Proceedings of the OSACT 2022 Workshop @LREC2022, pages 60–70
Marseille, 20 June 2022
© European Language Resources Association (ELRA), licensed under CC-BY-NC-4.0

Sa`7r: A Saudi Dialect Irony Dataset

Najla AlHazzani, Amaal AlDawod, Hala AlMazrou, Lama AlAwlaqi, Noura AlReshoudi, Hend Al-Khalifa, Luluh AlDhubayi
Information Technology Department, College of Computer and Information Sciences
King Saud University
Riyadh, Saudi Arabia
{442204257, 442203073, 442203082, 442202911, 442203043}@student.ksu.edu.sa,
{Hendk, laldubaie}@ksu.edu.sa

Abstract

In sentiment analysis, detecting irony is considered a major challenge. The key problem with detecting irony is the difficulty to recognize the implicit and indirect phrases which signifies the opposite meaning. In this paper, we present Sa`7r ساخرthe Saudi irony dataset, and describe our efforts in constructing it. The dataset was collected using Twitter API and it consists of 19,810 tweets, 8,089 of them are labeled as ironic tweets. We trained several models for irony detection task using machine learning models and deep learning models. The machine learning models include: K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Naïve Bayes (NB). While the deep learning models include BiLSTM and AraBERT. The detection results show that among the tested machine learning models, the SVM outperformed other classifiers with an accuracy of 0.68. On the other hand, the deep learning models achieved an accuracy of 0.66 in the BiLSTM model and 0.71 in the AraBERT model. Thus, the AraBERT model achieved the most accurate result in detecting irony phrases in Saudi Dialect.

Keywords: Irony detection, Twitter, Arabic tweets, Saudi Dialect, Arabic NLP, Transformer, Neural networks

1 Introduction

The content of social media networks such as Twitter shows unlimited daily feeds of millions of users' interactions (Rajadesingan et al., 2015). The massive amount of data attracts researchers to conduct several types of data and textual analysis for different purposes, such as detecting opinions, sentiment and irony. One of the fundamental NLP tasks is detecting ironic expressions which is considered one of the complex language phenomena and was widely studied in linguistics, philosophy, and psychology (Sigar, et.al;2012, Grice, et. al; 1975).

Although several studies were conducted on irony from different perspectives, the definition of irony has not reached a consensus yet (Filatova, 2012). One of the obstacles to defining irony is that irony has various vocabularies that undergo language changes (Nunberg,2001). In addition, the irony definition is affected by the variation of regional languages and dialects (Dress et al. 2008).

On the other hand, the literature shows a similar term to irony, which is sarcasm, and some studies used both terms interchangeably (Buschmeier et al., 2014, Duden, 2014). On the contrary, many studies tackled the delusion problem of sarcasm and irony, such as Kreuz and Glucksberg (1989). Kreuz and his colleague defined sarcasm as a prominent victim and the target of ridicule, whereas, in irony, there is no individual or victim.

In addition, Ironic language is usually less cruel, harmful, and aggressive than sarcasm. However, due to the high similarity between sarcasm and irony definitions and the complexity of distinguishing between the two phonenma, we considered, in this paper, both terms as synonym to define any expression in which a person uses words that deliver the opposite of literal meaning.

Detecting ironic expressions is important and fundamental, especially in sentiment analysis (Rosso et al., 2018). The automatic detection of irony can assist many essential domains, such as gaining business insights into public opinion to improve certain services. Moreover, detecting ironic expressions can help identify threats and distinguish between fake and real threatening messages (Al-Ghadhban et al., 2017).

Although social media companies provide analytic tools to analyze the vast amount of data available, these tools do not provide the best accuracy when applied to some text that contains irony and sarcasm or hidden meaning (Ghanem et al., 2019). Detecting this type of speech is considered difficult, especially in the Arabic language, because of its complexity and variations of the Arabic written styles. Additionally, Arabic language is also considered a challenging language in the field of NLP, due to its morphological richness, orthographic ambiguity and inconsistency, and dialectal variations (Darwish et al., 2021).

In this paper, we focus on collecting tweets for Saudi dialect to build an irony dataset extracted from Twitter. This work has two main contributions:

1. Creation of a public Saudi dialect dataset of 19,810 tweets with irony and non-irony labels.
2. Comparison of different neural network and machine learning models and reporting the best accurate model.

The rest of the paper is organized as follows: Section 2 gives an overview of related work in the area of irony and sarcasm detection for Arabic language. Section 3 presents the dataset generation stages, including data collection, dataset annotation, dataset statistics, and dataset evaluation. The experiment results and evaluation are described in Section 4. We then discuss the challenges faced through the experiment in Section 5. Finally, in section 6, we concluded the paper with suggestions for future works.

2 Related Work

Detecting Sarcasm and irony in the textual contents has been extensively studied in different languages, especially the English language. The increasing popularity of shared tasks for irony detection and sentiment analysis has increased the interest in this field and attracted more researchers to develop robust irony detection tools. The first shared task for irony detection in English tweets was proposed in 2018 (Van Hee, Lefever, and Hoste 2018, 20), the organizers proposed fine-grained multiclassification task on different types of irony instead of binary classification.

