AraNet: A Deep Learning Toolkit for Arabic Social Media

Muhammad Abdul-Mageed, Chiyu Zhang, Azadeh Hashemi, El Moatez Billah Nagoudi

Natural Language Processing Lab, University of British Columbia
{muhammad.mageeed, azadeh.hashemi, moatez.nagoudi}@ubc.ca
chiyuzh@mail.ubc.ca

Abstract

We describe AraNet, a collection of deep learning Arabic social media processing tools. Namely, we exploit an extensive host of both publicly available and novel social media datasets to train bidirectional encoders from transformers (BERT) focused at social meaning extraction. AraNet models predict age, dialect, gender, emotion, irony, and sentiment. AraNet either delivers state-of-the-art performance on a number of these tasks and performs competitively on others. AraNet is exclusively based on a deep learning framework, giving it the advantage of being feature-engineering free. To the best of our knowledge, AraNet is the first to performs predictions across such a wide range of tasks for Arabic NLP. As such, AraNet has the potential to meet critical needs. We publicly release AraNet to accelerate research, and to facilitate model-based comparisons across the different tasks.

1. Introduction

The proliferation of social media has made it possible to study large online communities at scale. This offers opportunities to make important discoveries, facilitate decision making, guide policies, improve health and well-being, aid disaster response, attend to population needs in pandemics such as the current COVID-19, etc. The wide host of languages, languages varieties, and dialects used on social media and the nuanced differences between users of various backgrounds (e.g., different age groups, gender identities) make it especially difficult to derive sufficiently valuable insights based on single prediction tasks. For these reasons, it is highly desirable to develop natural language processing (NLP) tools that can help piece together more complete pictures of events impacting individuals of different identities across different geographic regions. In this work, we propose AraNet , a suit of tools that has the promise to play such a role of Arabic social media processing.

For Arabic, a collection of languages and varieties spoken by a wide population of ~ 400 million native speakers covering a vast geographical region (shown in Figure 1), no such suite of tools currently exists. Many works have focused on sentiment analysis, e.g., (Abdul-Mageed et al., 2014a; Nabil et al., 2015; ElSahar and El-Beltagy, 2015; Al Sallab et al., 2015; Al-Moslmi et al., 2018; Al-Smadi et al., 2019; Al-Ayyoub et al., 2019; Farha and Magdy, 2019) and dialect identification (Elfardy and Diab, 2013; Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014; Cotterell and Callison-Burch, 2014; Zhang and Abdul-Mageed, 2019b; Bouamor et al., 2019a). However, there is rarity of tools for other tasks such as gender and age detection. This motivates our toolkit, which we hope can meet the current critical need for studying Arabic communities online. This is especially valuable given the waves of protests, uprisings, and revolutions that have swept the region during the last decade.

Although we create new models for tasks such as sentiment analysis and gender detection as part of AraNet, our primary focus is to provide strong baselines across the various tasks. We believe this will facilitate comparisons across models. This is particularly useful due to absence of standardization across datasets for many of the tasks, and given
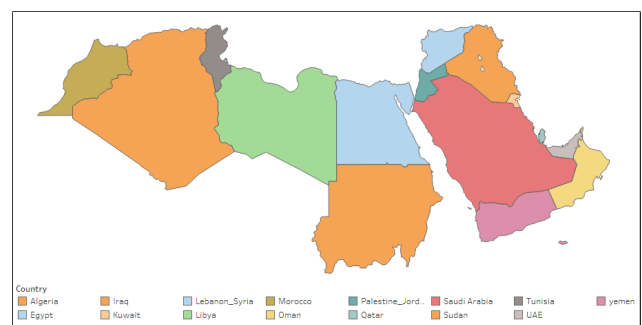

Figure 1: A map of Arab countries. Our different datasets cover varying regions of the Arab world as we describe in each section.

Figure 2: AraNet usage and output as a Python library.

the somewhat ephemeral nature of parts of some types of these data. In particular, many tasks are developed based on social media posts such as tweets that are distributed under restrictive conditions. For example, Twitter terms require release of data only in the form of tweet ids, making it challenging to acquire 100% of these tweets especially once the data are several months old. These reasons make *model-based comparisons* appealing, as a way to measure research progress in absence of easy benchmarking. Our general approach is to adopt sensible baselines across the various AraNet tasks, but we do not necessarily explicitly compare to all previous research. This is the case since most existing works either exploit *smaller data* (and so it will not be

a fair comparison), use *methods pre-dating BERT* (and so will likely be outperformed by our models). In addition, we note that although it would have been possible to acquire better results by feature engineering (especially on smaller datasets), our main goal is to keep our models free of laborious feature engineering. In some tasks, we even acquire better results that what is reported here by adopting more involved methods. But, again, we do our best here to keep all models relatively comparable (and as simple as possible) in terms of the methods employed to acquire them. Our hope is that, by adopting model-based comparisons, we can help accelerate progress on Arabic social media processing. For these reasons, we also package models from our recent works on dialect (Zhang and Abdul-Mageed, 2019b) and irony (Zhang and Abdul-Mageed, 2019a) as part of AraNet.

