DeepBlueAI at WANLP-EACL2021 task 2: A Deep Ensemble-based Method for Sarcasm and Sentiment Detection in Arabic

Bingyan Song **Chunguang Pan** **Shengguang Wang** **Zhipeng Luo**
DeepBlue Technology (Shanghai) Co., Ltd
{songby,panchg,wangshg,luozhp}@deepblueai.com

Abstract

Sarcasm is one of the main challenges for sentiment analysis systems due to using implicit indirect phrasing for expressing opinions, especially in Arabic. This paper presents the system we submitted to the Sarcasm and Sentiment Detection task of WANLP-2021 that is capable of dealing with both two subtasks. We first perform fine-tuning on two kinds of pre-trained language models (PLMs) with different training strategies. Then an effective stacking mechanism is applied on top of the fine-tuned PLMs to obtain the final prediction. Experimental results on ArSarcasm-v2 dataset show the effectiveness of our method and we rank third and second for subtask 1 and 2.

1 Introduction

Social media is growing as a communication medium where people can express online their feelings and opinions on a variety of topics in ways they rarely do in person. Detecting sarcasm and sentiment in text have become a crucial element for decision-makers and business leaders as well as for common users to understand public opinion. The significant role of the Arab region in international politics and in the global economy have led to the investigation on the task of detecting sarcasm and sentiment in Arabic. The task involves detecting whether a piece of text expresses a **positive**, a **negative**, or a **neutral** sentiment; and whether it is **sarcasm** or **non-sarcasm**.

Sarcasm or Irony is a form of speech that, in the context of sentiment analysis, mostly takes place when the speaker expresses a positive opinion but actually aims to complain about the opinion target (Majumder et al., 2019). Sarcasm is particularly hard to detect in Arabic language due to the use of positive indicators to express negative emotions. Sentiment detection is also challenging because the same sentiment word may have different polarity according to the domain (Oraby et al., 2013).

There are two main approaches for the monolingual approach of Sarcasm and Sentiment Detection in Arabic: corpus-based and lexicon-based (Oueslati et al., 2020). Then combination of these two can be referred to as the hybrid approach. The corpus-based method, typically trains sentiment classifiers. Several supervised learning algorithms (e.g. SVM) (Duwairi and El-Orfali, 2014) have been used to classify the sentiment label into positive or negative. These algorithms require handcrafted features including part-of-speech (POS) tags and social media-driven features. Recently, deep learning techniques such as recurrent neural network (RNN) (Al-Sallab et al., 2017; Al-Smadi et al., 2019) and convolutional neural network (CNN) (Alayba et al., 2017) emancipate researchers from feature engineering. The lexicon-based method commonly determines the sentiment or polarity of opinion by evaluating the sentiment words in the document or the sentence which is used when the data is unlabelled. It often uses predefined dictionaries of annotated sentiment terms to label each word in the document by its sentiment (Al-Ayyoub et al., 2015). Then lexicon scores are used as input features to the classifier and thus plays an important role in the hybrid approach (Elshakankery and Ahmed, 2019).

However, more recent word embedding techniques, such as fastText, ELMo, BERT, are yet to be fully explored for Sarcasm and Sentiment Detection in Arabic despite having pretrained Arabic vesrions of them publicly available like EMLo-ForManyLangs (Zeman et al., 2018), AraBERT (Antoun et al., 2020). Therefore, in this paper, we introduce our system for Sarcasm and Sentiment detection in Arabic, which leverages multiple PLMs with several training strategies. There are two main steps for our system, **(i)** fine-tuning two kinds of

Proceedings of the Sixth Arabic Natural Language Processing Workshop, pages 390–394
Kyiv, Ukraine (Virtual), April 19, 2021.

Dialect	Non-Sarcastic	Sarcastic	Negative	Neutral	Positive	Total
Egyptian	1,745	930	1,376	793	506	2,675
Gulf	487	157	264	259	121	644
Levantine	486	138	285	197	142	624
Maghrebi	28	15	25	12	6	43
MSA	7,634	928	2,671	4,486	1,405	8,562
Total	10,380	2,168	4,621	5,747	2,180	12,548

Table 1: Dataset statistics for sarcasm and sentiment over the dialects

PLMs, including XLM-R (Conneau et al., 2020) and AraBERT (Antoun et al., 2020), with various hyperparameters and training strategies, obtaining diverse models; **(ii)** applying an effective stacking mechanism on top of these PLMs to predict the final complexity scores.