A more profound analysis of linguistic phenomena of the ironic expression has been proposed by (Karoui et al. 2017) that analyzes different linguistic categories of irony in different languages in the social media contents. This approach was established by implementing a multilingual corpus annotated based on a multi-layered schema to measure the impact of different pragmatic phenomena used in the expression of irony in three Indo-European languages, including English, French, and Italian.

The efficiency of neural networks has been investigated to detect sarcastic texts (Ghosh and Veale 2016) by implementing a model composed of Convolutional Neural networks (CNN), Long Short Term Memory (LSTM), and Deep Neural Network (DNN) to detect sarcasm over social media contents, the proposed model compared against SVM-based models and showed an improvement for the neural networks. Another work (Dutta and Mehta 2021) applying deep learning techniques to detect sarcasm in the Twitter news dataset, the proposed model was implemented based on the Convolutional-Recurrent Neural network (C-RNN) to discover sarcastic pattern detection and achieved an accuracy of 84.73%.

For the Arabic language, there have been few papers that tackled Sarcasm and irony detection in the Arabic language. Twitter is the most widely used source for data collection for detecting irony due to the huge amount of textual and the large availability of ironic texts among different languages and cultures.

One of the earliest studies was conducted by Al-Ghadhban et al., (2017) and Karoui et al., (2017) where they both used supervised learning algorithm to develop a classifier model. Al-Ghadhban et al., (2017) used Naïve Bayes Multinomial Text algorithm for detecting tweets and the model evaluation achieved 0.659 in recall, 0.71 in precision, and 0.676 in f-score. While Karoui et al., (2017) used Random Forest with GainRatio algorithm to detect irony in Arabic tweets and achieved an accuracy of 72.36%.

Similarly, Allaith et al. (2019) proposed a system based on several language models: word-n-grams, topic models, sentiment models, statistical models, and embeddings of words. In addition, Bi-LSTM, Random Forest, and XGBoost were some of the classifiers that were used to evaluate the system. Based on the F1-score, the proposed system achieved 0.85. Also another submission has achieved 81.7% and 79.4% for two different neural networks models for word embedding respectively.

Recently, there has been renewed interest in detecting sarcasm and irony with a dedication on constructing datasets for Arabic ironic language. In a shred task conducted by (Abu Farha et al., 2021), they released ArSarcasm-v2 dataset, which consists of 15,548 tweets labelled for sarcasm, sentiment and dialect. The shared task received 27 submissions for the sarcasm detection subtask. Among the techniques used in the shared task is the work by El Mahdaouy et al. (2021). They used a deep multitask learning model to develop a model that allows knowledge to be accessed for sarcasm detection. Their work incorporated BERT model and multitask attention interaction module into a single model architecture which produced a better performance in detecting sarcasm. Furthermore, Wadhawan (2021) proposed an approach which consists of two phases: the dataset preprocessing phase which involves inserting, deleting, and segmenting various fragments of the text. The second phase was experimenting with two transformer-based models AraELECTRA and AraBERT. Author found out that AraBERT has the highest weighted F1-score while AraELECTRA has the worst weighted F1-score and accuracy. In addition, Abuzayed and Al-Khalifa (2021) employed seven BERT-based models which are: MARBERT, ArBERT, QARiB, AraBERTv02, GigaBERT, Arabic BERT and mBERT, also to fix the problem of imbalanced data they combined the shared task dataset with additional information.

Ameur and Aliane (2021) created a sarcasm and sentiment detection dataset for Arabic tweets during the pandemic, the dataset is called "AraCOVID19-SSD". They collected 5,162 tweets that are annotated with two labels related to the two tasks: Sarcasm detection (Yes or No) and sentiment analysis (Positive, negative, or neutral). They used three pre-trained transformer models for classification (AraBERT, mBERTm and XLM-Roberta) and other supervised models (SVM, LR, and Random Forest). Their experiments showed that the SVM and AraBERT models performed better

than other models by reaching an F1-score of more than 95%. Another work proposed by Talafha et al. (2021), they collected Arabic tweets for sarcasm detection. The prediction task was tackled as a regression problem instead of a classification problem by quantifying the level of sarcasm for a given tweet instead of deciding if a tweet is sarcastic or not. The experiment was evaluated using Mean Squared Error (MSE) as a loss function and it obtained a 0.011 loss value.

Table 1 summarizes the available Arabic irony datasets. We can see that few dialectical datasets tackled irony and sarcasm detection specifically in Saudi dialect. Most of these datasets collected tweets using hashtags only. Therefore, this paper proposes a new Arabic dataset for Saudi irony tweets collected from hashtags, phrases and words annotated by humans.

