Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 189 (2021) 92–99

Procedia Computer Science

www.elsevier.com/locate/procedia

5th International Conference on AI in Computational Linguistics

A Generative Adversarial Network for Data Augmentation: The Case of Arabic Regional Dialects

Xavier A. Carrasco[a], Ashraf Elnagar[a,], Mohammed Lataifeh[a]

[a]Department of CS, University of Sharjah, P O Box 27272, Sharjah, UAE

Abstract

Text Generation using Generative Adversarial Networks (GANs) has been successful in domains such as sentiment analysis using Sentimental GAN (SentiGAN) model. We adopt a similar model to generate sentences for five regional Arabic dialects (Egypt, Gulf, Maghreb, Levant, and Iraq). The objective is to overcome the scarcity of richly annotated Dialectal Arabic (DA) datasets by automatic generation of such corpora. The DA generation process for a specific dialect, relies on a generator to create new text, and a discriminator to evaluate that text, with a dynamic update that will allow the process to run automatically without supervision. Novelty and diversity are the two metrics used to verify the consistency and quality of the generated DA text before enriching the sought datasets. Experimental results confirm the reliability and value of the generated datasets when tested by different classifiers.

Keywords: Generative Adversarial Networks; Dialectal Arabic; Dataset Augmentation; Classification.

The Generative Adversarial Networks (GANs) were introduced some years ago by Ian Goodfellow [10], the generator objective is to learn the common features for images in a dataset and generate new images of similar features, while the discriminator job is to validate and classify these images as real or fake. Despite being thought for image generation, the adversarial algorithm has been successfully deployed with different types of data such as text [7, 11, 26, 28] and sounds [9, 27].

Based on the training process, one of the most recent works in text generation is SentiGAN [26], which was implemented and applied on several datasets, presenting good results in terms of quality, novelty, and diversity for the generated sentences. However, it still has many limitations like the number of sentiments that can be generated simultaneously by the generators, or the maximum allowed number of words required to obtain acceptable results.

The dataset used in this work is divided into five different regions, represented by different classes, each of which corresponds to a different dialect. We utilize a SentiGAN-like model to populate low-resource dialects. However, we used one generator per class instead of multiple generators. This change has several consequences that are going to be discussed further, but the results obtained were remarkable considering the metrics and classification results. Being the mother tongue of a population of more than 460 million [1], the Arabic language is the official language in 22 countries located primarily in the Middle East and Africa. Within each country, there are variations from the source Arabic language projected as different dialects that can be classified at city, country, and regional levels.

In this work, five regional dialects (Egypt, Gulf, Maghreb, Levant, and Iraq) are considered due to their representative coverage among the stated population. The objective here is to build richer datasets for DA that maintain the main features of the original colloquial text, while generating diverse novel sentences. To the best of our knowledge, this is the first work that uses GANs to generate dialectal text.

The remainder of this paper is organized as follows. In Section 1, we describe previous research work on the subject. Next, we describe the dataset and the generator model in Section 2. In Section 3, we demonstrate the experimental results. Finally, we conclude our research in Section 4.

10.1016/j.procs.2021.05.072

1. Related work

The first works on text generation were based on Recurrent Neural Networks [19]. However, RNNs-like models suffer from bias exposure, [3], and the ill-suited loss function for the sentence generator. Similarly, other approaches such as Bidirectional Encoder Representations from Transformer known as BERT [8], and Generative Pre-trained Transformer else known as GPT-2 [6] and GPT-3[5], and despite their successful utilization for English language and superior performance compared to RNN for tasks like classifications and machine translation [21], they suffer from different structural and algorithmic limitations highlighted by the authors and else where [30, 25] in addition to the large dataset often required for training. Therefore, GANs were proposed as a solution for the stated problems and for being better suited for the scope of this work.

Some of the early GANs examples used for text generation are SeqGAN [28], RankGAN [17], MaliGAN [7] and LeakGAN [11]. For this work, we adopt SentiGAN [26] that uses an LSTM layer [13] as a generator, while the discriminator is a Convolutional Neural Network (CNN) [16] with proven effectiveness in text classification [29].

Furthermore, the generator is also optimised using policy gradient methods which incorporate signals from the discriminator.

There are limited works that have been dedicated for Arabic text generation, a good example was presented by Souri in [22], which uses an RNN and some criteria of the Arabic language that help the network to improve the quality of the output. A different perspective was explored by Ismail in [14] that is based on a semantic representation of the language, and most recently AraGPT2 [12] where a combination of BERT and GPT-2 were used for modern standard Arabic.

