Character-level Adversarial Examples in Arabic

Basemah Alshemali
College of Computer Science and Engineering
Taibah University
Almadinah, KSA
College of Engineering and Applied Science
University of Colorado at Colorado Springs
Colorado Springs, USA
balshema@uccs.edu

Jugal Kalita
College of Engineering and Applied Science
University of Colorado at Colorado Springs
Colorado Springs, USA
jkalita@uccs.edu

***Abstract*—Several adversarial attacks have been proposed in the domains of computer vision and natural language processing (NLP). However, most attacks in the NLP domain have been applied to evaluate deep neural networks (DNNs) that were trained on English corpora. This paper proposes the first set of character-level adversarial attacks designed for models trained on Arabic. We present an efficient method to generate character-level adversarial examples against neural classifiers. Our method relies on flip operations that were designed based on the most common spelling mistakes that non-native Arabic learners make. We find that only a few manipulations are needed to mislead powerful and popular DNN-based classifiers trained on Arabic corpora.**

***Index Terms*—adversarial examples, adversarial attacks, Arabic, spelling mistakes, NLP, DNNs.**

I. Introduction

Studies have shown that DNNs are vulnerable to adversarial attacks – carefully-designed input samples that can trick even high-performing models. Small perturbations to the inputs can mislead such networks into making wrong decisions. This vulnerability has been exposed in the domains of computer vision [1], [2], speech, and NLP [3], [4].

Recently, a growing body of research has focused on textual adversarial attacks. Based on the attack's level of perturbation, three categories of adversarial attacks have been proposed in the NLP domain: Character-level, token-level, and sentence-level adversarial attacks [5]. Character-level attacks were designed to produce misspelled words. Although several character-level perturbations have been designed to evaluate DNNs [6]–[12], they have thus far only focused on models trained on English texts.

This paper proposes the first set of character-level adversarial attacks designed for models trained on Arabic. We present an efficient method to generate character-level adversarial examples against DNN-based text classification models. Our method relies on flip operations that were designed based on the most common spelling mistakes that non-native Arabic learners make. We find that only a few manipulations are needed to mislead powerful and popular DNN-based classifiers trained on Arabic corpora.

II. Related Work

As there are no studies on character-level adversarial examples targeting DNN-based classifiers with Arabic text, here we explore studies on English-based models. There has been some work on testing the robustness of DNNs against character-level adversarial examples (misspelled words). Early research demonstrated that NLP models are fragile to input perturbations.

Heigold et al. [13] evaluated DNN-based models with noisy inputs and showed that the performance of the models degraded from around 95.00% to around 80.00% classification accuracy, a decrease that was fairly consistent across all noise types. Ebrahimi et al. [14] generated character-level adversarial samples to evaluate neural text classifiers. They demonstrated that only a few single character changes were needed to trick the classifier.

Naik et al. [15] proposed an evaluation methodology for neural models based on spelling errors. Their evaluation revealed weaknesses within these models, and the performance of all models dropped across all adversaries. Liang et al. [16] developed a method to craft character-level adversarial samples against text classifiers. Their results illustrated that by applying their adversarial strategies, the classification of a given sample can be altered to any desired target class, or to an arbitrary target class. Gao et al. [6] generated character-level adversarial modifications on the input tokens to evaluate DNN-based models. On average, the prediction accuracy of their models was reduced by about 47.36%

Pruthi et al. [7] evaluated DNN-based models and stated that a single adversarially-chosen character attack can lower the classification accuracy of the BERT model [17] from 90.30% to 45.80%. Sun et al. [18] showed that even a small typographical error may confuse the most advanced models, such as BERT and XLNet [19], and hurt their performance. Jones et al. [8] concluded that attacking BERT using character-level adversaries results in dramatic performance drops. Typos reduced the average accuracy of BERT from 86.2% to 15.7% across six tasks.

Kumar et al. [9] showed that BERT's performance on

fundamental NLP tasks drops significantly in the presence of noisy data (misspelled words and typos). Misspellings in words chosen at random is good enough to cause substantial drop in BERT's performance. Aspillaga et al. [10] evaluated the robustness of XLNet and BERT models in the presence of misspelled words. They discovered that these models are still very fragile and demonstrate various unexpected behaviors. All types of misspelled words have a significant negative impact on accuracy on all tested models. For instance, swapping characters had between 46.30% and 59.00% reduction in models' accuracy.

