What Matters in Irony Detection: An Extended Feature Engineering for Irony Detection in English Tweets.

Linrui Zhang, Qixiang Pang, Belinda Copus
University of Central Missouri
W.C. Morris 222, Warrensburg, Missouri
{lzhang, qpang, copus}@ucmo.edu

Abstract

In recent years, large-scale language models (LLMs) have nearly become the dominant force in almost every natural language processing (NLP) task. The primary research approach has focused on selecting the most appropriate language model for specific NLP tasks and then incorporating linguistic features to enhance the model's performance. With swift progress in this field, new features and models are evolving rapidly, and outdated systems require timely updates. In this paper, we extended the accomplishments of SemEval-2018 Task 3, enhancing its irony detection systems with novel features and more sophisticated language models. Subsequently, we conducted an ablation study to showcase the contributions of these enhancements to the LLM-based system. Furthermore, we compared our leading system with the top performers in the SemEval-2018 competition, and our best model exhibited superior performance when compared to the leading performers applied to the same corpus.

Introduction

The identification of irony has a lengthy history involving the utilization of linguistic features, whether in the context of traditional rule-based or machine learning-based approaches (Joshi, Bhattacharyya, and Carman 2017). These features often include lexical features, sentiment features (Bouazizi and Ohtsuki 2015) (Farías, Patti, and Rosso 2016), (González-Ibánez, Muresan, and Wacholder 2011), Part-of-Speech (POS) tags (Reyes, Rosso, and Veale 2013) and so on. In the past few years, with the advancement of Deep Learning, researchers have developed a standardized two-step pipeline for constructing irony detection models: (1) performing feature engineering and (2) inputting the extracted features into neural network architectures, such as RNN (Schmidhuber 1989), LSTM (Hochreiter and Schmidhuber 1997). For example, in SemEval-2018 Task 3 (Van Hee, Lefever, and Hoste 2018), the second-ranked system, THU_NGN (Wu et al. 2018), was constructed using a Dense-LSTM model that incorporated POS tags and sentiment features. However, in recent years, with the rise of the pre-training and fine-tuning paradigm, there has been a growing number of proposals for pre-trained large-scale language models like BERT (Vaswani et al. 2017), and they are increasingly assuming a prominent role in the field of Natural Language Processing. This shift has led to the phasing out of proposed models from SemEval-2018 Task3. In this case, we have chosen novel features—those either not selected or only partially selected by SemEval teams—and applied them across a range of cutting-edge language models, including BERT, XLNET, and their variations. Subsequently, we demonstrated the impact of these newly incorporated features and language models on the task of irony detection, and compared our renovated models with the top performers in SemEval-2018. The primary contributions of this paper are as follows:

1. We enhanced the SemEval-2018 work by incorporating novel features and cutting-edge language models.
2. We demonstrated the impact of these enhancements on the LLM-based irony detection system.
3. Our best system outperformed the top participants in the SemEval-2018 Task 3 corpus.

Task Description

Irony detection refers to the process of identifying and understanding instances of irony in written or spoken language. The inception of this task traces back to SemEval-2018 Task3 (Van Hee, Lefever, and Hoste 2018),which was the first shared task on irony detection. In this initiative, ironic tweets were gathered through the use of irony-related hashtags (i.e. #irony, #not) and were subsequently annotated manually. There are two goals for this task: (1) Task A involves determining whether a given tweet is ironic (Binary Classification), and (2) Task B involves identifying which type of irony (if any) is expressed (Multilabel Classification). For example, consider the following tweet:

A wonderful day of starting work at 6 am.

The phrase "wonderful day" typically implies a positive or enjoyable experience. However, the addition of "starting work at 6 am" suggests an early and potentially undesirable or challenging start to the day. Thus, this tweet is classified as ironic (Task A), and its type is verbal irony realized through a polarity contrast (Task B).