The rest of the paper is organized as follows: In Section 2. we describe our methods. In Section 3., we describe or refer to published literature for the data we exploit for each task. Also in Section 3., we provide results from our models. Section 4. is about AraNet design and use, and Section 5. is about ethical considerations. We overview related works in Section 6., and conclude in Section 7..

2. Methods

2.1. Supervised BERT

Transformer. Across all our tasks, we use Bidirectional Encoder Representations from Transformers (BERT). BERT is based on the Transformer architecture of (Vaswani et al., 2017), which we briefly introduce here. The Transformer depends solely on self-attention, thus allowing for parallelizing the network (unlike RNNs). It is an encoder-decoder architecture where the *encoder* takes a sequence of symbol representations $x^{(i)} \ldots x^{(n)}$, maps them into a sequence of continuous representations $z^{(i)} \ldots x^{(n)}$ that are then used by the *decoder* to generate an output sequence $y^{(i)} \ldots y^{(n)}$, one symbol at a time. This is performed using *self-attention*, where different positions of a single sequence are related to one another. The Transformer employs an attention mechanism based on a function that operates on *queries*, *keys*, and *values*. The attention function maps a query and a set of key-value pairs to an output, where the output is a weighted sum of the values. For each value, a weight is computed as a compatibility function of the query with the corresponding key. This particular version of attention is a scaled dot product of queries and keys (each of d_k) that is scaled by a factor of $\frac{1}{\sqrt{d_k}}$ on which a softmax is applied to acquire the weights on the values. The scaled dot product attention is computed as as a set of queries, keys, and values in three matrices Q, K, and V, respectively, follows:

$$Attention\,(Q,\,K,\,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (1)$$

Encoder of the Transformer in (Vaswani et al., 2017) has 6 attention layers, each of which has *h* attention heads (*multi-head attention*) to allow the model to jointly attend to information from different representation subspaces across different positions. Each of the 6 layers also has a simple,

fully-connected feed-forward network (FFN) that is applied to each position separately and identically that different parameters across the different layers. Decoder of the Transformer is similar to the encoder but has a third sub-layer that performs multi-head attention over the encoder stack. Since the Transformer discards with both recurrence and convolution, it resorts to the so-called *positional encoding* (based on sin and cosine functions) at the bottoms of the encoder and decoder stacks as a way to capture order of the sequence. We now introduce BERT.

BERT. BERT involves two self-supervised learning tasks, (1) *masked language models (Masked LM)* and (2) *next sentence prediction*. Since BERT uses bidirectional conditioning, a given percentage of random input tokens are masked and the model attempts to predict these masked tokens. This is the Masked LM task, where masked tokens are simply replaced by a string [MASK] . (Devlin et al., 2018) mask 15% of the tokens (the authors use Word-Pieces) and feed the final hidden vectors of these masked tokens to an output softmax over the vocabulary. The next sentence prediction task is just binary classification. For a given sentence S, two sentences A and B are generated where A (positive class) is an actual sentence from the corpus and B is a randomly chosen sentence (negative class). Once trained on an unlabeled dataset, BERT can then be fine-tuned with supervised data for a downstream task (e.g., text classification, question answering).

All our models are trained in a fully supervised fashion, with dialect id being the only task where we leverage semi-supervised learning. We briefly outline our semi-supervised methods next.

2.2. Self-Training

Only for the dialect id task, we investigate augmenting our human-labeled training data with automatically-predicted data from self-training. Self-training is a wrapper method for semi-supervised learning (Triguero et al., 2015; Pavlinek and Podgorelec, 2017) where a classifier is initially trained on a (usually small) set of labeled samples D^l, then is used to classify an unlabeled sample set D^u. Most confident predictions acquired by the original supervised model are added to the labeled set, and the model is iteratively re-trained. We perform self-training using different confidence thresholds and choose different percentages from predicted data to add to our dialect training set. We only report best settings here, and the reader is referred to our winning system on the MADAR shared task for more details on these different settings (Zhang and Abdul-Mageed, 2019b).

2.3. Implementation & Models Parameters

For all our tasks, we use the BERT-Base Multilingual Cased model released by the authors [1]. The model is trained on 104 languages (including Arabic) with 12 layer, 768 hidden units each, 12 attention heads, and has 110M parameters in entire model. The model has 119,547 shared WordPieces vocabulary, and was pre-trained on the entire

---

[1] https://github.com/google-research/bert/blob/master/multilingual.md.