Although there is considerable progress in this area, it can be seen that this paper differentiates from previous works by evaluating the state-of-the-art DNN models, trained on Arabic corpora, when they are exposed to Arabic character-level adversarial attacks.

III. Methodology

It is important to highlight that our study is motivated by the most common spelling mistakes made by non-native Arabic learners. Different types of spelling mistakes have been examined in previous studies [20], [21]. Alyafii [22], Alshibayl and Banithhyab [23], and Yassin et al. [24] discussed the most common kinds of spelling mistakes in the written work of non-native Arabic learners. Overall, spelling errors can be a result of substituting letters when writing a particular word. The substitution errors occur when the speller substitutes one of the letters of the word with a similar one. The main cause of substitution errors of Arabic spelling is the high visual similarity between letters [25], [26]. For instance, learners may mix up the letter "حـ" (ħ) with the letter "جـ" (g) or "خـ" (x).

Table I shows groups of Arabic letters that are difficult to be distinguished by some non-native Arabic learners. In this table, letters are grouped based on their visual similarity. For example, in Group#1, the letters "بـ" (b), "تـ" (t), "ثـ" (th), "نـ" (n), and "يـ" (y) are mostly mixed up by non-native Arabic learners. Our work is motivated by this phenomenon. We wish, in this paper, to examine the effect of this phenomenon on DNN-based models. We aim to evaluate the performance of DNN-based models when they are exposed to misspellings similar to the ones made by non-native Arabic learners.

This work focuses on black-box adversarial attacks. We assume that the attacker cannot access the structure, parameters or gradient of the target model. This is a realistic setting, because most modern machine learning classifiers are deployed as a service to receive users' inputs and provide corresponding outputs. Therefore, we design a method to generate black-box character-level adversarial modifications on the input tokens directly. We propose a two-step approach to craft character-level adversarial examples in the black-box setting:

- Step 1: Determine the most important tokens in the input sequence by considering their effect on the targeted classifier prediction.

TABLE I
Groups of Arabic letters with high visual similarity which may cause non-native Arabic learners to make spelling mistakes. The IPA notation is provided within parentheses.

Group	Characters				
G#1	بـ (b)	تـ (t)	ثـ (th)	نـ (n)	يـ (y)
G#2	جـ (g)	حـ (ħ)	خـ (x)		
G#3	د (d)	ذ (dh)			
G#4	ر (r)	ز (z)			
G#5	سـ (s)	شـ (sh)			
G#6	صـ (s)	ضـ (dh)			
G#7	طـ (t)	ظـ (ð)			
G#8	عـ (a)	غـ (gh)			
G#9	فـ (f)	قـ (q)			
G#10	كـ (k)	ل (l)			
G#11	ه (h)	ة (t)			
G#12	و (w)	ؤ (u)			
G#13	ي (y)	ى (aa)	ئـ (a)		

- Step 2: Modify them slightly, by substituting letters with similar ones.

A. Scoring Function

First, a scoring function is used to determine which tokens are important for the model's prediction. The Replace-1 scoring function $R1S(\cdot)$ of Gao et al. [6] scores the importance of tokens in an input sequence according to the observed classification from a targeted classifier.

By assuming the input sequence $x = x_1x_2...x_n$, where x_i is the token at the i^{th} position, we need to measure the effect of the x_i token on the output of the targeted model (F). The scoring function $R1S(\cdot)$ measures the effect of x_i on the model by replacing x_i with x_i'. More formally:

$$R1S(x_i) = F(x_1, x_2, ..., x_{i-1}, x_i, ..., x_n) - F(x_1, x_2, ..., x_{i-1}, x_i', ..., x_n) \quad (1)$$

where x_i' is chosen to be out of vocabulary (OOV) and it is obtained by inserting, deleting, or substituting a letter in x_i for a random letter. By calculating the effect of replacing x_i with OOV, the importance of all tokens in the input sample can be measured and ranked.

Following the original definition of adversarial examples in computer vision, where imperceptibly small changes are made to the image such that the output classification changes [1], we are attacking only the most important token in a sample (the token with the largest $R1S$ value). More formally, we attack the token that holds:

$$\arg\max_{1 \le i \le n} R1S(x_i) \quad (2)$$

such that the adversarial sequence and its original sequence appear similar when read by human observers.

