X-PuDu at SemEval-2022 Task 6: Multilingual Learning for English and Arabic Sarcasm Detection

Yaqian Han, Yekun Chai, Shuohuan Wang, Yu Sun
Baidu, Inc.
{hanyaqian,chaiyekun,wangshuohuan,sunyu02}@baidu.com

Hongyi Huang, Guanghao Chen, Yitong Xu, Yang Yang
Shanghai Pudong Development Bank
{huanghy6,chengh13,xuyt3,yangy103}@spdb.com.cn

Abstract

Detecting sarcasm and verbal irony from people's subjective statements is crucial to understanding their intended meanings and real sentiments and positions in social scenarios. This paper describes the X-PuDu system that participated in SemEval-2022 Task 6, *iSarcasmEval - Intended Sarcasm Detection in English and Arabic*, which aims at detecting intended sarcasm in various settings of natural language understanding. Our solution finetunes pre-trained language models, such as ERNIE-M and DeBERTa, under the multilingual settings to recognize the irony from Arabic and English texts. Our system ranked second out of 43, and ninth out of 32 in Task A: one-sentence detection in English and Arabic; fifth out of 22 in Task B: binary multi-label classification in English; first out of 16, and fifth out of 13 in Task C: sentence-pair detection in English and Arabic.

1 Introduction

Sarcasm is the use of language that typically signifies the opposite to mock or convey contempt. As a narrow research field in natural language processing (NLP), sarcasm detection is a particular case in the spectrum of sentiment analysis, with important implications for a slew of NLP tasks, such as sentiment analysis, opinion mining, author profiling, and harassment detection. In the textual data, these tonal and gestural clues like heaving tonal stress and rolling of the eyes are missing, making it more difficult for machines.

The sarcastic intention of human annotators has potentially hindered the training and evaluation process in detecting the genuine emotions and positions of the natural language. Thus, this task (Abu Farha et al., 2022) adopted a novel data collection method (Oprea and Magdy, 2020), where authors themselves label the training samples. For sarcastic texts, the authors also rephrase them into non-sarcastic ones. Then, linguistic experts further checked the scathing pieces and labeled them into sub-categories of sarcasm defined by (Leggitt and Gibbs, 2000): sarcasm, irony, satire, understatement, overstatement, and rhetorical question.

This SemEval task requires the identification of sarcasm in either one sentence or sentence pairs in various language settings, which consists of three subtasks:

(1) Task A (English and Arabic): Given a text, determine whether it is sarcastic or non-sarcastic;

(2) Task B (English only): A binary multi-label classification task. Given a text, determine which ironic speech category it belongs to, if any;

(3) Task C (English and Arabic): Given a sarcastic text and its non-sarcastic rephrase, i.e. two texts that convey the same meaning, determine which is the sarcastic one.

Our method employed various multilingual or mono-lingual pre-trained language models, such as ERNIE-M (Ouyang et al., 2020) and DeBERTa (He et al., 2021) to address each component of this task, with a bunch of fine-tuning and ensemble techniques. Our system finally achieved

- 2nd out of 43 and 9th out of 32 in English and Arabic subtasks in *Task A*;
- 5th out of 22 in *Task B*;
- 1st out of 16 and 5th out of 13 in English and Arabic subtasks in *Task C*.

2 Previous Work

After detecting sarcasm in the speech was firstly proposed in (Tepperman et al., 2006), sarcasm detection has attracted extensive attention in the NLP community. Afterward, sarcasm detection in the

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 999 - 1004

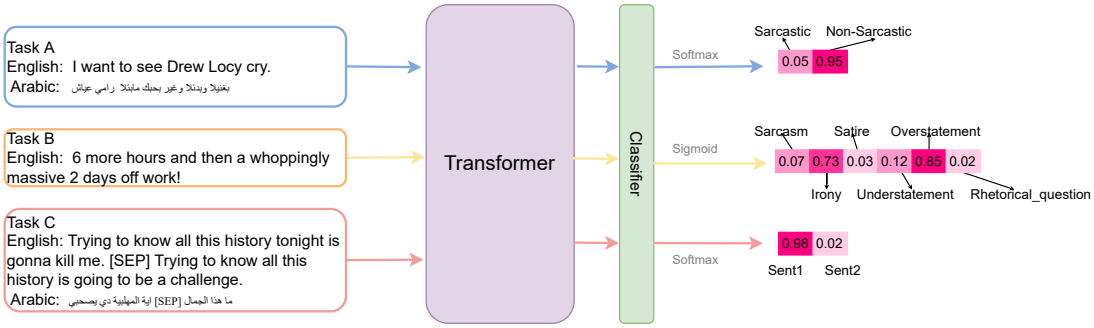

Figure 1: Fine-tuning pre-trained models on the *iSarcasmEval* data.

