AMERICAN UNIVERSITY OF BEIRUT

# Transformers for Arabic Natural Language Understanding and Generation

by

Wissam Fares Antoun

A thesis
submitted in partial fulfillment of the requirements
for the degree of Master of Engineering
to the Department of Electrical and Computer Engineering
of the Faculty of Engineering and Architecture
at the American University of Beirut

Beirut, Lebanon
September 2020

AMERICAN UNIVERSITY OF BEIRUT

# Transformers for Arabic Natural Language Understanding and Generation

by

Wissam Fares Antoun

Approved by: Hazem

Prof. Hazem Hajj, Associate Professor — Advisor

Electrical and Computer Engineering

Hazem

Signed by Prof. Hazem Hajj On Behalf Of Prof. Mazen Saghir

Prof. Mazen Saghir, Associate Professor — Member of Committee

Electrical and Computer Engineering

Hazem

Signed by Prof. Hazem Hajj On Behalf Of Prof. Wassim El Hajj

Prof. Wassim El-Hajj, Associate Professor — Member of Committee

Computer Science

Date of thesis defense: September 3, 2020

# AMERICAN UNIVERSITY OF BEIRUT

# THESIS, DISSERTATION, PROJECT RELEASE FORM

Student Name: Wissam Fares Antoun

Last First Middle

☑ Master's Thesis ○ Master's Project ○ Doctoral Dissertation

☑ I authorize the American University of Beirut to: (a) reproduce hard or electronic copies of my thesis, dissertation, or project; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes.

☑ I authorize the American University of Beirut, to: (a) reproduce hard or electronic copies of it; (b) include such copies in the archives and digital repositories of the University; and (c) make freely available such copies to third parties for research or educational purposes after: **One ___ year from the date of submission of my thesis, dissertation or project.**
**Two ___ years from the date of submission of my thesis , dissertation or project.**
**Three ✓ years from the date of submission of my thesis , dissertation or project.**

18-Sep-2020

Signature Date

This form is signed when submitting the thesis, dissertation, or project to the University Libraries

# Acknowledgements

I would first like to express my sincere gratitude to my advisor Prof. Hazem Hajj for the continuous support of my studies and related research, for his patience and motivation. His guidance and encouragement in this difficult time has been invaluable and has motivated me to persist and to achieve my goals. I could not have imagined having a better mentor. I would also like to thank Prof. Mazen Saghir and Prof. Wassim El-Hajj for being part of my thesis committee.

I thank my fellow colleagues and labmates for their continuous encouragement and help, and for their willingness to listen to my problems.

Last but not the least, I would like to thank my family: my parents and my sister for supporting me throughout my thesis.

# An Abstract of the Thesis of

Wissam Fares Antoun for Master of Engineering
Major: Electrical and Computer Engineering

Title: Transformers for Arabic Natural Language Understanding and Generation

Natural Language Processing (NLP) aims at advancing Artificial Intelligence by developing methods that enable machines to process language like humans do. While there has been significant breakthroughs in English NLP with the introduction of Machine Learning (ML) models called Transformers, Arabic NLP has been lagging behind, due to the lack of large scale data needed by these new models. Transformers represent special types of deep learning (DL) architectures, where the models learns to combine and weigh the different internal representations of a sentence. Furthermore, Arabic presents its own challenges such as the lexical sparsity, complex and concatenative morphology. This work aims to advance Arabic NLP tasks and bring the performances closer to English NLP. We propose multiple Transformer-based models that are specifically developed for Arabic Natural Language Understanding (NLU) and Generation (NLG). For Arabic NLU, we developed an Arabic centric Bidirectional Encoder Representations from Transformers, called AraBERT, bridging the gap with the English model BERT developed by Google. The model is comprised of 110 million parameters. For Arabic NLG, we proposed a Transformer-based encoder-decoder architecture to address challenges for Arabic open-domain chatbots. We built a large conversational dataset annotated for the gender of both interlocutors. The resulting model is the first open-domain gender-aware Arabic chatbot. For NLU experiments, we applied AraBERT to Arabic text classification and Arabic question answering. The performance showed state-of-the-art performance compared to multilingual BERT. For NLG Experiments, the results showed success of the model in achieving simple open-ended Arabic conversations, demonstrating basic world knowledge, and common-sense reasoning.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

NLP can be divided into two parts: Natural Language Understanding (NLU) and Natural Language Generation (NLG).

- **NLU:** deals with the machine's ability to comprehend and reason about spoken or written text. NLU can be applied to solve real-world problems like Sentiment Analysis (SA), Question Answering (QA), Named-Entity Recognition (NER)...

- **NLG:** deals with the machine's ability to produce human-readable text. NLG is used in tasks like Image Captioning, summarization, translation, Open-Domain QA (or Chatbots), etc. Although most NLG systems still require an understanding of the input text.

Recently, Natural Language Processing (NLP) experienced significant breakthroughs with advances in Deep Learning (DL) systems for extracting text semantics. These advances were marked by the introduction of neural network models such as the Recurrent Neural Network (RNN) [3] and Long Short-Term Memory (LSTM) [4] networks. The choice of such networks was motivated by their capabilities to deal with input with dynamic length and order-dependent variables which enable models to learn semantic text representations.

Early pretrained text representation models aimed at representing words by capturing their distributed syntactic and semantic properties using techniques like Word2vec [5] and GloVe [6]. However, these embedding models did not incorporate the context in which a word appears. This issue was addressed by generating contextualized representations using models like ELMO [7]. Recently, there has been a focus [8] on applying transfer learning by fine-tuning large pretrained language models for downstream NLP tasks with a relatively small number of examples, resulting in notable performance improvement for NLP tasks. This approach takes advantage of the language models that had been pre-trained in an unsupervised manner, sometimes called self-supervised. A key issue with these models is scalability, which breaks training optimization and parallelization. This is due to the inherent time dependency in the recurrent architecture. The issue was addressed with the introduction of the Transformer architecture [9], which replaces all the recurrent layers with deeper self-attention layers that are easier to train and scale. However, this advantage comes with drawbacks, particularly the huge corpora needed for pre-training, in addition to the high computational cost for training. Latest models required 500+ TPUs[1] or GPUs running for weeks [10, 11, 12].

There has been significant advances in NLP research for few languages, mainly English, Chinese, and Spanish. Other languages are lagging behind to the lack of quality data necessary to develop powerful systems. While Arabic is the 5th most spoken language in the world and the official language for more than 20 countries, it is still considered an under-resourced language for meeting the needs of new advanced NLP models. Additionally, Arabic is a morphologically rich language with a less explored syntax compared to English which often results in Out-of-Vocabulary (OOV) problems with DL systems. Given these limitations, Arabic Natural Language Processing (ANLP) tasks like Named Entity Recognition (NER), Question Answering (QA), Open-Domain chat have proven to be challenging to tackle and still lack behind their English counterparts. To remedy this gap, multilingual models have been trained to learn representations for more than 100 languages simultaneously, but still fall behind single-language models due to little data representation and small language-specific vocabulary. While languages with similar structure and vocabulary can benefit from shared representations [10], the difference in Arabic compared to other languages limits its ability. This thesis aims at advancing Arabic NLP by developing transformers for Arabic and achieving state-of-the-art (SOTA) in many ANLP tasks.

The high-level contributions of this work can be summarized as follows:

- Contributions for Arabic NLU:

    1. A methodology to pretrain a transformer-based universal language model for Arabic, which we name AraBERT.
    2. Using the developed AraBERT to achieve state of the art in four Arabic NLU downstream tasks: Sentiment Analysis, Offensive Language Identification, Hate Speech Detection, Question Answering.

- Contributions for Arabic NLG:

    1. A methodology for end-to-end training of the first open-domain Arabic conversational agent, using a transformer-based encoder decoder architecture.
    2. Address challenges in Arabic open-domain conversations by developing a model that can simultaneously handle gender-aware and Arabic OOV limitations in a generative DL model.

[1] `https://en.wikipedia.org/wiki/Tensor_processing_unit`

This work further contributed in valuable open-access resources for Arabic NLP:

- AraBERT is publicly released on popular NLP libraries[2].
- A large Arabic conversational dataset annotated for the speaker's and the listener's gender.
- The open-domain Arabic chatbot's model weights and code are publicly available on our repository[3].

[2] `github.com/aub-mind/arabert`, `huggingface.co/aubmindlab`

[3] `github.com/WissamAntoun/Arabic-Chatbot`

# Chapter 2

# Preliminary Work on English Transformers for Fake News Detection

This chapter[1] presents state of the art methods for addressing important challenges in automated fake news detection: fake news detection and domain identification. These models were developed as part of the Qatar International Cybersecurity Contest (QICC) on Fake News Detection. The proposed solutions for both tasks relies on advances in Natural Language Understanding (NLU) end to end deep learning models to identify stylistic differences between legitimate and fake news articles which proved that it outperforms the contest winning approach.

## 2.1 Introduction

Fake news articles are typically created with the goal of deceiving or misleading readers [14]. As an example, earlier cases for writing fake news articles were used to increase profit by directing web-traffic with "Clickbait" content[2], and were often designed and written to go viral by targeting controversial topics.

Nowadays, the majority of people rely on social media as their news source. In the USA, 62% of American adults get some of their news from social networking sites [15]. Generally, non-expert users can't infer the validity of news they read, and humans have 70% success rate in fake news detection [16]. As a result, social media networks have become fertile grounds for spreading fake news, which is an emerging type of cybersecurity threat. Accordingly, it has become important to

[1] This chapter is a slightly modified version of *State of the Art Models for Fake News Detection Tasks* [13]. The work that has been reproduces here only includes the work that I participated in.

[2] Clickbait: is a kind of deceptive or misleading false advertisement that exploits users curiosity to attract attention to follow a link - wikipedia.org/wiki/Clickbait

provide tools for automated detection of fake news.

There have been numerous research and industry efforts to automate and highlight fake news. Facebook is among the platforms with a large share of false news articles such as those preceding the 2016 US election [17]. To address the risk for fake news spreading, Facebook started adding tags to stories that can be flagged as false by fact-checkers. For the upcoming US election, Facebook is planning to label posts as "False Information", but this new policy will exclude Facebook ads placed by politicians [18]. In 2017, another effort was spearheaded for fact checking as the first step in fake news detection. Fake News Challenge (FNC-1) [19] was organized to develop new advances in intelligent systems for stance detection to predict one of four stance labels when comparing a document to its headline: Agree, disagree, discuss, and unrelated. The top ranked system used a weighted average model of a Convolutional Neural Network (CNN) and a gradient-boosting decision tree model [20]. The top three ranked models were further evaluated [21], and it was concluded that even the best performing features were not able to fully resolve the hard cases where even humans confused the agree, disagree and discuss labels. Another work on fake news detection focused on linguistic features such as Ngrams, punctuation, syntax, readability metrics and Psycholinguistic features (using the Linguistic Inquiry and Word Count (LIWC) tool)[16]. These features show that the difference in the style of writing can be exploited to detect the legitimacy of the content.

Despite the progress in fake news detection, accuracy performances remain limited in practical systems. To advance the field from this perspective, a contest was recently held as part of the Qatar International Cybersecurity Contest (QICC) [22]. The contest aimed at providing systems that have the best performances in practical applications with two challenging tracks:

- The Fake News Detection task aimed at detecting if an article is fake news or legitimate news.

- The News Domain Detection task aimed at detecting the news domain of an article: Politics, Business, Sports, Entertainment, Technology, or Education. The motivation is that different domains may require different approaches for the detection of fake news.

In this chapter, we present our approach that tackled these challenges. For fake news detection, our approach was motivated by the assumption that fake news would show stylistic differences hidden in the writing. The method consists of a classifier that uses state-of-the-art transformer-based language models to extract features from the text with the objective of detecting deep semantic differences in the writing. This method was inspired by the recent release of the 1.5B parameter GPT-2 natural language generation model along with a RoBERTa based detector that helps detect the output of the GPT-2 models [23]. For news domain identification, we also employ transformer-based models since these models can better distinguish between different domains by relying on their language comprehension gained from pretraining. The key contributions presented are the introduction of state-of-the-art approaches for fake news detection, news domain identification.

The rest of this chapter is organized as follows: In section 2.2, we present a brief overview of the literature for the two challenge areas of fake news detection. Section 2.3 describes the proposed methods and section 2.4 covers the experiments and results. Finally, the conclusion is presented in section 2.5.