Table 1: Summary of Arabic irony corpora

Datasets	**Dialect**	**Number of Tweets**	**Number of Ironic/ Sarcastic Tweets**
(Al-Ghadhban et al., 2017)	Saudi dialect	350	238
Soukhria (Karoui et al., 2017)	MSA, Egyptian, Syrian and Saudi dialect	5479	1733
IDAT (Ghanem et al., 2019)	MSA, Egypt, Gulf, Levantine and Maghrebi dialects.	22, 318	6, 809
DAICT (Abbes et al., 2020)	MSA, Egypt, Gulf, Levantine and Maghrebi dialects.	5358	4,809
ArSarcasm (Abu Farha and Magdy, 2020)	Egyptian, Gulf, LevantineMaghrebi and MSA	10,547	1682
ArSarcasm-v2 (Abu Farha et al., 2021)		15,548	2989
AraCOVID 19-SSD (Ameur and Aliane, 2021)	Multiple Arabic dialects (not specified) and MSA	5,162	1802
(Talafha et al., 2021)		1554	1165

3 Dataset Generation

3.1 Data Collection

The data collection was conducted using an open-source Python package called Twint[1]. Twint library enables scraping the raw data of interest from Twitter using a set of keywords. We aimed to collect Twitter data generated between 2011-03-16 and 2021-09-21 and the total collected tweets were 26,349 records. Hence, the date range specification was according to Twint library capability, which fixes the oldest date by default to 2011-03.

As for the keywords, we used 35 keywords that indicated irony in Saudi Dialect such as: كوميديا مسخرة, #سخرية, # سوداء, تعبير ساخر and hashtags like #تهكم, #دعابة, # and we searched for some words in phrases like: طبختيه يالرفلا to find tweets related to the ironic phrase: طبخ طبختيه يالرفلا اكليه. We also searched for the derivatives of the word, for example: تهكم، اتهكم. We found that Twint normalizes hamza and ta marbuta 'ة' or ha 'ه'. This means that there is no need to search for the same word in different orthographic forms.

The hashtags used along with their Buckwalter Arabic transliteration and translation and the keywords that inspired us to come up with other keywords are listed in Table 2.

Table 2: Hashtags and keywords used for data collection process

Hashtags		
Arabic text	**Transliteration**	**English translation**
#مزحة	mazha	Joke

[1] https://github.com/twintproject/twint

#دعابة	dueaba	Joke
#تهكم	tahakam	Irony
#استهزيء | #استهزاء	aistihzi' | aistihza'	Mockery
#مسخرة | #سخرية | #مصخرة	maskhara | sukhria | maskhara	Mockery
#اتهكم	aitahakum	Being ironic
#المضحك_المبكي	almudhik almabkiu	Laughing at the irony
#أمزح	'amzah	Joking
#أنكت	'ankat	Joking
#اتشمت	āttashamat	Gloated
#سخرية_القدر	sukhriat alqadr	Ironically
#كوميديا_سوداء	kumidia sawda'	Dark humor
Keywords		
Arabic text	**Transliteration**	**English translation**
لا ياشيخ | لايشيخ | لا يشيخ | لاياشيخ | لا يا شيخ	la yashykh | layshikh | la yashikh | layashykh | la ya shaykh	Oh Really!
اصفق لك	aisfaq lak	Should I clap for you?

اذا حجت البقرة على قرونها	adha hajat albaqarat ealaa quruniha	when a cow pilgrimage on its horns
قال تيس قال احلبه	qal tis qal ahlibh	I say this's a bull, he says milk it
طبخ طبختيه يالرفلا اكليه	tabkh tabkhatayh yalrafla akilih	hey bad cook, eat up what you cooked

اطقطق | طقطقة	aitqataq | taqtiqa	Mocking
الحمدلله والشكر	alhamudalaluh walshukr	Thank God
بس يابابا	bas yababa	Enough papa
بس ياشاطر | بس ياشاطرة	bas yashatir | bas yashatira	Stop it Smarty
خزياه بس	khizyah bas	Oh shameful
قل قسم	qul qasam	Swear to God
اتطنز	aitatanz	Making fun
جاب العيد | جابت العيد	jab aleid | jabat aleid	screwed up
هيا خدلك | هيا خذلك | يلا خذلك	hayaa khadalak | hayaa khadhalik | yala khadhalik	Oh here we go again
ياللهول	Yalllhwl	Oh my God
سوبهان الله	Subhan Allah	Subhan Allah (in mis-spilling)
Phrases		
Arabic text	**Transliteration**	**English translation**
اما حبى اما برك	ama habaa ama birak	Either crawling or sitting!

عنز لو طارت	aanz law tarat	Goat is a goat even if it flies

3.2 Dataset Description

As mentioned in the dataset collection section, the collected tweets file was about 3.7 MB in size. It is stored as a CSV file in which each row represents a tweet. Each tweet has five columns in which it is separated by a separator to ensure its correctness.

