SAIDS: A Novel Approach for Sentiment Analysis Informed of Dialect and Sarcasm

Abdelrahman Kaseb and **Mona Farouk**
Computer Engineering, Cairo University
Giza, Egypt
{abdelrahman.kaseb,mona_farouk}@eng.cu.edu.eg

Abstract

Sentiment analysis becomes an essential part of every social network, as it enables decision-makers to know more about users' opinions in almost all life aspects. Despite its importance, there are multiple issues it encounters like the sentiment of the sarcastic text which is one of the main challenges of sentiment analysis. This paper tackles this challenge by introducing a novel system (SAIDS) that predicts the sentiment, sarcasm and dialect of Arabic tweets. SAIDS uses its prediction of sarcasm and dialect as known information to predict the sentiment. It uses MARBERT as a language model to generate sentence embedding, then passes it to the sarcasm and dialect models, and then the outputs of the three models are concatenated and passed to the sentiment analysis model. Multiple system design setups were experimented with and reported. SAIDS was applied to the ArSarcasm-v2 dataset where it outperforms the state-of-the-art model for the sentiment analysis task. By training all tasks together, SAIDS achieves results of 75.98 FPN, 59.09 F1-score and 71.13 F1-score for sentiment analysis, sarcasm detection, and dialect identification respectively. The system design can be used to enhance the performance of any task which is dependent on other tasks.

1 Introduction

Sentiment analysis (SA) is one of the main tasks in the natural language processing (NLP) field. It is used for opinion mining which supports decision-makers. Working on sentiment analysis starts relatively early, for example, Pang et al. (2002) analysed the sentiment to positive and negative in movie reviews. Following this paper, sentiment analysis becomes one of the most important topics in NLP, especially with the increasing number of reviews on websites and social media platforms. Since then, a lot of work has been done in English sentiment analysis, while Arabic has relatively much less. Since Abbasi et al. (2008) started their work on Arabic SA, multiple researchers also began theirs. Now there are well-known Arabic SA models like (Alayba et al., 2018; Abdulla et al., 2013; Abu Farha and Magdy, 2021; Elshakankery and Farouk, 2019). Of course, working with Arabic has many challenges, one of the most challenging issues is the complex morphology of the Arabic language (Kaseb and Farouk, 2016; Abdul-Mageed, 2019). Another challenge is the variety of Arabic dialects (Abdul-Mageed, 2019). Moreover, one of the well-known challenges in SA for all languages is sarcasm, as the sarcastic person uses words and means the opposite of it. For example, "I'd really truly love going out in this weather!", does it reflect a positive or negative sentiment? because of the sarcasm, we cannot judge the sentiment correctly.

Several related works tackle English sarcasm detection with sentiment analysis (Oprea and Magdy, 2020; Abercrombie and Hovy, 2016; Barbieri et al., 2014). On the other hand, there are only a few works on both sentiment and sarcasm in Arabic. There are two shared tasks on sarcasm detection (Ghanem et al., 2019), but for both sarcasm and sentiment there was only one shared task Abu Farha et al. (2021) but each sub-task is independent, meaning that participating teams can submit a different model for each task. Some participants used the same model for both sentiment and sarcasm (El Mahdaouy et al., 2021).

Instead of training sentiment independently of sarcasm, this work introduces a new model architecture that works with multi-task training which trains both at the same time. There are other additions to the proposed architecture; firstly, it trains with dialect also. Secondly, the sarcasm and dialect that are initially predicted are used in the prediction of the sentiment. In other words, the sentiment model is informed by the sarcasm and dialect model output. The contributions offered by this work are:

- Design a novel model architecture that can be

Proceedings of the The Seventh Arabic Natural Language Processing Workshop (WANLP), pages 22 - 30
December 8, 2022

used for a complicated task that is dependent on another task, e.g. sentiment analysis which is dependent on sarcasm detection.

- Investigate the design setups for the new architecture and find the best setup that could be used.
- Train the model on ArSarcam-v2 dataset and achieve the state-of-the-art results recorded as 75.98 FPN on sentiment analysis.

This paper is organized as follows Section 2 shows the related work on sentiment analysis, sarcasm detection, and dialect identification. Section 3 describes the dataset used in this work and shows data statistics. Section 4 describes SAIDS model and all the design setups. Section 5 shows the experimental results and finally section 6 concludes the work.