Feature Selected

We have chosen four linguistic features and five language models with the potential to enhance the performance of the irony detection model. The selected components and the rationale behind their selection are outlined as follows:

- **Emoji**. Emojis serve as visual indicators of emotions, tone, or sarcasm, helping to disambiguate the intended meaning of a message. By incorporating emojis into irony detection models, they can leverage this additional layer of information, making it easier to discern between literal and ironic expressions (Shiha and Ayvaz 2017) (Chen et al. 2018).
- **Emojitext**.(Singh, Blanco, and Jin 2019) has shown that replacing emojis with their natural language description can significantly improve accuracy for tweet classification. Applying this concept, we utilized the Emoji for Python package[1] to transform emojis and emoticons into text representing their meanings. For example, ☻ will be converted into "smiling_face". This approach enables us to gather additional context from tweets.
- **Domain data**. The essence powering machine learning models lies in data (Li, Hou, and Che 2022). Numerous studies (Wei and Zou 2019) (Liu and Yu 2020) (Dong et al. 2021) have demonstrated that augmenting domain-specific data enhances the efficacy of the training process, resulting in improved model performance. As a result, we integrated additional data from the iSarcamEval corpus (Farha et al. 2022), specifically focusing on the sarcasm detection task sourced from SemEval-2022 Task 6. Given the resemblance between sarcasm and irony in text, we assumed that this incorporated corpus could boost the language model's ability to generalize, alleviate overfitting, and thereby enhance the overall system performance.
- **Hashtag**. Hashtags serve as contextual cues on social media platforms. In the context of irony, hashtags such as #irony, #sarcasm, and #not provide users with valuable signals by highlighting the intended ironic tone or context in a concise and recognizable manner.
- **Language models**. We have selected five prominent language models: BERT (Vaswani et al. 2017), BERTweet (Nguyen, Vu, and Nguyen 2020), TwHIN-BERT (Zhang et al. 2022), ALBERT (Lan et al. 2019), and XLNet (Yang et al. 2019). Notably, BERT serves as our baseline model. BERTweet and TwHIN-BERT represent variants of BERT specifically trained with Tweet language, illustrating the impact of word embeddings on the system. ALBERT, a lighter version of BERT, is utilized to demonstrate the influence of parameter size on the system. Additionally, we have included XLNet, which is not part of the BERT family, to enrich the diversity of the chosen models.

System and Approach

Data Preprocessing

The initial SemEval irony detection corpus provided 3,834 English tweets for training and an additional 784 tweets for testing. To enrich the dataset, we integrated data from the iSarcamEval corpus (Farha et al. 2022), expanding the training set with 4,335 instances of (non-)sarcastic data. Emojis and hashtags were already managed by the original corpus—removed or added as necessary. For converting emojis (e.g., ☹) into their text descriptions (e.g., "angry_face"), we utilized the Emoji for Python project. Additionally, we implemented a data cleaner program to rectify or eliminate incorrect, corrupted, improperly formatted, duplicate, or incomplete data within the dataset.

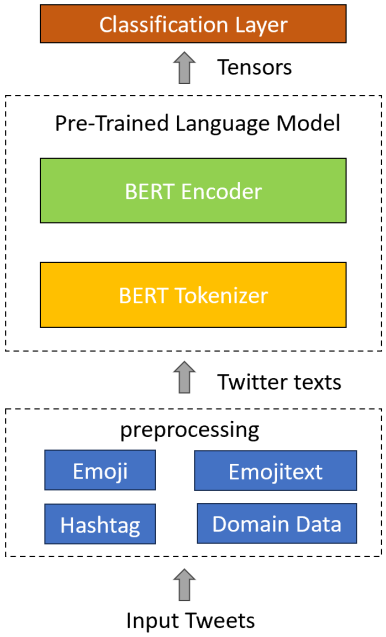


Figure 1: The main structure of our system.