text has been extended to a broad range of data forms in social media, such as tweets, comments, and TV dialogues, due to their public availability. Sarcasm detection spanned several approaches like rule-based, supervised, and semi-supervised (Joshi et al., 2016) methods, resulting in further development for automatic sarcasm detection. Rule-based methods mainly rely on linguistic information, and their classification accuracy is often not very high due to the presence of noisy data. Most previous work on sarcasm detection based on supervised machine learning tends to rely on different types of features, including sentence length, the number of capitalized words, punctuation (Davidov et al., 2010), pragmatic factors such as emoticons (González-Ibáñez et al., 2011), turn-level sentiment lexicon (Wilson et al., 2005), sarcasm markers (Ghosh and Muresan, 2018), and so on. Meanwhile, neural models have been applied to this task, relying on semantic relatedness (Amir et al., 2016) and neural intra-attention mechanism to capture the sarcasm (Tay et al., 2018) and thus reducing feature engineering efforts.

Recently, pre-trained language models such as BERT (Devlin et al., 2018), ERNIE (Sun et al., 2019), and GPT-3 (Brown et al., 2020), have set the new state-of-the-art in a wide range of NLP benchmarks, such as GLUE (Wang et al., 2018). Khatri et al. (2020) evaluated the performance of pre-trained model using feature-based and fine-tuning methods on irony detection in English tweets, finding the latter is better. Meanwhile, there is also a surge of applying pre-trained models in sarcasm detection (Dadu and Pant, 2020; Potamias et al., 2020; Javdan et al., 2020). Our system explored the multilingual and monolingual pre-trained language models to testify their fine-tuning performance on English and Arabic sarcasm detection tasks.

3 Approach

3.1 Pre-trained Language Models

We adopt *pretrain-then-finetune* paradigm for better leveraging the performance of large-scale pre-trained models. As illustrated in Figure 1, for all tasks, we utilize pre-trained models to extract the input representations, followed by a fully-connected feed-forward layer and a softmax/sigmoid activation after the `[CLS]` token for prediction. For sub-task A and B that input samples only contain one sentence, we directly fine-tune the pre-trained Transformers. For sub-tasks with two sentences, *i.e.*, sub-task C, we employ the multi-layer pre-trained Transformer blocks as the cross-encoder by concatenating sentence pairs and separating them with a `[SEP]` token.

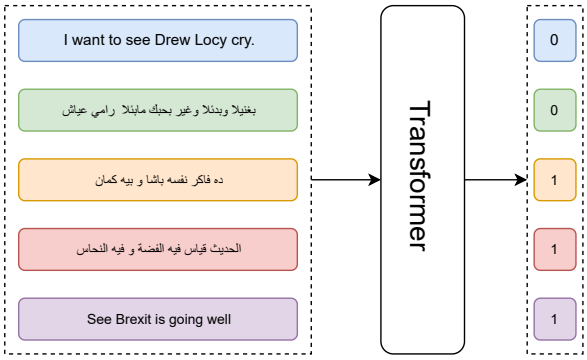

Figure 2: Multilingual learning on Task A. "0/1" indicate the non-sarcastic and sarcastic class.

3.2 Multilingual Learning

By observing that subtasks in task A and task C, we found that both subtasks in Task A and C are for the same objective but in different languages, *i.e.*, Arabic and English. Therefore, we adopt multilingual learning method by simultaneously fine-tuning the pre-trained models on both Arabic and

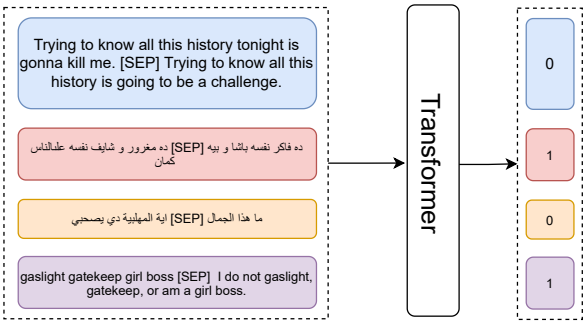

Figure 3: Multilingual learning on Task C. "0/1" indicate the first or second sentence belongs to sarcasm.