3.3 Dataset Annotation

As a first step, the tweets are classified based on two labels, "ironic" and "non-ironic". Nevertheless, we found that some tweets could not be clearly

classified as ironic or non-ironic, such as: " هه زيي سهرانة اطقطق وانام". Moreover, other tweets are written with different contexts, which are difficult for annotators to understand and interpret. To solve these problems, we decided to use the labeling criteria proposed by (Abbes et al., 2020), which classify the tweets into three labels: "ironic", "non-ironic" and "ambiguous". The ambiguous label helps annotators when they cannot decide with certainty whether a tweet is ironic or not.

The collected tweets were first cleaned by removing URLS, new lines "\n", punctuation, numbers, non-Arabic words, and duplicate tweets. Emojis were replaced with a decoded format using a python package[2]. We also performed the following normalization process using CAMeL tools [3], and PyArabic[4]:

- Unicode normalization, for example: ﷺ to صلى الله عليه وسلم.
- Normalize teh marbuta 'ة' to heh 'ه'.
- Normalize alef variants to 'ا'.
- Normalize double characters, for example: هههههه.
- Remove elongation '-'.
- Remove diacritics 'Tashkeel' (ً ، ٌ ، ٍ ، َ ، ُ ، ِ ، ْ).

After the preprocessing step we got 19,810 unique tweets. The annotation process was crowdsourced by dividing this task among different numbers of volunteers as needed.

However, we need to maintain a certain level of quality and reliability in the annotation process, therefore the annotators must be qualified for these conditions:

- Annotators must be familiar with the communication style of social media, especially Twitter.
- The age range of annotators is between 16 and 40 years old.
- The annotators must be Saudis so that they can understand the ambiguity behind the written words.
- The annotators should read the "annotators guideline".

The annotation has gone through two rounds as explained in Figure 1.

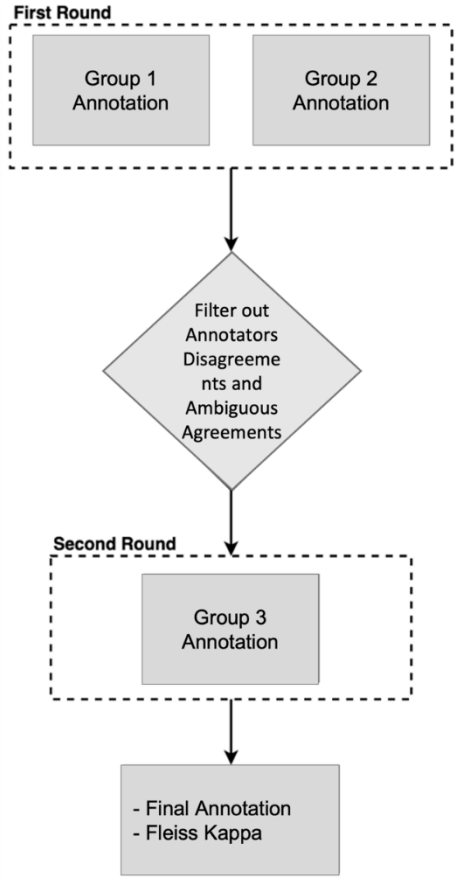


Figure 1: The annotation process

In the first round, we split the annotation process into two different groups to annotate 19,810 unique tweets. Each group consists of 7 annotators, with each annotator responsible for annotating 3,000 tweets, except for the seventh annotator who annotated the remaining 1,853 tweets. The number of annotated tweets with ironic tags was 7,425 and 8,573 for group one and two, respectively, while the number of non-ironic tweets is 11,026 for group one and 10671 for group two. This round of annotation took ten days and resulted in two annotations for each tweet. We then combined the annotations of the two groups and extracted the mismatched annotations; around 7753 tweets, and the tweets annotated as ambiguous; around 221 tweets. We offered the label "Ambiguous" to the annotators so that they could use it in case of uncertainty. After aggregation, we instructed five more annotators to perform the second round on a of total 7974 tweets to check for discrepancies, delete the "ambiguous" label, and clarify the new annotation considering emojis, punctuations, English words, and numbers for each tweet since they help to understand the tone of the tweet. Tables 3-6 show examples from the current dataset to proof that numbers, emojis,

[2] https://pypi.org/project/emojis/

[3] https://github.com/CAMeL-Lab/camel_tools

[4] https://pypi.org/project/PyArabic/

punctuations, and non-Arabic words are clarifying the tone of the tweet :

Table 3: Keep numbers in tweet example

Cleaned tweet with removing numbers:
تكفون يا عيال ارسلو له رابط
Tweet with keeping numbers:
٢٠٢١ تكفون يا عيال ارسلو له رابط
Translation of tweet with keeping numbers:
Please guys give him 2021 link
Transliteration of tweet with keeping numbers:
takufun ya eial arslu lah rabit 2021

Table 4: Keep emojis in tweet example

Cleaned tweet with removing emojis:
الحمد لله والشكر لله ع وجودنا في حياتك لولانا كانت حياتك
Tweet with keeping emojis:
الحمد لله والشكر لله ع وجودنا في حياتك لولانا كانت حياتك 🙈 😜 😂 😂
Translation of tweet with keeping emojis:
Thank God for our presence in your life, if it were not for us, your life would have been 🙈 😜 😂 😂
Transliteration of tweet with keeping emojis:
alhamd lilah walshukr lilah e wujuduna fi hayatik lawlana kanat hayatuk 🙈 😜 😂 😂

Table 5: Keep punctuations in tweet example

Cleaned tweet with removing punctuations :
الحب لا يشيخ الحب لا يا شيخ
Tweet with removing punctuations:
؟ الحب لا يشيخ = الحب … لا يا شيخ
Translation of tweet with keeping punctuations:
Love seriously = love … seriously?
Transliteration of tweet with keeping punctuations:
alhubu la yashikh = alhubu ... la ya shaykh ?