2 Related Work

SAIDS works on three tasks sentiment analysis, sarcasm detection, and dialect identification. In this section, the existing methods for each task are discussed.

2.1 Sentiment Analysis

Arabic sentiment analysis started with Abbasi et al. (2008) work. Since then, it is developed by multiple researchers. In the beginning, the main focus was on modern standard Arabic (MSA), but over time the researchers start to focus on dialectal Arabic (Mourad and Darwish, 2013; Kaseb and Farouk, 2021).

Regarding the datasets, based on Alyafeai et al. (2021), there are more than fifty datasets for sentiment analysis, including Elshakankery et al. (2021); Kaseb and Farouk (2019); Kiritchenko et al. (2016); Rosenthal et al. (2017); Elmadany et al. (2018) datasets. Because of the massive number of datasets, there are a massive number of system approaches for Arabic sentiments (Abu Farha and Magdy, 2019; Alayba et al., 2018; El-Beltagy et al., 2017). Based on Abu Farha and Magdy (2021) comparative study, using the word embedding with deep learning models outperform, the classical machine learning models and the transformer-based models outperform both of them. There is a reasonable number of Arabic transformer-based models like AraBERT (Antoun et al., 2020) and MARBERT (Abdul-Mageed et al., 2021) which are used by most Arabic sentiment analysis papers.

2.2 Sarcasm Detection

Unlike Arabic sentiment analysis, Arabic sarcasm detection has not gotten much attention yet. Only a few research works tackle the problem and still there is an obvious shortage of the Arabic sarcasm datasets, like Karoui et al. (2017); Abu Farha et al. (2022). Abbes et al. (2020) collected a dataset for sarcastic tweets, they used hashtags to collect the dataset for example #sarcasm. Then, they built multiple classical machine learning models SVM, Naive Bayes, and Logistic Regression, the best F1-score was 0.73.

After that, Ghanem et al. (2019) organized a shared task in a workshop on Arabic sarcasm detection. They built the dataset by collecting tweets on different topics and using hashtags to set the class. An additional step was added, by sampling some of the datasets and manually annotating them. In this shared task, eighteen teams were working on sarcasm detection. Khalifa and Hussein (2019) was the first team and achieved a 0.85 F1-score.

Then Abu Farha et al. (2021) made two tasks based on the ArSarcasm-v2 dataset; sentiment analysis and sarcasm detection. They have 27 teams participating in the workshop, the top teams achieved 62.25 F1-score and 74.80 FPN for sarcasm detection and sentiment analysis respectively.

2.3 Dialect Identification

Arabic dialect identification is an NLP task to identify the dialect of a written text. It can be on three levels, the first level is to identify MSA, classical Arabic (CA), and dialectical Arabic (McWhorter, 2004). The second level is to identify the dialect based on five main Arabic dialects EGY, LEV, NOR, Gulf, and MSA (El-Haj, 2020; Khalifa et al., 2016; Sadat et al., 2014; Al-Sabbagh and Girju, 2012; Egan, 2010). The third level is to identify the country-level dialect (Abdul-Mageed et al., 2020).

Regarding the datasets, there are datasets more than twenty Arabic datasets labeled with dialect. One of the most popular datasets is MADAR (Bouamor et al., 2018) where the data is labeled at the city-level for 25 Arab cities. Abdul-Mageed et al. (2020) built a shared task to detect the dialect, they published three different shared tasks. In the 2020 task, sixty teams participated, and the best results were 26.78 and 6.39 F1-score in the country-level and the city-level dialects respectively.

3 Dataset

ArSarcasm-v2 (Abu Farha et al., 2021) is the main dataset used in this work, it was released on WANLP 2021 shared task for two tasks sarcasm and sentiment analysis. It has about 15k tweets and is divided into 12k for training and 3k for testing, the same test set, as released on WANLP 2021, was used. Each tweet was labelled for the sentiment (positive (POS), neutral (NEU), and negative (NEG)), sarcasm (true, and false), and dialect (MSA, Egypt (EGY), Levantine (LEV), Maghreb (NOR), and Gulf). The authors of the dataset annotate it using a crowd-sourcing platform. This dataset originally consisted of a combination of two datasets, the first one is ArSarcasm (Abu Farha and Magdy, 2020) and the second one is DAICT (Abbes et al., 2020), Abu Farha et al. (2021) merged the two datasets.

