R2D2 at SemEval-2022 Task 6: Are language models sarcastic enough? Finetuning pre-trained language models to identify sarcasm

Mayukh Sharma, Ilanthenral Kandasamy and **W.B. Vasantha**
School of Computer Science and Engineering
Vellore Institute of Technology
Vellore - 632014, Tamil Nadu, India
04mayukh@gmail.com, ilanthenral.k@vit.ac.in,
vasantha.wb@vit.ac.in

Abstract

This paper describes our system used for SemEval 2022 Task 6: iSarcasmEval: Intended Sarcasm Detection in English and Arabic. We participated in all subtasks based on only English datasets. Pre-trained Language Models (PLMs) have become a de-facto approach for most natural language processing tasks. In our work, we evaluate the performance of these models for identifying sarcasm. For Subtask A and Subtask B, we used simple finetuning on PLMs. For Subtask C, we propose a Siamese network architecture trained using a combination of cross-entropy and distance-maximisation loss. Our model was ranked 7^{th} in Subtask B, 8^{th} in Subtask C (English), and performed well in Subtask A (English). In our work, we also present the comparative performance of different PLMs for each Subtask.

1 Introduction

Sarcasm, a form of figurative speech, allows people to express contempt or mock using irony. The irony is used to communicate the opposite of the intended meaning to express humour or mock something. Sarcasm plays an essential role in people's daily conversation and finds its use across social media to express thoughts. Recently, social media has drawn in millions of users around the world. Owing to its figurative nature, sarcasm poses a significant challenge to systems performing sentiment analysis on these social media platforms. Therefore, it is essential to design and develop systems that efficiently identify sarcasm. The most challenging aspect of sarcasm is the different socio-cultural backgrounds of people who drive it. Therefore, sarcasm intended by the author may not always be perceived by the audience from different backgrounds. Most datasets on sarcasm detection are collected using predefined criteria or human annotators (Oprea and Magdy, 2020). This is a sub-optimal approach that may not always capture the sarcasm intended by the author leading to noise in the models trained on it. SemEval 2022 Task 6: iSarcasmEval: Intended Sarcasm Detection in English and Arabic (Abu Farha et al., 2022) draws attention to the problem of identifying sarcasm using a dataset of intended sarcasm.

Our proposed approach for Subtask A and Subtask B approach uses a classification objective to finetune PLMs. PLMs learn semantic and syntactic features via training on large amounts of a text corpus. This information is used in downstream tasks by simply finetuning task-specific datasets. PLMs have shown remarkable performance on such downstream tasks using the simple finetuning approach (Sharma et al., 2021b). We extend the same idea to identify sarcasm (Subtask A) and identify the type of irony (Subtask B). Subtask C aimed at identifying sarcastic text from its non-sarcastic counterpart. We propose a Siamese network-based architecture using PLMs trained on a combination of cross-entropy and distance-maximisation loss. The classification objective(cross-entropy) identifies the sarcastic/non-sarcastic text, while the distance-maximisation loss maximises the distance between sarcastic/non-sarcastic features learnt by the model during training. We experimented with different PLMs, namely BERT, RoBERTa, MPNet, DeBERTa, to present a comparative study of their performance for the task of sarcasm detection.

Our final submissions for Subtask A and Subtask B used finetuning on MPNet, while we used an ELECTRA-based model for Subtask C. Our proposed system performed well in Subtask B and Subtask C(English), attaining a 7^{th} and 8^{th} rank, respectively on the official leaderboard and performed well in Subtask A. Our experiments show that all PLMs had almost similar performance with slight variation in results. The overall scores were low for Subtask A and Subtask B, indicating that sarcasm detection and identifying the type of irony is a difficult task for PLMs. However, Subtask C

Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1018 - 1024

results show that the models can efficiently differentiate between a sarcastic text and its non-sarcastic rephrase. Our code available at GitHub[1] for method replicability.