17

Wikipedia for each language. For fine-tuning, we use a maximum sequence size of 50 tokens and a batch size of 32. We set the learning rate to $2e-5$ and train for 15 epochs [2] and choose the best model based on performance on a development set. We use the same hyper-parameters in all of our BERT models. We fine-tune BERT on each respective labeled dataset for each task. For BERT input, we apply WordPiece tokenization, setting the maximal sequence length to 50 words/WordPieces. For all tasks, we use a TensorFlow implementation. An exception is the sentiment analysis task, where we used a PyTorch implementation with the same hyper-parameters but with a learning rate $2e-6$. [3]

Pre-processing. Most of our training data in all tasks come from Twitter. Exceptions are in some of the datasets we use for sentiment analysis, which we point out in Section 3.5.. Our pre-processing thus incorporates methods to clean tweets, other datasets (e.g., from the news domain) being much less noisy. For pre-processing, we remove all usernames, URLs, and diacritics in the data.

3. Data and Models

3.1. Age and Gender

Arab-Tweet. For modeling age and gender, we use Arap-Tweet (Zaghouani and Charfi, 2018) [4], which we will refer to as *Arab-Tweet*. Arab-tweet comprises 11 Arabic regions from 17 different countries. [5] For each region, data from 100 Twitter users were crawled. Users needed to have posted at least 2,000 tweets and were selected based on an initial list of seed words characteristic of each region. The seed list included words such as برشة /barsha/ 'many' for Tunisian Arabic and وايد /wayed/ 'many' for Gulf Arabic. (Zaghouani and Charfi, 2018) employed human annotators to verify that users do belong to each respective region. Annotators also assigned gender labels from the set *male, female* and age group labels from the set *under-25, 25-to34, above-35* at the user-level, which in turn is the tag for tweet level. Tweets with less than 3 words and re-tweets were removed. Refer to (Zaghouani and Charfi, 2018) for details about how annotation was carried out. We provide a description of the data in Table 1. Table 1 also provides class breakdown across our splits.We note that (Zaghouani and Charfi, 2018) do not report classification models exploiting the data. Although age and gender are user-level tasks, note that we train *tweet-level* age and gender models. However, tweet-level models can easily be ported to user-level by simply taking the majority class based on softmax-thresholding as we show in (Zhang and Abdul-Mageed, 2019b). [6]

We shuffle the Arab-tweet dataset and split it into 80% training (TRAIN), 10% development (DEV), and 10% test (TEST). The distribution of classes in our splits is in Table 1. For pre-processing, we reduce 2 or more consecutive repetitions of the same character into only 2 and remove diacritics. With this dataset, we train a small unidirectional GRU (*small-GRU*) with a single 500-units hidden layer and dropout$= 0.5$ as a baseline. Small-GRU is trained with the TRAIN set, batch size = 8, and up to 30 words of each sequence. Each word in the input sequence is represented as a trainable 300-dimension vector. We use the top 100K words from TRAIN which are weighted by mutual information as our vocabulary in the embedding layer. We evaluate the model on the blind TEST set. Table 2 shows that small-GRU obtains 36.29% acc. on age classification, and 53.37% acc. on gender detection. Table 2 also shows performance of the fine-tuned BERT model. BERT significantly outperforms our baseline on the two tasks. It improves 15.13% acc. (for age) and 11.93% acc. (for gender) over the small-GRU.

UBC Twitter Gender Dataset. We also develop an in-house Twitter dataset for gender. We manually labeled 1,989 users from each of the 21 Arab countries. The data had 1,246 "male", 528 "female", and 215 unknown users. We remove the "unknown" category and balance the dataset to have 528 from each of the two "male" and "female" categories. We ended with 69,509 tweets for "male" and 67,511 tweets for "female". We split the users into 80% TRAIN (110,750 tweets for 845 users), 10% DEV (14,158 tweets for 106 users), and 10% TEST (12,112 tweets for 105 users). We then model this dataset with BERT and evaluate on DEV and TEST. Table 3 shows that fine-tuned model obtains 62.42% acc. on DEV and 60.54% acc. on TEST. These results are 2.89% and 4.76% less than performance on Arab-Tweet, perhaps reflecting more diversity in UBC-Gender data which also makes it more challenging. Another potential reason for this accuracy drop could be that, for this tweet-level task, some tweets from the same user occur across our TRAIN/DEV/TEST splits. This was unavoidable since Arab-Tweet is distributed without user ids, thus not making it possible for us to prevent user-level data leakage into the two tweet-level classification tasks of age and gender we report here. We alleviate this issue for gender by annotating and developing on UBC-Gender where we control for user-level data distribution across the splits as explained earlier.

We also combine the Arab-tweet gender dataset with our UBC-Gender dataset for gender on training, development, and test, respectively, to obtain new TRAIN, DEV, and TEST. We fine-tune BERT on the combined TRAIN and evaluate on combined DEV and TEST. As Table 3 shows, the model obtains 65.32% acc. on combined DEV, and 65.32% acc. on combined TEST. This is the model we package in AraNet.