## 2.2 Literature Review

### 2.2.1 Fake News Detection

Current research on fake news detection can be divided into three types of approaches: propagation based, source analysis, and content based. Propagation-based research suggests that the spread of fake news behaves differently than reliable news. These dissemination patterns can be used to flag news as false or true based on the propagation map [24]. Source analysis approaches depend completely on analysing the source of the news piece and its behavior. This allows for early detection and for a more robust way to contain the spread of false news [25]. Content based techniques focuses on extracting linguistic features, both lexical or syntactic. It assumes that fake news articles are written using deceptive language and syntactic styles[26, 27, 28, 29, 16]. A new approach for stance detection was suggested by [30] combining multi-layer perceptron (MLP) representation with hand crafted features from the FNC-1 dataset. Skip-thought vectors are used to encode the headline and the body of each article. The hand crafted features include n-grams, char-grams, weighted TF-IDF score between body and heading of each article. Following the work of [30] on stance detection, [31] proposed to use bi-directional Recurrent Neural Networks (RNNs), together with neural attention, for encoding the headline of a news article, the first two sentences of a news article, and the entire news article. These representations are then combined with hand crafted features as used in [30].

### 2.2.2 News Domain Detection

Models for topic detection [32, 33, 34, 35, 36] can be divided into two main categories: deterministic models and probabilistic models. Deterministic models treat topics and texts as points in space through Vector Space Models (VSM) [37]. In [38], named entity recognition was used for new topic detection [33]. Some researchers usually use a tokenizer for word segmentation to model key topic detection [39, 40, 41]. However, new words and nonstandard writings make these models ineffective [33]. Probabilistic models treat topics and texts as probability distributions, which can be represented by Latent Dirichlet Allocation (LDA) [42], Author-LDA [43], Labeled LDA [44], TweetLDA [45], and other statistical representations [33]. Recent research uses deep learning algorithms for topic classification such as random multimodel deep learning (RMDL) introduced in [46], Trigger-aware Lattice Neural Network (TLNN) introduced in [47] and Dual-CNN [48].

## 2.3 Systems Description

This section describes the proposed methods to advance the accuracy of fake news detection in two different aspects, which were also included in the QICC competition [22]: Fake News Detection and News Domain Identification.

### 2.3.1 Fake News Detection

The objective for this task is to develop an approach that can accurately identify whether a news article is fake or legitimate. We present the model that outperformed the winning model at the contest which was developed after the competition ended.

In order for a model to detect the stylistic differences in the writing, the model has to have a deep understanding of the English language and its underlying semantics. Hence for the classifier choice, we chose to experiment with the state-of-the-art transformer-based language model and in particular XLNET[49] and RoBERTa[50]. We also compare against BERT[51]. Since the difference in the writing style can be hidden in every word, no pre-processing is needed for this model. We intentionally leave the articles as they are, and let the respective tokenizer of each language model handle the cleaning and tokenization. The deep learning classification models used are all based on pre-trained language models with a classification layer on top that will be fine-tuned for the fake news task.

### 2.3.2 News Domain Detection

The objective for this task is to develop a model that can accurately identify the topic domain of a news article into one of the following categories: Politics, Business, Sports, Education, Entertainment, or Technology. The proposed model also relies on a transformer-based language model in particulary RoBERTa with a classification layer added on top of it. Several other models are also considered including: N-gram and N-char TF-IDF with SVM, NB, RF, XGBoost, Deep Neural Network (DNN), stacked Convolutional Neural Network (CNN), LSTM, Gated Recurrent Unit (GRU), Bi-LSTM, 3 Concatenated CNN (3CCNN), BERT, XLNET, RoBERTa and RMDL which is a combination of TF-IDF with DNNs and word-to-vector embedding using Glove [52] with RNNs and CNNs. We also present experiment with pre-trained language models BERT, XLNET.

## 2.4 Experiments and Results

In this section, we show the results of the different approaches used in each task and highlight the insights behind the results. We also present an analysis of the different features from the fake news task. All experiments with transformer-based models were run on Google's Colab GPU enviroment, while the other experiments were conducted on local machines. Hyper-parameters' details for each model were obtained through randomized grid search and are available in our Github repository[3].

### 2.4.1 Datasets

The dataset, provided by QICC, consisted of 384 articles for training (192 fake and 192 legitimate) the models for fake news detection and domain identification (Tracks 1A.1 and 1A.2). Additionally, 48 articles (24 fake and 24 legitimate) were provided as the development set. Most of the articles consisted of heading and bodies however some of them were without heading. Moreover, the text was given in English. The News Domain dataset consisted of the same articles from the fake news track and are split into 6 categories: Politics, Business, Sports, Entertainment, Technology, and Education. There were 64 articles per category for training and 8 articles per category for development. The final submission for evaluation consisted of 45 articles.

### 2.4.2 Fake News Detection

The precision, recall and F1-score were used for model evaluation and the results of each classifier are summarized in Table 2.1.

Table 2.1: Fake news detection results.

| Model | Precision (%) | Recall (%) | F1-score (%) |
|---|---|---|---|
| mBERT-base | 92 | **100** | 96 |
| XLNET-base | **98** | 98 | **98** |
| RoBERTa | 92 | **100** | 96 |

The results show that the pre-trained deep learning language model classifier XLNET-base achieved the best performance, this is due to the superiotrity of the

[3] https://github.com/aub-mind/fake-news-detection

pre-training objective that XLNET uses. These results indicate that deep understanding of the language is necessary to detect the subtle stylistics differences in the writing of the fake articles. It was also noted that during the fine-tuning process, the pre-trained language models required only one epoch to learn the objective.

### 2.4.3 News Domain Detection

The hyper-parameters used for this task are detailed in our Github repository. The results of this task are shown in Table 2.2.

Table 2.2: News Domain Detection results.

| **Model** | **Precision(%)** | **Recall(%)** | **F1-score(%)** |
|---|---|---|---|
| TF-IDF with NB | 82 | 81 | 80 |
| TF-IDF with SVM | 73 | 67 | 68 |
| TF-IDF with RF | 70 | 69 | 68 |
| TF-IDF with Xgboost | 76 | 75 | 75 |
| CNN* | 83 | 82 | 82 |
| LSTM* | 83 | 79 | 79 |
| GRU* | 80 | 77 | 76 |
| Bi-LSTM* | 84 | 81 | 80 |
| 3CCNN* | 81 | 81 | 80 |
| Bi-LSTM/Attention* | 86 | 85 | 85 |
| mBERT-base | 91 | 90 | 90 |
| XLNET-base | 93 | 90 | 89 |
| RoBERTa | **94** | **94** | **94** |

* Pre-trained word Embeddings from Fasttext

The pretrained language models achieved the best results and specifically RoBERTa. Furthermore, when analyzing the results of the best models, we notice that, out of the 4 classification errors, 3 articles were consistently being mislabeled. As it turns out the reason for the mislabeling, can be attributed to the article being part of two domains.

## 2.5 Conclusion

In this chapter, we presented the state of the art models that beat first place in an international fake news competition, while tackling two challenges: Fake News Detection, Domain Identification. For fake news detection, we concluded with state-of-the-art approach based on XLNET. For news domain detection, RoBERTa was the best performer. Insights from the experiments showed that stylistic differences can be used to detect fake news. The experiments also showed the superiority of advances in language models that can provide a deep understanding of the language for multiple tasks. For future work, we suggest improving fake news performance by adding features from fact-checking websites in addition to Google searches.

# Chapter 3

# Transformers for Arabic NLU

In this chapter[1], we describe the process of pretraining the BERT transformer model [51] for the Arabic language, which we name AraBERT, we also describe the process of finetuning BERT multilingual (BERTm). We evaluate AraBERT and BERTm on three Arabic NLU downstream tasks that are different in nature: Sentiment Analysis, Offensive Language/Hate Speech Detection, and Question Answering. The experiments results show that AraBERT achieves state-of-the-art performances on most datasets, compared to several baselines including previous multilingual and single-language approaches. The datasets that we considered for the downstream tasks contained both Modern Standard Arabic (MSA) and Dialectal Arabic (DA).

Our contributions can be summarized as follows:

- A methodology to pretrain the BERT model on a large-scale Arabic corpus.
- Application of AraBERT to three Arabic NLU downstream tasks: Sentiment Analysis, Offensive Language/Hate Speech Detection, and Question Answering.
- Publicly releasing AraBert on popular NLP libraries[2].

The rest of the chapter is structured as follows. Section 3.1 provides a concise literature review of previous work on language representation for English and Arabic. Section 3.2 reviews related work on three Arabic NLU downstream tasks: Sentiment Analysis, Offensive Language/Hate Speech Detection, and Question Answering. Section 3.3 details the process of fine-tuning BERT multilingual for Arabic NLP tasks. Section 3.4 describes the methodology that was used to develop AraBERT. Section 3.5 describes the downstream tasks and benchmark

[1] This chapter is a slightly modified version of *AraBERT: Transformer-based Model for Arabic Language Understanding* [53], *hULMonA: The universal language model in Arabic* [54], and *Multi-Task Learning using AraBert for Offensive Language Detection* [55]. The work that has been reproduces here only includes the work that I participated in.

[2] `https://github.com/aub-mind/arabert`, `https://huggingface.co/aubmindlab`

datasets that are used for evaluation. Section 3.5.3 presents the experimental setup. Section 3.5.4 shows and discusses the results.. Finally, section 3.6 concludes and points to possible directions for future work.

## 3.1 Literature Review on Universal Language Models

**Evolution of Word Embeddings** The first meaningful representations for words started with the word2vec model developed by [5]. Since then, research started moving towards variations of word2vec like of GloVe [6] and fastText [56]. While major advances were achieved with these early models, they still lacked contextualized information, which was tackled by ELMO [7]. The performance over different tasks improved noticeably, leading to larger structures that had superior word and sentence representations. Ever since, more language understanding models have been developed such as ULMFit [8], BERT [51], RoBERTa [50], XLNet [49], ALBERT [57], and T5 [11], which offered improved performance by exploring different pretraining methods, modified model architectures and larger training corpora.

**Non-contextual Representations for Arabic** Following the success of the English word2vec [5], the same feat was sought by NLP researchers to create language specific embeddings. Arabic word2vec was first attempted by [58], and then followed by a Fasttext model [59] trained on Wikipedia data and showing better performance than word2vec. To tackle dialectal variations in Arabic [60] presented techniques for training multidialectal word embeddings on relatively small and noisy corpora, while [61, 62] provided Arabic word embeddings trained on ∼250M tweets.

**Contextualized Representations for Arabic** For non-English languages, Google released a multilingual BERT [51] supporting 100+ languages with solid performance for most languages. However, pre-training monolingual BERT for non-English languages proved to provide better performance than the multilingual BERT such as Italian BERT Alberto [63] and other publicly available BERTs [64, 65]. Arabic specific contextualized representations models, such as hULMonA [54], used the ULMfit structure, which had a lower performance that BERT on English NLP Tasks.

## 3.2 Literature Review on Arabic NLP tasks

### 3.2.1 Sentiment Analysis

Sentiment Analysis is a popular Arabic NLP task. Previous approaches relied on sentiment lexicons such as ArSenL [66], which is a large-scale lexicon of MSA words that is developed using the Arabic WordNet in combination with the English SentiWordNet. Recurrent and recursive neural networks were explored with different choices of Arabic-specific processing [67, 68, 69]. Convolutional Neural Networks (CNN) were trained with pre-trained word embeddings [70]. A hybrid model was proposed by [61], where CNNs were used for feature extraction, and LSTMs were used for sequence and context understanding. Current state-of-the-art results are achieved by the hULMonA model [54], which is an Arabic language model that is based on the ULMfit architecture [8].

### 3.2.2 Offensive Language and Hate Speech Detection

**Hate Speech Detection** An extensive overview of the different works on hate speech detection was done by [71], but very few works in the literature target the problem of Arabic hate speech detection. Albadi et al. [72] introduced the first dataset containing 6.6K Arabic hate-speech tweets targeting religious groups. The authors compared a lexicon-based classifier, SVM classifier trained with character n-gram features, and a Deep Learning approach consisting of a GRU trained on AraVec embeddings [58]. The GRU approach outperformed all other approaches with a 77% F1 score.