2 Background

Identifying sarcasm is an essential task in Natural Language Processing (NLP). Owing to its figurative nature, it affects the performance of sentiment analysis systems whose performance have significantly improved over the years (Rosenthal et al., 2017) (Sharma et al., 2021a). In social media, sarcasm is mainly used for humour but can hide hateful content, making the identification of sarcasm a vital topic. Methods to deal with sarcasm detection can be separated into two categories, i.e. content-based models and context-based models (Hazarika et al., 2018). Text-based models model the problem as a classification task using pragmatic and lexical features to identify sarcasm. (Riloff et al., 2013) shows sarcasm is expressed as a combination of positive sentiment words and negative situations. Work done in(Joshi et al., 2015) uses the concept of context incongruity for sarcasm detection. Contextual methods use information about the text and the context in which the text is used. (Khattri et al., 2015) uses the sentiment of the tweet as well as the history of the author's previous tweets on similar topics to identify sarcasm. Work done in (Wallace et al., 2015) uses nouns and sentiments presented in a forum towards irony/sarcasm detection. (Hazarika et al., 2018) worked on using both content as well as contextual information for identifying sarcasm. (Castro et al., 2019) presents work done to identify sarcasm from TV shows in a multimodal setting. (Sharma et al., 2020) used a multimodal feature fusion model using attention (Bahdanau et al., 2016) for identifying sarcasm in internet memes. (Felbo et al., 2017) use models trained on emoji using distant supervision to identify sentiment, sarcasm, and emotion. Another challenge in sarcasm detection is the availability of data. Due to the highly subjective nature of sarcasm, it is challenging to collect high-quality data. (Oprea and Magdy, 2020) tries to solve this problem by introducing a dataset of intended sarcasm where the sarcastic data is labelled by the authors removing any noise or ambiguity in labels. The subjective and figurative nature of sarcasm makes it a formidable task, and it poses a challenge for affective systems performing sentiment analysis (Satapathy et al., 2017). Therefore, the task of sarcasm detection is essential to advance state-of-the-art sentiment analysis systems. Moreover, most datasets for sarcasm detection contain much noise and are sub-optimal in capturing the sarcasm intended by the author of the text. SemEval 2022 Task 6: iSarcasmEval: Intended Sarcasm Detection In English and Arabic (Abu Farha et al., 2022) aims to use a dataset of intended sarcasm for identifying sarcasm in text. The task has three subtasks which we define as:

Subtask A (English and Arabic): Given a labelled dataset D of texts, the task aims to learn a classification function that can identify sarcastic/non-sarcastic texts.

Subtask B (English only): For a given labelled dataset D of texts, the objective of the task is to learn a multilabel classification function that can predict the type of irony I where $I \in \{$ Sarcasm, Irony, Satire, Understatement, Overstatement, Rhetorical $\}$.

Subtask C (English and Arabic): Given a dataset D of sarcastic texts and their non-sarcastic rephrase, i.e. both texts convey the same meaning, the objective of the task is to learn a classification function that can identify the sarcastic text from its non-sarcastic rephrase.

Our team participated in Subtask A(English), Subtask B, and Subtask C(English).

Dataset statistics: Table 1 and Table 2 contain the dataset statistics for all Subtasks (English). Dataset statistics for Subtask A and Subtask B show a clear data imbalance problem. To overcome the class imbalance, we used sklearn to compute class weights which are defined as: Let X be the vector containing counts of each class X_i where $i \in X$ and N be the total number of samples. Then the weights for each class were given as: $weight_i = N/(length(X) * X_i)$ where length function computes the number of classes in vector X. There was no imbalance for Subtask C for each sarcastic sample, the corresponding non-sarcastic rephrase was given.

3 System overview

3.1 *Pre-trained language models (PLMs):*

NLP, a diverse field, contains an array of tasks, but most datasets for these tasks contain only a few hundred or thousand human labelled samples. This makes training large models for these tasks

[1] https://github.com/04mayukh/R2D2-at-SemEval-2022-Task-6-iSarcasmEval

Type	Sarcasm	Irony	Satire	Understatement	Overstatement	Rhetorical	Total
Train	677	147	24	9	38	94	823
Validation	147	8	1	1	2	7	44
Test	180	20	49	1	10	11	1400

Table 1: Dataset statistics for SubTask B.

	Subtask A (English)			Subtask C (English)
Type	Sarcastic	Not Sarcastic	Total	Sarcastic/Rephrase
Train	794	2327	3121	780
Validation	73	274	347	87
Test	200	1200	1400	1400

Table 2: Dataset statistics for Subtask A and Subtask C.

a challenging task. Transfer learning using GloVe (Pennington et al., 2014) and FastText (Bojanowski et al., 2017) is one of the popular choices for solving this problem. Most recently, researchers came up with a method called pre-training (Qiu et al., 2020), which involves training general-purpose models from unannotated text data. This allows models to learn syntactic and semantic features in the text in an unsupervised setting. Transformer architecture proposed in (Vaswani et al., 2017) is the most common choice for training PLMs. These models can be finetuned on various downstream tasks using task-specific datasets. Finetuning allows models to adapt to small task-specific datasets easily and shows promising results (Sharma et al., 2021b). Next, we provide a summary of PLMs used in our approach.