Our experiments, merging PLMs in total, indicate that our method successfully utilizes weaker PLMs as well as high-performing PLMs. As a result, our system ranks third and second for the Subtask 1 and 2 of Sarcasm and Sentiment Detection in Arabic, WANLP-2021 (Abu Farha et al., 2021).

2 Data

In this paper, we use the dataset called **ArSarcasm-v2** (Abu Farha et al., 2021). This is an Arabic sarcasm detection dataset based on several other Arabic sentiment analysis datasets including ArSarcasm (Abu Farha and Magdy, 2020), SemEval's 2017 (Rosenthal et al., 2017) and ASTD (Nabil et al., 2015). For the annotation process, the annotators were asked to provide three labels for each tweet including **Sarcasm** (sarcastic or non-sarcastic), **Sentiment** (positive, negative or neutral) and **Dialect** (Egyptian, Gulf, Levantine, Maghrebi or Modern Standard Arabic). Each tweet was annotated by at least three different annotators.

ArSarcasm-v2 contains 12,548 training samples with annotations and 3,000 testing samples without annotations. Table 1 shows statistics of training set, where we can find that 17.28% of the data is sarcastic (2,168 tweets). Most of the data is either in MSA or the Egyptian dialect, while there are few examples of the Maghrebi dialect. Figure 1 shows the sentiment distribution over the sarcastic tweets. It illustrates that most of the sarcastic tweets have negative sentiment, and this agrees with the definition, which implies that sarcasm includes making ridicule of someone or something.

Since there is no official validation set, we use 7-fold cross-validation on the training dataset for performance estimation of our model.

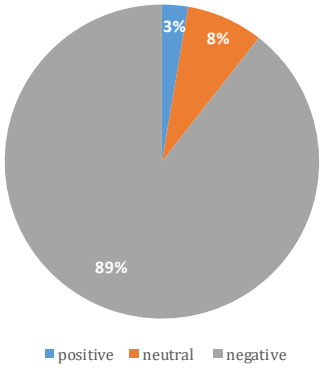

Figure 1: Sentiment distribution over sarcastic tweets.

3 System

3.1 Model

Architecture Figure 2 shows the architecture of our model that is capable of dealing with both subtask 1 and 2. We add dialect information before the tweet to construct the input segment. Since we utilize a BERT-like model as the encoder, we segment them with special tokens `[CLS]` and `[SEP]`. Then the embedding of `[CLS]`, which can stand for the whole input context, can be obtained. We feed it into a dense layer and get the final prediction through the Multi-Sample Dropout (Inoue, 2019). The output of dense layer x is depicted as below,

$$x = ReLU(W_0 dropout(x_{[CLS]})) \quad (1)$$

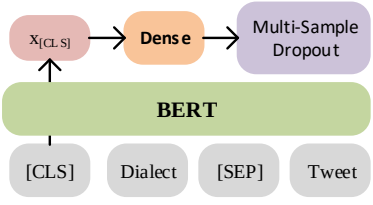

Figure 2: The overall architecture for sarcasm and sentiment detection.

where $W_0 \in R^{d \times k}$ is the learning weight, k is the dimension of $x_{[CLS]}$ and d is the hyperparameter which we set as 256.

Multi-Sample Dropout Dropout is a simple but efficient regularization technique for achieving better generalization of deep neural networks. During training, dropout randomly discards a portion of the neurons to avoid overfitting. The original dropout creates a randomly selected subset (called a dropout sample) from the input in each training iteration while the multi-sample dropout creates multiple dropout samples. The loss is calculated for each sample, and the sample losses are averaged to obtain the final loss.

Thus, the final prediction of subtask 1 and 2 can be calculated as following,

$$\begin{aligned} \hat{y}_{sub1} &= \frac{1}{N}\sum_{i=1}^{N} Sigmoid(W_i dropout_i(x)) \\ \hat{y}_{sub2} &= \frac{1}{N}\sum_{j=1}^{N} W_j dropout_j(x)) \end{aligned} \tag{2}$$

where $W_i \in R^{1 \times d}$ and $W_j \in R^{3 \times d}$ are the learning weights, N is the number of dropout values which we set as 5. By using this training mechanism, we can accelerate training and achieve lower error rates as well.