B. Token Transformer

Given the highest ranked tokens from a scoring function, the second part of creating our adversarial examples is to modify or perturb the tokens. We propose an efficient method to modify a given token, and we do this by deliberately creating misspelled words. The motivation

Algorithm 1: The process of substitution selection.

input : Input sequence x, classifier $F(\cdot)$, important token w, Transforming function $T(\cdot)$.

output: *AdversarialToken*

$p\_tokens = T(w)$ // here we substitute the letters of w with similar letters, one substitution at a time, and store the new perturbed tokens in p_tokens

for t_k *in* p_tokens **do**

  $candidate(k)$ = replace w with t_k in x

  $score(k) = F(x)$ - $F(candidate(k))$

end

$AdversarialToken = \arg\max_{t_k} score(k)$

Return *AdversarialToken*

as mentioned before is to mutate the misspelled words mimicking what non-native Arabic learners are likely to do and evaluate DNN-based models with them.

To generate character-level adversarial examples, many operations can be used. However, we prefer small changes to the original tokens as we require the generated adversarial sentence to be visually and semantically similar to the original one for human understanding. Therefore, we propose two attacks:

- Ar-Flip: Substitutes an Arabic character in the token with a visually similar Arabic character (based on the groups in Table I).
- Ar-Flip2: Substitutes two Arabic characters in the token with two visually similar characters, based on the groups in Table I.

Based on the groups of visually similar characters (Table I), we substitute all the letters of the token with similar letters, one substitution at a time. For Ar-Flip attack, we choose the optimal substitution according to the change of the model's confidence value, i.e., choosing the substitution that decreases the confidence value of the ground truth class the most. For Ar-Flip2 attack, we choose the two substitutions that decrease the confidence value of the model the most. Then, we replace the token with the token that carries the optimal substitution to obtain perturbed text. Algorithm 1 summarizes the adversarial substitution selection process. Using two Arabic dictionaries: The Gulf Arabic Corpus (Gumar) [27] and the Universal Dependency of Arabic Treebank (NUDAR) [28], we exclude any real-word misspellings that were produced by the substitution operations.

Words are symbolic, and learned DNN-based classification models usually employ a dictionary to represent a finite set of possible words. The size of the typical NLP word dictionary is much smaller than the possible combinations of characters at a similar length (e.g., about 28n for the Arabic case, where n is the length of the word). This means if we deliberately misspell important words, we can easily convert those important words to "unknown" (i.e., words not in the dictionary). Unknown words are mapped to the "unknown" embedding vector in DNN models, which is likely to be vastly different than the embedding for the original word. Our results in Section V strongly indicate that such simple strategy can effectively force text classification models to behave incorrectly.

IV. Experiments

We implemented our character-level adversarial attacks using Python, Pandas, Numpy, and PyTorch libraries.

A. Corpus

We used the Book Reviews in Arabic Dataset (BRAD) [29]: A large-scale textual corpus designed for sentiment analysis task. BRAD is a balanced corpus that includes book reviews in Arabic collected from www.goodreads.com. The reviews in BRAD are mapped into four rating classes: 1 (the least) to 4 (the highest). This dataset includes 156,507 Arabic reviews. The average length of the reviews is 60 words, while the average length of tokens is 4 letters.

B. Targeted Classification Models

We evaluated the effectiveness of the proposed character-level attacks on the word-level Bi-LSTM model of Gao et al. [6]. This model is a Bi-directional LSTM which contains an LSTM in both directions, reading from the first word to last, and from the last word to first. We also experimented with the word-level CNN model of Kim [30]. We replicated Kim's CNN architecture, which contains three convolutional layers, a max-pooling layer, and a fully-connected layer.

C. Word Embeddings

We used the Word2vec model [31] to generate word vectors of 300 dimensions.

V. Results

In this section, we investigate the practical performance of the proposed method for generating adversarial texts. Following the practices of previous studies that have explored adversarial examples [3], [11], [12], [32], [33], and because the process of generating adversarial examples is time and resource-consuming, we randomly sampled 1280 examples from the BRAD testing set to evaluate the efficiency of the proposed attacks.