3.2 *Brief overview of used PLMs:*

BERT: It is a bidirectional language model developed by Google that uses transformers. BERT (Devlin et al., 2019) stands for Bidirectional Encoder Representation using Transformer. It uses the auto-encoding modelling technique. It uses Masked Language Modelling (MLM) and the Next Sentence Prediction (NSP) objective for pre-training the model.

RoBERTa: A Robustly Optimized BERT Pre-training Approach was proposed by Facebook in (Liu et al., 2019) and used the BERT architecture with slight modifications to improve its performance. They replaced MLM with dynamic masking and removed the NSP objective during pre-training. They also found that BERT was undertrained, so they trained the model for longer durations with more data and bigger batch size. RoBERTa outperformed BERT on several downstream tasks.

ELECTRA: It was inspired by generative adversarial networks and introduced a new pre-training objective called Replaced Token Detection (RTD) (Clark et al., 2020). Unlike MLM, which introduces <MASK> tokens, ELECTRA replaces specific tokens with plausible fakes. The pre-training objective is to identify if the given token is replaced or the original one. Unlike BERT, where only prediction for the masked token is done, replaced token detection objective is applied to all tokens in ELECTRA, making RTD more efficient than MLM.

MPNet: It was proposed by Microsoft in (Song et al., 2020) and uses a combination of auto-regressive and auto-encoding strategies for pre-training. It solves the problem of MLM in BERT and permuted language modelling in XLNet (Yang et al., 2019) and achieves better performance. It models dependency between tokens using permuted language modelling (vs MLM in BERT) and uses the auxiliary position information to allow the model to see complete sentence reducing position discrepancy (vs permuted language modelling in XLNet). Thus, it uses a combination of masked language modelling and permuted language modelling to jointly model the dependency among predicted tokens and use positional information of complete sentences.

3.3 *Finetuning (Subtask A and Subtask B):*

For Subtask A and Subtask B, we finetuned the pre-trained models defined above. Subtask A was a binary classification task. We added a simple classification head on top of PLMs for Subtask A. It consisted of a 64-neuron dense layer followed by a batch normalisation layer and a final 1-neuron

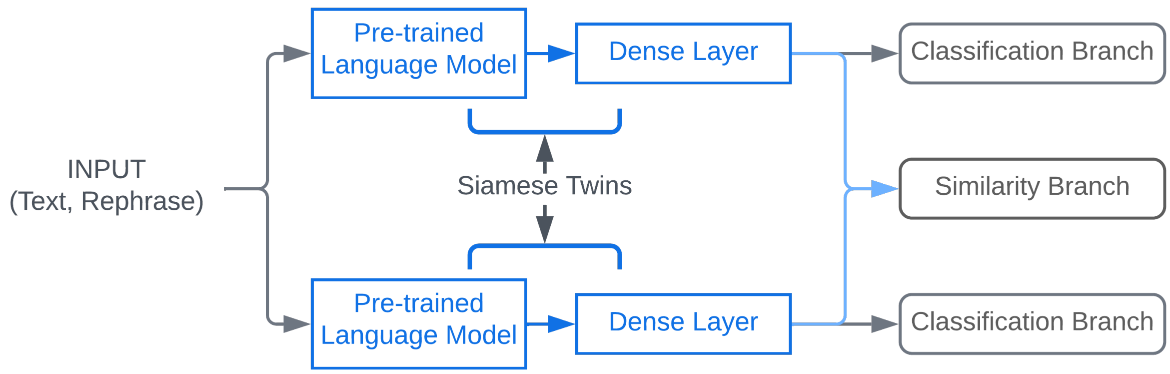

Figure 1: Overview of our Siamese network architecture for Subtask C.

layer with sigmoid activation. Subtask B was a multilabel task aimed at identifying the type of irony. We used a multi-branch model using features from the PLMs, with each branch trying to identify one of the given categories of irony. We used the same classification head for each branch defined in Subtask A.

3.4 *Siamese Network (Subtask C):*

The goal of Subtask C was to identify a sarcastic text from its non-sarcastic rephrase. We use a model based on Siamese network trained using a combination of cross-entropy and distance maximisation loss. Figure 1 shows our Siamese network architecture. A Siamese neural network comprises twin networks that can accept distinct inputs. These twin networks share the same weights, which is known as weight tying (Koch, 2015). Weight tying ensures that the features generated for distinct input are in the same feature space because each network calculates the same function. We use the PLMs to define the twin network of our Siamese architecture. We use the twin network to generate features corresponding to sarcastic text and its non-sarcastic rephrase. Next, these features are separated into two different branches, which we define as:

Distance Maximisation (Similarity branch): Since we want to separate the sarcastic text from its non-sarcastic rephrase, we want to maximise the distance between their features learned by the twin network. We merge the features generated from the twin network using Euclidean distance. It is then passed through a single neuron layer with sigmoid activation, which helps to normalise the distance within a known range of 0-1. We maximise this distance using the distance maximisation loss function, which we define as: Let d_i be the output from the final layer(1-neuron with sigmoid activation), then the loss function L is defined as:

$$L_i = (max[1 - d_i, 0])^2$$

and $L = \sum L_i$ for i belongs to N (Total samples). This ensures that sarcastic/non-sarcastic texts with similar meanings learn different sets of features.

