://arxiv.org/abs/2007.13184

Mulki, H., Haddad, H., Bechikh Ali, C., & Alshabani, H. (2019). L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language. 111–118. https://doi.org/10.18653/v1/w19-3512

Hassan, S., Samih, Y., Mubarak, H., Abdelali, A., Rashed, A., & Chowdhury, S. A. (2020). ALT Submission for OSACT Shared Task on Offensive Language Detection. Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection, May, 61–65. https://www.aclweb.org/anthology/2020.osact-1.9

Alakrot, A., Murray, L., & Nikolov, N. S. (2018). Dataset Construction for the Detection of Anti-Social Behaviour in Online Communication in Arabic. Procedia Computer Science, 142, 174–181. https://doi.org/10.1016/j.procs.2018.10.473

Husain, F., & Uzuner, O. (2021). Transfer Learning Approach for Arabic Offensive Language Detection System. http://arxiv.org/abs/2102.05708

Alshalan, R., & Al-Khalifa, H. (2020). A deep learning approach for automatic hate speech detection in the saudi twittersphere. Applied Sciences (Switzerland), 10(23), 1–16. https://doi.org/10.3390/app10238614

Faris, H., Aljarah, I., Habib, M., & Castillo, P. A. (2020). Hate speech detection using word embedding and deep learning in the Arabic language context. ICPRAM 2020 - Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods, March, 453–460. https://doi.org/10.5220/0008954004530460

Nguyen, T. T., & Thesis, M. (2019). Machine Translation with Transformers. 1–64.

Sharma, S., Asri, L. El, Schulz, H., & Zumer, J. (2017). Relevance of Unsupervised Metrics in Task-Oriented Dialogue for Evaluating Natural Language Generation. http://arxiv.org/abs/1706.09799

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 2017-December(Nips), 5999–6009.

Handloser, D. (2020). Towards Diversity and Relevance in Neural Natural Language Response Generation. 多, August 2019.

Rothman, D. (2021). Transformers for Natural Language Processing.

Abdelali, A., Darwish, K., Durrani, N., & Mubarak, H. (2016). Farasa: A fast and furious segmenter for arabic. *NAACL-HLT 2016 - 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session*, *2016*, 11–16. https://doi.org/10.18653/v1/n16-3003

Aluru, S. S., Mathew, B., Saha, P., & Mukherjee, A. (2020). *Deep Learning Models for Multilingual Hate Speech Detection*. 1–16. http://arxiv.org/abs/2004.06465

Barbieri, F., Anke, L. E., & Camacho-Collados, J. (2021). *XLM-T: A Multilingual Language Model Toolkit for Twitter*. *2015*. http://arxiv.org/abs/2104.12250

Ruder, S., Søgaard, A., & Vulic, I. (2019). Unsupervised cross-lingual representation learning. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Tutorial Abstracts*, 31–38. https://doi.org/10.18653/v1/p19-4007

RING, C. E. (2013). *HATE SPEECH IN SOCIAL MEDIA : AN EXPLORATION OF THE PROBLEM AND ITS PROPOSED SOLUTIONS by CAITLIN ELIZABETH RING B . A ., Clemson University , 2001 M . S ., University of Denver , 2004 A dissertation submitted to the Faculty of the Graduate School of the*.

Ma, L., & Zhang, Y. (2015). Using Word2Vec to process big text data. *Proceedings - 2015 IEEE International Conference on Big Data, IEEE Big Data 2015, October 2015*, 2895–2897. https://doi.org/10.1109/BigData.2015.7364114