Table 6: Keep non-Arabic languages example

Cleaned tweet with removing non-Arabic languages:
من اقوال سيخلدها التاريخ لا ياشيخ
Tweet with keeping non-Arabic language:
من اقوال سيخلّدها التاريخ: " People who are dying who have never been died before. Donald trump لا_ياشيخ" 🐸 😂
Translation of tweet with keeping non-Arabic language:
Sayings that will be immortalized by history: "People who are dying who have never been died before. Donald trump " Seriously 🐸 😂
Transliteration of tweet with keeping non-Arabic language:
min aqwal sykhlldha altaarikhu: "People who are dying who have never been died before. Donald trump "la_yashikh 🐸 😂

The second round lasted for five days. We measured the inter-annotator agreement between the two annotators using Fleiss's Kappa which is a statistical measure of agreement between categorical values. It is commonly used to measure the inter-annotator reliability of the annotation of a dataset (Abbes et al., 2020). The Fleiss's Kappa inter-annotator agreement value was 0.54 which is a moderate level. The final annotated collection consists of 8,089 ironic tweets, and 11,715 non-ironic tweets.

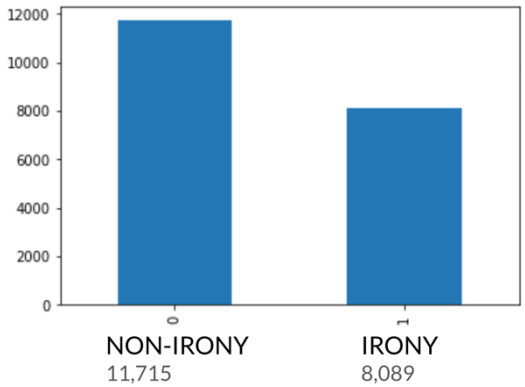


Figure 2 shows the distribution of the final corpus, we can see that the dataset has about 10% more on the non-irony class.

4 Experiments and Results

In this section we conducted different experiments to set a baseline system for the new dataset. We started with a set of machine learning algorithms, then a classifier built using word embeddings

vectorization technique with BiLSTM, lastly, we tested a BERT-based model. We have split the dataset into two parts, the first is the training set, which represents 80% of the data which equals 15843 entries, while the testing set represents the remaining 20% of the data which equals 3961 entries. For the evaluation, we used the F1-score to compare the results of all the models, since F1-score delivers a realistic score that does not get affected by the data imbalance (Ibrahim, Torki, and El-Makky 2018).

4.1 Machine Learning Models

There are many options for classification algorithms that can be used for binary classification of tweets into irony or non-irony. We implemented K-Nearest Neighbor (KNN), Logistic Regression (LR), Support Vector Machine (SVM), and Naïve Bayes (NB) with several variations (Bernoulli, Multinomial and Gaussian).

4.1.1 K-Nearest Neighbor (KNN)

For this algorithm, we set the k value as 10, as it is a reasonable value to avoid noise, as well as avoiding the reduction of boundaries between each neighbor and the other (Ikram and Chakir, 2019).

4.1.2 Logistic Regression (LR)

LR is another classification algorithm that can be employed to classify text, this algorithm measures the statistical significance of each independent variable in accordance with the probability (Shah et al., 2020), we set the inverse of regularization strength (c parameter) to 0.01, to increase the regularization.

4.1.3 Naïve Bayes (NB)

Naive Bayes is a classification method based on the Bayes theorem (Lewis, 1998). NB has different types of classifiers, including Multinomial, Gaussian, and Bernoulli. In this experiment, we validated all three NB variations to identify which one gives better accuracy. Multinomial gained the best accuracy of 0.66 compared to others. To optimize accuracy, tuning the hyperparameters will affect the performance of the model and it might improve it (Yang and Shami, 2020). Hence, we changed the value of the Bernoulli hyperparameter (binarize) to be 0.1 to optimize the accuracy and then its accuracy increased to 0.67.

4.1.4 Support Vector Machine (SVM)

In this experiment, we used the linear SVM algorithm with the linear kernel and regularization parameter equals to 2 to determine how much misclassification should be avoided in the SVM optimization.

4.2 Deep Learning Models

Our aim in this experiment is to use an algorithm that can deal with the peculiarities of text data, as in the experimentations of (Abu Farha et al., 2021), and (Allaith et al., 2019). Where the Bidirectional Long-short-term memory (BiLSTM) model has proven its ability in dealing with sequential data.