English training data based on multilingual pre-trained models, *i.e.*, ERNIE-M. Specifically, we combine both tasks in Task A or C as a single task, that is, training on Arabic and English sarcasm detection within the same subtask at the same time. As shown in Figure 2, we combine the one-sentence binary sarcasm detection subtasks in English and Arabic together and fine-tune the multilingual pre-trained models in one forward pass. Similarly, as illustrated in Figure 3, we conduct the identical settings for Task C. We found that this approach can achieve obvious performance gain on some specific settings and will discuss it in Secion 4.5.

3.3 Ensemble Learning

Considering the limited training data, we split the training data into k-fold with disparate random seeds, selecting one out of k data blocks for evaluation and using the rest $k-1$ for data training, as shown in Figure 4. Then, we choose the optimal model evaluated on various folds and random seeds. Finally, we apply ensemble techniques by averaging all outputs of test sets using optimal models.

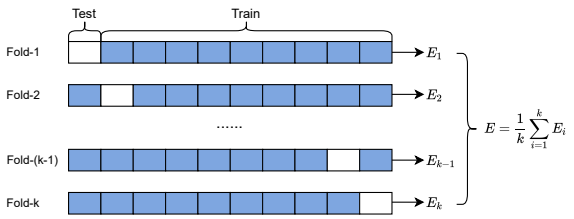

Figure 4: Illustration of ensemble learning. E_i indicates the prediction of the i-th model on the test set.

4 Experiments

4.1 Task Description

4.1.1 Task A: Binary Sarcasm Detection

The first task is binary text classification: given a tweet sample, the system needs to predict whether it is sarcastic or non-sarcastic. The following examples respectively present a sarcastic and non-sarcastic tweet.

(1) `The only thing I got from college is a caffeine addiction.` `(#sarcastic)`

(2) `I want to see Drew Lock cry.` `(#non-sarcastic)`

Example 1 is a sarcastic tweet where the author's true intention is "College is really hard, expensive, and exhausting, and I often wonder if the degree is worth the stress."

4.1.2 Task B: Multi-label Sarcasm Detection

The second task is a multi-label classification task, where the system requires to predict multiple categories out of six labels, such as `#Sarcasm`, `#Irony`, `#Satire`, `#Understatement`, `#Overstatement`, and `#Rhetorical_question`. The following examples provide examples for multiple sub-categories:

(1) `Falling asleep at your laptop is always fun.` `(#Sarcastic)`

(2) `Wow Bdubs can bench press 150 kilometers.` `(#Irony)`

(3) `Lil Pump is the Nelson Mandela of our generation.` `(#Satire #Sarcastic)`

(4) `Lucky for 2nd placed Brentford that there's no stand out team like Leeds this year, or they might have no chance of winning the league.` `(#Understatement #Sarcastic)`

(5) `6 more hours and then a whoppingly massive 2days off work! wowzers!` `(#Overstatement #Irony)`

(6) `wait what the fuck that solo yolo is mad? (#Rhetorical_question #Sarcastic)`

In the above examples, the types of sarcasm are subdivided into six categories. `#Sarcasm`, which is an ironic remark meant to mock by saying something different than what the speaker really means. For example, in example 1, the speaker hates falling asleep on his laptop. `#Irony` is when something happens that is the opposite of what was expected. As shown in example 2, the fact is that Bdubs cannot bench press 150 kilometers. `#Satire` is a type of wit that is meant to mock human vices or mistakes, often through hyperbole, understatement and sarcasm, as shown in Example 3. `#Understatement` is often a way of being critical. In example 4, because Norwich is the standout this year, Brentford cannot win the league. `#Overstatement` is an act of stating something more profound than it actually is, to make the point more serious, important, or beautiful. In example 5, a whoppingly massive two days off work means regret, and the genuine emotion ought not to require overstatement. `#Rhetorical_question` is a question that is asked even if the person doing the asking knows what the answer is. The solos in example 6 was truly expressed to be awful.