3.2. Dialect

The dialect identification model in AraNet is based on our winning system in the MADAR shared task 2 (Bouamor et al., 2019b) as described in (Zhang and Abdul-Mageed, 2019b). The corpus is divided into training, development,

---

[2] For dialect id, we trained only for 10 epochs. This was based on monitoring loss on a development set.

[3] We find this learning rate to work better when we use PyTorch.

[4] The resource is an Arabic **p**rofiling dataset, and hence the sequence "Arap" with an "p".

[5] Counts are based on the distribution we received from the authors.

[6] Arab-Tweet is also distribute only with tweet-level labels (i.e., without user ids), thus making it not possible to model age and gender at the user level exploiting the data.

Data split	Under 25		25 until 34		35 and up		# of tweets
	Female	Male	Female	Male	Female	Male	
TRAIN	215,950	213,249	207,184	248,769	174,511	226,132	1,285,795
DEV	27,076	26,551	25,750	31,111	21,942	28,294	160,724
TEST	26,878	26,422	25,905	31,211	21,991	28,318	160,725
ALL	269,904	266,222	258,839	311,091	218,444	282,744	1,607,244

Table 1: Distribution of age and gender classes in our Arab-Tweet data splits

	Age		Gender	
	DEV	TEST	DEV	TEST
small-GRU	36.13	36.29	53.39	53.37
BERT	**50.95**	**51.42**	**65.31**	**65.30**

Table 2: Model performance in accuracy of Arab-Tweet age and gender classification tasks.

	DEV	TEST
UBC_TW_Gender	62.42	60.54
Gender_comb	**65.32**	**65.32**

Table 3: Model performance in accuracy. UBC_TW_Gender refers to the model trained on UBC Twitter Gender dataset. Gender_Comb denotes the model trained on the Arab-Tweet and UBC-Gender combined TRAIN data split. Each model is evaluated on the corresponding DEV and TEST sets.

and test; and the organizers masked test set labels. We lost some tweets from TRAIN when we crawled using tweet ids, ultimately acquiring 2,036 (TRAIN-A), 281 (DEV) and 466 (TEST). We also make use of the task 1 corpus (95,000 sentences (Bouamor et al., 2018)). More specifically, we concatenate the task 1 data to the training data of task 2, to create TRAIN-B. Again, note that TEST labels were only released to participants after the official task evaluation. Table 4 shows statistics of the data. More information about the data is in (Bouamor et al., 2018). We use TRAIN-A to perform supervised modeling with BERT and TRAIN-B for self training, under various conditions. We refer the reader to (Zhang and Abdul-Mageed, 2019b) for more information about our different experimental settings on dialect id. We acquire our best results with self-training, with a classification accuracy of 49.39% and F_1 score at 35.44. This is the winning system model in the MADAR shared task and we showed in (Zhang and Abdul-Mageed, 2019b) that our tweet-level predictions can be ported to user-level prediction. On user-level detection, our models perform superbly, with 77.40% acc. and 71.70% F_1 score on unseen MADAR TEST.

	# of tweets		
	TRAIN	DEV	TEST
TRAIN-A	193,086	26,588	43,909
TRAIN-B	288,086	–	–

Table 4: Distribution of classes within the MADAR twitter corpus.

3.3. Emotion

We make use of two datasets, LAMA-DINA and LAMA-DIST (Alhuzali et al., 2018). The LAMA-DINA dataset is a Twitter dataset with a combination of gold labels from (Abdul-Mageed et al., 2016) and distant supervision labels. The tweets are labeled with the Plutchik 8 primary emotions from the set: {*anger, anticipation, disgust, fear, joy, sadness, surprise, trust*}. The distant supervision approach depends on use of seed phrases with the Arabic first person pronoun انا (Eng. "I") + a seed word expressing an emotion, e.g., فرحان (Eng. "happy"). The manually labeled part of the data comprises tweets carrying the seed phrases verified by human annotators 9,064 tweets for inclusion of the respective emotion. LAMA-DIST (182,605 tweets) [7] is only labeled using distant supervision. For more information about the dataset, readers are referred to (Alhuzali et al., 2018). The data distribution over the emotion classes is in Table 5. We combine LAMA+DINA and LAMA-DIST training set and refer to this new training set as LAMA-D2 (189,903 tweets). We fine-tune BERT on the LAMA-D2 and evaluate the model with same DEV and TEST sets from LAMA+DINA. On DEV set, the fine-tuned BERT model obtains 61.43% acc. and 58.83 F_1. On TEST set, we acquire 62.38% acc. and 60.32% F_1.