A. Effectiveness of Character-level Adversarial Examples

We ran our attacks on the testing samples with results summarized in Table II. It is clear that when modifying a character in the most important token of input sequence, our method successfully generates samples that are able to evade DNN-based classifiers. The proposed method could successfully flip the classifiers' output and reduce the

TABLE II
EFFECTIVENESS OF THE CHARACTER-LEVEL ADVERSARIAL ATTACKS, AR-FLIP AND AR-FLIP2, ON BI-LSTM AND CNN MODELS. THE NUMBERS REPRESENT THE CLASSIFICATION ACCURACY OF THE MODEL UNDER THE SPECIFIED ATTACKS. AVERAGE DECREASE IS THE AVERAGE PERCENT DECREASE OF THE MODELS ACCURACY.

Attack	**Bi-LSTM**	**CNN**
No-attack	80.00%	82.00%
Ar-Flip	54.53%	61.10%
Ar-Flip2	50.00%	56.40%
Average Decrease	27.73%	23.25%

TABLE III
EXAMPLE OF AR-FLIP ATTACK RESULT. ATTACKED TOKENS WERE HIGHLIGHTED IN RED.

Prediction	**Text**
3 (positive)	**Original Text:** كتاب ممتع وأسلوب من أبسط ما يمكن ومعاني روحانية جميلة.
	ktab mmta w:aslub mn absat ma yumkn w:maani ruħanyh jamilah.
	An interesting book, one of the simplest possible writing styles, and beautiful spiritual meanings.
1 (negative)	**Adversarial Text:** كتاب مميع وأسلوب من أبسط ما يمكن ومعاني روحانية جميلة.
	ktab mmya w:aslub mn absat ma yumkn w:maani ruħanyh jamilah.
	An OOV book, one of the simplest possible writing styles, and beautiful spiritual meanings.

models' classification accuracy. For the Bi-LSTM model, our method reduced the model's accuracy from 80.00% to around 54.50% under the Ar-Flip attack, and to 50.00% under the Ar-Flip2 attack. For the CNN model, the proposed method reduced the model's accuracy from 82.00% to around 61.00% and 56.40%, under the Ar-Flip and Ar-Flip2 attacks respectively. The results indicate that DNN-based classifiers are susceptible to spelling mistakes including the mistakes made by Arabic learners who flip a character with a similar one.

It is worth mentioning that we aim to measure the effect of changing the most important token per sample. As seen in Table II, the proposed attacks yielded success after changing only a single word per sample. These attacks can be even more aggressive by attacking more than one token per sample as discussed in Section V-B. Samples of outputs produced by the Ar-Flip attack and the Ar-Flip2 attack are shown Tables III and IV.

B. Attacking Several Tokens

To make our adversarial examples more aggressive, we attempted to attack several tokens in the input sample. As we did in Section V-A, we attacked only the most important word in each testing sample. However, here, we attack the five most important tokens per testing sample. Table V shows the results of applying the attacks on several important tokens per testing sample. Compared to attacking only one token per sample (Table II), we see that attacking several important tokens has an average of 5.74% greater reduction in accuracy of the Bi-LSTM model, and an average of 5.35% greater reduction in accuracy of CNN.

TABLE IV
EXAMPLE OF AR-FLIP2 ATTACK RESULT. ATTACKED TOKENS WERE HIGHLIGHTED IN RED.

Prediction	**Text**
1 (negative)	**Original Text:** بذلت مجهودا جبارا في محاولة قراءة الكتاب لكنني فشلت. القصة غريبة جداً لدرجة أنني توقفت ولم أستطع إكمال القراءة.
	bdhalt majhud jbar fi mħawlt qra'at alktab lknni fshlt. alqsah ghribh ɖ ɪda:n ldrjat anni twqft w:lm astta ekmal alqraah.
	I made a great effort trying to read the book, but I failed. The story is so strange that I stopped reading and couldn't continue.
4 (positive)	**Adversarial Text:** بذلت مجهودا جبارا في محاولة قراءة الكتاب لكنني فشلت. القصة غزينة جداً لدرجة أنني توقفت ولم أستطع إكمال القراءة.
	bdhalt majhud jbar fi mħawlt qra'at alktab lknni fshlt. alqsah ghzinh ɖ ɪda:n ldrjat anni twqft w:lm astta ekmal alqraah.
	I made a great effort trying to read the book, but I failed. The story is so OOV that I stopped reading and couldn't continue.