Loss Function Since subtask 1 is a binary classification task and subtask 2 is a multiclass classification task, we choose the Binary Cross Entropy Loss and Cross Entropy Loss for the two subtasks respectively.

3.2 Training strategies

In order to further improve the performance of our model, we adopt two training strategies and are introduced below.

Task-Adaptive Pre-training Task-adaptive pre-training (TAPT) can effectively improve model performance (Gururangan et al., 2020). The data used in general pre-training usually does not contain task-specific data. Thus we do task-adaptive pre-training by pre-training the masked language model task on the given Arabic dataset.

Knowledge Distillation Inspired by Hinton et al. (2015), we adopt the knowledge distillation mechanism into our system. The whole procedure consists of three steps. First, we train the original big model using a hard target, which is the true label (e.g. [0,1,0]). Then, we use the trained model to predict the soft target, which is the probability of each class. After this, we train a small model by minimizing the loss between the scores predicted by the small model and the soft target. We choose MSE or Smoothl as the loss function. Finally, we utilize the small model to predict the final results.

3.3 Stacking Trained Models

Model stacking is commonly used in competitions to improve mode accuracy. The main procedure of stacking trained models in our method including five steps. First, since the dataset we use is in Arabic, we choose XLM-R (Conneau et al., 2020) which is a new state-of-the-art multilingual masked language model and AraBERT (Antoun et al., 2020) which is a transformer-based model for Arabic as base models. Second, we do TAPT on these PLMs to achieve new PLM models. Third, we perform 7-fold cross-validation during the whole training process to avoid overfitting or selection bias. Fourth, we train multiple models with different hyperparameters (e.g. learning rate) and different training strategies to improve the model diversity. Ultimately, we train a simple Linear Regression (LR) model and a Support Vector Machine (SVM) model as the final estimator for subtask 1 and 2 respectively.

4 Results and Discussion

Evaluation Metrics As mentioned in the evaluation procedure of WANLP-2021 Sarcasm task, the F-score of the sarcastic class is the official metric for subtask 1 and F-PN (Marco average of the F-score of the positive and negative classes) is the official one for subtask 2.

PLMs with Training Strategies As shown in Table 2, for both subtask 1 and 2, we use two kinds of PLMs which are XLM-R and AraBERT. The results are the average scores of 7-fold cross-validation on the training dataset. XLM-R performs better on this task.

We evaluate the performance of adding different training strategies as well. By adding TAPT, F1 scores of both XLM-R and AraBERT in subtask 1 and 2 increase. It demonstrates that further pre-train the language model with task-specific data will improve the performance. We then add knowledge distillation to the models after TAPT and achieves the best F1 scores on the cross-validation

	Subtask 1		Subtask 2	
Tpye	**Model**	**F1-sarcastic**	**Model**	**F1-PN**
XLM-R	XLM-R$_{\text{LARGE}}$	0.5725	XLM-R$_{\text{LARGE}}$	0.7348
	XLM-R$_{\text{LARGE}}$+TAPT	0.6027	XLM-R$_{\text{LARGE}}$+TAPT	0.7460
	XLM-R$_{\text{LARGE}}$+TAPT+Distillation	**0.6687**	**XLM-R$_{\text{LARGE}}$+TAPT+Distillation**	**0.7919**
AraBERT	AraBERT$_{\text{LARGE}}$	0.5724	AraBERT$_{\text{LARGE}}$	0.7413
	AraBERT$_{\text{LARGE}}$+TAPT	0.6079	AraBERT$_{\text{LARGE}}$+TAPT	0.7451
	AraBERT$_{\text{LARGE}}$+TAPT+Distillation	0.6538	AraBERT$_{\text{LARGE}}$+TAPT+Distillation	0.7790
Ensemble	mean	0.5964	mean	0.7782
	LR	0.6627	-	-
	LR+ Threshold	**0.6899**	**SVM**	**0.8024**

Table 2: Comparison of F1 scores for stacking different models of subtask 1 and 2.