	LAMA+DINA		LAMA-DIST	
	#	%	#	%
anger	1,038	11.45	3,650	2.00
anticipation	933	10.29	24,672	13.51
disgust	1,069	11.79	2,478	1.36
fear	1,434	15.82	28,315	15.51
happy	1,364	15.05	55,253	30.26
sad	1,195	13.18	27,584	15.11
surprise	1,167	12.88	15,106	8.27
trust	864	9.53	25,547	13.99
total	9,064	100.00	182,605	100.00

Table 5: Emotion class distribution in LAMA+DINA and LAMA-DIST datasets.

3.4. Irony

We use the dataset for irony identification on Arabic tweets released by IDAT@FIRE2019 shared task (Ghanem et al., 2019). The shared task dataset contains 5,030 tweets related to different political issues and events in the Middle East taking place between 2011 and 2018. Tweets are

---

[7] These statistics are based on minor cleaning of the data to remove short tweets < 3 words and residuals of the seeds used for collecting the data.

collected using pre-defined keywords (i.e., targeted political figures or events) and the positive class involves ironic hashtags such as #sokhria, #tahakoum, and #maskhara (Arabic variants for "irony"). Duplicates, retweets, and non-intelligible tweets are removed by organizers. Tweets involve both MSA as well as dialects at various degrees of granularity such as *Egyptian*, *Gulf*, and *Levantine*.

IDAT@FIRE2019 (Ghanem et al., 2019) is set up as a binary classification task where tweets are assigned labels from the set {*ironic*, *non-ironic*}. A total of $4,024$ tweets were released by organizers as training data. In addition, a total of $1,006$ tweets were used by organizers as TEST data. TEST labels were not release; and teams were expected to submit the predictions produced by their systems on the TEST split. For our models, we split the $4,024$ released training data into 90% TRAIN ($n = 3,621$ tweets; 'ironic'$= 1,882$ and 'non-ironic'$= 1,739$) and 10% DEV ($n = 403$ tweets; 'ironic'$= 209$ and 'non-ironic'$= 194$). We use the same small-GRU architecture of Section 3.1 as our baselines. We fine-tune BERT on our TRAIN, and evaluate on DEV. The small-GRU obtain 73.70% acc. and 73.47% F_1 score. BERT model significantly outperforms the small-GRU, acquiring 81.64% acc. and 81.62% F_1 score.

	Acc	**F_1**
small-GRU	73.70	73.47
BERT	**81.64**	**81.62**

Table 6: Model performance on irony detection.

3.5. Sentiment

We collect 15 datasets related to sentiment analysis of Arabic, including MSA and dialects (Abdul-Mageed and Diab, 2012; Abdulla et al., 2013; Abdul-Mageed et al., 2014b; Nabil et al., 2015; Kiritchenko et al., 2016; Aly and Atiya, 2013; Salameh et al., 2015; Rosenthal et al., 2017; Alomari et al., 2017; Mohammad et al., 2018; Baly et al., 2019). Table 8 shows all the corpora we use. The datasets involve different types of sentiment analysis tasks such as binary classification (i.e., negative or positive), 3-way classification (i.e., negative, neutral, or positive), and subjective language detection. To combine these datasets for binary sentiment classification, we normalize different types of labels to binary tags in the set {'$positive'$, '$negative'$} using the following rules:

- Map {Positive, Pos, or High-Pos} to 'positive'

- Map {Negative, Neg, or High-Neg} to 'negative'

- Exclude samples whose label is not 'positive' or 'negative' such as 'obj', 'mixed', 'neut', or 'neutral'.

After label normalization, we obtain $126,766$ samples. We split this resulting dataset into 80% training (TRAIN), 10% development (DEV), and 10% test (TEST). The distribution of classes in our splits is presented in Table 7. We fine-tune pre-trained BERT on the TRAIN set using PyTorch implementation with $2e - 6$ learning rate and 15 epochs, as explained in Section 2.. Our best model on the DEV set

obtains 80.24% acc. and 80.24% F_1. We evaluate this best model on TEST set and obtain 77.31% acc. and 76.67% F_1.

	TRAIN	**DEV**	**TEST**
# pos	61,555	7,030	7,312
# neg	39,044	7,314	4,511
Total	100,599	14,344	11,823

Table 7: Distribution of sentiment classes in our data splits.

4. AraNet Design and Use

AraNet consists of identifier tools including age, gender, dialect, emotion, irony and sentiment. Each tool comes with an embedded model. The tool comes with modules for performing normalization and tokenization. AraNet can be used either as (1) a Python library or (2) a command-line and interactive tool, as follows:

AraNet as a Python Library: Importing AraNet module as a Python library provides identifier functions. Prediction is based on a text input or a path to a file, and returns the identified class label. The library also returns the probability distribution over all available class labels if needed. This probability is the outcome of the softmax function applied to the last layer (with logits) in each model. Figure 2 shows two examples of using the tool as Python library.