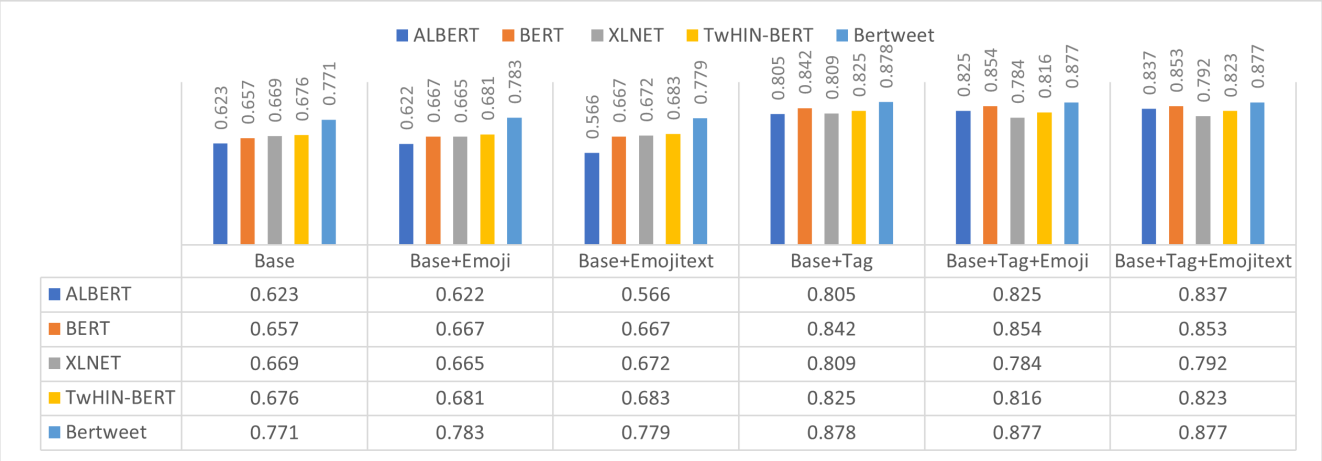
System Overview

We utilized five large-scale language models: BERT, BERTweet, TwHIN-BERT, ALBERT, and XLNet. Here, we use the BERT model as an example to illustrate the fundamental structure of our system. The pipeline of the structure can be found in Figure 1.

We initially employed the previously discussed method to process the tweets and then passed them to the tokenizer. Using the pre-trained BERT tokenizer, the input tweets underwent tokenization, breaking them into smaller tokens in preparation for input into the BERT encoder. Subsequently, the BERT encoder generated a condensed representation summarizing the entire input sequence. Finally, this representation was fed into a classification layer for the ultimate task of determining whether the given tweets are ironic or not.

We adopted the implementation of the BERT tokenizer and encoder from the Hugging Face Model Hub (Wolf et al. 2020) and initialized the tokenizer and encoder with the `Bert-base-uncased` checkpoint. We fine-tuned it for a maximum of 10 epochs, employing an early stopping

[1]https://pypi.org/project/emoji/



	Base	Base+Emoji	Base+Emojitext	Base+Tag	Base+Tag+Emoji	Base+Tag+Emojitext
ALBERT	0.623	0.622	0.566	0.805	0.825	0.837
BERT	0.657	0.667	0.667	0.842	0.854	0.853
XLNET	0.669	0.665	0.672	0.809	0.784	0.792
TwHIN-BERT	0.676	0.681	0.683	0.825	0.816	0.823
Bertweet	0.771	0.783	0.779	0.878	0.877	0.877

Figure 2: The performance of the irony detection models with the four chosen linguistic features and five language models on SemEval-2018 Task 3 Task A.