**Offensive Language Detection** For offensive language detection in Arabic, different approaches can be found in the literature. Alakrot et al. [73], introduced a dataset for offensive speech in Arabic collected from 15K YouTube comments. For classifying the different comments, the data was preprocessed by removing stop words and diacritics, correcting misspelled words, then tokenization and stemming was performed in order to extract features that are used by a binary SVM classifier. Mohaouchane et al. [74], explored the use of different Deep Learning architectures for offensive language detection. AraVec embeddings of each comment were used to train several models: CNN-LSTM, CNN-BiLSTM with attention, Bi-LSTM, and CNN model on the dataset proposed in [73] where the CNN model was found to provide the best F1 score. In Mubarak et al. [75] 36 million tweets were collected and used it to train a FastText deep learning model and SVM classifier on character n-gram features where it was found that the Arabic FastText DL model provided the best results.

### 3.2.3 Question Answering

Question Answering is one of the goals of artificial intelligence, this goal can be achieved by leveraging natural language understanding and knowledge gathering [76]. English QA research has been fueled by the release of large datasets such as Stanford Question Answering Dataset (SQuAD) [77]. On the other hand, research in Arabic QA has been hindered by the lack of such massive datasets, and by the fact that Arabic presents its own challenges such as:

- Inconsistent name spelling (ex: Syria in Arabic can be written as “سوريا - *sOriyA*” and “سورية - *sOriyT*” )
- Name de-spacing (ex: The name is written as “عبدالعزيز - *AbdulAzIz*” in the question, and “عبد العزيز - *Abdul AzIz*” in the answer)
- Dual form “المثنى”, which can have multiple forms (ex: “قلمان” - “*qalamAn*” or “قلمين” - “*qalamyn*” meaning “*two pencils*”)
- Grammatical gender variation: all nouns, animate and inanimate objects are classified under two genders either masculine or feminine (ex: “كبير” - “*kabIr*” and “كبيرة” - “*kabIrT*”

## 3.3 BERT

Bidirectional Encoder Representations from Transformers or (BERT) [51] is an architecture and pre-training method that achieved state-of-the-art results on 11 English NLP tasks when released by Google. The BERT architecture was inspired by the Transformer architecture [9], stacks multi-head self-attention layers to form a better text representation then older recurrent architecture. Another advantage of such architecture, is the ability to scale such models into the billion-parameter range which has shown to improve performance and scores. Two versions of BERT were initially released BERT-base, BERT-large for English and Chinese separately and another version that support 100+ languages (incl. Arabic) called BERT-base multilingual (BERTm).

BERTm was trained on the top 100 largest Wikipedia languages, and uses a 110K shared WordPiece vocabulary. Both the training data and the vocabulary weighted by the size of the training data for a given language, in order to balance the discrepancy in the data size between languages.

**Data Pre-processing and Tokenization** BERT model requires a special format for the input data before feeding it into the model. A special token, called [CLS], is added at the beginning of every sentence and a special token, called [SEP] is added at the end of every sequence. For Arabic tokenization, we chose WordPiece[78] tokenizer as it was also used during the pre-training of BERT. Figure 3.1 presents a sentence before and after going through the BERT tokenizer.

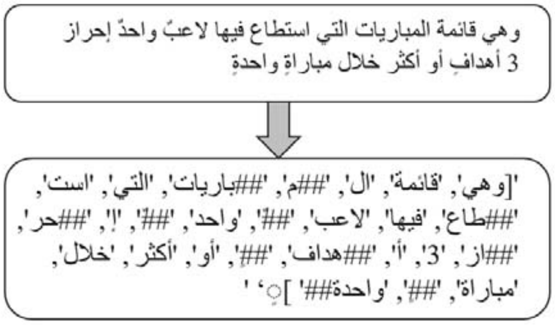

وهي قائمة المباريات التي استطاع فيها لاعبٌ واحدٌ إحراز
3 أهدافِ أو أكثر خلال مباراةٍ واحدةٍ
['وهي', 'قائمة', 'ال', '##م', '##باريات', 'التي', 'است',
'##طاع', 'فيها', 'لاعب', '##ٌ', 'واحد', '##ٌ', 'إ', '##حر',
'##از', '3', 'أ', '##هداف', '##ِ', 'أو', 'أكثر', 'خلال',
'مباراة', '##ٍ', 'واحدة##' ]ٍ' '

Figure 3.1: BERT Tokenizer Results

The tokenizer splits words into WordPiece tokens separated by ##. After tokenization, each word is mapped to an index using a 110k token vocabulary file that is provided by BERT for all the languages.

Since BERTm uses a shared weighted vocabulary between all languages, Arabic tokens account for only 4% or 4873 out of 110K, which, given how rich is the Arabic language, greatly under-represent the Arabic Language. Also these token are shared with the Urdu language since both use the same set of alphabets which creates an even sparser representation of Arabic text.

## 3.4 AraBERT: Pre-training Methodology

We create AraBERT based on the BERT model architecture. We use the BERT-base configuration that has 12 encoder blocks, 768 hidden dimensions, 12 attention heads, 512 maximum sequence length, and a total of ∼110M parameters[3]. We also introduced additional preprocessing prior to the model's pre-training, in order to better fit the Arabic language. Below, we describe the pre-training setup, the pre-training dataset for AraBERT, the proposed Arabic-specific preprocessing, and the fine-tuning process.

[3] Further details about the transformer architecture can be found in [9]

### 3.4.1 Pre-training Setup

Following the original BERT pre-training objective, we employ the *Masked Language Modeling* (MLM) task by adding whole-word masking where; 15% of the $N$ input tokens are selected for replacement. Those tokens are replaced 80% of the times with the [MASK] token, 10% with a random token, and 10% with the original token. Whole-word masking improves the pre-training task by forcing the model to predict the whole word instead of getting hints from parts of the word. We also employ the *Next Sentence Prediction* (NSP) task that helps the model understand the relationship between two sentences, which can be useful for many language understanding tasks such as Question Answering.

### 3.4.2 Pre-training Dataset

The original BERT was trained on 3.3B words extracted from English Wikipedia and the Book Corpus [79]. Since the Arabic Wikipedia Dumps are small compared to the English ones, we manually scraped Arabic news websites for articles. In addition, we used two publicly available large Arabic corpora: (1) the 1.5 billion words Arabic Corpus [80], which is a contemporary corpus that includes more than 5 million articles extracted from ten major news sources covering 8 countries, and (2) OSIAN: the Open Source International Arabic News Corpus [81] that consists of 3.5 million articles (∼1B tokens) from 31 news sources in 24 Arab countries.

The final size of the pre-training dataset, after removing duplicate sentences, is 70 million sentences, corresponding to ∼24GB of text. This dataset covers news from different media in different Arab regions, and therefore can be representative of a wide range of topics discussed in the Arab world. It is worth mentioning that we preserved words that include Latin characters, since it is common to mention named entities, scientific or technical terms in their original language, to avoid information loss.

### 3.4.3 Sub-Word Units Segmentation

Arabic language is known for its lexical sparsity which is due to the complex concatenative system of Arabic [68]. Words can have different forms and share the same meaning. For instance, while the definite article "ال - *Al*", which is equivalent to "the" in English, is always prefixed to other words, it is not an intrinsic part of that word. Hence, when using a BERT-compatible tokenization, tokens will appear twice, once with "*Al-*" and once without it. For instance, both

"كتاب- *kitAb*" and "الكتاب-*AlkitAb*" need to be included in the vocabulary, leading to a significant amount of unnecessary redundancy.

To avoid this issue, we first segment the words using Farasa [82] into stems, prefixes and suffixes. For instance, "اللّغة - *Alloga*" becomes ال+ لغ +ة - *Al+ log +a*". Then, we trained a SentencePiece (an unsupervised text tokenizer and detokenizer [83]), in unigram mode, on the segmented pre-training dataset to produce a subword vocabulary of ∼60K tokens. To evaluate the impact of the proposed tokenization, we also trained SentencePiece on non-segmented text to create a second version of AraBERT (AraBERTv0.1) that does not require any segmentation. The final size of vocabulary was 64k tokens, which included nearly 4K unused tokens to allow further pre-training, if needed.

## 3.5 Applications of AraBERT for NLP tasks

Although AraBERT was pre-trained on MLM and NSP, the model needs to be further trained and fine-tuned for specific NLP tasks, since the pre-training step only serves to "familiarize" the model with the Arabic language. We now describe the fine-tuning process and the use cases and applications of AraBERT, by evaluating AraBERT on three Arabic language understanding downstream tasks: Sentiment Analysis, Offensive Language/Hate Speech Detection, and Question Answering. As a baseline, we compared AraBERT to the multilingual version of BERT, and to other state-of-art results on each task.

### 3.5.1 Fine-tuning

**Sequence Classification** To fine-tune AraBERT for sequence classification (Sentimen Analysis and Offensive Language/Hate Speech Detection), a linear (fully-connected) layer with a standard softmax activation function is added to the last hidden state of the first token (the [CLS] token) as shown in Figure 3.2. With a hidden state vector $C \in R^H$ where H is the dimension of the hidden state and a fully-connected classification layer with weights $W \in R^{K \times H}$ where K is the number of classification labels, the label probability after applying the softmax function is then $P = softmax(CW^T)$. During fine-tuning, the classifier and the pre-trained model weights are trained jointly to maximize the log-probability of the correct class.

**Question Answering** In the QA, given a question and a passage containing the answer, the model needs to select a span of text that contains the answers. This is done by predicting a "start" token and an "end" token on condition that the "end" token should appear after the "start" token. During training, the final embedding of every token in the passage is fed into two classifiers, each with a single set of weights, which are applied to every token. The dot product

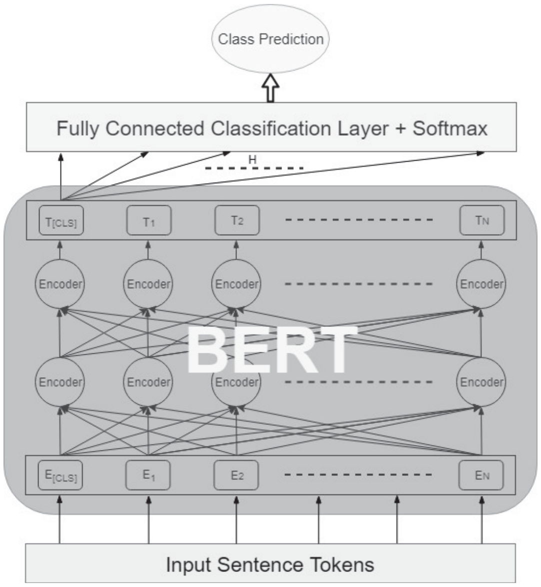


Figure 3.2: BERT Fine-Tuning Model Architecture

of the output embeddings and the classifier is then fed into a softmax layer to produce a probability distribution over all the tokens. The token with the highest probability of being a "start" toke is then selected, and the same process is repeated for the "end" token.

### 3.5.2 Evaluation Datasets

**A. Sentiment Analysis:**

We evaluated AraBERT on the following Arabic sentiment datasets that cover different genres, domains and dialects.

- **HARD:** The Hotel Arabic Reviews Dataset [84] contains 93,700 hotel reviews written in both Modern Standard Arabic (MSA) and in dialectal Arabic. Reviews are split into positive and negative reviews, where a negative review has a rating of 1 or 2, a positive review has a rating of 4 or 5, and neutral reviews with rating of 3 were ignored.

- **ASTD:** The Arabic Sentiment Twitter Dataset [85] contains 10,000 tweets written in both MSA and Egyptian dialect. We tested on the balanced version of the dataset, referred to as ASTD-B.

- **ArSenTD-Lev:** The Arabic Sentiment Twitter Dataset for LEVantine [86] contains 4,000 tweets written in Levantine dialect with annotations for sentiment, topic and sentiment target. This is a challenging dataset as the collected tweets are from multiple domains and discuss different topics.

- **LABR:** The Large-scale Arabic Book Reviews dataset [87] contains 63,000 book reviews written in Arabic. The reviews are rated between 1 and 5. We benchmarked our model on the unbalanced two-class dataset, where reviews with ratings of 1 or 2 are considered negative, while those with ratings of 4 or 5 are considered positive.

- **AJGT:** The Arabic Jordanian General Tweets dataset [88] contains 1,800 tweets written in Jordanian dialect. The tweets were manually annotated as either positive or negative.

We compare the results of AraBERT to those of hULMonA.

### B. Offensive Language and Hate Speech Detection

We evaluate AraBERT on the Offensive Language and Hate Speech Detection dataset, which was part of the shared task in the 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4) [89]. The task is split up into two Subtasks: Subtask A) which aimed at detecting whether a tweet is offensive or not and Subtask B) which aimed at detecting whether a tweet is hate-speech or not. The organizers labeled a tweet as offensive if it contained explicit or implicit insults directed towards other people or inappropriate language. While a tweet labeled as hate speech contains targeted insults towards a group based on their nationality, ethnicity, gender, political or sport affiliation.