3.1 Dataset Statistics

In this subsection, we introduce some dataset statistics that motivated us to work on SAIDS. The ArSarcasm-v2 dataset has 15,548 tweets, 3000 tweets are kept for testing and the rest of the tweets for training. Table 1 shows the number of examples for all task labels on the training set, as we can see, most of the data is labeled as MSA and non-sarcastic in dialect and sarcasm respectively.

Task	Label	Count
Sentiment	Positive	2,180
	Neutral	5,747
	Negative	4,621
Sarcasm	Sarcastic	2,168
	Non-sarcastic	10,380
Dialect	MSA	8,562
	EGY	2,675
	Gulf	644
	LEV	624
	NOR	43
Total		12,548

Table 1: Number of labels of sentiment, sarcasm and dialect on the training set

The relationship between sentiment labels and both sarcasm and dialect independently can be shown from Table 2. For the sentiment/sarcasm part, we can see that about 90 percent of sarcastic tweets are sentimentally labeled as negative, and about 50 percent of non-sarcastic tweets are sentimentally labeled as neutral. On the other hand, for the sentiment/dialect part, we can see that about 50 percent of MSA tweets are sentimentally labeled as neutral and about 50 percent of EGY tweets are sentimentally labeled as negative. From this table, we can conclude that the information we can get on sarcasm and dialect will benefit the sentiment analysis task.

	POS	NEU	NEG
Non-sarcastic	2,122	5,576	2,682
Sarcastic	58	171	1,939
MSA	1,405	4,486	2,671
EGY	506	793	1,376
Gulf	121	259	264
LEV	142	197	285
NOR	6	12	25

Table 2: Cross tabulation between sentiment labels and both sarcasm and dialect labels on the training set

Table 3 shows the percentage of sarcastic tweets on each dialect. As the number of NOR tweets is limited, its percentage is not reliable, so we can see that Egyptians' tweets are the most sarcastic. This supports the facts from table 2 that most EGY tweets are negative and most of the sarcastic tweets are negative tweets.

Dialect	Sarcasm percentage
MSA	10.83 %
EGY	34.77 %
Gulf	24.38 %
LEV	22.12 %
NOR	34.88 %

Table 3: Percentage of sarcastic tweets for each dialect on the training set

4 Proposed System

This section presents a detailed description of the proposed system. SAIDS learns sentiment analysis, sarcasm detection, and dialect identification at the same time (multi-task training), in addition, it uses the sarcasm detection and dialect outputs as an additional input to the sentiment analysis model which is called "informed decision". SAIDS decides the sentiment class using the information of sarcasm and dialect class which are both outputs itself. The main idea behind SAIDS is based on analyzing the dataset statistics, as shown in section

3, which says that most sarcastic tweets are classified as negative tweets and most MSA tweets are classified as neutral tweets.

4.1 System Architecture

Figure 1 shows the SAIDS architecture. The architecture consists of four main modules, the first module is MARBERTv2 (Abdul-Mageed et al., 2021), it is a transformer-based model, its input is the tweet, and its output is a sentence embedding which is a vector of length 768. The second module is the "Sarcasm Model", it is a binary classifier for sarcasm, its input is the sentence embedding, and its output is two values one for sarcastic tweets and another for non-sarcastic tweets. The third module is the "Dialect Model", which is identical to the "Sarcasm Model" except that it outputs five classes (EGY, LEV, NOR, Gulf, and MSA). The fourth module is the "Sentiment Model", it is a classifier for sentiment, its input is the concatenation of the sentence embedding, sarcasm model outputs and dialect model outputs.

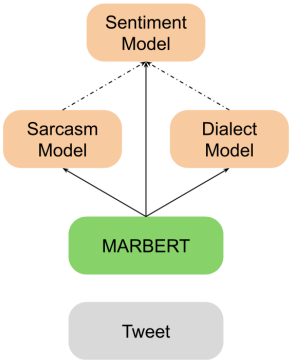

Figure 1: SAIDS architecture

The loss function used is Cross-Entropy for sentiment and dialect. Of course, since sarcasm is binary, we used binary Cross-Entropy for it.

4.2 Training Setups

This subsection describes the multiple setups that were used to arrive at the best model performance. The experiments carried out utilized multiple setups regarding the architecture and the training strategies.

