# Sarcasm Detection in English and Arabic

**Priya Pande**
Dept. of Computer Science
Stony Brook University

**Aditi Raj Kandoi**
Dept. of Computer Science
Stony Brook University

**Yiyun Yang**
Dept. of Computer Science
Stony Brook University

## Abstract

Sarcasm is a form of linguistic irony that has been broadly adopted in social media. It occurs when there is a discrepancy between the literal and intended meanings of an utterance. The proposed work explores several methods built on the the pre-trained DistilBERT model and the AraBERT model to detect sarcasm in English and Arabic respectively. It further employs a multilingual model to compare the sarcasm detection results for English and Arabic tweets.

## 1 Introduction

In the world of digitalization, people often express their thoughts on social media. For instance, it is estimated that over 500 million tweets are posted on the Twitter platform every single day. The data on these platforms are of great importance as it provides opportunities to businesses for conducting marketing research, information categorization, etc. One of the main challenges in analyzing the user behaviour and emotions through this data is the presence of a figurative language, sarcasm which contrasts the implicit meaning of the sentence.

Consider the following example:
"This is the best laptop bag ever. It is so good that within two months of use, it is worthy of being used as a grocery bag."

The above sentence is a review on a laptop bag purchased by a customer. The innate sarcasm in the review is evident as the user is not happy with the quality of the bag. However, as the sentence contains words like 'best', 'good' and 'worthy', the review can easily be mistaken to be positive.

Recognizing sarcasm is critical for understanding the actual sentiment and meaning of the discourse. The difficulty in the recognition of sarcasm causes misunderstandings in everyday communication. In recent years, there has been a growing trend to address the Sarcasm Detection problem among Natural Language Processing (NLP) researchers. (Kumar et al., 2020) develop a feature rich SVM model that uses the semantic, sentiment and punctuation based hand-crafted features and a multi-head attention-based bidirectional long-short memory (MHA-BiLSTM) network to detect sarcastic comments in a given corpus. (Cai et al., 2019) investigate instilling attribute information from social media posts to propose a model leveraging hierarchical fusion.

Models such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), exploit the use of pretraining and bidirectional transformers to enable efficient solutions obtaining state-of-the-art performance. Pretrained embeddings significantly outperform the previous state-of-the-art in similar problems such as humor detection and subjectivity detection. Two deep learning models are used for sarcasm detection (Babanejad et al., 2020) extending the architecture of BERT by incorporating both effective and contextual features which (ataei et al., 2020) use a bi-directional BERT based model trained on sentiment analysis.

Till date, the sarcasm detection task is extremely complex as it largely dependent on context, prior knowledge and the tone in which the sentence was spoken or written. The objective of this project is to use the pretrained transformer models to detect sarcasm in English and Arabic sentences.

## 2 Sarcasm Detection

This is a binary classification task which has a text based input in the form of tweet. The output is a binary label of 1 if the tweet is sarcastic and 0 for non-sarcastic tweet. Sarcasm is considered like an Achilles' Heel in Sentiment Analysis. Classifying text as sarcastic is even more difficult than audio

or video. This is because the other two forms of input contain features like tone or expressions of face that can be extracted for an accurate classification. In understanding features for text we heavily rely on social media text containing hashtags which may not be accurate. Text from these social media sites can also be very noisy and is difficult to analyse correctly. Thus, feature selection is an important task in text classification for sarcasm. A deeper study into the semantics, hyperbole features and punctuation is required for more precise results.

Several models for sarcasm detection have been presented that incorporate statistical, machine learning, and rule-based approaches but predominantly they utilize simple datasets (Mandal and Mahto , 2019). However, such approaches are not capable of perceiving the figurative meaning of words (Joshi A, 2017). Furthermore, these approaches require handcrafted features and are unable to understand the patterns in passive voice sentences (Bajwa IS, 2006).

(Ghosh et al., 2017) has thoroughly explored the role of conversational context and various LSTM models in sarcasm detection. Prior to this paper, Ghosh et al. has also explored on the topic of sarcasm detection quite a bit, including his previous publication at EMNLP [Debanjan Ghosh and Muresan., 2015], which serves as the baseline for his paper in 2017 (Ghosh et al., 2017).