The dataset for both tasks is the same containing 10K tweets that were annotated for offensiveness with labels (OFF or NOT_OFF) and hate speech with labels (HS or NOT_HS). The data was split by the competition organizers into 70% training set, 10% development set, and 20% test set. Table 3.1 shows the data distribution among the different labels and splits. By examining Table 3.1, it can be seen that the data is very imbalanced having only 5% of the examples labeled as hate speech and 20% of the examples labeled as offensive in the training dataset, which makes the tasks much harder and calls for methods that can learn efficiently from little data.

Table 3.1: Data distribution for both tasks

| **Class** | **Training** | **Developement** |
|---|---|---|
| NOT_OFF | 5468 | 821 |
| OFF | 1371 | 179 |
| NOT_HS | 6489 | 956 |
| HS | 350 | 44 |

**C. Question Answering:**

We evaluate QA on the Arabic Reading Comprehension Dataset (ARCD) [90], where the task is to find the span of the answer in a document for a given question. ARCD contains 1395 questions on Wikipedia articles along with 2966 machine translated questions and answers from the SQuAD dubbed (Arabic-SQuAD). We train on the whole Arabic-SQuAD and on 50% of ARCD and test on the remaining 50% of ARCD.

### 3.5.3 Experimental Setup

**Pretraining** In our experiments, the original implementation of BERT on TensorFlow was used. The data for pre-training was sharded, transformed into TFRecords, and then stored on Google Cloud Storage. Duplication factor was set to 10, a random seed of 34, and a masking probability of 15%. The model was pre-trained on a TPUv2-8 pod for 1,250,000 steps. To speed up the training time, the first 900K steps were trained on sequences of 128 tokens, and the remaining steps were trained on sequences of 512 tokens. The decision of stopping the pre-training was based on the performance of downstream tasks. We follow the same approach taken by the open-sourced German BERT [91]. Adam optimizer was used, with a learning rate of 1e-4, batch size of 512 and 128 for sequence length of 128 and 512 respectively. Training took 4 days, for 27 epochs over all the tokens.

**Fine-tuning** Fine-tuning was done independently using the same configuration for all tasks. We do not run extensive grid search for the best hyper-parameters due to computational and time constraints. We use the splits provided by the dataset's authors when available. and the standard 80% and 20% when not[4].

### 3.5.4 Results and Discussion

Table 3.2 illustrates the experimental results of applying AraBERT to multiple Arabic NLU downstream tasks, compared to state-of-the-art results and the multilingual BERT model (mBERT).

**Sentiment Analysis** For Arabic sentiment analysis, the results in Table 3.2 show that both versions of AraBERT outperform mBERT and other state-of-the-art approaches on most tested datasets. Even though AraBERT was trained on MSA, the model was able to preform well on dialects that were never seen before.

[4] The scripts used to create the datasets are available on our Github repo `https://github.com/aub-mind/arabert`

Table 3.2: Performance of AraBERT on Arabic downstream tasks compared to mBERT and previous state of the art systems

| Task | metric | prev. SOTA | mBERT | AraBERTv0.1/ v1 |
|---|---|---|---|---|
| **SA (HARD)** | Acc. | 95.7 [54] | 95.7 | **96.2** / 96.1 |
| **SA (ASTD)** | Acc. | 86.5 [54] | 80.1 | 92.2 / **92.6** |
| **SA (ArsenTD-Lev)** | Acc. | 52.4 [54] | 51.0 | 58.9 / **59.4** |
| **SA (AJGT)** | Acc. | 92.6 [92] | 83.6 | 93.1 / **93.8** |
| **SA (LABR)** | Acc. | **87.5** [92] | 83.0 | 85.9 / 86.7 |
| **Offensive Lang.** | macro-F1 | **90.5** [93] | 83.3 [93] | -/90.0 |
| **Hate Speech** | macro-F1 | **95.1** [94] | 73.0 | -/80.6 |
| **QA (ARCD)** | Exact Match | mBERT | 34.2 | 51.1 / **54.8** |
| | macro-F1 | | 61.3 | 82.1 / **82.2** |
| | Sent. Match | | 90.0 | 95.5 / **95.6** |

**Offensive Language and Hate Speech Detection** Although AraBERT scored slightly lower on offensive language detection than SOTA approaches, and came far behind previous SOTA on hate speech, the results show a huge improvement over the BERT multilingual. The results for the hate speech task fell behind due to the small number of the positive training examples, which only constitute 5% of the training data.

**Question Answering** The results in Table 3.2 show huge improvement in F1-score and in exact match scores. Upon further examination of the results, the majority of the erroneous answers differed from the true answer by one or two words with no significant impact on the semantics of the answer. Examples are shown in Tables 3.3 and 3.4. We also report a 2% absolute increase in the sentence match score over mBERT, which is the previous state-of-the-art. Sentence Match (SM) measures the percentage of predictions that are within the same sentence as the ground truth answer.

Table 3.3: Example of an erroneous results from the ARCD test set: the only difference is the preposition "في - *In*".

| **Question** | أين تأسست منظمة الأمم المتحدة؟<br>*where was the united nations established?* |
|---|---|
| **Ground Truth** | *In San Francisco* – في سان فرنسيسكو |
| **Predicted Answer** | *San Francisco* – سان فرنسيسكو |

**Discussion** The jump in performance over BERT multinigual has multiple explanations. First, data size is a clear factor for the boost in performance.

Table 3.4: Another example of an erroneous results from the ARCD test set: the predicted answer does not include “introductory” words.

| | |
|---|---|
| **Question** | ما هو النظام الخاص بدولة النمسا؟<br>*What is the type of government in Austria?* |
| **Ground Truth** | *Austria is a federal republic* – النمسا هي جمهورية فيدرالية |
| **Predicted Answer** | *A federal republic* – جمهورية فيدرالية |

AraBERT used around 24GB of data in comparison with the 4.3GB Wikipedia used for the multilingual BERT. Second, the vocab size used in the multilingual BERT is 2K tokens in comparison with 64k vocab size used for developing AraBERT. Third, with the large data size, the pre-training distribution has more diversity. As for the fourth point, the pre-segmentation applied before BERT tokenization improved performance on SA and QA. It is also noted that the pre-processing applied to the pre-training data took into consideration the complexities of the Arabic language. Hence, increased the effective vocabulary by excluding unnecessary redundant tokens that come with certain common prefixes, and help the model learn better by reducing the language complexity. We believe these factors largely contributed to the success of AraBERT over BERTm in all tested tasks and helped reach state-of-the-art results on 2 different tasks and 7 different datasets. Obtained results indicate that the advantage we got in the datasets considered are better understood in a monolingual model than of a general language model trained on Wikipedia crawls such as multilingual BERT.

## 3.6 Conclusion

AraBERT achieved state-of-the-art performance on sentiment analysis and question answering tasks, and surpasses BERTm’s results on all tested tasks. This adds truth to the assumption that pre-trained language models on a single language only surpass the performance of a multilingual model. It is also 300MB smaller than multilingual BERT. By publicly releasing our AraBERT models, we hope that it will be used to serve as the new baseline for the various Arabic NLP tasks, and hope that this work will act as a stepping stone to building and improving future Arabic language understanding models. Future work direction involves training models with a better understanding of the various dialects that the Arabic language has across different Arabic countries.

# Chapter 4

# Transformers for Arabic NLG: The Case of Open-Domain Chatbots

One of the goals of Artificial Intelligence (AI) is to develop a conversational agent (CA), also called chatbot, that can converse with humans in a way that is indistinguishable from a real human being.

According to Turing, a chatbot should be able to converse in any topic (Open-Domain), yet research advances have only been able to achieve designs that targets specific topics with limited domain knowledge such as booking systems or navigation, and only recently we started seeing a resurgence in open-domain chatbots research. Furthermore, almost all research work has been done on only a few languages, mainly English, Chinese, and Spanish. Other languages have lagged behind due to the lack of quality conversational data necessary to develop chatbots capable of open-domain conversations.

Recently research on English open-domain chatbots achieved human-level abilities using end-to-end training of large models on large dataset [95]. This approach solved a critical issue in earlier chatbots: the responses often didn't make sense and were not specific.

Current Arabic chatbots employ retrieval-based (pattern matching) approaches which, even though they excel in task-specific context, fail in an open-domain or social chat context [96, 97, 98, 99]. Furthermore, commercial Arabic chatbots are also lagging behind their English counterparts in terms of adoption and user experience [100]. This is caused by requiring users to converse with the chatbots in Modern Standard Arabic (MSA), and not in their spoken dialect.

One of the open challenges in Arabic chatbots is the need to provide Open-domain chat while correctly handling gender inflections and covering a rich set of Arabic vocabulary also knows as the Out-of-Vocabulary (OOV) problem. In this chapter, we aim to address these challenges, by proposing an "End-to-End Open-Domain Gender-Aware Chatbot". Open-domain chatting problem is framed as a text-to-text mapping problem, also known as Seq2Seq model that learns to map an utterance to a response. Gender inflections are also mapped as a text-to-text problem by appending the gender labels of the interlocutors to the input sequence.

Another challenge in Arabic chatbots is the scarcity of large conversational datasets that are annotated with speaker information. Hence, we propose enriching a noisy open-domain movie transcript dataset with the gender labels using state-of-the-art text-based classifiers. Two classifiers were trained on an existing subset of the movie transcript dataset annotated for the speaker gender and on a second subset that we create and annotate for the listener's gender.

In summary, the contributions of this work are as follows:

- Developing the first open-domain Arabic chatbot while simultaneously addressing open-domain, gender-aware and Arabic OOV problem in a generative DL model.
- Building the first large Arabic conversational dataset annotated for the speaker's and the listener's gender, which can be used for end-to-end training of open-domain chatbot

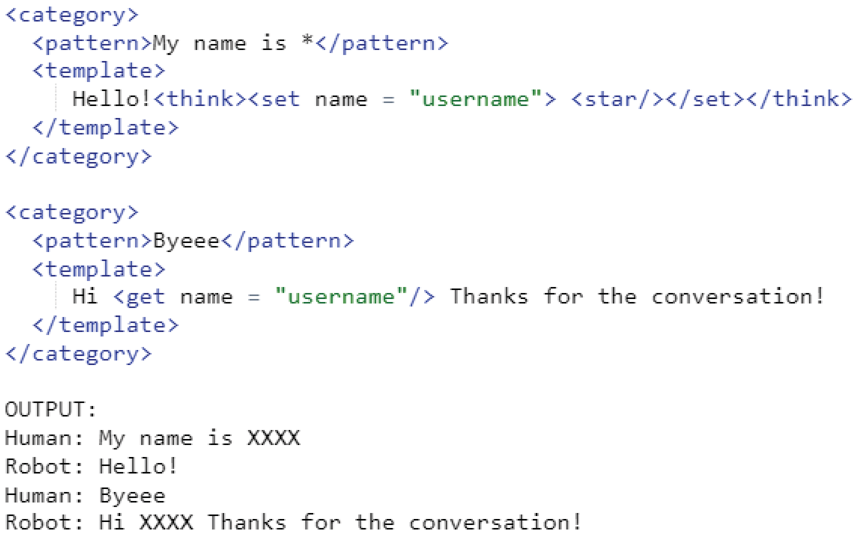
The rest of this chapter is organized as follows: Section 4.1 describes in details recent seminal work on English Open-domain chatbots and related works on Arabic chatbots. Section 4.2 describes the proposed open-domain gender-aware Arabic chatbot, and the creation of a gender-annotated Arabic conversational dataset. Section 4.3 presents and discusses the experiments and results. Finally, conclusion and future work are presented in Section 4.4 .

## 4.1 Literature Review

Although multiple types of chatbots exist, such as task-oriented systems, intelligent personal assistant..., we focus our literature review on open-domain chatbots, seeing that other types of chatbots target different end-goals and excel only within well-defined domains.