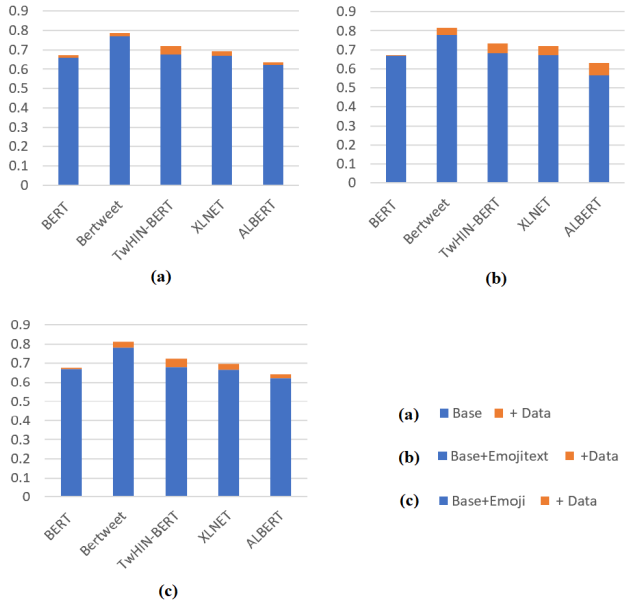


Figure 3: The impact of domain-specific data on the irony detection systems for SemEval-2018 Task 3 Task A

callback. The evaluation metric is the F-1 score (macro-averaged F-1 for Task B), which is the original assessment method in SemEval-2018 Task 3. The experiments were conducted using a Google Colab T4 GPU.

Experimental Results and Analysis

Figure 2 illustrates the performance of the irony detection models with the four chosen linguistic features and five language models on Task A.

The Impact of Pre-trained Language Models

From Figure 2, we can observe that the top-performing models are BERTweet and TwHIN-BERT, with BERTweet particularly standing out as it clearly outperforms the other models. This distinction is due to the fact that the above-mentioned two models are pre-trained on a large corpus of tweets, which allows them to capture the specific linguistic nuances and characteristics of Twitter language. Regular BERT, on the other hand, is trained on a diverse range of text from the internet, which may not adequately capture the unique features of tweets.

In addition, we observed that ALBERT exhibits inferior performance compared to BERT and XLNet. This discrepancy can be attributed to ALBERT being a lightweight version of BERT [2], featuring smaller models with fewer parameters and training on a less extensive dataset. The reduced capacity of ALBERT limits its ability to acquire richer and more detailed contextual representations compared to BERT and XLNET, leading to less than satisfactory performance. However, on the positive side, lightweight language models require less computational power and achieve faster training times.

The Impact of Emoji, Emojitext, Hashtag

Illustrated in the first three columns of Figure 2, emojis and emojitext can enhance system performance. However, their positive impact is negligible, and in some cases, it may even have a negative influence. Our understanding is that the influence of emojis (and emojitext) is absorbed by the BERT and XLNET encoder. Since they are not specifically trained on tweet text, it does not attribute meaningful embeddings to emojis. Instead, it treats them as normal words (or even noise), making it challenging to extract substantial information from them. Conversely, the outcomes appear promis-

[2]The BERT-base model contains 110 million parameters, while ALBERT, with only 11 million parameters, is 10 times smaller than BERT.

ing when considering BERTweet and TwHIN-BERT, both trained on tweet text and endowed with the capability to assign meaningful embeddings to emojis.

Hashtags serve as significant indicators reflecting people's attitudes, and users frequently use them to convey specific themes or topics, such as celebrating achievements with #BossMoves or expressing indifference with #IDontCare. In fact, in many studies on human attitudes, hashtags serve as the gold standard for data collection (Rosenthal, Farra, and Nakov 2019) (Ghosh et al. 2015) (Farha et al. 2022). Therefore, it is unsurprising to observe, as depicted in the last three columns of Figure 2, that the inclusion of hashtags significantly increases system performance.

The Impact of Domain Data

Figure 3 illustrates the impact of domain specific data on the system for Task A. The orange color indicates the performance improvements attained through the utilization of data from iSarcamEval corpus (Farha et al. 2022). From the figure, we can observe that incorporating (non-)sarcastic data benefits the system performance of each language model under three scenarios (i.e., base, base+emoji, base+emojitext) [3]. Considering the similarity that both irony and sarcasm tasks involve a gap between the literal meaning of the words and the intended meaning, the iSarcamEval corpus qualifies as domain-specific data and thus contributes to the performance gains.