Modules Architecture Multiple architectures were tested for the "Sentiment Model", "Sarcasm Model" and "Dialect Model". As a proof of concept for the idea, we first built a simple random forest model in each task model (random forest version). For the real scenario, we used multi-layer neural network (MNN) models. The first and the simplest is one output layer model and zero hidden layers. The second is one or two hidden layers, then the output layer. The third is one or two hidden layers the output of the module is the output of the hidden layer, which means that "Sentiment Model" inputs is not the output layer of the "Sarcasm Model" but the last hidden layer of it. The fourth setup is to concatenate the last hidden layer with the output layer and then pass it to "Sentiment Model".

What Should Be Informed The SAIDS architecture Figure 1 shows that the "Sentiment Model" inputs are "Sarcasm Model" and "Dialect Model" outputs but we experimented with multiple settings in this part; sentiment analysis informed of sarcasm only, dialect only, and both sarcasm and dialect.

Limited Backpropagation We limited the backpropagation over the dotted lines in Figure 1. It is used to ensure that the "Sarcasm Model" and the "Dialect Model" learn their main target correctly. When the model predicts sentiment incorrectly, its loss propagates directly to the MARBERTv2 model via the solid line and does not propagate via the dotted lines. Also, we evaluate SAIDS without limiting backpropagation which means the loss propagates everywhere, and with partial limiting. The partial limiting can be only set when the "Sarcasm Model" has hidden layers. We then limit the backpropagation through the sarcasm model's output layer but propagate it through the hidden layers.

Activation Function The experiments were carried out with Softmax as the activation function for the output of all modules. However, for the sake of comparison, we run the training without Softmax for the modules outputs, which means that the values are not from one to zero.

Task By Task Training As we train all the three tasks together with the same model, we experimented to train the first layer models, "Sarcasm Model" and "Dialect Model", for some epochs first, then train the full system together for multiple epochs. The motivation behind this idea is that as long as the first layer models work correctly, the sentiment analysis will correspondingly work correctly. We train in multiple orders like alternating between first layer models and full system and so on.

Other Training Parameters In our experiments, we built SAIDS and used the MARBERTv2 model provided by HuggingFace's transformers library (Wolf et al., 2020). Most of the experiments trained

for five epochs except for a low learning rate where it was twenty epochs. For the learning rate, we used a range from $1e^{-4}$ to $1e^{-6}$. The sequence was truncated to a maximum length of 128 tokens. Adam (Kingma and Ba, 2015) was used as an optimizer for all models.

5 Results

In this section, the results achieved with SAIDS are discussed. For the sake of comparison, baselines were built for the system. To initially evaluate the idea itself, a random forest model baseline was built and compared with the random forest version of SAIDS. Baselines for real scenario are baseline one (B1) which is identical to BERTModelForSequenceClassification class in HuggingFace's (Wolf et al., 2020), which takes the MARBERTv2 sentence embedding and passes it to the output layer for classification, and baseline two (B2) which uses two hidden layers before the classification layer, the hidden layer size is equal to the "Sentiment Model" hidden layer size, and baseline three (B3) which uses a larger hidden layer size to match the total number of trained parameters of SAIDS model.

For evaluation, we used the original metrics described for the dataset (Abu Farha et al., 2021). For sentiment analysis, the metric is the average of the F1-score for the negative and positive classes (FPN). For sarcasm detection, the metric is F1-score for the sarcastic class only (FSar). For dialect identification, we used the weighted average of the F1-score for all dialects (WFS).

5.1 Results of Different Training Setups

This subsection presents the results of the training setups and describes the best setup that was chosen for the proposed model. For each part of this subsection, every other setup was not changed to make the comparison fair.

Modules Architecture As a proof of concept for our system, the random forest (RF) model baseline was compared with the informed random forest (IRF) which is the random forest version of SAIDS. Table 4 shows that IRF outperforms RF where the FPN is improved by 3 percent which is due to the proposed architecture. The information gained from the new inputs, "outputs of sarcasm model" and "outputs of dialect model", was 5 and 4 percent respectively. This means that about 10 percent of the sentiment analysis decision came from the newly added information.

Model	FPN
Random Forest	59.36
Informed Random Forest	62.34

Table 4: Performance comparison for the proof of concept on the validation set

For the MNN architecture of the modules, multiple numbers of hidden layers were trained. At each experiment, all the modules have the same number of hidden layers. Table 5 shows that using zero hidden layers gives the best results. So no hidden layer setup was used in SAIDS.