AraNet as a Command-Line and Interactive Tool: AraNet provides scripts supporting both command-line and interactive mode. Command-line mode accepts a text or file path. Interaction mode is good for quick interactive line-by-line experiments and also pipeline re-directions.


```
python araNet.py --batch infile.tsv

python araNet.py < infile.tsv

python araNet.py

أهلاً يا عمنا. تعاشبب شاي <<<

[('Egypt', 0.84)]
```

Figure 3: AraNet usage examples as command-line mode, pipeline, and interactive mode.

AraNet is available through pip or from source on GitHub [12] with detailed documentation.

5. Ethical Considerations

AraNet is trained on data collected from publicly available sources. The distribution of classes across the different tasks are reasonably balanced as listed in the respective sections in the current paper. Meanwhile, we note that we have not used AraNet in real-world situations, nor tested any bias its decisions could involve. As a result, we advise against using AraNet in decision

---

[12]`https://github.com/UBC-NLP/aranet`

Authors	Task	Sources	# Data	#Class	Classes	MSA/DIA
Abdul-Mageed and Diab (2012)	SSA	Wiki.[8], PAT[9], Forums	5, 382	4	Obj, Subj, Pos, Neg and Neut	MSA
Abdulla et al. (2013)	SA	Twitter,	2000	2	Pos, Neg	MSA
Abdul-Mageed et al. (2014b)	SSA	Maktoob[10], Twitter	11918	3	Obj, Subj Pos, Subj Neg and Subj Mixed	MSA+DIA
Nabil et al. (2015)	SSA	Twitter	10000	4	Obj, Subj Pos, Subj Neg and Subj Mixed	MSA
Kiritchenko et al. (2016)	SI	Twitter	1, 366	—	Regression [0,1]	MSA
Aly and Atiya (2013)	SA	Book reviews	63, 000	3	Pos, Neg, or Neut	MSA
Salameh et al. (2015)	SA	BBN Parallel Text[11]	1200	3	Pos, Neg, or Neut	DIA
Salameh et al. (2015)	SA	Twitter	2000	3	Pos, Neg, or Neut	DIA
Rosenthal et al. (2017)	SA	Twitter	9,500	2	Pos, or Neg	MSA
Rosenthal et al. (2017)	SA	Twitter	3,400	3	Pos, Neut, or Neg	MSA
Rosenthal et al. (2017)	SA	Twitter	9,450	5	High-Pos, Pos, Neut, Neg, Hihg-Neg	MSA
Alomari et al. (2017)	SA	Twitter	1800	3	Pos or Neg	DIA
Mohammad et al. (2018)	SA	Twitter	1,800	7	Various levels of Pos, Neg or Neut [-3,3]	MSA
Saad (2019)[*]	SA	Twitter	58,751	2	Pos, or Neg	DIA
Baly et al. (2019)	SA	Twitter	4,000	5	High-Pos, Pos, Neut, Neg, Hihg-Neg	DIA

Table 8: Sentiment analysis datasets. **SA**: Sentiment analysis. **SSA**: Subjectivity and sentiment analysis. [*]Dataaet from Saad (2019) is available at `https://www.kaggle.com/mksaad/arabic-sentiment-twitter-corpus`.

making without prior research as to what its deployment could involve and how best it can be tested. ***We also do not approve any use of the AraNet or its decisions in any form for manipulative, unfair, malicious, dangerous, or otherwise unlawful (including by international standards) causes by individuals or organizations.*** Our conviction is that machine-learning-based software can be very powerful and useful, if not at times necessary, but must be tested and deployed *only* carefully and ethically. AraNet is no exception.

6. Related Works

As we pointed out earlier, there are several works on some of the tasks but less on others. By far, Arabic sentiment analysis has been the most popular task. Works focused on both MSA (Abdul-Mageed et al., 2011; Abdul-Mageed et al., 2014a) and dialects (Nabil et al., 2015; ElSahar and El-Beltagy, 2015; Al Sallab et al., 2015; Al-Moslmi et al., 2018; Al-Smadi et al., 2019; Al-Ayyoub et al., 2019; Farha and Magdy, 2019). A number of studies have been published on dialect detection, including (Zaidan and Callison-Burch, 2011; Zaidan and Callison-Burch, 2014; Elfardy and Diab, 2013; Cotterell and Callison-Burch, 2014). Some works took as their target the tasks of age detection (Zaghouani and Charfi, 2018; Rangel et al., 2019), gender detection (Zaghouani and Charfi, 2018; Rangel et al., 2019), irony identification (Karoui et al., 2017; Ghanem et al., 2019), and emotion analysis (Abdul-Mageed et al., 2016; Alhuzali et al., 2018).