### 4.1.1 Early days of Conversational Agents

Chatbots have greatly progressed since their creation in the 1960s. The earliest instance of a CA, ELIZA [101], was developed by Joseph Weizenbaum, in 1966. It was a computer program that mimicked a psychiatrist by selecting a pre-written response through pattern matching. The program successfully fooled a great many people giving the illusion of understanding. In 1975, Colby [102] developed another chatbot Parry with the personality of a paranoid person. Parry was the



```
<category>
  <pattern>My name is *</pattern>
  <template>
     Hello!<think><set name = "username"> <star/></set></think>
  </template>
</category>

<category>
  <pattern>Byeee</pattern>
  <template>
     Hi <get name = "username"/> Thanks for the conversation!
  </template>
</category>

OUTPUT:
Human: My name is XXXX
Robot: Hello!
Human: Byeee
Robot: Hi XXXX Thanks for the conversation!
```

Figure 4.1: Simple AIML example script [1]

first chatbot to pass the Turing test. It also relied on hand-written rules and patterns, but was designed to respond aggressively and with hostility.

The Artificial Linguistic Internet Computer Entity, or ALICE for short, was a chatbot created in 2009 by Wallace [103] and allowed users to customize their experience using an Artificial Intelligence Markup Language (AIML), a dialect of Extensive Markup Language (XML). AIML, Figure 4.1, uses tags to define patterns, categories, responses... which are used to simplify the creation of chatbots. Although ALICE won the Loebner Prize[1] multiple times, the system failed to pass the Turing Test partially due to the use of AIML as the dialogue engine, which fails when it encounters long open-domain or chitchat conversations.

### 4.1.2 English Open-Domain Chatbots

This section reviews in detail recent seminal work on English open-domain chatbot, by presenting the motivation, challenges, implementation details, and performance of each of the included work.

#### A. A Neural Conversation Model

In 2015, Vinyals & Le [104] presented a rather simple approach for conversation modeling, motivated by the idea that previous approaches were domain-restricted (booking, weather checking...) and required tedious hand-crafted rules. The presented approach uses a sequence to sequence (Seq2Seq) [105] LSTM network [4] that converses by predicting the response given the previous utterance/s. The

[1] https://en.wikipedia.org/wiki/Loebner_Prize

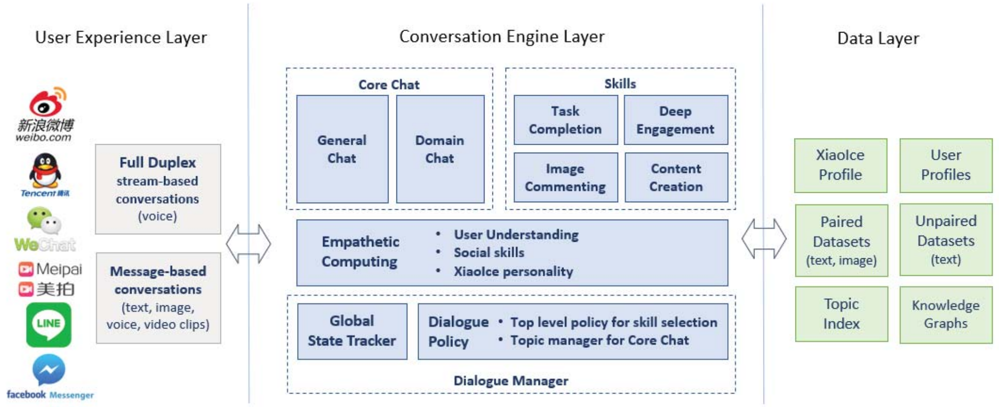


Figure 4.2: XiaoIce's underlying architecture [2]

benefits of using such architecture are that the end-to-end training requires very few hand-crafted rules since the model is trained on a large conversational dataset. The datasets were a domain-specific IT helpdesk Troubleshooting English dataset and an open-domain noisy dataset that consists of English movie subtitles [106]. Performance was measured in validation "perplexity"[2], which at convergence was 17. Li et al. [107] improved on this work by adding a third module to the encoder/decoder architecture, which they call the speaker model that encodes the personality of each speaker as a vector.

### B. XiaoIce: an Empathetic Social Chatbot

XiaoIce [2] takes the concept of open-domain chatbots to the next level, by adding the ability to understand the user's emotions which allows it to offer a friendly conversation that can cheer, encourage, sympathize with the user. The underlying architecture of the 6th XiaoIce generation is rather complicated compared to previous chatbots, as shown in Figure 4.2. It consists of 3 layers: user experience, conversation engine, and a data layer.

The Core Chat functionality, from the conversation engine module, is of special importance for our work since it provides the text communication capabilities by generating the appropriate response given at text input. It consists of two sub-modules: General Chat and a set of Domain Chats. General Chat handles open-domain conversations, while the domain-specific conversations (music, art, books, movies...) are handled by Domain Chats sub-modules. We again focus on General Chat since it is a data-driven response generation system, which employs a two-stage response system.

- **First Stage:** Three candidate responses are generated by three different

[2] Perplexity is the inverse probability if the predicted sentence normalized by the number of words. High perplexity is bad since the perplexity is the exponentiation of the entropy

system *(i)* a retrieval-based system that uses previously collected query-response pairs from human conversations available on the internet or previous interactions with the chatbots (with some additional filtering). *(ii)* a neural response generator that is based on a Seq2Seq architecture with additional empathetic vectors added for conditioning the response. It was trained using the human-human conversational database. *(iii)* another retrieval-based system that uses non-conversational data which can improve response coverage.

- **Second Stage:** The three generated candidates are ranked using a boosted tree ranker, based on features such as local cohesion, global coherence, empathy matching, and retrieval matching.

XiaoIce dialogue formulation is a hierarchical decision-making process that maximizes long-term user engagement which can be measured in Conversation-turns Per Session (CPS), the larger the CPS the more engaging is the chatbot. During deployment, the achieved CPS was 23 despite the addition of more than 230 new skills.

### C. Meena Chatbot

Meena is a recent work [95] that revisits the concept of end-to-end training of open-domain chatbot and pushes its limits by training on 40B words (or 867M context/response pairs) filtered from social media conversations. The model uses is a Seq2Seq architecture based on the Evolved Transformer [108], with 1 encoder block and 13 decoder blocks (2.6B parameters). The model is trained on multi-turn conversations as a single input sequence with context windows of 7 previous utterances, with the output sequence forming the response. Training achieves a perplexity of 10.2 with only 8K BPE [109] vocabulary.

This work also introduces a new quality metric for chatbot called Sensibleness and Specificity Average (SSA). SSA is a human evaluation process where judges are asked to label each response for “making sense” and having the response specific to the context since a chatbot can always reply with “I don’t know” [107] and be sensible. The results also shows that perplexity is highly correlated with SSA, hence in can be used as an automatic proxy for human judgement.

**Conclusion** End-to-end training of simple Seq2Seq language models can be used to model conversations, with large transformer models achieving high human-likeness. Other recent works follow the trend of end-to-end training of chatbots with similar approaches, bigger models and incremental improvements such as DialoGPT [110], Blender [111], PLATO-2 [112].

### 4.1.3 Arabic Conversational Systems

This section reviews recent work on Arabic conversational systems. It should also be noted that at the time of writing, there is no existing published work on Arabic open domain chatbot, hence the reviewed systems are domain-specific chatbots.

#### A. ArabChat

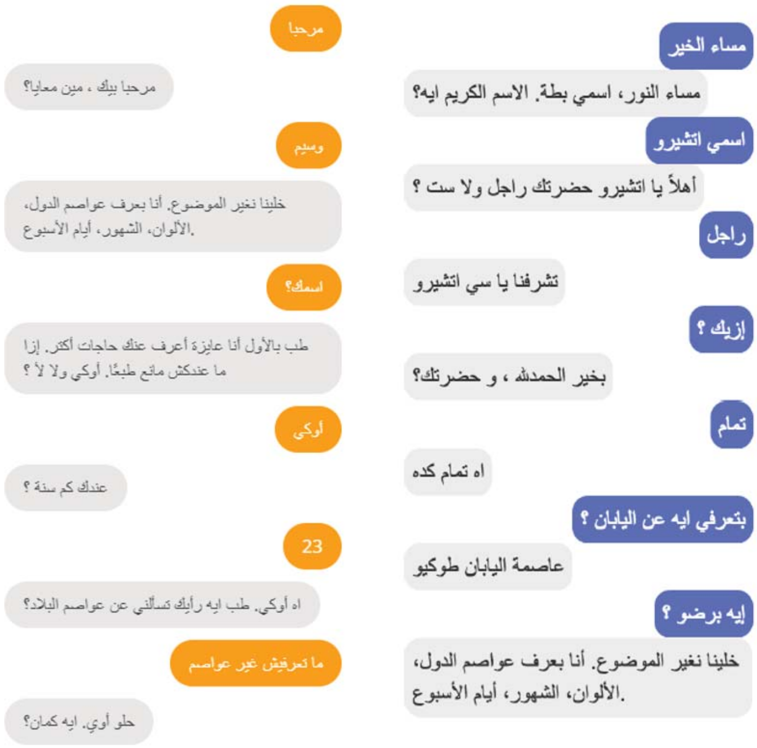
ArabChat [96, 97, 98] is a closed-domain Arabic MSA chatbot, deployed in the University of Jordan as an information point. ArabChat is built on top of a proprietary scripting engine and language which is responsible for detecting the topics of the conversations. Based on the detected topic the engine chooses a context from the knowledge base that contains rules. The rules are patterns which are used to generate the text response. In case the engine couldn't find a match or encountered a non-question, it would then output the highest rated rule regardless. ArabChat achieves a 73.56% match rate which was increased to 82% in the enhanced version. During deployment, unserious users were causing the system to fail by trying to open discussions instead of asking questions.

#### B. Botta

Botta [99], is the only Arabic chatbot that implement chitchat instead of a task oriented system. The chatbot is a modified version of the English chatbot Rosie [3]. Botta and Rosie shares the same AIML files, which were partially translated to Arabic and modified to support the Egyptian dialect. Botta also tries to identify the gender of the user using a name-to-gender mapping database, the gender of the user is then used to accurately provide gender inflection. Capital guessing functionality, is added using a capital database. It also tries to provide humorous responses using Arabic proverbs when it fails to understand an input. The chatbot was publicly available on the Pandorabots platforms, we show in Figure 4.3 a sample conversation with Botta. Although the chatbot correctly addresses the user with the proper gender inflections and Egyptian dialect, it always tries to steer to conversation towards the capital guessing functionality, since it is the only one implemented.

**Conclusion** Current research on Arabic conversational systems only focused on task-oriented systems that still rely on pattern-matching which hinders the ability of a system or designer due to the complexity of the Arabic language. There are no work in the literature that benefits from the advances in NLP and

[3] Pandorabots, Rosie: Base content for AIML 2.0 chatbot, 2018 `https://github.com/pandorabots/rosie`



مرحبا

مرحبا بيك ، مين معايا؟

وسيم

خلينا نغير الموضوع. أنا بعرف عواصم الدول،
الألوان، الشهور، أيام الأسبوع.

اسمك؟

طب بالأول أنا عايزة أعرف عنك حاجات أكتر. إزا
ما عندكش مانع طبعًا. أوكي ولا لأ ؟

أوكي

عندك كم سنة ؟

23

اه أوكي. طب ايه رأيك تسألني عن عواصم البلاد؟

ما تعرفيش غير عواصم

حلو أوي. ايه كمان؟

مساء الخير

مساء النور، اسمي بطة. الاسم الكريم ايه؟

اسمي اتشيرو

أهلاً يا اتشيرو حضرتك راجل ولا ست ؟

راجل

تشرفنا يا سي اتشيرو

إزيك ؟

بخير الحمدلله ، و حضرتك؟

تمام

اه تمام كده

بتعرفي ايه عن اليابان ؟

عاصمة اليابان طوكيو

إيه برضو ؟

خلينا نغير الموضوع. أنا بعرف عواصم الدول،
الألوان، الشهور، أيام الأسبوع.

Figure 4.3: Botta sample conversations. Left(Yellow): Our own test with the public version. Right(Blue): Example from the published paper

DL, which might be due to the lack of published datasets for Arabic chatbot (although commercial applications exist with more complex systems).

## 4.2 Proposed Open-Domain Gender-Aware Chatbot

We are presented with the problem of building an open-domain Arabic chatbot, and since conversing in Arabic requires the knowledge of the gender of both sides of the conversation, the chatbot need to have gender-aware responses that cover a wide variety of topics. The challenges in building such a system are as follows:

- Arabic conversational datasets are very scarce which makes end-to-end training a challenge.
- Conversational text needs to be labeled for the gender of both parties in a conversation
- Out-of-Vocabulary words during training and deployment, since Arabic is rich morphologically.

