AraBERT and Farasa Segmentation Based Approach For Sarcasm and Sentiment Detection in Arabic Tweets

Anshul Wadhawan
Flipkart Private Limited
anshul.wadhwan@flipkart.com

Abstract

This paper presents our strategy to tackle the EACL WANLP-2021 Shared Task 2: Sarcasm and Sentiment Detection. One of the subtasks aims at developing a system that identifies whether a given Arabic tweet is sarcastic in nature or not, while the other aims to identify the sentiment of the Arabic tweet. We approach the task in two steps. The first step involves pre processing the provided ArSarcasm-v2 dataset by performing insertions, deletions and segmentation operations on various parts of the text. The second step involves experimenting with multiple variants of two transformer based models, AraELECTRA and AraBERT. Our final approach was ranked seventh and fourth in the Sarcasm and Sentiment Detection subtasks respectively.

1 Introduction

During the last two decades, work on subjective language processing has been very common in literature. The work on sentiment analysis was a major theme that was pursued during this time. Sentiment Analysis is a process, according to (Liu, 2012), in which we extract out and examine the emotional stance in a particular piece of text. With the introduction of user-driven networks such as websites for social media, the work on Sentiment Analysis flourished. Most of this work was based on English, while not much attention was gathered by the Arabic language. The work on Arabic Sentiment Analysis was initiated by (Abdul-Mageed et al., 2011), but as compared to English it still needs development. This can be due to the many complexities pertaining to the language, including the broad range of dialects (Shaalan, 2010; Darwish, 2014) and the complicated language morphology (Abdul-Mageed et al., 2011). Some of the prominent problems while tackling the task of Sentiment Analysis are domain dependency, negation handling, lack of sarcasm and knowledge (Hussein, 2018). Sarcasm is described as a form of verbal irony designed to convey disdain or ridicule (Joshi et al., 2017). There has been a lot of work on the identification of English sarcasm, including datasets, like the works of (Barbieri et al., 2014b; Ghosh et al., 2015; Abercrombie and Hovy, 2016; Filatova, 2012; Joshi et al., 2016; Barbieri et al., 2014a) and detection systems like (Joshi et al., 2015; Rajadesingan et al., 2015; Amir et al., 2016). Work on Arabic sarcasm is, to our knowledge, restricted to the work of (Karoui et al., 2017), an irony detection task (Ghanem et al., 2019), and sarcasm datasets by (Abbes et al., 2020; Abu Farha and Magdy, 2020).

This paper puts forth the approach we applied to handle the WANLP-2021 Shared Task 2. The paper is ordered in the following manner: The problem statement, along with details of the ArSarcasm-v2 dataset are presented in Section 2. The methodology that we propose as our solution is described in Section 3. The experiments which were carried out, dataset statistics, system settings and results of the experiments are provided in Section 4. The paper ends with a brief section that talks about the conclusion and future directions of our research, in section 5.

2 Task Definition

The WANLP-2021 Shared Task 2 (Abu Farha et al., 2021) is based on a text classification problem, based on identifying sentiment and sarcasm in Arabic tweets. The provided training and test datasets have a total of 12,548 and 3,000 tweets respectively.

The shared task is divided into two subtasks:

Subtask 1 (Sarcasm Detection): The aim is to identify whether a tweet is sarcastic or not. Given a tweet, the task is to return TRUE if there is sarcasm present in the tweet and FALSE otherwise. This is

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 395–400
Kyiv, Ukraine (Virtual), April 19, 2021.

a binary classification problem. Precision/Recall/F-score/Accuracy are the evaluation metrics, where F-score of the sarcastic class is the official metric of evaluation.

Subtask 2 (Sentiment Analysis): The aim is to identify the sentiment of a tweet by assigning one of three labels (Positive, Negative, Neutral). Given a tweet, the task is to return POS if the tweet has a positive sentiment, NEG if the tweet has a negative sentiment or NEU if the tweet has a neutral sentiment. This is a multiclass classification problem. Precision/Recall/F-score/Accuracy are the evaluation metrics, where macro average of the F-score of the positive and negative classes (F-PN) is the official metric of evaluation.

3 Methodology

This section describes the process we employed to tackle the task. The process is divided into two steps: data preprocessing and transformer based models application. The first step involves processing the provided ArSarcasm-v2 dataset to convert it into a format better processed by the models. The second step involves experimenting with different models to decide which model performs the best for the ArSarcasm-v2 dataset. Details about these steps have been provided in the following sub-sections.