4.1.3 Task C: Binary Irony Classification on Two Sentences

The third subtask is binary classification: given a sarcastic tweet and its non-sarcastic rephrase (i.e., two tweets that convey the same meaning), the system needs to predict the sarcastic one. The following examples present a sarcastic sentence and its non-sarcastic paraphrase.

(1) `Trying to know all this history tonight is gonna kill me. (#Sarcastic)`

(2) `Trying to know all this history is going to be be a challenge. (#Rephrase)`

4.2 Evaluation Metrics

For these three sub-tasks, standard evaluation metrics including accuracy and F1 score are used to evaluate the participating system, calculated as follows:

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

Task	#Instances	#Metric
`Task A`	1400	F1-score
`Task B`	1400	Macro-F1 score
`Task C`	200	accuracy

Table 1: Summary of official test set in SemEval-2022 Task6.

Class Label	#Instances
`sarcastic`	867
`non-sarcastic`	2601
`total`	3468

Table 2: Satirical and non-satirical categories in training data.

$$precision = \frac{TP}{TP + FP} \quad (2)$$

$$recall = \frac{TP}{TP + FN} \quad (3)$$

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (4)$$

where TP, FP, TN, FN represent true positive, false positive, true negative, and false negative, respectively.

As shown in Table 1, task A, B and C use the F1-score for the sarcastic class, the Macro-F1 score over all classes, and accuracy, respectively. The Macro-F1 score implies that all class labels have equal weights in the final score.

4.3 Data

The detailed statistics of the sarcasm detection dataset are summarized in Table 2 and 3. As shown in Table 2, the training data are shown to be imbalanced, with 867 positive samples vs. 2601 negative ones. We only remove extra spaces, tabs, and line breaks for pre-processing. All emojis that contain emotional factors in training texts are kept without any change.

Multi-class Label	#Instances
`sarcasm`	713
`irony`	155
`satire`	25
`understatement`	10
`overstatement`	40
`rhetorical_question`	101

Table 3: Six satirical sub-categories in Task B.

4.4 Experiment Details

Due to the long-tailed nature of training data, we tried to use outside data[1] for data augmentation. Particularly, we merge the classes of "figurative""irony" in the exta data into a "sarcasm" class, but found it of no benefit. We conjecture that this is due to manual annotators' subjective intention can give the same samples various interpretations and therefore introduce some noise.

Still, the training data is relatively small and insufficient to achieve an unbiased performance estimate with a random train/test split. Instead, we use a k-fold cross-validation procedure ($k = 10$), a common model evaluation scheme in machine learning. The k-fold cross-validation procedure involves splitting the training dataset into k folds. In which $k - 1$ folds are used to train a model, and the rest one fold is used as the evaluation set. Finally, the final output of k models is the mean of these runs.

For English tasks, we compare ERNIE-M (Ouyang et al., 2020) and DeBERTa (He et al., 2021) as the pre-trained workhorse, while for Arabic tasks, we only consider ERNIE-M. We use the AdamW optimizer (Loshchilov and Hutter, 2017) and weight decay of 0.01. We warm up the learning rate for the first 10% of the update to a peak value of 1e-5 and 5e-6, respectively, and then linearly decay it afterward. We also use dropout (Srivastava et al., 2014) with a rate of 0.15 to prevent overfitting. We adopt a total batch size of 64 by running gradient accumulation on each GPU device with a step size of 8 and a batch size of 1, sharded across 8 NVIDIA V100 GPU chips. Our final solution is to ensemble all the model results obtained using a 10-fold cross-validation strategy with different learning rates (1e-5 and 5e-6) and training epochs (20 and 30), respectively.

4.5 Results

Table 4 compares the final performance on the official test set of task A,B,C under proposed model settings. It is obvious that DeBERTa outperforms ERNIE-M on English task since it is pre-trained only on English corpus. As to the multilingual learning in Task A and C, we observe the significant performance gain (*i.e.*, +6 absolute percentage point on F1 measure) on Task C while find it on par with monolingual fine-tuning on Task A. We guess this is because Task C are given two sentences for comparison, which is more straightforward than Task A (single sentence) to capture the ironic pattern for sarcasm detection. Due to the time limit, we only submit the monolingual fine-tuning results of ERNIE-M (*i.e.*, 84% acc.), which ranks 5th out of 13 in the Arabic subtask of Task C. Instead, the performance of our multilingual learning can achieve 2nd in Task C (Arabic). We contend that it would be worthwhile further exploring multilingual learning methods in various language settings in the future.