This model was implemented by utilizing a pretrained Arabic word embeddings "AraVec" which is trained using skip-gram algorithm, these word embeddings are then fed into deep learning model of BiLSTM, its hyperparameters are described in Table 7. This model resulted in 0.66 accuracy and F1 score of 0.59.

Table 7: AraVec BiLSTM model hyperparameters

Embedding layer	300
Bidirectional LSTM	128
Dropout	0.2
Activation	Sigmoid
Optimizer	SGD
Loss	Binary_crossentropy
Learning Rate	0.001
Epochs	5
Batch Size	100

4.3 Transformers Model

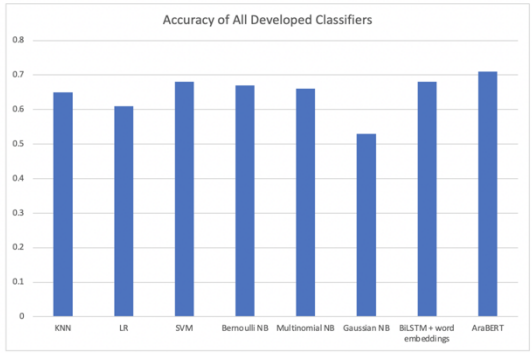
In this experiment, we used AraBERT which is a pretrained language model that was trained with large data from Twitter (Antoun, Baly, and Hajj 2021). We used AraBERTv0.2-Twitter-base, which was trained using 60 million multi-dialect words obtained from Twitter, which suits the problem of irony classification, since our dataset was obtained from twitter as well. The AraBERT model was fine-tuned using our dataset and the resulted accuracy was 71%.

4.4 Models' Results

All models used were configured manually, using random values to initialize the hyperparameters. Table 8 shows the performance all the developed classifiers.

Table 8: Comparison of evaluation results using A: accuracy, P: precision, R: recall, and F1: F1-score macro.

Model	A	P	R	F1
KNN	0.65	0.63	0.62	0.62
LR	0.61	0.66	0.53	0.44
SVM	0.68	0.67	0.67	0.67
Bernoulli NB	0.67	0.66	0.67	0.66
Multinomial NB	0.66	0.65	0.61	0.61
Gaussian NB	0.53	0.56	0.56	0.53
AraVec BiLSTM	0.69	0.59	0.58	0.59
AraBERT	0.71	0.70	0.70	0.70



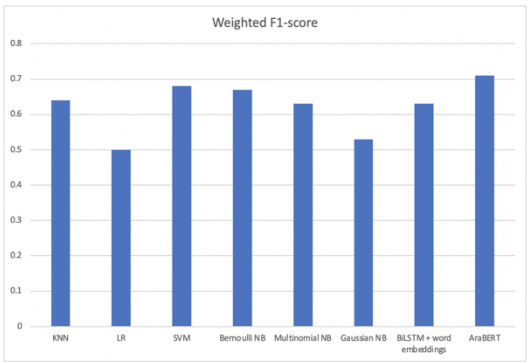


Figure 3: Comparison of the classifiers' accuracies and F1-Score

5 Discussion and Limitation

Figure 3 shows that the AraBERT-based model yielded the highest result of 71% F1-score macro, also it has the highest f1-score on the irony class detection with 65% as shown in Table 9, this indicates the power of transformers in handling text classification issues. The second-best model is the SVM model with F1-score of 68%. Followed by Bernoulli NB, KNN, Multinomial NB and BiLSTM, Gaussian NB and lastly logistic regression. We hypothesize that the fine-tuning of the hyperparameters would be in favor of improving the performance of the models, also increasing the number of annotated data for the training process could benefit the classifiers in general.

In terms of challenges, the collected tweets are based on Dialectal Arabic (DA) words that are common among Saudis; to obtain Arabic tweets from Saudi dialects it is important to mention that we totally relied on the words that are commonly used in the colloquial Saudi dialects, since lots of tweets were retrieved with no location tag. Pre-processing may affect the meaning, but its main benefit is to remove duplicate tweets and normalize the text. Even though the usage of some phrases would cause collecting similar tweets, the context of these phrases is still different, and a single emoji or punctuation or number or non-Arabic characters could change the whole meaning as shown in Table 3, Table 4, Table 5 and Table 6.

Also, it is unavoidable to collect tweets from another dialect or languages, but since these keywords are common in Saudi dialects, in addition to the fact that Saudis represent the largest number of users within the Arab region on Twitter ("Twitter: Most Users by Country" 2022), we have considered these collected tweets as Saudi tweets. However, the ironic words and phrases are huge, and this work is limited to only 35 keywords, more keywords may be included in future work.

In addition, the misspelled words such as (سوبهان الله) gave more ironic results than direct ironic words such as (تهكم). We noticed that misspelled words are intentionally used in the context of irony. The existence of English words and punctuation have high impact on understanding tweets, especially in dialects.