3.1 Data Pre-Processing

The data that is used for pre training the transformer based models is processed to create a better representation of the data. Thus, for the models to perform at their best ability, the data has to be processed in the same manner in the fine tuning process also. Raw data that is fetched from social media websites is diverse due to vast differences in expressions of opinions among users from different parts of the world. ArSarcasm-v2 dataset has these variations in different forms, evident from manual analysis. Social media users often make use of slang words, and non ascii characters such as emojis. URLs, user mentions and spelling errors are prominent in many posts on social media platforms. These attributes do not qualify as discerning features for classification tasks like Sarcasm and Sentiment Detection, and contribute to noise within the dataset. Therefore, we employ different pre processing techniques to clear this noise, so that the transformer based models only receive the relevant features. These techniques are as follows:

1. Remove HTML line breaks and markup, unwanted characters like emoticons, repeated characters (> 2) and extra spaces.

2. Perform Farasa segmentation (for select models only) (Abdelali et al., 2016).

3. Insert whitespace before and after all non Arabic digits or English Digits and Alphabet and the 2 brackets, and between words and numbers or numbers and words.

4. Replace all URLs with [رابط], emails with [بريد], mentions with [مستخدم].

3.2 Transformer Based Models

Deep learning methods have shown promising results in different machine learning domains such as Computer Vision (Krizhevsky et al., 2012) and Speech Recognition (Graves et al., 2013). The traditional machine learning methods have been over taken by deep learning techniques in the recent past, because of their superior performance owing to architectures inspired by the human brain. On the lines of Natural Language Processing, most deep learning techniques have been making use of word vector representations (Yih et al., 2011; Bengio et al., 2003; Mikolov et al., 2013) mainly as a way of representing the textual inputs. These techniques are further being replaced by transformer based techniques (Vaswani et al., 2017) due to significant improvements on most NLP tasks like text classification (Chang et al., 2020), which is the task at hand. Transformer based techniques have the ability to produce efficient embeddings as an output of the pre training process, which makes them proficient language models.

3.2.1 AraBERT

(Antoun et al.) Pre trained to handle Arabic text, AraBERT is a language model that is inspired from the Google's BERT architecture. Six variants of the same model are available for esxperimentation: AraBERTv0.2-base, AraBERTv1-base, AraBERTv0.1-base, AraBERTv2-large, AraBERTv0.2-large, and AraBERTv2-base. The architectural attributes of each of these models have been highlighted in Table 1.

3.2.2 AraELECTRA

(Antoun et al., 2020) With reduced computations for pre training the transformers, ELECTRA is a method aimed towards the task of self-supervised

Model	Size		Pre-Segmentation	Dataset		
	MB	Params		#Sentences	Size	#Words
AraBERTv0.1-base	543MB	136M	No	77M	23GB	2.7B
AraBERTv1-base	543MB	136M	Yes	77M	23GB	2.7B
AraBERTv0.2-base	543MB	136M	No	200M	77GB	8.6B
AraBERTv0.2-large	1.38G	371M	No	200M	77GB	8.6B
AraBERTv2-base	543MB	136M	Yes	200M	77GB	8.6B
AraBERTv2-large	1.38G	371M	Yes	200M	77GB	8.6B

Table 1: Architectural Attributes of Models

language representation learning. ELECTRA models are inspired from the two primary components of Generative Adversarial Networks: generator and discriminator. They aim at distinguishing between real input tokens and the fake ones. These models have shown convincing state-of-the art results on Arabic QA data.

For the pretraining process of all new AraBERT and AraELECTRA models, the same data is used, which has a size of 77GB. It has in total 8,655,948,860 words or 82,232,988,358 characters or 200,095,961 lines, before Farasa segmentation. For an initial pre training dataset, several websites were crawled, which are: Assafir news articles, OSCAR unshuffled and filtered, The OSIAN Corpus, Arabic Wikipedia dump from 2020/09/01, and The 1.5B words Arabic Corpus. For newer models, a fresh dataset was developed for the pre training process, which did not include the data crawled from the above websites, but had unshuffled and properly filtered OSCAR corpus in addition to the dataset used in AraBERTv1.