Model	FPN
0 Hidden Layer	75.23
1 Hidden Layer	74.90
2 Hidden Layer	74.89

Table 5: Performance comparison for the number of hidden layers in modules on the validation set

What Should Be Informed Experiments were also done to find the best features to use while analysing sentiment. Table 6 shows that using both dialect and sarcasm is better than using only one of them and of course better than not using any of them which is the baseline. With a quick observation, it was found out that the dialect benefits the sentiment more than the sarcasm, this can be obvious when speaking about MSA tweets because most of them are labeled as neutral on sentiment. Accordingly, sarcasm and dialect information was used in SAIDS.

Model	FPN
Not Informed (B1)	72.40
Informed of sarcasm	73.67
Informed of dialect	74.41
Informed of sarcasm and dialect	75.23

Table 6: Performance comparison for what should be informed on the validation set

Limited Backpropagation Experiments were also done to find the best path for backpropagation to work with. "Full limit" is when the loss does not propagate through the "Sarcasm model" and "Dialect Model", "Partial limit" is when it propagates through some layers, and "Unlimited" is when it propagates through all layers. The model was composed of two hidden layers while running these experiments. Table 7 shows that "Partial limit" gets

better results than the others, but on SAIDS we did not use it as we used a no hidden layer setup, so we used the "Full limit" backpropagation.

Model	FPN
Full limit	74.23
Partial limit	74.89
Unlimited	72.31

Table 7: Performance comparison for limiting backpropagation on the validation set

Activation Function For the sake of comparison, the Softmax layer was removed from the output layer of the model in the experiments. Table 8 compares both setups, it shows that, as expected, using Softmax is better than not using it, as it quantify the probability of being sarcasm or being a certain dialect. So in SAIDS, Softmax was used on each module.

Model	FPN
With Softmax	75.23
Without Softmax	72.15

Table 8: Performance comparison for the activation function setting on the validation set

Task By Task Training Experiments were also done with training the three tasks together at the same time (All tasks), and multiple sets of the training sequence. The first is one epoch of training for sarcasm and dialect, and the rest for the full system (Seq 1). The second is odd epochs for sarcasm and dialect and even epochs for the full system (Seq 2). The third is two epochs of training for sarcasm and dialect and the rest for sentiment only (Seq 3). Table 9 shows that Seq 1 performs better than the other sequences, so we used it for the final model training.

Model	FPN
All tasks	74.35
Seq 1	75.23
Seq 2	73.49
Seq 3	73.01

Table 9: Performance comparison for different model training sequences on the validation set

Summary of Used Setups SAIDS used information from sarcasm and dialect models, which are both one classification layer with no hidden layers, the sentiment loss does not propagate through sarcasm and dialect models, and the Softmax activation function was used on each model output. The used training sequence was one epoch of training for sarcasm and dialect, and the rest epochs for the full system.

5.2 Results comparison with literature

SAIDS was trained and compared to the baselines we built and also the state-of-the-art models. Table 10 shows that SAIDS outperforms the existing state-of-the-art models on the sentiment analysis task. SAIDS's main task is sentiment analysis, the sarcasm detection and dialect identification are considered secondary outputs. Although the FSar score for SAIDS is considerably high, it is ranked third in the state-of-the-art models. On the other hand, most works that achieve state-of-the-art results are using different models for each task but in the proposed architecture, one model is used for both. The model also outputs the dialect, it achieves 71.13 percent on the weighted F1-score metric, but the literature has not reported the dialect performance so it is not included in the table.

Model	FPN	FSar
Baseline 1	71.60	58.41
Baseline 2	72.53	58.61
Baseline 3	73.11	58.62
El Mahdaouy et al. (2021)	74.80	60.00
Song et al. (2021)	73.92	**61.27**
Abdel-Salam (2021)	73.21	56.62
Wadhawan (2021)	72.55	58.72
SAIDS	**75.98**	59.09

Table 10: Performance comparison for the state-of-the-art models and SAIDS on the test set