TABLE V
EFFECTIVENESS OF ATTACKING THE FIVE MOST IMPORTANT TOKENS PER TESTING SAMPLE USING AR-FLIP AND AR-FLIP2 ATTACKS. THE NUMBERS REPRESENT THE CLASSIFICATION ACCURACY OF THE MODEL UNDER THE SPECIFIED ATTACKS.

Attack	**Bi-LSTM**	**CNN**
No-attack	80.00%	82.00%
Ar-Flip	48.44%	56.80%
Ar-Flip2	44.61%	50.00%
Average Decrease	33.47%	28.60%

The attacks we proposed in this paper generate adversarial tokens whose only difference with their seed tokens is one or two character-level modifications. While the readability of our adversarial samples is subjective to the reader, we believe that they can be well understood by most readers, thus resulting in valid adversarial samples. Humans in general can decipher the originally intended word. Psychologists have shown that people can accurately read paragraphs containing words constructed with flipped and swapped letters with relatively small time-cost of around 11.00% slowdown in reading speed [34]. Current DNN-based NLP models have not achieved the human-level of processing and understanding of natural language inputs. Although some models use complicated feature extraction structures that can potentially catch the semantic expression of the words, they still fail to model the similarity among words.

C. News Categorization Task

In Section V-A, we evaluated the effectiveness of the discussed attacks on the sentiment analysis task. Here, we evaluated them on the news categorization task, using the Bidirectional Encoder Representations from Transformers (BERT) model [17] and the XLNet model [19].

TABLE VI
EFFECTIVENESS OF THE CHARACTER-LEVEL ADVERSARIAL ATTACKS ON BERT AND XLNET MODELS.

Attack	**BERT**	**XLNet**
No-attack	90.00%	92.11%
Ar-Flip	67.50%	75.00%
Ar-Flip2	65.00%	72.50%
Average Decrease	23.75%	18.36%

BERT is an unsupervised, bidirectional system for pre-training language representations which obtains state-of-the-art results on a variety of NLP tasks. BERT can be a general-purpose language-understanding model trained on a large textual corpus (such as Wikipedia), and can then be used for downstream NLP tasks such as news categorization. XLNet model is a generalized autoregressive language model that uses a permutation language modeling objective to combine the advantages of autoregressive and autoencoding based pre-training methods like BERT. According to NLP-progress[1], the XLNet model scored the state-of-the-art results in several NLP tasks, including news categorization task.

In the experiments of this section, BERT and XLNet models were trained on the Single-labeled Arabic News Articles Dataset (SANAD) [35]. SANAD is a news categorization dataset which contains Arabic news articles categorized into seven classes: Culture, Finance, Medical, Politics, Religion, Sports and Technology. We used a balanced SANAD dataset that contains 98,154 Arabic news articles. Each category includes 14,022 news articles. The average number of words per article is 275.

We randomly selected 1280 samples from the SANAD testing set to evaluate the effectiveness of the proposed attacks, and the results are shown in Table VI. Even for the powerful BERT and XLNet models, which have achieved great performance in various NLP tasks, adversarial attacks can still drop their classification accuracy. For the BERT model, the Ar-Flip attack lowered its classification accuracy by about 22.50%. The Ar-Flip2 attack also reduced BERT's accuracy by 25.00%. For the XLNet model, there was about 17.00% reduction in its classification accuracy when it was under the Ar-Flip attack, and 19.60% accuracy reduction when it was under the Ar-Flip2 attack.

We show that, just a few replace operations can reduce the accuracy of the BERT and XLNet classifiers. Our results align with prior research in that misspellings are harmful even to the most powerful models such as the BERT and XLNet models (Section II).

Another important result of our study is that regardless of the neural architecture of the models: Bi-LSTM, CNN, BERT, and XLNet, or the algorithm used for word embeddings: Word2vec, BERT, and XLNet, our method is able to evade DNN-based classifiers. This proves that word-embedding based models are vulnerable to easy attacks that introduce non-words.

[1]http://nlpprogress.com

VI. Statistical Analysis

Although the results show that the studied models are susceptible to the proposed attacks and the under-attack models appear to perform worse than their baseline counterparts, an important question to ask is whether the differences in relative performance are statistically significant, i.e. could they have occurred simply due to chance?