4 Experiments

We split the provided training dataset in a 90:10 ratio to create our training and validation splits. This is followed by experimenting with eight transformer based models, for which performance on the validation split is compared. The pre trained models are fine tuned on the training split and metrics for the validation split are calculated. The final test set predictions are made from the model that performs the best on the validation split, among the eight models. This section includes the dataset distribution, system settings, results and a brief analysis of the system, for both the subtasks: sarcasm and sentiment detection.

	Sarcasm		Total
	FALSE	TRUE	
Train	9356	1937	11293
Dev	1024	231	1255
Total	10380	2168	12548

Table 2: Data Distribution w.r.t Sarcasm

	Sentiment			Total
	NEG	NEU	POS	
Train	4139	5197	1957	11293
Dev	482	550	223	1255
Total	4621	5747	2180	12548

Table 3: Data Distribution w.r.t Sentiment

4.1 Dataset

Tables 2 and 3 show the class wise distribution of the 90:10 training-validation splits created from the provided ArSarcasm-v2 dataset, for the tasks of sarcasm detection which has class labels TRUE and FALSE and sentiment detection which has class labels NEG, NEU and POS, respectively.

4.2 System Settings

We make use of hugging-face[1] API to fetch the pre-trained AraBERT and AraELECTRA models. The API provides the six variants of AraBERT

[1] https://huggingface.co/transformers/

Parameter	Value
Epsilon (Adam optimizer)	1e-8
Learning Rate	1e-5
Batch Size (for base models)	40
Batch Size (for large models)	4
Maximum Sequence Length	256
#Epochs	10

Table 4: Parameter Values

Model	Subtask 1				Subtask 2			
	P	R	F1	A	P	R	F1	A
AraBERTv0.1-base	84.15	84.62	84.36	84.62	72.38	72.43	72.40	72.43
AraBERTv0.2-base	84.31	85.42	84.62	85.42	74.90	74.98	74.90	74.98
AraBERTv0.2-large	84.57	85.90	84.63	85.90	76.25	76.41	76.20	76.41
AraBERTv1-base	84.73	85.02	84.86	85.02	73.27	73.55	73.17	73.55
AraBERTv2-base	85.39	85.74	85.55	85.74	75.93	75.86	75.89	75.86
AraBERTv2-large	85.16	86.37	85.07	86.37	74.28	74.50	74.31	74.50
AraELECTRA-base-generator	82.72	83.19	82.94	83.19	72.45	72.03	71.99	72.03
AraELECTRA-base-discriminator	84.75	85.34	84.99	85.34	74.70	74.82	74.57	74.82

Table 5: Results on Validation Set

models by the names of bert-base-arabertv02, bert-base-arabert, bert-base-arabertv2, bert-base-arabertv01, bert-large-arabertv2, and bert-large-arabertv02, and the two variants of AraELECTRA models by the names of araelectra-base-discriminator and araelectra-base-generator. We fine tune these models on the training split with hyper parameter values specified in Table 4.

4.3 Results and Analysis

This section provides the detailed results obtained on the created validation and provided test sets.

4.3.1 Validation Set Results

The experimental results in terms of weighted Precision(P), weighted F1 scores(F1), weighted Recall(R) and Accuracy(A) on the created validation split have been depicted in Table 5, for both subtasks: Sarcasm Detection (Subtask 1) and Sentiment Detection (Subtask 2).

Our observations from Table 5 are as follows:

1. When comparing all the models, AraELECTRA generator model has the worst performance in terms of both F1 scores and Accuracy, for both the subtasks. This is possibly due to its forte of handling GAN related tasks rather than general classification tasks.

2. When comparing all AraBERT models, one of the large models seem to perform the best in terms of accuracy for both the subtasks. This is possibly due to superior architectures and heavier models.

3. For Subtask 1, AraBERTv2-base has the highest weighted F1-score and for Subtask 2, AraBERTv0.2-large has the highest weighted F1-score.

	C_E	A	P	R	M-F1
T_S^1	58.72	78.30	72.64	71.47	72.00
T_S^2	72.55	69.83	65.15	66.23	65.31

Table 6: Official Results on Test Set

4.3.2 Test Set Results

From the above observations, we select AraBERTv2-base for Subtask 1 and AraBERTv0.2-large for Subtask 2 as final models to formulate our officially submitted predictions. Table 6 presents the final test set results, with T_S^1 denoting Subtask 1, T_S^2 denoting Subtask 2, P denoting Precision, R denoting Recall, A denoting Accuracy, M-F1 denoting Macro-F1 score, and C_E denoting the criteria of evaluation for the two subtasks (F-score of the sarcastic class for T_S^1, and Macro averaged F score of positive and negative classes for T_S^2).