Despite being relatively new, research on dialectal Arabic certainly continues to gain momentum. The focus here is on generating dialectal Arabic text, so for detection and classification, we refer the reader to a recent survey [20] and further examples of robust classifiers used for dialectal Arabic on Colloquial Arabic datasets [18, 15].

2. Datasets and Generative Model

2.1. Datasets

The original dataset for the Arabic language used is called MADAR [4], which covers the dialects for 25 cities from the Arab World, created by translating selected sentences from the Basic Travel Expression Corpus introduced in [23]. The regional datasets were compiled by collecting several sources of data from the cities that makeup a specific regional dialect. In order to prepare the datasets for training the GAN, every sentence is cleaned and limited to a minimum length of 2 words and a maximum of 18.

The punctuation marks (Latin and Arabic), Latin characters and numbers were removed from the sentences. As mentioned earlier, the regional group comprising of 5 specific datasets: Egyptian, Gulf Arabic, Maghrebi, Levantine and Iraqi. The statistics of sentences for the five regional dialects are shown Figure 1.

2.2. GAN Model

SentiGAN [26] was designed to generate different sentiments by using multiple generators and one discriminator. Although, such architecture is limited by the number of sentiments and maximum words in a sentence, high-quality output was reported in the paper while working with 2 sentiments and a maximum of 18 words per sentence. In this work, 5 dialects are used instead of 2, therefore, we chose to treat each dialect independently having a total of 5 generators/discriminators working independently on each specific dialect, without cross-groups mixing.

A general diagram of the model adopted can be seen in Figure 2, followed by the implemented algorithm for the multiple discriminators.

Nonetheless, the improvements and differences between SentiGAN and other GANs for text revolves around the objective functions and penalties briefly explained below.

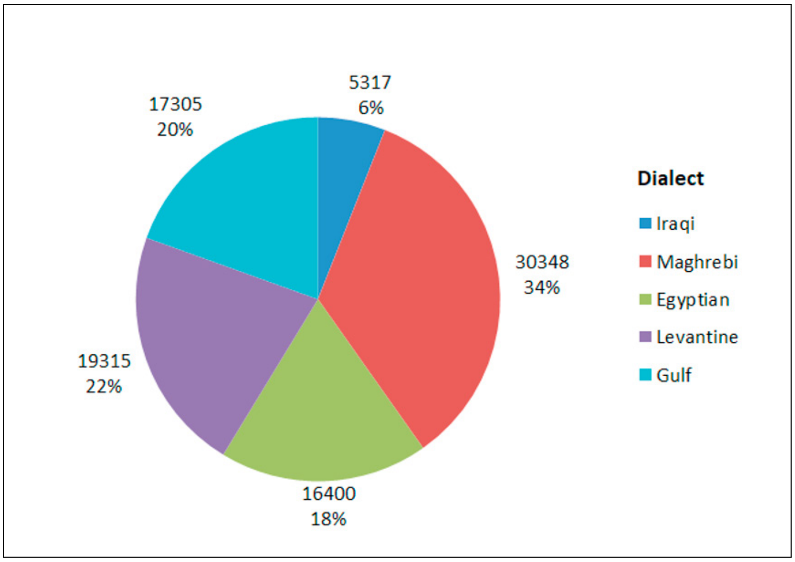

Fig. 1. Number of sentences for each regional dialect.