Moreover, the dataset is imbalanced where the number of non-ironic tweets is larger than ironic ones which requires further consideration during the model training and evaluation, therefore, we relied on F1-score for evaluation purposes and avoided accuracy.

Another issue is that ironic tweets depend on the topic; this issue should be considered when hiring annotators. Also, the annotator's personalities and mood is another issue, this could affect the annotation process. Yet, we mitigated this issue by making multiple annotation rounds.

Table 9 F1-score per class, for each model.

Model	**class**	**F1**

KNN	Non-irony	0.73
	Irony	0.51
LR	Non-irony	0.75
	Irony	0.13
SVM	Non-irony	0.74
	Irony	0.60
Bernoulli NB	Non-irony	0.71
	Irony	0.61
Multinomial NB	Non-irony	0.75
	Irony	0.46
Gaussian NB	Non-irony	0.52
	Irony	0.54
AraVec BiLSTM	Non-irony	0.79
	Irony	0.38
AraBERT	Non-irony	0.75
	Irony	0.65

6 Conclusion

In the era of social media, irony detection is considered a challenging and important task to understand a person's intentions. In this paper, we presented a new Arabic irony detection dataset for the Saudi dialects called Sa`7r[5]. We collected the corpus from Twitter using specific hashtags, keywords, and phrases related to irony based on the Saudi dialects. We plan to expand this dataset to include more linguistic content in the future. Additionally, we would like to determine whether there are any similarities and differences between ironic Arabic expressions used by speakers in other countries. In terms of modelling, we aim to solve the dataset imbalance in order to obtain more accurate results with a model trained with balanced dataset, we also aim to manipulate the hyperparameters of the BiLSTM model so that we can enhance the models learning abilities. For the transformer-based model, other options of pretrained Arabic BERT-based models do exist, and it is worth to experiment with such different models to find the best fit with the Saudi dialect dataset.

[5] https://github.com/iwan-rg/Saudi-Dialect-Irony-Dataset

7 References

Ibrahim Abu Farha, Wajdi Zaghouani, and Walid Magdy. 2021. Overview of the WANLP 2021 Shared Task on Sarcasm and Sentiment Detection in Arabic. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 296–305, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Abeer Abuzayed and Hend Al-Khalifa. 2021. Sarcasm and Sentiment Detection In Arabic Tweets Using BERT-based Models and Data Augmentation. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 312–317, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Hadj Ameur, Mohamed & Aliane, Hassina. (2021). AraCOVID19-SSD: Arabic COVID-19 Sentiment and Sarcasm Detection Dataset.

Kareem Darwish, Nizar Habash, Mourad Abbas, Hend Al-Khalifa, Huseein T. Al-Natsheh, Houda Bouamor, Karim Bouzoubaa, Violetta Cavalli-Sforza, Samhaa R. El-Beltagy, Wassim El-Hajj, Mustafa Jarrar, and Hamdy Mubarak. 2021. A panoramic survey of natural language processing in the Arab world. Communications of the ACM, 64(4):72–81, March.

Abdelkader El Mahdaouy, Abdellah El Mekki, Kabil Essefar, Nabil El Mamoun, Ismail Berrada, and Ahmed Khoumsi. 2021. Deep Multi-Task Model for Sarcasm Detection and Sentiment Analysis in Arabic Language. In Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 334–339, Kyiv, Ukraine (Virtual), April. Association for Computational Linguistics.

Talafha, B., Za'Ter, M. E., Suleiman, S., Al-Ayyoub, M., & Al-Kabi, M. N. (2021, November). sarcasm detection and quantification in arabic tweets. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 1121-1125). IEEE.

Wadhawan, A. (2021, April). AraBERT and Farasa Segmentation Based Approach For Sarcasm and Sentiment Detection in Arabic Tweets. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop* (pp. 395-400).

Ines Abbes, Wajdi Zaghouani, Omaima El-Hardlo, and Faten Ashour. 2020. DAICT: A dialectal Arabic irony corpus extracted from Twitter. In Proceedings of the 12th language resources and evaluation conference, pages 6265–6271, Marseille, France, May. European Language Resources Association. Citation Key: abbes-etal-2020-daict.

Ibrahim Abu Farha and Walid Magdy. 2020. From Arabic Sentiment Analysis to Sarcasm Detection: The ArSarcasm Dataset. In Proceedings of the 4th Workshop on Open-Source Arabic Corpora and

Processing Tools, with a Shared Task on Offensive Language Detection, pages 32–39, Marseille, France, May. European Language Resource Association.

Kanish Shah, Henil Patel, Devanshi Sanghvi, and Manan Shah. 2020. A Comparative Analysis of Logistic Regression, Random Forest and KNN Models for the Text Classification. Augmented Human Research, 5(1):12, March.

Li Yang and Abdallah Shami. 2020. On hyperparameter optimization of machine learning algorithms: Theory and practice. Neurocomputing, 415:295–316, November.