(Kumar and Garg , 2019) perform sarcasm detection in typo-graphic memes of Instagram posts using the lexical-based supervised techniques integrated with pragmatic and semantic features. (Khatri and Anand, 2020) point out that word embedding elevates the performance of a model higher than the traditional feature extraction techniques. In the light of these findings, several works utilize the pragmatic and lexical features for sarcasm detection.

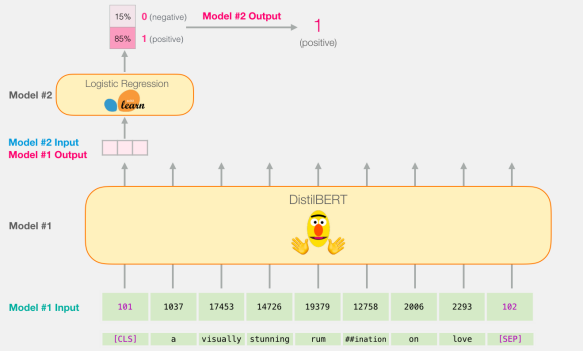
### 2.1 Baseline Model(s)

We have explored five models, two baselines for English, two for Arabic and one Multilingual Language Model.

#### 2.1.1 English Language Models

- **T5-base-fine-tune-sarcasm-twitter**
  The T5 model was presented in the paper (Raffel et al., 2020). This model explores the landscape of transfer learning techniques for NLP by introducing a unified framework that converts every language problem into a text-to-text format. It has done a systematic study by comparing pre-training objectives, architectures, unlabeled datasets, transfer approaches, and other factors on dozens of language understanding tasks. By combining the insights from exploration with scale and new "Colossal Clean Crawled Corpus", it achieves state-of-the-art results on many benchmarks covering summarization, question answering, text classification, and more.

Figure 1: DistilBERT Transformer Model



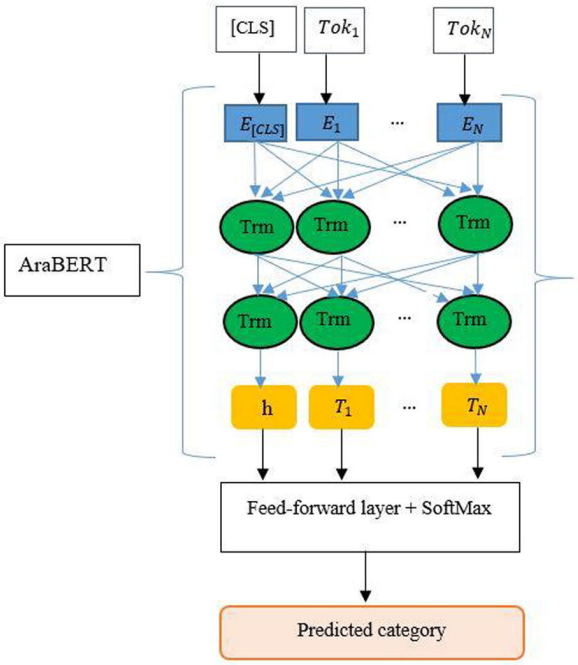

- **DistilBERT-base-uncased**
  The DistilBERT model was proposed in the paper (Sanh et al., 2020). DistilBERT is a small, fast, cheap and light Transformer model trained by distilling BERT base. It has 40% less parameters than bert-base-uncased, runs 60% faster while preserving over 95% of BERT's performances as measured on the GLUE language understanding benchmark.

#### 2.1.2 Arabic Language Models

- **AraBERT**
  It is an Arabic pretrained language model based on Google's BERT architechture described in this paper(Antoun et al., 2020). It uses the same BERT-Base config, and segments the words into stems, prefixes and suffixes. That is because Arabic words can have different forms and share the same meaning, hence, when using a BERT-compatible tokenization, tokens will appear twice then. So they trained a SentencePiece (an unsupervised text tokenizer and detokenizer (Kudo, 2018)) in unigram mode. See Figure 2.