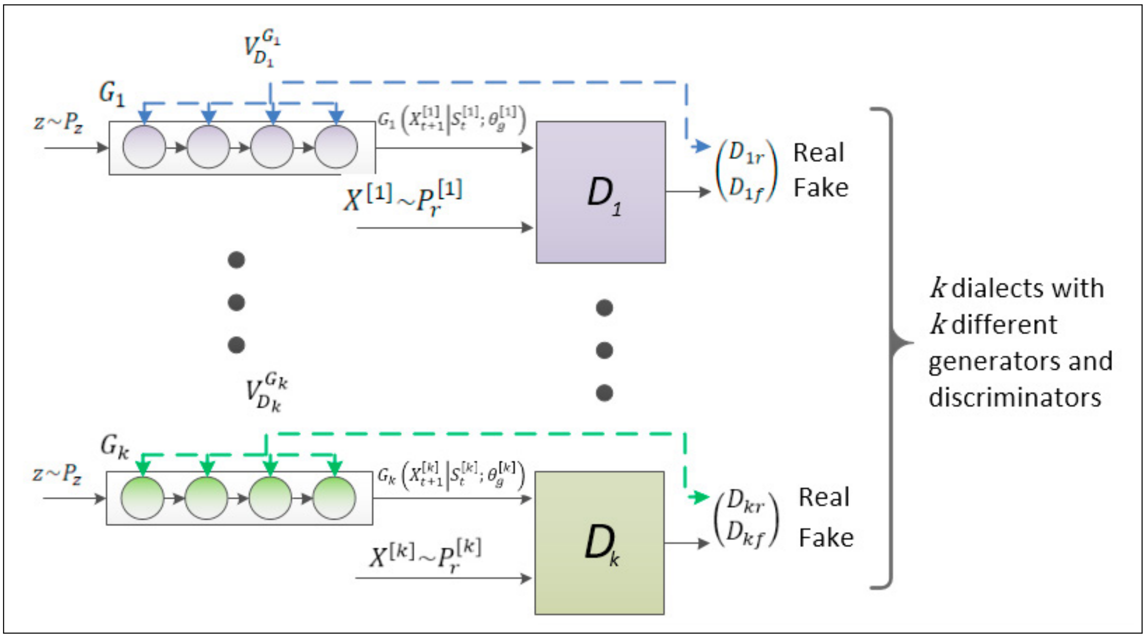

Fig. 2. General architecture of the GAN for k independent dialects.

Penalty instead of reward for the discriminator.
A difference between SeqGAN and SentiGAN is the use of a penalty function instead of a reward for the discriminator, which helps to generate a larger variety of sentences, the original reward function was $D_i(X^{[i]}; \theta_d^{[i]})$ changed by the penalty in Eq. 1, the use of this function will be shown in the next subtitle.

$$V_{D_i}^{G_i}(X^{[i]}) = 1 - D_i(X^{[i]}; \theta_d^{[i]}) \tag{1}$$

Generator objective function.
One of the most difficult problems that appear when dealing with GANs is the mode collapse of the generator [24].

Such issue is manifested by the generation of an output without any changes through epochs. Even if the discriminator is fooled, the quality of the output is normally very low in quality and variety.

A partial solution for this problem is changing the loss function [2]. A modified version of this function is used for the generator in SentiGAN, Eq 2.

$$J_{G_i}(X^{[i]}) = G_i(X^{[i]}_{t+1}|S^{t+1}_{t};\theta^{[i]}_{g}) \cdot V^{G_i}_{D_i}(S^{[i]}_{t};X^{[i]}_{t+1}) \tag{2}$$

The generator is trained to generate a sequence $X^{[i]}_{1:t} = \{X^{[i]}_1, \ldots, X^{[i]}_t\}$, where $X^{[i]}_t$ represents a word token in the given vocabulary C. $G_i(X^{[i]}_{t+1}|S^{[i]}_t;\theta^{[i]}_g)$ means the probability that selecting the $(t+1)^{th}$ word depends on the previously generated words (its current state), denoted as $S^{[i]}_t = \{X^{[i]}_1, \ldots, X^{[i]}_t\}$. $V^{G_i}_{D_i}(S^{[i]}_t;X^{[i]}_{t+1})$ is the penalty for a sequence $X_{1:t+1}$, which is calculated by the discriminator. Finally, the objective of the i^{th} generator $G_i(X^{[i]}_{t+1}|S^{[i]}_t;\theta^{[i]}_g)$ is to minimize the value in Equation 1. In this work, each generator is updated with its own penalty and has its own gradients. The X^i_t represents a token that belongs to the i^{th} dataset. As for the discriminator, the loss function to be reduced is shown in Eq 3.

$$J_{D_i}(\theta^{[i]}_d) = -E_{X^{[i]}\sim P^{[i]}_g} \log(D_{k+1}(X;\theta^{[i]}_d)) - E_{X^{[i]}\sim P^{[i]}_r} \log(D_i(X^{[i]};\theta^{[i]}_d]) \tag{3}$$

Where $P^{[i]}_g$ is the text generated by the i^{th} generator, $P^{[i]}_r$ represents the real texts from the respective dialect.

Another important change made by SentiGAN to the original GAN is the addition of pre-training on both, the generator and discriminator as detailed in the algorithm below. The generator is trained by using the Maximum Likelihood Estimation (MLE), while the discriminator is fed with real sentences from the dataset and fake examples from the previously pre-trained generator. This is a common technique to speed up the training process. However, it tends to mode collapse if the discriminator is over-trained.