Classification Branch: The primary role of this branch is to classify the generated features as sarcastic/non-sarcastic. It contains two classification heads, each using features generated from the twin network. The classification head comprises a 64-neuron dense layer followed by a batch normalisation layer and a final 1-neuron classification layer with sigmoid activation. Each head independently classifies the features from the respective outputs of the twin network. The classification head with the best performance on the development set was used for making predictions on the test set.

4 Experimental Setup

Text pre-processing: The text was first passed through a pre-processing pipeline to remove noise and normalize into standard features. We removed any website names in the text as they add noise to the data. We also found certain chat words like LOL (laugh out loud) present in the data and converted them into their respective full forms. Emojis are converted to their actual meanings. We used the emoji[2] library for emoji conversion. Lastly, we used ekphrasis (Baziotis et al., 2017) to normalize date, numbers to a standard format and perform spelling correction. PLMs require the text to be tokenized as part of pre-processing step. We use

[2] https://github.com/carpedm20/emoji/

Model	Sarcasm	Irony	Satire	Understatement	Overstatement	Rhetorical	Macro-F1
MPNet*	**.248**	**.032**	**.139**	**.003**	**.0**	**.034**	**.076**
BERT	.219	.042	.126	.0	.031	.022	.074
RoBERTa	.256	.060	.086	.0	.014	.118	.089
ELECTRA	.232	.037	.080	.0	.015	.023	.064

Table 3: Test set results(F1 score) for different PLMs using simple finetuning on Subtask B. (MPNet* was used for the official submission and has been highlighted in bold). The underlined score respresent the best performing models for each sarcasm category.

	Subtask A			Subtask C	
Model	F1 Sarcastic	F1 Macro	Accuracy	F1 Macro	Accuracy
ELECTRA*	.330	.541	.637	**.741**	**.750**
BERT	.323	.522	.605	.763	.765
RoBERTa	.324	.523	.606	.716	.720
MPNet∼	**.3276**	**.5265**	**.610**	.728	.735

Table 4: Test set results for Subtask A and Subtask C. Our final submission for Subtask A was done using MPNet∼ and for Subtask C using ELECTRA*. We have highlighted the official submissions in bold and underlined the individual best metrics across different PLMs.

hugging face's implementation of Fast tokenizers[3] for each pre-trained model. Sequence length was fixed to 70 tokens. Samples greater/smaller than the defined length were truncated or padded.

Data preparation for Siamese network (Subtask C): The dataset for Subtask C consisted of sarcastic texts and their rephrase. We rearranged this data to make sure the input to the Siamese network contains samples in the form of (sarcastic, rephrase) and (rephrase, sarcastic). This is important because the classification layers on top of the Siamese network are used to make predictions on the inputs of the twin network independently using two classification heads. If we do not rearrange the data, each of the two heads will learn to simply predict the output as always sarcastic and non-sarcastic, thereby not learning from training data.

Finetuning: Our approach for all subtasks involves finetuning PLMs and using their features. We used features of [CLS] token for BERT, ELECTRA and start token (<s>) features for MPNet and RoBERTa. These features are then passed to further layers of models as per the architecture we defined above.

Hyperparameters and Training: We developed our models on Keras[4] (Chollet et al., 2015) and used Hugging Face's[5] implementation of transformer[6](Wolf et al., 2020) models. Finetuning was performed on Colab using TPUs. For finetuning we used Adam (Kingma and Ba, 2015) optimiser. We experimented with learning rates ranging from 2e-5 to 5e-5. For Subtask A and Subtask B, we used a binary cross-entropy loss. For Subtask C, we used a binary cross-entropy loss on the classification branch and distance maximization objective on the similarity branch. We finetuned the models for ten epochs and used the weights with the best performance on the development set to make predictions on the test set.

Evaluation metric: Subtask A uses the F1 score of the sarcastic class as an evaluation metric. For Subtask B, the macro averaged F1 score is the official metric, while for Subtask C, accuracy is used as the evaluation metric.