- **AraBERT-v0.1**
  Based on the AraBERT model above, we

Figure 2: AraBERT Transformer Model (zahra El-Alami et al., 2021)




also use another model named AraBERTv0.1 (Antoun et al., 2020) that was trained on non-segmented text that does not require any segmentation. In this way can we do a comparison on the impact of segmentation on Arabic language.

#### 2.1.3 Multilingual Language Models

- **BERT multilingual base model**
  We also use multilingual model on both English and Arabic tasks. This model was introduced in paper (Devlin et al., 2018). It was pretrained on the 104 languages with the largest Wikipedias and was learned on two objectives: Masked language modeling (MLM) and Next sentence prediction (NSP). So this model learns an inner representation of the languages, which can be used to classify the sarcastic sentence.

### 2.2 The Issues

- **Low accuracy**
  The use of baseline models used on the original English dataset give a very low accuracy from 0.59 to 0.73.

- **Overfitting**
  High capacity of Language Models makes the baseline models prone to overfitting the noisy labels generated by weak supervision. This indeed occurs during our fine-tuning session. Possible resolution includes Contrastive Self-training with All Data, Contrastive Learning on Sample Pairs, etc (Yu et al., 2020).

- **Large Memory Overhead**
  While we use baseline models for training, we can often come into an error of "RuntimeError: CUDA out of memory". That is because BERT base model has large amount of parameters and others that requires lots of memories. We address this by applying smaller batch size and lighter models of BERT.

## 3 Approach

In the proposed work, several approaches have been carried out to compare the results and find out the best model for sarcasm detection task.

### 3.1 Pragmatic and Lexical Features

To detect sarcasm, understanding the sentiments of the sentence is very important as it highlights the behaviour and the emotion of the user while typing the tweet. The emotions are often expressed by using different emojis and emoticons. Hence, as the first step of the approach, we use the emoji package to replace the emojis with the expression they represent and build a custom dictionary to replace the emoticons with their intended emotion. Additionally, the irrelevant content of the data such as hashtags, URLS and mentions are also removed using re and BeautifulSoup packages.

### 3.2 Contextual incongruity

Computational models for sarcasm detection have often relied on the content of utterances in isolation. However, sarcastic intent is not always obvious without additional context. Therefore, to make our models learn the important context in the tweets more efficiently, we use the additional feature in the dataset which consists of the paraphrase of the sarcastic tweets written by the authors of the tweet.

### 3.3 Segmentation

Segmentation is a construction based on tokens. Tokens are not enough for language understand-

ing because it is common in many language that a word's meaning can only be completely delivered under several tokens' combination, such as "New York". We use the AraBERT models with and without segmentation to see if it helps in understanding the meaning of the language better.

## 4 Evaluation

We are building a system that should be able to identify features in text such that it can be classified as sarcastic. The evaluation of our system is based on how good our model does on the above task. Is it accurately able to detect semantic and contextual hints of sarcasm? Or, does it cheat based on some ambiguous features it has learnt to classify as sarcastic. In such a scenario, the model has learned on the data and not the sarcasm detection task.

For evaluating our system, one should input a set of sarcastic/non-sarcastic tweet not seen in our dataset and see if the model is able to make a reasonable prediction on it.

### 4.1 Dataset Details

SemEval 2022 is a new dataset consisting of approximately 3500 instances that has been released with more accurate labelling of sarcastic and non sarcastic tweets. The labelling is done by the author of the tweets themselves. We plan to submit our fine tuned model and check performance on this new dataset. This is a rich dataset as it contains paraphrased text of the sarcastic tweet which could help in training differences in sarcasm vs direct speech.

This task also contains a similar dataset for Sarcasm detection in Arabic. We have extended our approach to a multilingual model and tested the accuracy of the model on the Arabic dataset.