Algorithm 1: Adversarial training with multiple discriminators

Input: Input noise z; Generators $G_i(X^{[i]}|S^{[i]};\theta^{[i]}_g)^{(i=12)}_{(i=1)}$; Discriminators $D_i(X^{[i]};\theta^{[i]}_d)$; Real text dataset with the k dialects, $T = T_1, \ldots, T_{12}$

Output: Well trained generators, $G_i(X^{[i]}|S^{[i]};\theta^{[i]}_g)^{(i=k)}_{(i=1)}$

```
for i in [1 : k] do
    Initialize G_i and D_i with random weights
    Pre-train G_i using MLE on T
    Generate fake texts F_i using G_i
    Pre-train D_i(X^[i]; θ_d^[i]) using T_i, F_i;
    while GAN is not converging do
        for g-steps do
            Generate fake texts using G_i(z; θ_g^[i])
            Calculate penalty V_{D_i}^{G_i}
            Update G_i(z; θ_g^[i]) by minimizing Eq. (2);
        for d-steps do
            Generate fake texts using F_i using G_i(X^[i]|S^[i]; θ_g^[i])
            Update D(X; θ_d^[i]) using T_i by minimizing Eq. (3);
```

3. Experiments

The generative process provided many different sentences of different lengths. To show the effectiveness of the algorithm, we explore a subset of 2 sentences generated in different regional Arabic dialects. To have comparable results, all selected examples are on the subject of reserving a room in some hotel. The generated sentences, in Table 1, reflect the use of specific words from each dialect, which makes it easy to identify each source region; superb performance.

Table 1. Sample sentences from each regional dialect.

Region	Regional Dialectal Sentence
EGY	عايز اوضه لأتنين
	كام تكلفة الغرفة اتناشر صفر سبعة
GLF	ابغي احجز غرفة حق اليوم
	بغيت احجز غرفة حق الليلتين الجايات
MAG	الغرفة ديالي فيها الصداع بزاف
	بغيت غرفة عفاك
LEV	بدي غرفة لتنين مع تختين مفرد
	في عندك غرفة فاضية الليلة
IRQ	اريد غرفة يم المصعد
	غرفة سرير واحد علمود مرتاح

The number of possible generated sentences was limited by the number of sentences in the dataset. The main reason behind this is the use of independent generators and discriminators, in order to avoid the possibility of mixing common features between the dialects. The generated sentences are added to the original datasets to enrich each single dialect-set by well annotated sentences. Achieving this objective enables the extension of each dataset. However, the next step is to measure the novelty and diversity of the newly added sentences.

3.1. Novelty and Diversity

Novelty evaluation is carried out to ensure that the generator is creating new sentences instead of copying the ones in the dataset. It is an independent metric that cannot be measured by the fluency or logic in the sentence. This measure, [26], is expressed as follows:

$$Novelty(S_i) = 1 - \max\{\varphi(S_i, C_j)\}_{j=1}^{j=|C|} \tag{4}$$

Where C is the sentences set, φ is the Jaccard similarity function and S_i is a generated sentence.

On the other hand, diversity measures how different are the generated sentences from the original ones in the same dataset. Given a collection of generated sentences, S, we define the diversity of sentences S_i as follows:

$$Diversity(S_i) = 1 - \max\{\varphi(S_i, S_j)\}_{j=1}^{j=|S|, j \neq i} \tag{5}$$

The average results for novelty and diversity are shown in Table 2, in both cases a higher value means more novel and diversified sentences, and the demonstrated results are encouraging for building a new dataset.

Fluency however, is a difficult metric to obtain. It is mainly used to determine if the sentence has the right meaning. Such metric is usually evaluated by human readers.

Another test to verify the effectiveness of the generated datasets is to build classifiers and test if they can predict the correct dialect for a given sentence. This would confirm whether the dialect-specific features were kept through the generation process. This will be discussed next.

3.2. Classification

We implemented four different classifiers, which are: Support Vector Machines (SVM), Multinomial Naïve Bayes (MNB), Long-Short Term Memory network (LSTM), General Deep Neural Network (DNN), and Convolutional Neural Network (CNN). The models were trained and validated 4 times, covering the following scenarios:

Table 2. Mean values of novelty and diversity (originals → generated) metrics.