Comparison with the SemEval-2018 Participants

We also compared our best system with the top performers in SemEval-2018 Task 3 (Ghosh and Veale 2018) (Wu et al. 2018) (Baziotis et al. 2018) (Rohanian et al. 2018). The comparison results can be found in Table 1. It should be emphasized that we did not leverage domain data to enhance our system for Task B since it involves a multiclass irony classification problem, whereas the iSarcamEval corpus focuses solely on binary classification. Furthermore, in the original SemEval-2018 competition, hashtags were excluded from the training data. To maintain consistency in our comparison, we also employed our non-hashtag models for evaluation.

From the comparison results depicted in Table 1, we observed that our best model achieved F-1 scores of 0.813 and 0.550 in Task A and Task B, respectively. The achieved scores exceed those of the 1st ranked performer, UCDCC (Ghosh and Veale 2018), in both tasks. This is primarily attributed to our use of more powerful language models. Even though the concept of transfer learning was established as early as 2014 (Taigman et al. 2014) (Antol et al. 2015), the widespread adoption of large-scale models began around mid-2017. The leading participants in SemEval indeed employed the transfer learning approach, pre-training their models on external datasets. Nevertheless, both the size of their training set and the number of trainable parameters in their structures are not at the same level as the latest language models, such as BERT. For example, the 3rd performer, NTUA-SLP (Baziotis et al. 2018), pretrained their model on the SemEval2017Task4A dataset (Rosenthal, Farra, and Nakov 2019), which consists of only 50,333 tweets. This is notably smaller when contrasted with the total training corpus of around 3.3 billion words used for models like BERT. Larger training datasets and parameters mean greater model capacity or complexity, resulting in the superior performance of our model over the SemEval-2018 participants.

[3] It's important to note that the iSarcamEval corpus lacks hashtag information; thus, we have excluded it from the comparison scenario.

Task A		
Rank	**Teams**	**F1**
1	**BERTweet**	0.813
2	UCDCC	0.724
3	**TwHIN-BERT**	0.723
4	THU-NGN	0.705
5	NTUA-SLP	0.672
6	WLV	0.650

Task B		
Rank	**Teams**	**F1**
1	**BERTweet**	0.550
2	**TwHIN-BERT**	0.536
3	UCDCC	0.507
4	NTU-SLP	0.496
5	THU-NGN	0.495

Table 1: The comparsion between our top performed irony detection models with the leading teams in SemEval-2018 Task 3.

Potential Improvement and Future Work

There are several potential improvements that we did not pursue in the experiment due to constraints in time and hardware. For instance, we opted not to utilize the more recent language model GPT-3 (Brown et al. 2020), given its substantial 175 billion parameters, which present a notable challenge for Google Colab. In terms of linguistic features, there are also several additional aspects to consider, such as misspelled words or negation words like "a loooot of" or "so goood". As for domain-specific data, contemplating the utilization of the Twitter sentiment analysis corpora from past SemEval competitions (Ghosh et al. 2015) (Rosenthal, Farra, and Nakov 2019) is also a possibility. These shortcomings will be deferred for future investigation.

Conclusion

In this paper, we have built upon the achievements of the SemEval-2018 Task3 teams, enriching their studies by integrating additional linguistic features and utilizing more recent language models. We showcased the effects of these modifications on the irony detection task. Through experiments, we highlighted the performance improvements associated with these features (and language models) and offered explanations for the observed results. Additionally, we formulated our own irony detection model, surpassing the performance of the leading systems in SemEval-2018.

References

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Baziotis, C.; Athanasiou, N.; Papalampidi, P.; Kolovou, A.; Paraskevopoulos, G.; Ellinas, N.; and Potamianos, A. 2018. Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns. *arXiv preprint arXiv:1804.06659*.

Bouazizi, M., and Ohtsuki, T. 2015. Sarcasm detection in twitter:" all your products are incredibly amazing!!!"-are they really? In *2015 IEEE global communications conference (GLOBECOM)*, 1–6. IEEE.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems* 33:1877–1901.

Chen, Y.; Yuan, J.; You, Q.; and Luo, J. 2018. Twitter sentiment analysis via bi-sense emoji embedding and attention-based lstm. In *Proceedings of the 26th ACM international conference on Multimedia*, 117–125.