The open-domain chatbot problem can be formulated as a text-to-text problem, where the model takes as input a user's utterance and outputs a meaningful response on diverse topics. Our solution employs a Seq2Seq architecture trained to map input queries to output responses. We also condition the responses on the gender labels for both interlocutors, requiring the text-to-text mapping to append labels with the input tokens.

In sub-section 4.2.1, we describe the underlying model architecture, followed by a detailed description of the training data creation process in sub-section 4.2.2.

### 4.2.1 Model Architecture

We chose the SOTA Transformer-base Seq2Seq architecture [9] with 6 encoders and 6 decoders each with 8 self-attention heads, with an embedding size of 512 and 30K vocabulary, for a total of 75M parameters. The loss function used is the cross entropy loss. High level model architgecture is show in Figure 4.4

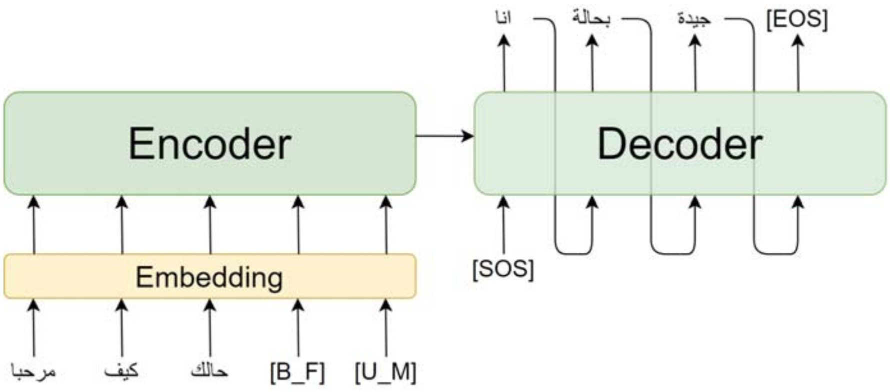


Figure 4.4: Proposed Model Architecture: The appended gender labels corresponds to a female bot [B_F] and a male user[U_M]

**Word Tokenization** To reduce the impact of Out-of-Vocabulary words in Arabic word tokenization, we first morphologically splitting prefixes and suffixes using the Farasa Segmenter [82] after which we apply BPE segmentation [109] to further reduce the impact for rare segmented words. This approach greatly increases vocabulary coverage compared to using BPE segmentation only. Input Sequence length is then limited to 25 tokens after segmentation.

### 4.2.2 Training Data

For reference we chose the OpenSubtitles 2018 dataset [113], which has been commonly used for early English chatbots. The dataset consists of a large collection of movie subtitles in many languages including Arabic, covering 83.6M time-stamped sentences, written in MSA. The dataset holds two issues:

**Turn Identifiers** Although subtitles are transcripts of conversations between actors, the dataset provides no turn identifiers. We address this issue by assuming that consecutive sentences as two utterance pairs, which effectively doubles the size of the training examples. This approach was used successfully in [104].

**Gender Labels** Since speaker annotation is missing, no gender labels were provided. To solve this issue, we resort into labeling all the subtitle sentences with gender labels for the speaker and the listener (both sides of the conversation) using text-based gender classifiers.

### A. Gender Labeling

Since sentences needs to labeled with two separate labels, one for each the speaker and another one for the listener, we propose using two text-based gender classifiers, as manually labeling 86M sentences is infeasible. The classifiers are based on the AraBERT [53] model. We use AraBERTv1[4] which was pretrained on a large morphologically segmented Arabic text dataset and achieved state-of-the-art performance on multiple Arabic text classification tasks.

For the speaker gender classification training data, we use a speaker gender dataset [114] that contains 12K sentences extracted from the OpenSubtitles 2018 dataset, which only include first-person-singular pronouns. The selected sentences were then annotated for the gender of the first person (the speaker) as follows: F for Female, M for Male, and B for ambiguous.

As for the listener's gender classification, we created our own dataset with a similar strategy to [114]. We selected 1300 sentences from the Arabic OpenSubtitles 2018 dataset, and we manually labeled each utterance for the listener's gender (F/M/B). The resulting dataset consisted of 706 ambiguous, 386 males, and 215 for females, from which we created a balanced test set, between males and females, with 150/50/50 for B/M/F gender labels. The developed AraBERT classifiers were then used to label all 83M sentences in the OpenSubtitles corpus, for the speaker and the listener's gender.

### B. Dataset Cleanup

The assumption of considering consecutive sentences as an utterance pair, will result in noisy examples because consecutive sentences may be coming from the same character or from two different movie scenes. To try and mitigate this issue, we employed automatic cleaning for the dataset by discarding sentence pairs that:

[4] `https://huggingface.co/aubmindlab/bert-base-arabert`

- Have matching gender labels between pairs. Ex: if the speaker was a male in the first utterance, then listener in the second utterance can't be a female

- Contain words longer than 10 characters, since some of the sentences had words that had been joined into a single word.

- Contain a very common response like: انا - لا اعرف - نعم - ماذا ؟ - اللّعنة - كلا - حسنا - لا - اسف. This keeps the model from learning to always generate common responses. Note: We keep 10% of these responses in order to note lose all the information

All the sentences in the original corpus were pre-processed to reduce the vocabulary variation, by removing diacritics including the "shadda", removing elongations, removing all non Arabic characters except the question and exclamation marks. After pre-processing and de-duplication the resulting dataset contained 32M pairs, annotated for the speaker and listener's gender.

We also create a second dataset by further filtering the noisy conversational datasets, to create a Question/Answer dataset, by only considering pairs with the first sentence ending with a question mark, and the second sentence having no question marks. The resulting dataset has 4.2M QA pairs.

### 4.2.3 Training via Transfer Learning

Another issue with our proposed approach is that the "Ambiguous" gender label is part of the training data, and should not appear in the deployment since the user will be required to choose their gender before engaging with the chatbot. We address this issue by using a Transfer Learning approach, where the model is first trained on a dataset without adding the gender labels, we then continue training the model on a filtered version of the dataset that only contains Male or Female labels. This Transfer Learning training approach has been proven to work in an English-to-Arabic machine translation setting [115]. We then extract 10K sentences from the filtered dataset to be used as a validation set. These sentences were also removed from the unfiltered dataset to prevent contamination. The same validation set is used for pre-training and finetuning, during pretraining the gender labels are removed.

We apply the same gender label filtering on both datasets, the conversational dataset, and the Question/Answer pairs dataset. Which results in the following datasets and splits:

- Conversational dataset: 32M pairs with no gender labels, used for pretraining, 700K pairs with gender labels used for finetuning and 10K pairs with gender labels as a validation set.
- Question/Answer dataset: 4.2M pairs with no gender labels, used for pretraining, 100K pairs with gender labels used for finetuning and 10K pairs with gender labels as a validation set.

## 4.3 Experiments & Results

In this section, we show and discuss the results of conducted experiments on gender classification and on the final Open-domain Arabic Chatbot.

### 4.3.1 Experimental Setup

As a baseline, we train an LSTM Seq2Seq model with 2 stacked uni-directional LSTM layers with 1024 hidden units, Luong general attention [116], 512 embedding size and 30K vocabulary, for a total of 96M parameters.

Transformer models were optimized using Adam [117], while LSTM models used Stochastic Gradient Descent (SGD).

**Pre-training** During pre-training, the model is trained until convergence (validation perplexity stops decreasing) with Transformer models taking 100K steps, and LSTM models taking 150K steps with a batch size of 1024 for all models.

**Fine-tuning** After pre-training the models were fine-tuned on the gender labeled datasets for one epoch on both datasets since the performance starts to go down after the first epoch due to over-fitting.

Training was done on Google's Colaboratory GPU environment, with each model pre-training taking 24 hours of training on a P100 16GB NVidia GPU, using the OpenNMT Pytorch Library [118].

### 4.3.2 Gender Labeling

Table 4.1 shows that on the speaker classification task, the model achieves a high F1-score compared to previous SOTA. For the listener's gender, the scores are lower than the speaker's score, which is expected since the number of samples in the listener is a magnitude lower. The results also show, that the performance on females is better than males, which can be attributed to the Arabic female inflections that are usually added to words (ex. the addition of ة+ to the end of the words). The confusion matrices in Figure 4.5, indeed show that both models confuse ambiguous with males more than females. It also shows that there is very little confusion between males and females in both models, which is a key requirement since confusion between males and females will later impact the chatbot's performance more than confusion with the ambiguous class.

| | Prev-SOTA | | AraBERT | |
|---|---|---|---|---|
| Label | Speaker | Listener | Speaker | Listener |
| B | 96 | - | **98** | **86** |
| F | 80 | - | **94** | **88** |
| M | 71 | - | **87** | **71** |
| Average | 83 | - | **93** | **82** |

Table 4.1: F1-score of the speaker and listener classifiers. Note: Average is the Macro-Average

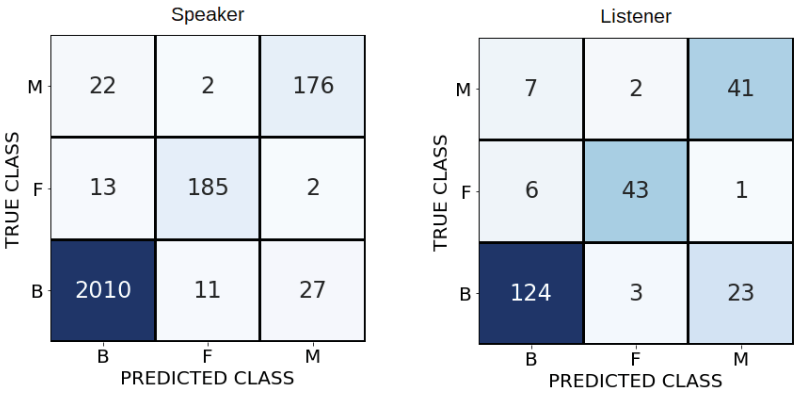


Figure 4.5: Confusion matrix for the speaker and listener classifiers

### 4.3.3 Open-Domain Arabic Chatbot

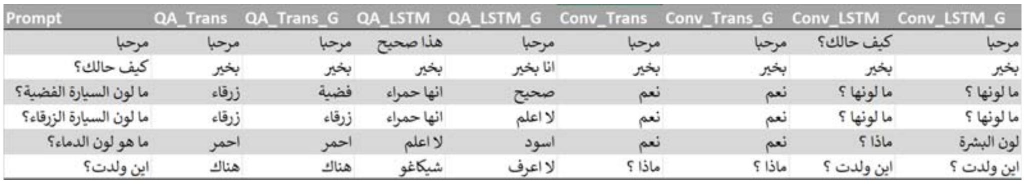
Table 4.2 illustrates the results of the pre-training and fine-tuning experiments on both datasets. The transformer-based model performed better than the LSTM baseline in all experiments. Also, the results show that the addition of the gender labels benefits the models by lowering the perplexity and increasing the accuracy on both datasets.