Dialect	Novelty	Diversity
EGY	0.770	0.594 → 0.400
GLF	0.771	0.579 → 0.399
MAG	0.771	0.617 → 0.427
LEV	0.768	0.636 → 0.406
IRQ	0.794	0.632 → 0.488

Table 3. Classification results of the 4 experiments on the testing datasets.

Classifier	1st		2nd		3rd		4th	
	Loss	Acc. (%)	Loss	Acc. (%)	Loss	Acc. (%)	Loss	Acc. (%)
SVM	0.46	**83.35**	0.45	83.02	0.26	90.20	0.26	**90.65**
MNB	0.54	**84.50**	0.52	84.38	0.39	92.12	0.36	**92.38**
DNN	1.17	81.75	0.80	**82.15**	0.44	**91.20**	0.65	89.95
LSTM	1.30	81.63	1.28	**82.73**	0.62	**90.93**	0.61	90.55
CNN	1.28	82.17	1.16	**82.65**	0.54	**90.18**	0.72	89.42

- *1^{st} scenario*: Trained and validated on original sentences; tested on original sentences.
- *2^{nd} scenario*: Trained and validated on original + generated sentences; tested on original sentences.
- *3^{rd} Scenario*: Trained and validated on generated sentences; tested on original sentences.
- *4^{th} Scenario*: Trained and validated on original + generated sentences; tested on generated sentences.

The 1^{st} scenario shows the initial expected performance for a classifier trained only using the original dataset. This information is helpful to study the impact of adding the generated sentences (2^{rd} scenario). Similarly, we study the 3^{rd} and 4^{th} scenarios but testing on the generated dataset.

It should be noted that the amount of available data for regional dialects is satisfactory for building robust classifiers. However, our aim here is to study the quality of generated sentences. Therefore, we created a dataset that is composed of original and generated sentences. We have conducted the following experiments:

- *1^{st} experiment*: 10000 original sentences per regional dialect for training and validation; 1000 original sentences per class for testing.
- *2^{nd} experiment*: 5000 original and 5000 generated sentences per regional dialect for training and validation; 1000 original sentences per class for testing.
- *3^{rd} experiment*: 10000 original sentences per regional dialect for training and validation; 1000 generated sentences per class for testing.
- *4^{th} experiment*: 5000 original and 5000 generated sentences per regional dialect for training and validation; 1000 generated sentences per class for testing.

The resulting performance of all classifiers using accuracy and loss metrics are demonstrated in Table 3. The results confirm that the generated dialectal sentences are of similar high-quality of original ones. For example, 1^{st} experiment (that uses original dataset) and 2^{nd} experiment (that uses original and generated datasets) report comparable accuracy scores between 81% and 85% while testing on the original dataset only. In fact, the three deep learning models of the 2^{nd} experiment outperform the scores of the 1^{st} experiment. On the other hand,the 3^{rd} experiment (that uses original dataset) and 4^{th} experiment (that uses original and generated datasets) reported a comparable but higher accuracy scores between 89% and 93% while tested on the generated dataset only. The nearly 8% jump when testing on the generated sentences compared to the original ones confirms the efficiency of the generator in preserving key features of each dialect.

4. Conclusions

In this paper, we utilized a modified SentiGAN to enrich low-size datasets by generating a diversity of high-quality sentences for twelve different Arabic dialects obtained from MADAR dataset. The number of generated sentences was higher than the number in the original dataset. The procedure reduces the vocabulary size and only uses the most common words. Although the reduction affects marginally the diversity of the generated text, the detected features are key to each dialect, which results in robust classification performance. The novelty score is very satisfactory for all the generated datasets, which suggests that the generative process is capable of producing new sentences based on the detected key features. Experimental results demonstrate not only the consistency and effectiveness of the generator process, but also the enhancement of classification results when using the original and generated datasets.The improvement in the accuracy score is significant as shown in the experiments.

An extension of this work will be to generate country-level dialects with data augmentation for the DA datasets.

References

[1] Ali, H.H.E.S., Dana, A.M.Y., 2020. The bridging language between diglossia of classical and colloquial arabic. Al-Arabi: Journal of Teaching Arabic as a Foreign Language 4, 130–162.

[2] Arjovsky, M., Chintala, S., Bottou, L., 2017. Wasserstein generative adversarial networks, in: 34th Int. Conf. Mach. Learn. ICML 2017.

[3] Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. arXiv preprint arXiv:1506.03099 .