Task	Lang	ERNIE-M (multilingual)	ERNIE-M (monolingual)	DeBERTa	Rank
Task A	en	36.75	38.46	**56.91**(∗)	2/43
	ar	40.36	**41.87**(∗)	-	9/32
Task B	en	N/A	-	**7.99**(∗)	5/22
Task C	en	82.50	75.00	**87.00** (∗)	1/16
	ar	**90.50**	84.00(∗)	-	5/13

Table 4: Official test-set performance under various experimental settings. The "ERNIE-M (multilingual)" column indicates the performance of multilingual learning in Task A and C. Scores with asterisk indicate final submitted results. The official evaluation metrics for Task A,B,C are F1-score, macro F1-score, and accuracy, respectively.

5 Conclusion

We present our system that participated in SemEval Task 6 and employ the multilingual learning method to train the English and Arabic tasks jointly. We empirically find that it confers benefits in specific scenarios and outranks the monolingual pre-trained models on Arabic tasks. However, we do not adopt other Arabic-specific pre-trained models, which is also worth comparing. In the future, it is a promising direction to explore different sarcasm detection approaches under multilingual settings.

References

Ibrahim Abu Farha, Silviu 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)*. Association for Computational Linguistics.

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. *CoRR*, abs/1607.00976.

[1] https://www.kaggle.com/c/gse002/data?select=test.csv

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tanvi Dadu and Kartikey Pant. 2020. Sarcasm detection using context separators in online discourse. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 51–55.

Dmitry Davidov, Oren Tsur, and Ari Rappoport. 2010. Semi-supervised recognition of sarcasm in Twitter and Amazon. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 107–116, Uppsala, Sweden. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Debanjan Ghosh and Smaranda Muresan. 2018. "with 1 follower I must be AWESOME : P". exploring the role of irony markers in irony recognition. *CoRR*, abs/1804.05253.

Roberto González-Ibáñez, Smaranda Muresan, and Nina Wacholder. 2011. Identifying sarcasm in Twitter: A closer look. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 581–586, Portland, Oregon, USA. Association for Computational Linguistics.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Soroush Javdan, Behrouz Minaei-Bidgoli, et al. 2020. Applying transformers and aspect-based sentiment analysis approaches on sarcasm detection. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 67–71.

Aditya Joshi, Pushpak Bhattacharyya, and Mark James Carman. 2016. Automatic sarcasm detection: A survey. *CoRR*, abs/1602.03426.

Akshay Khatri, Pranav P, and Anand Kumar M. 2020. Sarcasm detection in tweets with BERT and glove embeddings. *CoRR*, abs/2006.11512.

John S. Leggitt and Raymond W. Gibbs. 2000. Emotional reactions to verbal irony. *Discourse Processes*, 29(1):1–24.

Ilya Loshchilov and Frank Hutter. 2017. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101.

Silviu Oprea and Walid Magdy. 2020. isarcasm: A dataset of intended sarcasm. *ArXiv*, abs/1911.03123.

Xuan Ouyang, Shuohuan Wang, Chao Pang, Yu Sun, Hao Tian, Hua Wu, and Haifeng Wang. 2020. Ernie-m: enhanced multilingual representation by aligning cross-lingual semantics with monolingual corpora. *arXiv preprint arXiv:2012.15674*.

Rolandos Alexandros Potamias, Georgios Siolas, and Andreas-Georgios Stafylopatis. 2020. A transformer-based approach to irony and sarcasm detection. *Neural Computing and Applications*, 32(23):17309–17320.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. 2019. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*.

Yi Tay, Luu Anh Tuan, Siu Cheung Hui, and Jian Su. 2018. Reasoning with sarcasm by reading in-between. *CoRR*, abs/1805.02856.

Joseph Tepperman, David Traum, and Shrikanth Narayanan. 2006. Yeah right: Sarcasm recognition for spoken dialogue systems.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, British Columbia, Canada. Association for Computational Linguistics.