5 Conclusion and Future Work

In this paper, we present our strategy to approach the EACL WANLP-2021 Shared Task 2. We tackle the task in two steps. In the first step, the ArSarcasm-v2 dataset is pre-processed by altering different parts of text. This is followed by running experiments with multiple variants of two transformer based models pre-trained on Arabic text, AraELECTRA and AraBERT. The final submissions for the tasks of Sarcasm and Sentiment Detection are based on that variant of model which performs the best. Our approach fetches a private leaderboard rank 7 and 4 in the Sarcasm and Sentiment Detection tasks respectively. As future scope, we plan to explore other features which may be relevant for this task, and inculcate ensemble learning taking into consideration both word vector based and transformer based embeddings.

References

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.

Ahmed Abdelali, Kareem Darwish, Nadir Durrani, and Hamdy Mubarak. 2016. Farasa: A fast and furious segmenter for Arabic. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 11–16, San Diego, California. Association for Computational Linguistics.

Muhammad Abdul-Mageed, Mona Diab, and Mohammed Korayem. 2011. Subjectivity and sentiment analysis of Modern Standard Arabic. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 587–591, Portland, Oregon, USA. Association for Computational Linguistics.

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. 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, 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*.

Silvio Amir, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Silva. 2016. Modelling context with user embeddings for sarcasm detection in social media. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 167–177, Berlin, Germany. Association for Computational Linguistics.

Wissam Antoun, Fady Baly, and Hazem Hajj. Arabert: Transformer-based model for arabic language understanding. In *LREC 2020 Workshop Language Resources and Evaluation Conference 11–16 May 2020*, page 9.

Wissam Antoun, Fady Baly, and Hazem Hajj. 2020. Araelectra: Pre-training text discriminators for arabic language understanding.

Francesco Barbieri, Francesco Ronzano, and Horacio Saggion. 2014a. Italian irony detection in twitter: a first approach.

Francesco Barbieri, Horacio Saggion, and Francesco Ronzano. 2014b. 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.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *J. Mach. Learn. Res.*, 3(null):1137–1155.

Wei-Cheng Chang, Hsiang-Fu Yu, Kai Zhong, Yiming Yang, and Inderjit Dhillon. 2020. Taming pretrained transformers for extreme multi-label text classification.

Kareem Darwish. 2014. Arabic information retrieval. *Foundations and Trends® in Information Retrieval*, 7:239–342.

Elena Filatova. 2012. Irony and sarcasm: Corpus generation and analysis using crowdsourcing. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 392–398, Istanbul, Turkey. European Language Resources Association (ELRA).

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.

Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan. 2015. Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1003–1012, Lisbon, Portugal. Association for Computational Linguistics.

Alex Graves, Abdel rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks.

Doaa Mohey El-Din Mohamed Hussein. 2018. A survey on sentiment analysis challenges. *Journal of King Saud University - Engineering Sciences*, 30(4):330–338.

Aditya Joshi, Pushpak Bhattacharyya, and Mark J. Carman. 2017. Automatic sarcasm detection: A survey. *ACM Comput. Surv.*, 50(5).

Aditya Joshi, Vinita Sharma, and Pushpak Bhattacharyya. 2015. Harnessing context incongruity for sarcasm detection. In *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), pages 757–762, Beijing, China. Association for Computational Linguistics.

Aditya Joshi, Vaibhav Tripathi, Pushpak Bhattacharyya, and Mark J. Carman. 2016. Harnessing sequence labeling for sarcasm detection in dialogue from TV series 'Friends'. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 146–155, Berlin, Germany. Association for Computational Linguistics.

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.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.

Bing Liu. 2012. Sentiment analysis and opinion mining. volume 5.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

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*, WSDM '15, page 97–106, New York, NY, USA. Association for Computing Machinery.

Khaled Shaalan. 2010. Nizar y. habash, introduction to arabic natural language processing (synthesis lectures on human language technologies). *Machine Translation*, 24(3–4):285–289.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Wen-tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. 2011. Learning discriminative projections for text similarity measures. In *Proceedings of the Fifteenth Conference on Computational Natural Language Learning*, pages 247–256, Portland, Oregon, USA. Association for Computational Linguistics.