6 Conclusion

Sentiment analysis is an important system that is being used extensively in decision-making, though it has different drawbacks like dealing with sarcastic sentences. In this work, we propose SAIDS which is a novel model architecture to tackle this problem. SAIDS essentially improves the sentiment analysis results while being informed of sarcasm and dialect of the sentence. This was achieved by training on the ArSarcasm-v2 dataset which is labeled for sentiment, sarcasm, and dialect. SAIDS's main target is to predict the sentiment of a tweet. It is trained to predict dialect and sarcasm, and then make use of them to predict the sentiment of the

tweets. This means that while the model is predicting the sentiment, it is informed of its sarcasm and dialect prediction. SAIDS achieved state-of-the-art performance on the ArSarcasm-v2 dataset for predicting the sentiment; 75.98 percent average F1-score for negative and positive sentiment. For sarcasm detection, SAIDS achieved a 59.09 percent F1-score for the sarcastic class, whereas for dialect identification it achieved a 71.13 percent weighted F1-score for all the dialects. We believe that this model architecture could be used as a starting point to tackle every challenge in sentiment analysis. Not only sentiment analysis but also this is a general architecture that can be used in any context where the prediction of a task depends on other tasks. The idea behind the architecture is intuitive, train for both tasks and inform the model of the dependent task with the output of the independent task.

References

Ahmed Abbasi, Hsinchun Chen, and Arab Salem. 2008. Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums. *ACM Trans. Inf. Syst.*, 26(3).

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. European Language Resources Association.

Reem Abdel-Salam. 2021. WANLP 2021 shared-task: Towards irony and sentiment detection in Arabic tweets using multi-headed-LSTM-CNN-GRU and MaRBERT. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 306–311, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Muhammad Abdul-Mageed. 2019. Modeling arabic subjectivity and sentiment in lexical space. *Information Processing & Management*, 56(2):291–307. Advance Arabic Natural Language Processing (ANLP) and its Applications.

Muhammad Abdul-Mageed, AbdelRahim Elmadany, and El Moatez Billah Nagoudi. 2021. ARBERT & MARBERT: Deep bidirectional transformers for Arabic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7088–7105, Online. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Chiyu Zhang, Houda Bouamor, and Nizar Habash. 2020. NADI 2020: The first nuanced Arabic dialect identification shared task. In *Proceedings of the Fifth Arabic Natural Language Processing Workshop*, pages 97–110, Barcelona, Spain (Online). Association for Computational Linguistics.

Nawaf A. Abdulla, Nizar A. Ahmed, Mohammed A. Shehab, and Mahmoud Al-Ayyoub. 2013. Arabic sentiment analysis: Lexicon-based and corpus-based. In *2013 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, pages 1–6.

Gavin Abercrombie and Dirk Hovy. 2016. Putting sarcasm detection into context: The effects of class imbalance and manual labelling on supervised machine classification of Twitter conversations. In *Proceedings of the ACL 2016 Student Research Workshop*, pages 107–113, Berlin, Germany. Association for Computational Linguistics.

Ibrahim Abu Farha and Walid Magdy. 2019. Mazajak: An online Arabic sentiment analyser. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 192–198, Florence, Italy. Association for Computational Linguistics.

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. European Language Resource Association.

Ibrahim Abu Farha and Walid Magdy. 2021. A comparative study of effective approaches for arabic sentiment analysis. *Information Processing & Management*, 58(2):102438.

Ibrahim Abu Farha, Silviu Vlad Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 task 6: iSarcasmEval, intended sarcasm detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, pages 802–814, Seattle, United States. Association for Computational Linguistics.

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). Association for Computational Linguistics.

Rania Al-Sabbagh and Roxana Girju. 2012. YADAC: Yet another dialectal Arabic corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2882–2889, Istanbul, Turkey. European Language Resources Association (ELRA).

Abdulaziz M. Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. 2018. A combined cnn and lstm model for arabic sentiment analysis. In *Machine Learning and Knowledge Extraction*, pages 179–191, Cham. Springer International Publishing.

Zaid Alyafeai, Maraim Masoud, Mustafa Ghaleb, and Maged Saeed AlShaibani. 2021. Masader: Metadata sourcing for arabic text and speech data resources. *CoRR*, abs/2110.06744.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. AraBERT: Transformer-based model for Arabic language understanding. In *Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, pages 9–15, Marseille, France. European Language Resource Association.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014. Modelling sarcasm in Twitter, a novel approach. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 50–58, Baltimore, Maryland. Association for Computational Linguistics.

Houda Bouamor, Nizar Habash, Mohammad Salameh, Wajdi Zaghouani, Owen Rambow, Dana Abdulrahim, Ossama Obeid, Salam Khalifa, Fadhl Eryani, Alexander Erdmann, and Kemal Oflazer. 2018. The MADAR Arabic dialect corpus and lexicon. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Kathleen Egan. 2010. Cross lingual Arabic blog alerting (COLABA). In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas: Government MT User Program*, Denver, Colorado, USA. Association for Machine Translation in the Americas.