[4] 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).

[5] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. arXiv preprint arXiv:2005.14165 .

[6] Budzianowski, P., Vulić, I., 2019. Hello, it's gpt-2–how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. arXiv preprint arXiv:1907.05774 .

[7] Che, T., Li, Y., Zhang, R., Hjelm, R.D., Li, W., Song, Y., Bengio, Y., 2017. Maximum-likelihood augmented discrete generative adversarial networks. arXiv preprint arXiv:1702.07983 .

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

[9] Donahue, C., McAuley, J., Puckette, M., 2018. Adversarial audio synthesis. arXiv preprint arXiv:1802.04208 .

[10] Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial networks. arXiv preprint arXiv:1406.2661 .

[11] Guo, J., Lu, S., Cai, H., Zhang, W., Yu, Y., Wang, J., 2018. Long text generation via adversarial training with leaked information, in: Proceedings of the AAAI Conference on Artificial Intelligence.

[12] Harrag, F., Debbah, M., Darwish, K., Abdelali, A., 2021. Bert transformer model for detecting arabic gpt2 auto-generated tweets. arXiv preprint arXiv:2101.09345 .

[13] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

[14] Ismail, S.S., Aref, M., Moawad, I.F., 2015. A model for generating arabic text from semantic representation, in: 2015 11th International Computer Engineering Conference (ICENCO), IEEE. pp. 117–122.

[15] Khalifa, Y., Elnagar, A., 2020. Colloquial arabic tweets: Collection, automatic annotation, and classification, in: 2020 International Conference on Asian Language Processing (IALP), IEEE. pp. 163–168.

[16] LeCun, Y., Haffner, P., Bottou, L., Bengio, Y., 1999. Object recognition with gradient-based learning, in: Shape, contour and grouping in computer vision. Springer, pp. 319–345.

[17] Lin, K., Li, D., He, X., Zhang, Z., Sun, M.T., 2017. Adversarial ranking for language generation. arXiv preprint arXiv:1705.11001 .

[18] Lulu, L., Elnagar, A., 2018. Automatic arabic dialect classification using deep learning models. Procedia computer science 142, 262–269.

[19] Mikolov, T., Kombrink, S., Burget, L., Černockỳ, J., Khudanpur, S., 2011. Extensions of recurrent neural network language model, in: 2011 IEEE international conference on acoustics, speech and signal processing (ICASSP), IEEE. pp. 5528–5531.

[20] Nassif, A.B., Elnagar, A., Shahin, I., Henno, S., 2020. Deep learning for arabic subjective sentiment analysis: Challenges and research opportunities. Applied Soft Computing , 106836.

[21] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI blog 1, 9.

[22] Souri, A., El Maazouzi, Z., Al Achhab, M., El Mohajir, B.E., 2018. Arabic text generation using recurrent neural networks, in: International Conference on Big Data, Cloud and Applications, Springer. pp. 523–533.

[23] Takezawa, T., Kikui, G., Mizushima, M., Sumita, E., 2007. Multilingual spoken language corpus development for communication research, in: International Journal of Computational Linguistics & Chinese Language Processing, Volume 12, Number 3, September 2007: Special Issue on Invited Papers from ISCSLP 2006, pp. 303–324.

[24] Theis, L., Oord, A.v.d., Bethge, M., 2015. A note on the evaluation of generative models. arXiv preprint arXiv:1511.01844 .
[25] de Vries, W., Nissim, M., 2020. As good as new. how to successfully recycle english gpt-2 to make models for other languages. arXiv preprint arXiv:2012.05628 .
[26] Wang, K., Wan, X., 2019. Automatic generation of sentimental texts via mixture adversarial networks. Artificial Intelligence 275, 540–558.
[27] Yang, S.H., Chung, M., 2019. Self-imitating feedback generation using gan for computer-assisted pronunciation training. arXiv preprint arXiv:1904.09407 .
[28] Yu, L., Zhang, W., Wang, J., Yu, Y., 2017. Seqgan: Sequence generative adversarial nets with policy gradient, in: Proceedings of the AAAI conference on artificial intelligence.
[29] Zhang, X., LeCun, Y., 2015. Text understanding from scratch. arXiv preprint arXiv:1502.01710 .
[30] Zhao, T.Z., Wallace, E., Feng, S., Klein, D., Singh, S., 2021. Calibrate before use: Improving few-shot performance of language models. arXiv preprint arXiv:2102.09690 .