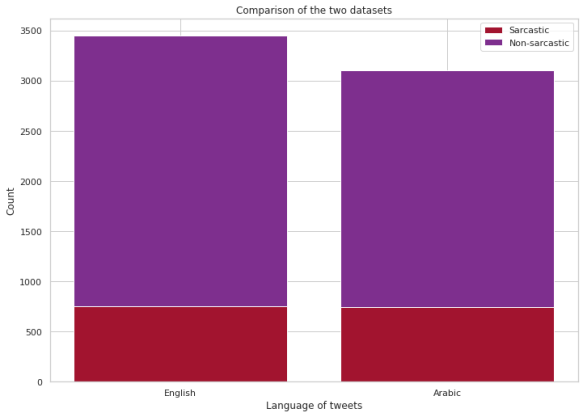
### 4.2 Evaluation Measures

We are using the following quantitative evaluation methods.

- **F1 Score** - The F1 score is the harmonic mean of the precision and recall. The highest possible value of an F-score is 1.0, indicating perfect precision and recall, and the lowest possible value is 0, if either the precision or the recall is zero.

- **Confusion Matrix** - It is also known as an error matrix, and is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one. It tells us calculate percentage of true positives and false negatives predicted for each class.

Figure 3: Class Distribution of Tweets




- **Weighted Accuracy** - This is computed by taking the average, over all the classes, of the fraction of correct predictions in this class

### 4.3 Baselines

Based on our understanding of sarcastic text in the data we wanted to finetune it for the below scenarios and analyse the results.

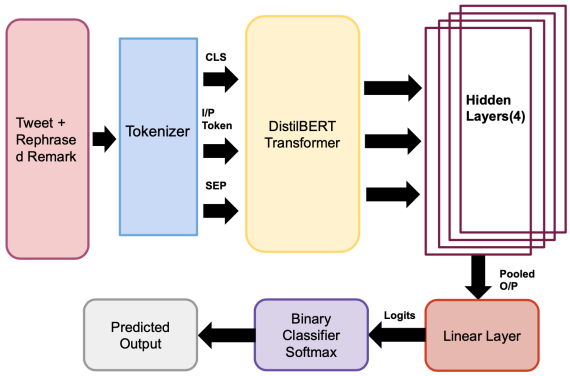
#### 4.3.1 DistilBERT base model + Layers of features + Contextual Information

As the first step, information on the pragmatic features of the data was extracted by using the emoji package. The irrelevant content of the data such as hashtags, URLS and mentions were removed using re and BeautifulSoup. The DistilBERT model was then trained on the preprocessed data consisting of the tweets and the paraphrased tweets sequentially by setting the default number of layers to 6, batch size to 32 and epochs to 3.

#### 4.3.2 DistilBERT base model + Hidden Layers + Layers of features + Randomization of input features

To address the issue of class imbalance, a stratified K fold cross validation technique was used. We trained the model on a shuffled input concatenation of rephrase and tweet and have evaluated the model only on the tweets. Additionally, we concatenated the output of the last four hidden layers and passed it to a linear layer before classification.

Figure 4: DistilBERT Transformer Model




As a result, the performance of the model was reasonably good and worked well on other similar datasets with the same task.

#### 4.3.3 AraBERT baseline model + Segmentation

For Arabic language, we first use the embbeded preprocessing functions inside AraBERTv0.2-Twitter-base(Antoun) model, but we do not use this for training because it was pre-trained on a very large corpus that requires a large amount of memories. We then divide the data into multiple folds for cross-validation during training. We use AdamW (a Weight Decay Regularization in Adam (Loshchilov and Hutter, 2017)) as the optimizer for faster convergence. We also apply the step-decay learning rate(Ge et al., 2019) as well as the half-precision floating point (FP16)(Kalamkar et al., 2019) for improving the training performance. All off the approaches mentioned above are easily implemented through a BERT based framework FastBert (Utterworks, 2019), which enables adjustable parameters such as epochs, learning rate, batch size and others, and the saving of the model afterwards.

#### 4.3.4 Comparison of BERT multilingual model on English and Arabic datasets

Since the BERT multilingual model was pre-trained on a couple of languages including English and Arabic, we experimented on the English and Arabic datasets to compare their results. The preprocessing techniques were similar to the above models for both the datasets. The model setup was such that the batch size was 16, maximum sequence length was 512, learning rate was 0.00001 and the number of epochs for which the model was trained was 5.