Ali Allaith, Muhammad Shahbaz, and Mohammed Alkoli. 2019. Neural network approach for irony detection from arabic text on social media. In Parth Mehta, Paolo Rosso, Prasenjit Majumder, and Mandar Mitra, editors, Working notes of FIRE 2019 - forum for information retrieval evaluation, kolkata, india, december 12-15, 2019, volume 2517, pages 445–450. CEUR-WS.org.

Bilal Ghanem, Jihen Karoui, Farah Benamara, Véronique Moriceau, and Paolo Rosso. 2019. IDAT at FIRE2019: Overview of the Track on Irony Detection in Arabic Tweets. In Proceedings of the 11th Forum for Information Retrieval Evaluation, pages 10–13, New York, NY, USA, December. Association for Computing Machinery.

AIT YAHIA Ikram and LOQMAN Chakir. 2019. Arabic Text Classification in the Legal Domain. In 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS), pages 1–6. October.

Paolo Rosso, Francisco Rangel, Irazu Hernández Farías, Leticia Cagnina, Wajdi Zaghouani, and Anis Charfi. 2018. A survey on author profiling, deception, and irony detection for the Arabic language. Language and Linguistics Compass, 12(4):e12275.

Dana Al-Ghadhban, Eman Alnkhilan, Lamma Tatwany, and Muna Alrazgan. 2017. Arabic sarcasm detection in Twitter. In 2017 International Conference on Engineering MIS (ICEMIS), pages 1–7. May.

Ilseyar Alimova, Elena Tutubalina, Julia Alferova, and Guzel Gafiyatullina. 2017. A Machine Learning Approach to Classification of Drug Reviews in Russian. In 2017 Ivannikov ISPRAS Open Conference (ISPRAS), pages 64–69. November.

Jihen Karoui, Farah Banamara Zitoune, and Véronique Moriceau. 2017. SOUKHRIA: Towards an Irony Detection System for Arabic in Social Media. Procedia Computer Science, 117:161–168, January.

Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach. In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining, pages 97–106, New York, NY, USA, February. Association for Computing Machinery.

Buschmeier, K., Cimiano, P., & Klinger, R. (2014, June). An impact analysis of features in a classification approach to irony detection in product reviews. In Proceedings of the 5th workshop on computational approaches to subjectivity, sentiment and social media analysis (pp. 42-49).

Duden. 2014. Duden Verlag. Online: http://www. duden.de/rechtschreibung/Ironie. ac- cessed April 28, 2014.

Sigar, A., Taha, Z., 2012. A contrastive study of ironic expressions in english and arabic. College of Basic Education Researchers Journal.

Aurangzeb Khan, Baharum Baharudin, Lam Hong Lee, and Khairullah Khan. 2010. A review of machine learning algorithms for text-documents classification. Journal of advances in information technology, 1(1):4–20.

David D. Lewis. 1998. Naive (Bayes) at forty: The independence assumption in information retrieval. In Claire Nédellec and Céline Rouveirol, editors, Machine Learning: ECML-98, pages 4–15, Berlin, Heidelberg. Springer.

Kreuz, R. J., & Glucksberg, S. (1989). How to be sarcastic: The echoic reminder theory of verbal irony. Journal of experimental psychology: General, 118(4), 374.

Grice, H. P., Cole, P., & Morgan, J. L. (1975). Syntax and semantics. Logic and conversation 3,41–58.

Torki, Marwan & Ibrahim, Mai & El-Makky, Nagwa. (2018). Imbalanced Toxic Comments Classification Using Data Augmentation and Deep Learning. 10.1109/ICMLA.2018.00141.

Dutta, Shawni, and Akash Mehta. 2021. “Unfolding Sarcasm in Twitter Using C-RNN Approach.” *Bulletin of Computer Science and Electrical Engineering* 2 (1): 1–8. https://doi.org/10.25008/bcsee.v2i1.1134.

Ghosh, Aniruddha, and Tony Veale. 2016. “Fracking Sarcasm Using Neural Network.” In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, 161–69. San Diego, California: Association for Computational Linguistics. https://doi.org/10.18653/v1/W16-0425.

Karoui, Jihen, Farah Benamara, Veronique Moriceau, Viviana Patti, Cristina Bosco, and Nathalie Aussenac-Gilles. 2017. “Exploring the Impact of Pragmatic Phenomena on Irony Detection in Tweets: A Multilingual Corpus Study.” In *15th Conference of the European Chapter of the Association for Computational Linguistics*, 1:262–72. Valencia, Spain. https://hal.archives-ouvertes.fr/hal-01686475.

“Twitter: Most Users by Country.” 2022. Statista. 2022. https://www.statista.com/statistics/242606/number-of-active-twitter-users-in-selected-countries/.

Van Hee, Cynthia, Els Lefever, and Véronique Hoste. 2018. “SemEval-2018 Task 3: Irony Detection in English Tweets.” In

Proceedings of The 12th International Workshop on Semantic Evaluation, 39–50. New Orleans, Louisiana: Association for Computational Linguistics. https://doi.org/10.18653/v1/S18-1005.