Subtask 1			**Subtask 2**		
Team	**F1-sarcastic**	**Macro-F1**	**Team**	**F-PN**	**Macro-F1**
BhamNLP	0.6225	0.7268	CS-UM6P	0.7480	0.6625
SPPU-AASM	0.6140	0.7096	**DeepBlueAI**	**0.7392**	**0.6570**
DeepBlueAI	**0.6127**	**0.7310**	rematchka	0.7321	0.6587
CS-UM6P	0.6000	0.7183	Phonemer	0.7255	0.6531
dalya	0.5989	0.7251	IDC	0.7190	0.6446

Table 3: Leaderboard

of training data. It indicates that the soft target predicted by the original big model contains useful information that needs to be learned.

Stacking trained models For subtask 1, we adopt a linear regression (LR) model as the final estimator to stack all the trained models with different training strategies. We train the weights of each model in LR on the training set and then use the learning weights to predict final scores of test set. We compare the result of LR model and the result of averaging all the trained models, the former one is much more better which proves the necessity of using LR estimator. Due to the imbalanced of positive and negative samples, we adjust the threshold as 0.41. Samples with predicted scores higher than 0.41 are sarcastic. For subtask 2, we train a SVM model as the final predictor to stack all the different trained models. The score of SVM is also better than the score obtained by averaging the trained models. The scores of all the ensemble methods are higher than the scores predicted by one model, this shows that stacking trained models is an effect way to further improve the system performance.

5 Official Submission

We submitted the scores predicted by the ensemble method introduced above. The official ranking have been released and the top five teams are presented in Table 3. We rank third in subtask 1 and second in subtask 2, which demonstrates the effectiveness of our system.

6 Conclusion

In this paper, we propose a top performing model for the task of Sarcasm and Sentiment Detection. We fine-tune two kinds of pre-trained language models including XLM-R and AraBERT with different training strategies contains task-adaptive pre-training and knowledge distillation. Then we stack them with a simple linear regression model in subtask 1 and a SVM model in subtask 2. Experimental results show the validity of this ensemble method and we rank third and second for subtask 1 and 2.

References

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*.

Mahmoud Al-Ayyoub, Safa Bani Essa, and Izzat Alsmadi. 2015. Lexicon-based sentiment analysis of arabic tweets. *International Journal of Social Network Mining*, 2(2):101–114.

Ahmad Al-Sallab, Ramy Baly, Hazem Hajj, Khaled Bashir Shaban, Wassim El-Hajj, and Gilbert Badaro. 2017. Aroma: A recursive deep learning model for opinion mining in arabic as a low resource language. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)*, 16(4):1–20.

Mohammad Al-Smadi, Bashar Talafha, Mahmoud Al-Ayyoub, and Yaser Jararweh. 2019. Using long short-term memory deep neural networks for aspect-based sentiment analysis of arabic reviews. *International Journal of Machine Learning and Cybernetics*, 10(8):2163–2175.

Abdulaziz M Alayba, Vasile Palade, Matthew England, and Rahat Iqbal. 2017. Arabic language sentiment analysis on health services. In *2017 1st international workshop on arabic script analysis and recognition (asar)*, pages 114–118. IEEE.

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.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451.

Rehab Duwairi and Mahmoud El-Orfali. 2014. A study of the effects of preprocessing strategies on sentiment analysis for arabic text. *Journal of Information Science*, 40(4):501–513.

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

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hiroshi Inoue. 2019. Multi-sample dropout for accelerated training and better generalization. *arXiv preprint arXiv:1905.09788*.

Navonil Majumder, Soujanya Poria, Haiyun Peng, Niyati Chhaya, Erik Cambria, and Alexander Gelbukh. 2019. Sentiment and sarcasm classification with multitask learning. *IEEE Intelligent Systems*, 34(3):38–43.

Mahmoud Nabil, Mohamed Aly, and Amir Atiya. 2015. Astd: Arabic sentiment tweets dataset. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 2515–2519.

Shereen Oraby, Yasser El-Sonbaty, and Mohamad Abou El-Nasr. 2013. Finding opinion strength using rule-based parsing for arabic sentiment analysis. In *Mexican International Conference on Artificial Intelligence*, pages 509–520. Springer.

Oumaima Oueslati, Erik Cambria, Moez Ben HajHmida, and Habib Ounelli. 2020. A review of sentiment analysis research in arabic language. *Future Generation Computer Systems*, 112:408–430.

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.

Daniel Zeman, Jan Hajic, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. Conll 2018 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies*, pages 1–21.