Dong, X. L.; Zhu, Y.; Fu, Z.; Xu, D.; and de Melo, G. 2021. Data augmentation with adversarial training for cross-lingual nli. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 5158–5167.

Farha, I. A.; Oprea, S.; Wilson, S.; and Magdy, W. 2022. Semeval-2022 task 6: isarcasmeval, intended sarcasm detection in english and arabic. In *The 16th International Workshop on Semantic Evaluation 2022*, 802–814. Association for Computational Linguistics.

Farías, D. I. H.; Patti, V.; and Rosso, P. 2016. Irony detection in twitter: The role of affective content. *ACM Transactions on Internet Technology (TOIT)* 16(3):1–24.

Ghosh, A., and Veale, T. 2018. Ironymagnet at semeval-2018 task 3: A siamese network for irony detection in social media. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, 570–575.

Ghosh, A.; Li, G.; Veale, T.; Rosso, P.; Shutova, E.; Barnden, J.; and Reyes, A. 2015. Semeval-2015 task 11: Sentiment analysis of figurative language in twitter. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, 470–478.

González-Ibánez, R.; Muresan, S.; and Wacholder, N. 2011. Identifying sarcasm in twitter: a closer look. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 581–586.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Joshi, A.; Bhattacharyya, P.; and Carman, M. J. 2017. Automatic sarcasm detection: A survey. *ACM Computing Surveys (CSUR)* 50(5):1–22.

Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; and Soricut, R. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Li, B.; Hou, Y.; and Che, W. 2022. Data augmentation approaches in natural language processing: A survey. *Ai Open* 3:71–90.

Liu, C., and Yu, D. 2020. Blcu-nlp at semeval-2020 task 5: Data augmentation for efficient counterfactual detecting. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 633–639.

Nguyen, D. Q.; Vu, T.; and Nguyen, A. T. 2020. Bertweet: A pre-trained language model for english tweets. *arXiv preprint arXiv:2005.10200*.

Reyes, A.; Rosso, P.; and Veale, T. 2013. A multidimensional approach for detecting irony in twitter. *Language resources and evaluation* 47:239–268.

Rohanian, O.; Taslimipoor, S.; Evans, R.; and Mitkov, R. 2018. Wlv at semeval-2018 task 3: Dissecting tweets in search of irony. Association for Computational Linguistics.

Rosenthal, S.; Farra, N.; and Nakov, P. 2019. Semeval-2017 task 4: Sentiment analysis in twitter. *arXiv preprint arXiv:1912.00741*.

Schmidhuber, J. 1989. A local learning algorithm for dynamic feedforward and recurrent networks. *Connection Science* 1(4):403–412.

Shiha, M., and Ayvaz, S. 2017. The effects of emoji in sentiment analysis. *Int. J. Comput. Electr. Eng.(IJCEE.)* 9(1):360–369.

Singh, A.; Blanco, E.; and Jin, W. 2019. Incorporating emoji descriptions improves tweet classification. 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)*, 2096–2101.

Taigman, Y.; Yang, M.; Ranzato, M.; and Wolf, L. 2014. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1701–1708.

Van Hee, C.; Lefever, E.; and Hoste, V. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, 39–50.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems* 30.

Wei, J., and Zou, K. 2019. Eda: Easy data augmentation techniques for boosting performance on text classification tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6382–6388.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al.

2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, 38–45.

Wu, C.; Wu, F.; Wu, S.; Liu, J.; Yuan, Z.; and Huang, Y. 2018. Thu_ngn at semeval-2018 task 3: Tweet irony detection with densely connected lstm and multi-task learning. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, 51–56.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems* 32.

Zhang, X.; Malkov, Y.; Florez, O.; Park, S.; McWilliams, B.; Han, J.; and El-Kishky, A. 2022. Twhin-bert: a socially-enriched pre-trained language model for multilingual tweet representations. *arXiv preprint arXiv:2209.07562*.