Samhaa R. El-Beltagy, Mona El Kalamawy, and Abu Bakr Soliman. 2017. NileTMRG at SemEval-2017 task 4: Arabic sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 790–795, Vancouver, Canada. Association for Computational Linguistics.

Mahmoud El-Haj. 2020. Habibi - a multi dialect multi national Arabic song lyrics corpus. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1318–1326, Marseille, France. European Language Resources Association.

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). Association for Computational Linguistics.

AbdelRahim A. Elmadany, Hamdy Mubarak, and Walid Magdy. 2018. An arabic speech-act and sentiment corpus of tweets. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA). The 3rd Workshop on Open-Source Arabic Corpora and Processing Tools, OSACT3 ; Conference date: 08-05-2018.

Kariman Elshakankery and Mona Farouk. 2019. Hilatsa: A hybrid incremental learning approach for arabic tweets sentiment analysis. *Egyptian Informatics Journal*, 20(3):163–171.

Kariman Elshakankery, Magda Fayek, and Mona Farouk. 2021. Lastd: A manually annotated and tested large arabic sentiment tweets dataset. In *2021 the 5th International Conference on Information System and Data Mining*, ICISDM 2021, page 62–66, New York, NY, USA. Association for Computing Machinery.

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*, FIRE '19, page 10–13, New York, NY, USA. Association for Computing Machinery.

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. Arabic Computational Linguistics.

Gehad S. Kaseb and Mona Farouk. 2016. Arabic sentiment analysis approaches: An analytical survey. *International Journal of Scientific & Engineering Research*, 7(10).

Gehad S. Kaseb and Mona Farouk. 2019. Extended-atsd: Arabic tweets sentiment dataset. *Journal of Engineering and Applied Sciences*, 14.

Gehad S. Kaseb and Mona Farouk. 2021. An enhanced svm based approach for sentiment classification of arabic tweets of different dialects. *International Journal of Advances in Electronics and Computer Science*, 8.

Muhammad Khalifa and Noura Hussein. 2019. Ensemble learning for irony detection in arabic tweets. In *Working Notes of FIRE 2019 - Forum for Information Retrieval Evaluation, Kolkata, India, December 12-15, 2019*, volume 2517 of *CEUR Workshop Proceedings*, pages 433–438. CEUR-WS.org.

Salam Khalifa, Nizar Habash, Dana Abdulrahim, and Sara Hassan. 2016. A large scale corpus of Gulf Arabic. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4282–4289, Portorož, Slovenia. European Language Resources Association (ELRA).

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Svetlana Kiritchenko, Saif Mohammad, and Mohammad Salameh. 2016. SemEval-2016 task 7: Determining sentiment intensity of English and Arabic phrases. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 42–51, San Diego, California. Association for Computational Linguistics.

John H. McWhorter. 2004. Review of the syntax of spoken arabic: A comparative study of moroccan, egyptian, syrian, and kuwaiti dialects. In *Language (Volume 80)*, pages 338–339. Association for Computational Linguistics.

Ahmed Mourad and Kareem Darwish. 2013. Subjectivity and sentiment analysis of Modern Standard Arabic and Arabic microblogs. In *Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 55–64, Atlanta, Georgia. Association for Computational Linguistics.

Silviu Oprea and Walid Magdy. 2020. iSarcasm: A dataset of intended sarcasm. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1279–1289, Online. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 task 4: Sentiment analysis in Twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada. Association for Computational Linguistics.

Fatiha Sadat, Farnazeh Kazemi, and Atefeh Farzindar. 2014. Automatic identification of arabic dialects in social media. In *Proceedings of the First International Workshop on Social Media Retrieval and Analysis*, SoMeRA '14, page 35–40, New York, NY, USA. Association for Computing Machinery.

Bingyan Song, Chunguang Pan, Shengguang Wang, and Zhipeng Luo. 2021. DeepBlueAI at WANLP-EACL2021 task 2: A deep ensemble-based method for sarcasm and sentiment detection in Arabic. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 390–394, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Anshul Wadhawan. 2021. AraBERT and farasa segmentation based approach for sarcasm and sentiment detection in Arabic tweets. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 395–400, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.