Many researchers recommend McNemar's test [36] for comparing the performance of two classifiers [37], [38] as it has a lower probability of Type I error. McNemar's is a non-parametric pairwise test designed for comparing two populations, or in this case, the predictions from two different classifiers on the same test dataset.

In this paper, we performed McNemar's test to compare the performance of the under-attack models (CNN, Bi-LSTM, BERT, and XLNet) with their baseline counterparts (studied in Sections V-A and V-C). Here, we compared the performance of the under-attack Bi-LSTM model with the baseline Bi-LSTM model, the under-attack CNN model with the baseline CNN model, the under-attack BERT model with the baseline BERT model, etc. We tested the null hypothesis, which states that there is no difference in the accuracy of each of the models studied, and we adjusted the significance threshold for each individual pairwise test to 0.05.

In all cases, the difference between the under-attack models and their baseline counterparts (the p-values) were significant (<0.05). Thus, we reject the null hypothesis which assumed there was no difference between the classifiers, in favor of the alternative. The results show that there was a statistically significant difference in the accuracy of all models, which indicates that the proposed attacks effectively compromised the performance of the studied models.

VII. Conclusion

In this paper, we proposed the first set of character-level adversarial attacks to evaluate text classifiers trained on Arabic corpora. The proposed attacking method is designed based on the most common spelling mistakes that non-native Arabic learners make. Our findings show that only a few substitutions are needed to mislead state-of-the-art DNN-based classifiers trained on Arabic corpora. Our experimental results indicate that flipping only one character per testing sample can result in a significant accuracy decrease from the original classification accuracy for neural models. In future work, we hope to evaluate more DNN-based models in an adversarial setting for a variety of NLP tasks, such as machine translation and question answering systems, with languages beyond English.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

[2] B. Alshemali, A. Graham, and J. Kalita, "Toward robust image classification," in *Proceedings of SAI Intelligent Systems Conference*. Springer, 2019, pp. 483–489.

[3] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," in *the Association for the Advancement of Artificial Intelligence*, 2020.

[4] B. Alshemali and J. Kalita, "Adversarial examples in Arabic," in *International Conference on Computational Science and Computational Intelligence*, 2019, pp. 371–376.

[5] ——, "Improving the reliability of deep neural networks in NLP: A review," *Knowledge-Based Systems*, vol. 191, no. 105210, pp. 1–19, 2020.

[6] J. Gao, J. Lanchantin, M. L. Soffa, and Y. Qi, "Black-box generation of adversarial text sequences to evade deep learning classifiers," in *IEEE Security and Privacy Workshops*, 2018, pp. 50–56.

[7] D. Pruthi, B. Dhingra, and Z. C. Lipton, "Combating adversarial misspellings with robust word recognition," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5582–5591.

[8] E. Jones, R. Jia, A. Raghunathan, and P. Liang, "Robust encodings: A framework for combating adversarial typos," in *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2752–2765.

[9] A. Kumar, P. Makhija, and A. Gupta, "Noisy text data: Achilles' heel of BERT," in *Proceedings of the Workshop on Noisy User-generated Text*. Association for Computational Linguistics, 2020, pp. 16–21.

[10] C. Aspillaga, A. Carvallo, and V. Araujo, "Stress test evaluation of Transformer-based models in natural language understanding tasks," in *Proceedings of the Language Resources and Evaluation Conference*. European Language Resources Association, 2020, pp. 1882–1894.

[11] I. Mondal, "BBAEG: Towards BERT-based biomedical adversarial example generation for text classification," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2021, pp. 5378–5384.

[12] X. He, L. Lyu, L. Sun, and Q. Xu, "Model extraction and adversarial transferability, your BERT is vulnerable!" in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*, 2021, pp. 2006–2012.

[13] G. Heigold, G. Neumann, and J. van Genabith, "How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse?" in *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas*, 2018, pp. 68–80.

[14] J. Ebrahimi, A. Rao, D. Lowd, and D. Dou, "Hotflip: White-box adversarial examples for NLP," in *The Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 31–36.

[15] A. Naik, A. Ravichander, N. Sadeh, C. Rose, and G. Neubig, "Stress test evaluation for natural language inference," in *Proceedings of the International Conference on Computational Linguistics*, 2018, pp. 2340–2353.

[16] B. Liang, H. Li, M. Su, P. Bian, X. Li, and W. Shi, "Deep text classification can be fooled," in *The International Joint Conference on Artificial Intelligence*, 2018.