Figure 4.6 shows the response of all the trained models to prompts that cover greetings, reasoning, and world knowledge. Overall the models trained on the QA dataset have consistent performance compared to the models trained on the noisy

|  | **Conv (Acc./PPL)** |  | **QA(Acc./PPL)** |  |
|---|---|---|---|---|
| **Model** | **Pre-Training** | **Fine-Tuning** | **Pre-Training** | **Fine-Tuning** |
| **LSTM** | 38.2/37.6 | 38.8/36.1 | 37.5/38.5 | 37.9/37.3 |
| **Transformer** | 38.6/35.3 | **39.1/34.2** | 38.3/37 | **38.6/36.3** |

Table 4.2: Accuracy and Perplexity during the pre-training and fine-tuning stage for all datasets

conversational dataset, which is expected since the QA dataset is better filtered and less noisy. The results show that the transformer model trained on the Conv dataset performed the worst even though it had the lowest perplexity. In fact, during our testing, almost all responses were dull response with low confidence scores.



| Prompt | QA_Trans | QA_Trans_G | QA_LSTM | QA_LSTM_G | Conv_Trans | Conv_Trans_G | Conv_LSTM | Conv_LSTM_G |
|---|---|---|---|---|---|---|---|---|
| مرحبا | مرحبا | مرحبا | هذا صحيح | مرحبا | مرحبا | مرحبا | كيف حالك؟ | مرحبا |
| كيف حالك؟ | بخير | بخير | بخير | انا بخير | بخير | بخير | بخير | بخير |
| ما لون السيارة الفضية؟ | زرقاء | فضية | انها حمراء | صحيح | نعم | نعم | ما لونها ؟ | ما لونها ؟ |
| ما لون السيارة الزرقاء؟ | زرقاء | زرقاء | انها حمراء | لا اعلم | نعم | نعم | ما لونها ؟ | ما لونها ؟ |
| ما هو لون الدماء؟ | احمر | احمر | لا اعلم | اسود | نعم | نعم | ماذا ؟ | لون البشرة |
| اين ولدت؟ | هناك | هناك | شيكاغو | لا اعرف | ماذا ؟ | ماذا ؟ | اين ولدت ؟ | اين ولدت ؟ |

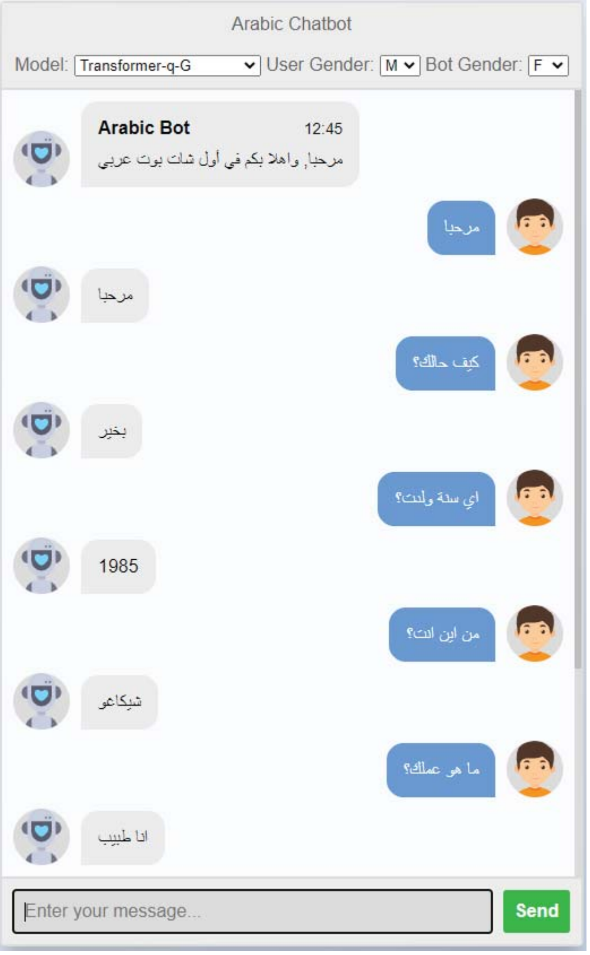
Figure 4.6: Response of all the trained chatbot models to different prompts. '_G' indicates the models fine-tuned with gender labels

Figure 4.7 shows a short sample conversation using the chatbot's user interface (UI). The UI allows changing the gender of the chatbot and the user during the conversation, and allows experimenting with multiple loaded models at the same time[5].

### 4.3.4 Discussion

Our modest results show that end-to-end training for Arabic chatbot systems is a success, despite the limited resources. The model was able to successfully deal question about simple reasoning and world knowledge on a wide variety of domains. The biggest limiting factor of such approach is the dataset source. Although movie subtitles contain Arabic conversational text, the texts is a human-translated version of English subtitles, hence it doesn't provide a good representation of Arabic conversation structure and vocabulary. We also experimented with adding prior utterances as context during training which resulted in worse responses, since the context may have added more noise and confusion to the model.

[5] Code and trained models will be available on `github.com/WissamAntoun/Arabic-Chatbot`



Model: Transformer-q-G User Gender: M Bot Gender: F

Arabic Bot 12:45
مرحبا, واهلا بكم في أول شات بوت عربي

مرحبا

مرحبا

كيف حالك؟

بخير

اي سنة ولدت؟

1985

من اين انت؟

شيكاغو

ما هو عملك؟

انا طبيب

Figure 4.7: Sample conversation using the chatbot's user interface (UI)

## 4.4 Conclusion

In this chapter, we show that end-to-end training for open-domain Arabic chatbot with gender-aware response is possible using a simple model architecture. We also show that training on the human translated noisy dataset can be improved by training on question answer pairs extracted from the noisy dataset, which results in a chatbot that can handle simple and basic conversations. Nonetheless, the model still requires modifications to be able to output deeper and more realistic conversations. We however believe that further research needs to be done to create Arabic conversational dataset that covers a wide variety of domains and dialects.

# Chapter 5

# Conclusion

This work addresses the gaps in modern Arabic NLP research, by proposing state-of-the-art solutions and architectures that solve Arabic specific problems (i.e. morphological richness, gender inflections, dialects). Our approach is grounded on two NLP domain: NLU and NLG. We set a new state-of-the-art in Arabic text classification and question answering, using transformer language models trained on large-scale Arabic text datasets. We also propose combining morphological segmentation and byte-pair encoding to address the out-of-vocabulary problem in Arabic. We show that creating open-domain Arabic chatbots based on the transformer Seq2Seq architecture, using end-to-end training with a limited amount of resources, can overcome the limitations of current approaches, and can simultaneously address open-domain, gender-aware and Arabic OOV problem in a generative DL model. We also make all of our pre-trained models and datasets publicly available hoping to encourage future research and applications for Arabic NLP to bridge the gap with English NLP.

As future work, we believe that creating benchmark datasets and tasks, similar to GLUE [119] for Arabic, is crucial since it enables evaluating and comparing the performance of different models of multiple tasks.

# Appendix A

# Abbreviations

| | |
|---|---|
| AI | Artificial Intelligence |
| AIML | Artificial Intelligence Markup Language |
| ANLP | Arabic Natural Language Processing |
| BERT | Bidirectional Encoder Representations from Transformers |
| BERTm | BERT Multilingual |
| BPE | Byte-Pair Encoding |
| CA | Conversational Agents |
| CPS | Conversation-turns Per Session |
| DA | Dialectal Arabic |
| DL | Deep Learning |
| LSTM | Long-Short Term Memory |
| MLM | Masked Language Modeling |
| MSA | Modern Standard Arabic |
| NER | Named-Entity Recognition |
| NLG | Natural Language Generation |
| NLP | Natural Language Processing |
| NLU | Natural Language Understanding |
| NSP | Next Sentence Prediction |
| OOV | Out-of-Vocabulary |
| PPL | Perplexity |
| QA | Question Answering |
| RNN | Recurrent Neural Network |
| SA | Sentiment Analysis |
| Seq2Seq | Sequence to Sequence |
| SGD | Stochastic Gradient Descent |
| SOTA | State-Of-The-Art |
| SSA | Sensibleness and Specificity Average |
| UI | User Interface |
| XML | Extensive Markup Language |

# Bibliography

[1] "Aiml tutorial." https://medium.com/@pemagrg/aiml-tutorial-a8802830f2bf. Accessed: 2020-08-11.

[2] L. Zhou, J. Gao, D. Li, and H.-Y. Shum, "The design and implementation of xiaoice, an empathetic social chatbot," *Computational Linguistics*, vol. 46, no. 1, pp. 53–93, 2020.

[3] J. L. Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[4] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[5] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, pp. 3111–3119, 2013.

[6] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[7] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of NAACL-HLT*, pp. 2227–2237, 2018.

[8] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[10] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, and V. Stoyanov, "Unsupervised cross-lingual representation learning at scale," 2019.

[11] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," 2019.

[12] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, and Q. V. Le, "Towards a human-like open-domain chatbot," 2020.

[13] W. Antoun, F. Baly, R. Achour, A. Hussein, and H. Hajj, "State of the art models for fake news detection tasks," in *2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT)*, pp. 519–524, IEEE, 2020.

[14] S. Tavernise, "As fake news spreads lies, more readers shrug at the truth," Dec 2016.

[15] J. Gottfried and E. Shearer, "News use across social media platforms 2016," Dec 2017.

[16] V. Pérez-Rosas, B. Kleinberg, A. Lefevre, and R. Mihalcea, "Automatic detection of fake news," in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 3391–3401, 2018.

[17] C. Silverman, "This analysis shows how viral fake election news stories outperformed real news on facebook," Nov 2016.

[18] L. H. OWEN, "Facebook is just gonna come out and start calling fake news fake (well, "false")."

[19] D. Pomerleau and D. Rao, "Fake news challenge stage 1 (fnc-i): Stance detection." `http://www.fakenewschallenge.org/`.

[20] B. Sean, S. Doug, and P. Yuxi, "Talos targets disinformation with fake news challenge victory," 2017.

[21] A. Hanselowski, A. PVS, B. Schiller, F. Caspelherr, D. Chaudhuri, C. M. Meyer, and I. Gurevych, "A retrospective analysis of the fake news challenge stance detection task," *arXiv preprint arXiv:1806.05180*, 2018.

[22] "Qatar international fake news detection and annotation contest." `https://sites.google.com/view/fakenews-contest`, 2019.

[23] I. Solaiman, "Gpt-2: 1.5b release," Nov 2019.

[24] S. Vosoughi, D. Roy, and S. Aral, "The spread of true and false news online," *Science*, vol. 359, no. 6380, pp. 1146–1151, 2018.

[25] R. Baly, G. Karadzhov, D. Alexandrov, J. Glass, and P. Nakov, "Predicting factuality of reporting and bias of news media sources," *arXiv preprint arXiv:1810.01765*, 2018.

[26] S. Afroz, M. Brennan, and R. Greenstadt, "Detecting hoaxes, frauds, and deception in writing style online," in *2012 IEEE Symposium on Security and Privacy*, pp. 461–475, IEEE, 2012.

[27] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the second workshop on computational approaches to deception detection*, pp. 7–17, 2016.

[28] H. Rashkin, E. Choi, J. Y. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language in fake news and political fact-checking," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2931–2937, 2017.

[29] M. Potthast, J. Kiesel, K. Reinartz, J. Bevendorff, and B. Stein, "A stylometric inquiry into hyperpartisan and fake news," *arXiv preprint arXiv:1702.05638*, 2017.

[30] G. Bhatt, A. Sharma, S. Sharma, A. Nagpal, B. Raman, and A. Mittal, "Combining neural, statistical and external features for fake news stance identification," in *Companion Proceedings of the The Web Conference 2018*, pp. 1353–1357, International World Wide Web Conferences Steering Committee, 2018.

[31] L. Borges, B. Martins, and P. Calado, "Combining similarity features and deep representation learning for stance detection in the context of checking fake news," *Journal of Data and Information Quality (JDIQ)*, vol. 11, no. 3, p. 14, 2019.

[32] J. G. Fiscus and G. R. Doddington, "Topic detection and tracking evaluation overview," in *Topic detection and tracking*, pp. 17–31, Springer, 2002.

[33] P. Han and N. Zhou, "A framework for detecting key topics in social networks," in *Proceedings of the 2nd International Conference on Big Data Technologies*, pp. 235–239, ACM, 2019.

[34] Y. Cha and J. Cho, "Social-network analysis using topic models," in *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pp. 565–574, ACM, 2012.

[35] T. N. Rubin, A. Chambers, P. Smyth, and M. Steyvers, "Statistical topic models for multi-label document classification," *Machine learning*, vol. 88, no. 1-2, pp. 157–208, 2012.

[36] R. Ibrahim, A. Elbagoury, M. S. Kamel, and F. Karray, “Tools and approaches for topic detection from twitter streams: survey,” *Knowledge and Information Systems*, vol. 54, no. 3, pp. 511–539, 2018.

[37] J. M. Schultz and M. Y. Liberman, “Towards a “universal dictionary” for multi-language information retrieval applications,” in *Topic detection and tracking*, pp. 225–241, Springer, 2002.

[38] G. Kumaran and J. Allan, “Text classification and named entities for new event detection,” in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 297–304, ACM, 2004.

[39] S. I. Nikolenko, S. Koltcov, and O. Koltsova, “Topic modelling for qualitative studies,” *Journal of Information Science*, vol. 43, no. 1, pp. 88–102, 2017.

[40] G. Fuentes-Pineda and I. V. Meza-Ruiz, “Topic discovery in massive text corpora based on min-hashing,” *Expert Systems with Applications*, 2019.

[41] H.-J. Choi and C. H. Park, “Emerging topic detection in twitter stream based on high utility pattern mining,” *Expert Systems with Applications*, vol. 115, pp. 27–36, 2019.

[42] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[43] M. Rosen-Zvi, T. Griffiths, M. Steyvers, and P. Smyth, “The author-topic model for authors and documents,” in *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 487–494, AUAI Press, 2004.