A number of resources and tools exist for Arabic natural language processing, including Penn Arabic treebank (Maamouri et al., 2004), Buckwalter Morphological Analyzer (Buckwalter, 2002), segmenters (Abdelali et al., 2016), POS taggers (Abumalloh et al., 2016; Diab et al., 2004), morpho-syntactic analyzers (Abdul-Mageed et al., 2013; Pasha et al., 2014), subjectivity and sentiment analysis (Abdul-Mageed, 2019; Farha and Magdy, 2019), offensive and hateful language (Elmadany et al., 2020), and

dangerous speech (Alshehri et al., 2020).

7. Conclusion

We presented AraNet, a deep learning toolkit for a host of Arabic social media processing. AraNet predicts age, dialect, gender, emotion, irony, and sentiment from social media posts. It delivers either state-of-the-art or competitive performance on these tasks. It also has the advantage of using a unified, simple framework based on the recently-developed BERT model. AraNet has the potential to alleviate issues related to comparing across different Arabic social media NLP tasks, by providing one way to test new models against AraNet predictions (i.e., model-based comparisons). Our toolkit can be used to make important discoveries about the Arab world, a vast geographical region of strategic importance. It can enhance also enhance our understating of Arabic online communities, and the Arabic digital culture in general.

8. Bibliographic References

Abdelali, A., Darwish, K., Durrani, N., and Mubarak, H. (2016). 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*, pages 11–16.

Abdul-Mageed, M. and Diab, M. T. (2012). Awatif: A multi-genre corpus for modern standard arabic subjectivity and sentiment analysis. In *LREC*, volume 515, pages 3907–3914. Citeseer.

Abdul-Mageed, M., Korayem, M., and YoussefAgha, A. (2011). "yes we can?": Subjectivity annotation and tagging for the health domain. In *Proceedings of RANLP2011*.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2013). Asma: A system for automatic segmentation and morpho-syntactic disambiguation of modern standard arabic. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 1–8.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014a). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M., Diab, M., and Kübler, S. (2014b). Samar: Subjectivity and sentiment analysis for arabic social media. *Computer Speech & Language*, 28(1):20–37.

Abdul-Mageed, M., AlHuzli, H., and Duaa'Abu Elhija, M. D. (2016). Dina: A multidialect dataset for arabic emotion analysis. In *The 2nd Workshop on Arabic Corpora and Processing Tools*, page 29.

Abdul-Mageed, M. (2019). Modeling arabic subjectivity and sentiment in lexical space. *Information Processing & Management*, 56(2):291–307.

Abdulla, N., Mahyoub, N., Shehab, M., and Al-Ayyoub, M. (2013). Arabic sentiment analysis: Corpus-based and lexicon-based. In *Proceedings of The IEEE conference on Applied Electrical Engineering and Computing Technologies (AEECT)*.

Abumalloh, R. A., Al-Sarhan, H. M., Ibrahim, O., and Abu-Ulbeh, W. (2016). Arabic part-of-speech tagging. *Journal of Soft Computing and Decision Support Systems*, 3(2):45–52.

Al-Ayyoub, M., Khamaiseh, A. A., Jararweh, Y., and Al-Kabi, M. N. (2019). A comprehensive survey of arabic sentiment analysis. *Information Processing & Management*, 56(2):320–342.

Al-Moslmi, T., Albared, M., Al-Shabi, A., Omar, N., and Abdullah, S. (2018). Arabic senti-lexicon: Constructing publicly available language resources for arabic sentiment analysis. *Journal of Information Science*, 44(3):345–362.

Al Sallab, A., Hajj, H., Badaro, G., Baly, R., El Hajj, W., and Shaban, K. B. (2015). Deep learning models for sentiment analysis in arabic. In *Proceedings of the second workshop on Arabic natural language processing*, pages 9–17.

Al-Smadi, M., Talafha, B., Al-Ayyoub, M., and Jararweh, Y. (2019). Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *International Journal of Machine Learning and Cybernetics*, 10(8):2163–2175.

Alhuzali, H., Abdul-Mageed, M., and Ungar, L. (2018). Enabling deep learning of emotion with first-person seed expressions. In *Proceedings of the Second Workshop on Computational Modeling of People's Opinions, Personality, and Emotions in Social Media*, pages 25–35.

Alomari, K. M., ElSherif, H. M., and Shaalan, K. (2017). Arabic tweets sentimental analysis using machine learning. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 602–610. Springer.

Alshehri, A., Nagoudi, E. M. B., and Abdul-Mageed, M. (2020). Understanding and detecting dangerous speech in social media. In *The 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4), LREC*.

Aly, M. and Atiya, A. (2013). 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)*, volume 2, pages 494–498.

Baly, R., Khaddaj, A., Hajj, H., El-Hajj, W., and Shaban, K. B. (2019). Arsentd-lev: A multi-topic corpus for target-based sentiment analysis in arabic levantine tweets. *arXiv preprint arXiv:1906.01830*.

Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., et al. (2018). The madar arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Bouamor, H., Hassan, S., and Habash, N. (2019a). The madar shared task on arabic fine-grained dialect identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19), Florence, Italy*.

Bouamor, H., Hassan, S., and Habash, N. (2019b). The MADAR Shared Task on Arabic Fine-Grained Dialect Identification. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop (WANLP19)*, Florence, Italy.

Buckwalter, T. (2002). Buckwalter arabic morphological analyzer version 1.0. *Linguistic Data Consortium, University of Pennsylvania*.

Cotterell, R. and Callison-Burch, C. (2014). A multi-dialect, multi-genre corpus of informal written arabic. In *LREC*, pages 241–245.

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

Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic tagging of arabic text: From raw text to base phrase chunks. In *Proceedings of HLT-NAACL 2004: Short papers*, pages 149–152. Association for Computational Linguistics.

Elfardy, H. and Diab, M. T. (2013). Sentence level dialect identification in arabic. In *ACL (2)*, pages 456–461.

Elmadany, A., Zhang, C., Abdul-Mageed, M., and Hashemi, A. (2020). Leveraging affective bidirectional transformers for offensive language detection. In *The 4th Workshop on Open-Source Arabic Corpora and Processing Tools (OSACT4), LREC*.

ElSahar, H. and El-Beltagy, S. R. (2015). Building large arabic multi-domain resources for sentiment analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 23–34. Springer.

Farha, I. A. and Magdy, W. (2019). Mazajak: An online arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198.

Ghanem, B., Karoui, J., Benamara, F., Moriceau, V., and Rosso, P. (2019). Idat@fire2019: Overview of the track on irony detection in arabic tweets. In *Mehta P., Rosso P., Majumder P., Mitra M. (Eds.) Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS.org, Kolkata, India, December 12-15*.

Karoui, J., Zitoune, F. B., and Moriceau, V. (2017). Soukhria: Towards an irony detection system for arabic in social media. *Procedia Computer Science*, 117:161–168.

Kiritchenko, S., Mohammad, S., and Salameh, M. (2016). Semeval-2016 task 7: Determining sentiment intensity of english and arabic phrases. In *Proceedings of the 10th international workshop on semantic evaluation (SEMEVAL-2016)*, pages 42–51.

Maamouri, M., Bies, A., Buckwalter, T., and Mekki, W. (2004). The penn arabic treebank: Building a large-scale annotated arabic corpus. In *NEMLAR conference on Arabic language resources and tools*, volume 27, pages 466–467. Cairo.

Mohammad, S., Bravo-Marquez, F., Salameh, M., and Kiritchenko, S. (2018). Semeval-2018 task 1: Affect in tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 1–17.

Nabil, M., Aly, M., and Atiya, A. (2015). Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2515–2519.

Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). Madamira: A fast, comprehensive tool for morphological analysis and disambiguation of arabic. In *LREC*, volume 14, pages 1094–1101.

Pavlinek, M. and Podgorelec, V. (2017). Text classification method based on self-training and lda topic models. *Expert Systems with Applications*, 80:83–93.

Rangel, F., Rosso, P., Charfi, A., Zaghouani, W., Ghanem, B., and Sánchez-Junquera, J. (2019). Overview of the track on author profiling and deception detection in arabic. In *Mehta P., Rosso P., Majumder P., Mitra M. (Eds.) Working Notes of the Forum for Information Retrieval Evaluation (FIRE 2019). CEUR Workshop Proceedings. In: CEUR-WS.org, Kolkata, India, December 12-15*.

Rosenthal, S., Farra, N., and Nakov, P. (2017). Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th international workshop on semantic evaluation (SemEval-2017)*, pages 502–518.

Salameh, M., Mohammad, S., and Kiritchenko, S. (2015). Sentiment after translation: A case-study on arabic social media posts. In *Proceedings of the 2015 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 767–777.

Triguero, I., García, S., and Herrera, F. (2015). Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information Systems*, 42(2):245–284, Feb.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Zaghouani, W. and Charfi, A. (2018). Arap-tweet: A large multi-dialect twitter corpus for gender, age and language variety identification. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Zaidan, O. F. and Callison-Burch, C. (2011). The arabic online commentary dataset: an annotated dataset of informal arabic with high dialectal content. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 37–41. Association for Computational Linguistics.

Zaidan, O. F. and Callison-Burch, C. (2014). Arabic dialect identification. *Computational Linguistics*, 40(1):171–202.

Zhang, C. and Abdul-Mageed, M. (2019a). Multi-task bidirectional transformer representations for irony detection. In *The 11th meeting of the Forum for Information Retrieval Evaluation 2019*.

Zhang, C. and Abdul-Mageed, M. (2019b). No army, no navy: Bert semi-supervised learning of arabic dialects. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 279–284.