[17] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformers for language understanding," in *The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 4171–4186.

[18] L. Sun, K. Hashimoto, W. Yin, A. Asai, J. Li, P. Yu, and C. Xiong, "Adv-BERT: BERT is not robust on misspellings! generating nature adversarial samples on BERT," in *arXiv preprint arXiv:2003.04985*, 2020.

[19] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "XLNet: Generalized autoregressive pretraining for language understanding," in *Advances in Neural Information Processing Systems*, 2019, pp. 5753–5763.

[20] K. Shaalan, R. Aref, and A. Fahmy, "An approach for analyzing and correcting spelling errors for non-native Arabic learners," in *The International Conference on Informatics and Systems*. IEEE, 2010, pp. 1–7.

[21] J. A. Jassem, "Tahalyl alakhta alkutabyt fi aleadad (almufrad walmthna waljame) [Analysis of written mistakes in number (singular, dual and plural)]," in *International Conference on Arabic Language*, 2013, pp. 67–117.

[22] M. N. Alyafii, "Alakhta altrkybyh lada mutaealimin allughah aleurbh [Syntax mistakes of learners of Arabic]," Master's thesis, Qatar University, 2016.

[23] A. Alshibayl and M. Banithhyab, "Alakhta altarkibiah ladaa mutaealimi allughah alearabiah lilnnatiqin bighayriha [Mistakes of Arabic learners in Arabic syntax]," *Journal of Linguistic and Literary Studies*, vol. 11, no. 1, pp. 101–122, 2019.

[24] R. Yassin, D. L. Share, and Y. Shalhoub-Awwad, "Learning to spell in Arabic: The impact of script-specific visual-orthographic features," *Frontiers in Psychology*, vol. 11, no. 7, pp. 205–215, 2020.

[25] F. Aloliemat, "Tahlil akhta mutaealimi allughat alearabiat min alnnatiqin bighiriha (ala mustawa aladad) [Analyzing Arabic learners' mistakes (at the number level)]," *Humanities and Social Sciences Journal*, vol. 45, no. 4, pp. 24–33, 2018.

[26] H. b. L. Alotaibi, "Eiktisab alaadad lada mutaealimi alearabiah lughat thanyh [Comprehending Arabic number features in Arabic learners]," *Educational and Human Sciences Journal*, vol. 43, no. 4, pp. 334 – 360, 2019.

[27] S. Khalifa, N. Zalmout, and N. Habash, "Morphological analysis and disambiguation for Gulf Arabic: The interplay between resources and methods," in *Proceedings of the Language Resources and Evaluation Conference*, 2020, pp. 3895–3904.

[28] D. Taji, N. Habash, and D. Zeman, "Universal Dependencies for Arabic," in *Proceedings of the Arabic Natural Language Processing Workshop*. Association for Computational Linguistics, 2017, pp. 166–176.

[29] A. Elnagar, L. Lulu, and O. Einea, "An annotated huge dataset for standard and colloquial Arabic reviews for subjective sentiment analysis," *Procedia Computer Science*, vol. 142, no. 9, pp. 182–189, 2018.

[30] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1746–1751.

[31] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *International Conference on Learning Representations*, 2013.

[32] B. Alshemali and J. Kalita, "Toward mitigating adversarial texts," *International Journal of Computer Applications*, vol. 178, no. 50, pp. 1–7, 2019.

[33] ——, "Generalization to mitigate synonym substitution attacks," in *Deep Learning Inside Out Workshop at the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1–7.

[34] K. Rayner, S. J. White, and S. Liversedge, "Raeding wrods with jubmled lettres: There is a cost," *Psychological Science*, vol. 17, no. 3, pp. 192 – 193, 2006.

[35] O. Einea, A. Elnagar, and R. Al Debsi, "SANAD: Single-label Arabic news articles dataset for automatic text categorization," *Data in Brief*, vol. 25, no. 3, pp. 20–36, 2019.

[36] Q. McNemar, "Note on the sampling error of the difference between correlated proportions or percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.

[37] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[38] J. Costa, C. Silva, M. Antunes, and B. Ribeiro, "Adaptive learning models evaluation in Twitter's timelines," in *International Joint Conference on Neural Networks*. IEEE, 2018, pp. 1–8.