[44] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled lda: A supervised topic model for credit attribution in multi-labeled corpora,” in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pp. 248–256, Association for Computational Linguistics, 2009.

[45] D. Quercia, H. Askham, and J. Crowcroft, “Tweetlda: supervised topic classification and link prediction in twitter,” in *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 247–250, ACM, 2012.

[46] K. Kowsari, M. Heidarysafa, D. E. Brown, K. J. Meimandi, and L. E. Barnes, “Rmdl: Random multimodel deep learning for classification,” in *Proceedings of the 2nd International Conference on Information System and Data Mining*, pp. 19–28, ACM, 2018.

[47] N. Ding, Z. Li, Z. Liu, H. Zheng, and Z. Lin, "Event detection with trigger-aware lattice neural network," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 347–356, 2019.

[48] G. Burel, H. Saif, M. Fernandez, and H. Alani, "On semantics and deep learning for event detection in crisis situations," *Open Research Online*, 2017.

[49] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *arXiv preprint arXiv:1906.08237*, 2019.

[50] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[51] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[52] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1532–1543, Association for Computational Linguistics, Oct. 2014.

[53] W. Antoun, F. Baly, and H. Hajj, "Arabert: Transformer-based model for arabic language understanding," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 9–15, 2020.

[54] O. ElJundi, W. Antoun, N. El Droubi, H. Hajj, W. El-Hajj, and K. Shaban, "hulmona: The universal language model in arabic," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 68–77, 2019.

[55] M. Djandji, F. Baly, W. Antoun, and H. Hajj, "Multi-task learning using arabert for offensive language detection," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pp. 97–101, 2020.

[56] T. Mikolov, E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin, "Advances in pre-training distributed word representations," *arXiv preprint arXiv:1712.09405*, 2017.

[57] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2019.

[58] A. B. Soliman, K. Eissa, and S. R. El-Beltagy, "Aravec: A set of arabic word embedding models for use in arabic nlp," *Procedia Computer Science*, vol. 117, pp. 256–265, 2017.

[59] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 135–146, 2017.

[60] A. Erdmann, N. Zalmout, and N. Habash, "Addressing noise in multidialectal word embeddings," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 558–565, 2018.

[61] I. Abu Farha and W. Magdy, "Mazajak: An online Arabic sentiment analyser," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, (Florence, Italy), pp. 192–198, Association for Computational Linguistics, Aug. 2019.

[62] M. Abdul-Mageed, H. Alhuzali, and M. Elaraby, "You tweet what you speak: A city-level dataset of arabic dialects," in *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[63] M. Polignano, P. Basile, M. de Gemmis, G. Semeraro, and V. Basile, "AlBERTo: Italian BERT Language Understanding Model for NLP Challenging Tasks Based on Tweets," in *Proceedings of the Sixth Italian Conference on Computational Linguistics (CLiC-it 2019)*, vol. 2481, CEUR, 2019.

[64] L. Martin, B. Muller, P. J. O. Suárez, Y. Dupont, L. Romary, Éric Villemonte de la Clergerie, D. Seddah, and B. Sagot, "Camembert: a tasty french language model," 2019.

[65] W. de Vries, A. van Cranenburgh, A. Bisazza, T. Caselli, G. van Noord, and M. Nissim, "Bertje: A dutch bert model," *arXiv preprint arXiv:1912.09582*, 2019.

[66] G. Badaro, R. Baly, H. Hajj, N. Habash, and W. El-Hajj, "A large scale arabic sentiment lexicon for arabic opinion mining," in *Proceedings of the EMNLP 2014 workshop on arabic natural language processing (ANLP)*, pp. 165–173, 2014.

[67] A. Al Sallab, H. Hajj, G. Badaro, R. Baly, W. El-Hajj, and K. Shaban, "Deep learning models for sentiment analysis in arabic," in *Proceedings of the second workshop on Arabic natural language processing*, pp. 9–17, 2015.

[68] A. Al-Sallab, R. Baly, H. Hajj, K. B. Shaban, W. El-Hajj, and G. Badaro, "Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 16, no. 4, pp. 1–20, 2017.

[69] R. Baly, H. Hajj, N. Habash, K. B. Shaban, and W. El-Hajj, "A sentiment treebank and morphologically enriched recursive deep models for effective sentiment analysis in arabic," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 16, no. 4, pp. 1–21, 2017.

[70] A. Dahou, M. A. Elaziz, J. Zhou, and S. Xiong, "Arabic sentiment classification using convolutional neural network and differential evolution algorithm," *Computational intelligence and neuroscience*, vol. 2019, 2019.

[71] A. Al-Hassan and H. Al-Dossari, "Detection of hate speech in social networks: a survey on multilingual corpus," in *6th International Conference on Computer Science and Information Technology*, 2019.

[72] N. Albadi, M. Kurdi, and S. Mishra, "Are they our brothers? analysis and detection of religious hate speech in the arabic twittersphere," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 69–76, IEEE, 2018.

[73] A. Alakrot, L. Murray, and N. S. Nikolov, "Towards accurate detection of offensive language in online communication in arabic," *Procedia computer science*, vol. 142, pp. 315–320, 2018.

[74] H. Mohaouchane, A. Mourhir, and N. S. Nikolov, "Detecting offensive language on arabic social media using deep learning," in *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pp. 466–471, IEEE, 2019.

[75] H. Mubarak and K. Darwish, "Arabic offensive language classification on twitter," in *International Conference on Social Informatics*, pp. 269–276, Springer, 2019.

[76] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, M. Kelcey, J. Devlin, K. Lee, K. N. Toutanova, L. Jones, M.-W. Chang, A. Dai, J. Uszkoreit, Q. Le, and

S. Petrov, "Natural questions: a benchmark for question answering research," *Transactions of the Association of Computational Linguistics*, 2019.

[77] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[78] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv preprint arXiv:1609.08144*, 2016.

[79] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler, "Aligning books and movies: Towards story-like visual explanations by watching movies and reading books," 2015.

[80] I. A. El-Khair, "1.5 billion words arabic corpus," *arXiv preprint arXiv:1611.04033*, 2016.

[81] I. Zeroual, D. Goldhahn, T. Eckart, and A. Lakhouaja, "OSIAN: Open source international Arabic news corpus - preparation and integration into the CLARIN-infrastructure," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, (Florence, Italy), pp. 175–182, Association for Computational Linguistics, Aug. 2019.

[82] A. Abdelali, K. Darwish, N. Durrani, and H. Mubarak, "Farasa: A fast and furious segmenter for arabic," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 11–16, 2016.

[83] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," 2018.

[84] A. Elnagar, Y. S. Khalifa, and A. Einea, "Hotel arabic-reviews dataset construction for sentiment analysis applications," in *Intelligent Natural Language Processing: Trends and Applications*, pp. 35–52, Springer, 2018.

[85] M. Nabil, M. Aly, and A. Atiya, "ASTD: Arabic sentiment tweets dataset," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 2515–2519, Association for Computational Linguistics, Sept. 2015.

[86] R. Baly, A. Khaddaj, H. Hajj, W. El-Hajj, and K. B. Shaban, "Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets," in *OSACT 3: The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools*, p. 37, 2018.

[87] M. Aly and A. Atiya, "LABR: A large scale Arabic book reviews dataset," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, (Sofia, Bulgaria), pp. 494–498, Association for Computational Linguistics, Aug. 2013.

[88] K. M. Alomari, H. M. ElSherif, and K. Shaalan, "Arabic tweets sentimental analysis using machine learning," in *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pp. 602–610, Springer, 2017.

[89] H. Mubarak, K. Darwish, W. Magdy, T. Elsayed, and H. Al-Khalifa, "Overview of OSACT4 Arabic offensive language detection shared task," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, (Marseille, France), pp. 48–52, European Language Resource Association, May 2020.

[90] H. Mozannar, E. Maamary, K. El Hajal, and H. Hajj, "Neural arabic question answering," in *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 108–118, 2019.

[91] DeepsetAI, "Open sourcing german bert."

[92] A. Dahou, S. Xiong, J. Zhou, and M. A. Elaziz, "Multi-channel embedding convolutional neural network model for arabic sentiment classification," *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, vol. 18, no. 4, pp. 1–23, 2019.

[93] S. Hassan, Y. Samih, H. Mubarak, A. Abdelali, A. Rashed, and S. A. Chowdhury, "ALT submission for OSACT shared task on offensive language detection," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, (Marseille, France), pp. 61–65, European Language Resource Association, May 2020.

[94] F. Husain, "OSACT4 shared task on offensive language detection: Intensive preprocessing-based approach," in *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, (Marseille, France), pp. 53–60, European Language Resource Association, May 2020.

[95] D. Adiwardana, M.-T. Luong, D. R. So, J. Hall, N. Fiedel, R. Thoppilan, Z. Yang, A. Kulshreshtha, G. Nemade, Y. Lu, *et al.*, "Towards a human-like open-domain chatbot," *arXiv preprint arXiv:2001.09977*, 2020.

[96] M. Hijjawi, Z. Bandar, K. Crockett, and D. Mclean, "Arabchat: an arabic conversational agent," in *2014 6th International Conference on Computer Science and Information Technology (CSIT)*, pp. 227–237, IEEE, 2014.

[97] M. Hijjawi, H. Qattous, and O. Alsheiksalem, "Mobile arabchat: An arabic mobile-based conversational agent," *Int. J. Adv. Comput. Sci. Appl. IJACSA*, vol. 6, no. 10, 2015.

[98] M. Hijjawi, Z. Bandar, and K. Crockett, "The enhanced arabchat: An arabic conversational agent," *International Journal of Advanced Computer Science and Applications*, vol. 7, 2016.

[99] D. A. Ali and N. Habash, "Botta: An arabic dialect chatbot," in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pp. 208–212, 2016.

[100] S. AlHumoud, A. Al Wazrah, and W. Aldamegh, "Arabic chatbots: A survey," *International Journal Of Advanced Computer Science And Applications*, vol. 9, no. 8, pp. 535–541, 2018.

[101] J. Weizenbaum, "Eliza—a computer program for the study of natural language communication between man and machine," *Communications of the ACM*, vol. 9, no. 1, pp. 36–45, 1966.

[102] K. M. Colby, *Artificial paranoia: a computer simulation of paranoid process.* Pergamon Press, 1975.

[103] R. S. Wallace, "The anatomy of alice," in *Parsing the Turing Test*, pp. 181–210, Springer, 2009.

[104] O. Vinyals and Q. Le, "A neural conversational model," *arXiv preprint arXiv:1506.05869*, 2015.

[105] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.

[106] J. Tiedemann, "News from opus-a collection of multilingual parallel corpora with tools and interfaces," in *Recent advances in natural language processing*, vol. 5, pp. 237–248, 2009.

[107] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan, "A persona-based neural conversation model," *arXiv preprint arXiv:1603.06155*, 2016.

[108] D. So, Q. Le, and C. Liang, "The evolved transformer," in *International Conference on Machine Learning*, pp. 5877–5886, 2019.

[109] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.

[110] Y. Zhang, S. Sun, M. Galley, Y.-C. Chen, C. Brockett, X. Gao, J. Gao, J. Liu, and B. Dolan, "Dialogpt: Large-scale generative pre-training for conversational response generation," in *ACL, system demonstration*, 2020.

[111] S. Roller, E. Dinan, N. Goyal, D. Ju, M. Williamson, Y. Liu, J. Xu, M. Ott, K. Shuster, E. M. Smith, *et al.*, "Recipes for building an open-domain chatbot," *arXiv preprint arXiv:2004.13637*, 2020.

[112] S. Bao, H. He, F. Wang, H. Wu, H. Wang, W. Wu, Z. Guo, Z. Liu, and X. Xu, "Plato-2: Towards building an open-domain chatbot via curriculum learning," 2020.

[113] P. Lison and J. Tiedemann, "Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pp. 923–929, 2016.

[114] N. Habash, H. Bouamor, and C. Chung, "Automatic gender identification and reinflection in arabic," in *Proceedings of the First Workshop on Gender Bias in Natural Language Processing*, pp. 155–165, 2019.

[115] M. Elaraby, A. Y. Tawfik, M. Khaled, H. Hassan, and A. Osama, "Gender aware spoken language translation applied to english-arabic," in *2018 2nd International Conference on Natural Language and Speech Processing (ICNLSP)*, pp. 1–6, IEEE, 2018.

[116] M.-T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015.

[117] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[118] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[119] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," *arXiv preprint arXiv:1804.07461*, 2018.