5 Results

Table 3 and Table 4 describe the results of our experiments using different pre-trained models. Our official submission for the task used MPNet for Subtask A, Subtask B and ELECTRA for Subtask C. Our system was ranked 7^{th} in Subtask B, attaining a macro F1 score of 0.076 with the highest F1 score for satire and second highest score for sarcasm and understatement category. Our Siamese architecture-based system also performed well, attaining accuracy of 75% and ranked 8^{th} on the leaderboard for Subtask C.

We performed experiments using BERT, RoBERTa, MPNet, and ELECTRA during the

[3]Hugging Face's Fast Tokenizers
[4]https://keras.io
[5]https://huggingface.co
[6]https://huggingface.co/transformers

evaluation phase. We evaluated these models using the validation set, and the model with the best performance was used to make a submission on the test data. Table 3 and Tabel 4 summarise results using different pre-trained models. We have highlighted the official test submissions in bold while underlined the best performing metric across different models. Subtask A results show that all PLMs have similar performance for sarcasm detection. They perform well, but there is considerable scope for improvement. PLMs are trained on general text corpus making it difficult for them to understand figurative content like sarcasm. For Subtask B, RoBERTa performs better than other PLMs. Results for Subtask B show that it is comparatively easy to identify sarcasm and satire compared to other types of irony, which have very low performance on evaluation metrics. Another reason for this could be the high imbalance in the dataset, making it difficult for models to identify different types of irony. For Subtask C, all models have similar performance with slight variations. The models perform significantly better on evaluation metrics when compared to Subtask A and Subtask B, indicating that models can distinguish between sarcastic and non-sarcastic content having similar meanings.

6 Conclusion

This paper describes our proposed model used for SemEval 2022 Task 6: iSarcasmEval: Intended Sarcasm Detection In English and Arabic. Different PLMs were used to do a comparative analysis of their performance for the sarcasm detection task. Our fine-tuning approach worked well for Subtask B, with the best score for satire and second-best performance for the sarcasm and understatement category. For Subtask C, we proposed a novel Siamese network architecture to identify sarcastic content from it's non-sarcastic rephrase. It performed well, attaining 8^{th} rank on the leaderboard. Our comparative analysis shows that sarcasm detection is a difficult task for the PLMs, and there is scope for further improvements, which we will take up in future works.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _Obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4619–4629, Florence, Italy. Association for Computational Linguistics.

François Chollet et al. 2015. Keras. `https://keras.io`.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bjarke Felbo, Alan Mislove, Anders Søgaard, Iyad Rahwan, and Sune Lehmann. 2017. Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark. Association for Computational Linguistics.

Devamanyu Hazarika, Soujanya Poria, Sruthi Gorantla, Erik Cambria, Roger Zimmermann, and Rada Mihalcea. 2018. CASCADE: Contextual sarcasm detection in online discussion forums. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1837–1848, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

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.

Anupam Khattri, Aditya Joshi, Pushpak Bhattacharyya, and Mark Carman. 2015. Your sentiment precedes you: Using an author's historical tweets to predict sarcasm. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 25–30, Lisboa, Portugal. Association for Computational Linguistics.

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*.

Gregory R. Koch. 2015. Siamese neural networks for one-shot image recognition.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. 2020. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897.

Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as contrast between a positive sentiment and negative situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 704–714, Seattle, Washington, USA. 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.

Ranjan Satapathy, Claudia Guerreiro, Iti Chaturvedi, and Erik Cambria. 2017. Phonetic-based microtext normalization for twitter sentiment analysis. In *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 407–413.

Mayukh Sharma, Ilanthenral Kandasamy, and W.b. Vasantha. 2020. Memebusters at SemEval-2020 task 8: Feature fusion model for sentiment analysis on memes using transfer learning. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1163–1171, Barcelona (online). International Committee for Computational Linguistics.

Mayukh Sharma, Ilanthenral Kandasamy, and W.B. Vasantha. 2021a. Comparison of neutrosophic approach to various deep learning models for sentiment analysis. *Knowledge-Based Systems*, 223:107058.

Mayukh Sharma, Ilanthenral Kandasamy, and W.b. Vasantha. 2021b. YoungSheldon at SemEval-2021 task 7: Fine-tuning is all you need. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1146–1152, Online. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *NeurIPS 2020*. ACM.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Byron C. Wallace, Do Kook Choe, and Eugene Charniak. 2015. Sparse, contextually informed models for irony detection: Exploiting user communities, entities and sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1035–1044, Beijing, China. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi 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 M. 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.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.