### 4.4 Results

Table 1 contains the accuracy of all the models and experiments conducted. There are some interesting things happening in some of the results. We see that on fine-tuning DistilBERT it overfits on our data and gives shockingly good results. The concatenated hidden layers approach does not work as well as we have expected giving a low accuracy of 0.55. The best performing model for us is the MBERT model which outperforms the existing twitter benchmarks on both English and Arabic.

### 4.5 Analysis

Our analysis is divided into 2 portions based on the approaches we tried. After establishing a baseline result of 0.73 in English on DistilBERT, we fine-tuned the model. We also tried to tweak the DistilBERT output layers as shown in architecture diagram in 5

1. The fine-tuned model on 5 epochs and with context information passed to it performed exceptionally well giving an accuracy of 0.98 on the evaluation dataset. This seemed too good to be true, so ran it on another similar dataset consisting of tweets and its labels. The performance of the model degraded as it gave an F1-score of 0.05 for the label of sarcasm. From the above analysis, we deduced that our tuned model overfitted on the dataset as it was learning the sarcastic tweet and its paraphrased sentence sequentially. In order to resolve this issue, instead of passing the tweet and its context sequentially, it was shuffled and used as an input parameter for training the model.

2. After retraining the model on shuffled input, we tried to modify the default architecture and pooled the output of the last 4 hidden layers. This strategy worked the best in text classification as mentioned in the BERT paper. It did not do that well for us giving a lower accuracy of 0.55. On further analysis, we plotted the validation and training loss for. We can see that the model did not generalise well on the sarcastic tweet. It was doing very well when the paraphrase was being passed to it during training. But, on evaluating against

Table 1: Baseline Comparison with Benchmarked results

| Model | Accuracy | Benchmarks |
|---|---|---|
| T5 base Twitter Fine-tuned(Baseline) | 0.59 | |
| DistilBERT base(Baseline) | 0.73 | |
| DistilBERT base(hyperparameter tuning) | 0.98 | |
| DistilBERT(Tuned)SemEval 2018 | **0.6** | **0.725** |
| DistilBERT(concat) | **0.55** | |
| AraBERTv0.1-base (Baseline) | 0.85 | 0.57 |
| AraBERTv1-base (Baseline) | 0.89 | 0.57 |
| MBERT Arabic | 0.82 | 0.57 |
| MBERT English | 0.75 | 0.725 |

the tweet it gave a higher loss thus proving that it did not learn the contextual features for sarcasm we intended it to.

3. With more epochs training on the baseline models during Arabic Language training, accuracy gradually drop for all of the 3 models at around epoch 3-4. This is possible because fine tuning on one language may break the existing relationship of cross-language word embedding (Wu, 2020).

4. AraBERT model with Segmentation gives an 0.89 accuracy against the 0.85 that without. But Multilingual Model gives the 0.82 among all, which proves segmentation is effective in understanding Arabic language, and multilingual also takes effect but not as well as the above two models.

### 4.6 Code

The code, dataset and models has been uploaded on Google Drive. More details can be found in the README file.

## 5 Conclusions

The aim of the project was to detect sarcasm in English and Arabic sentences. In the paper, we performed two main experiments. Firstly, we tried to improve sarcasm classification with different

Figure 5: Loss evaluated by the model

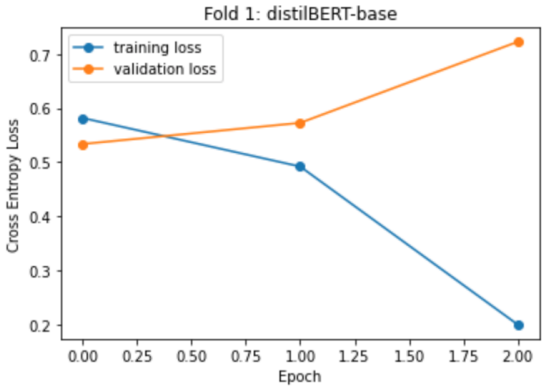
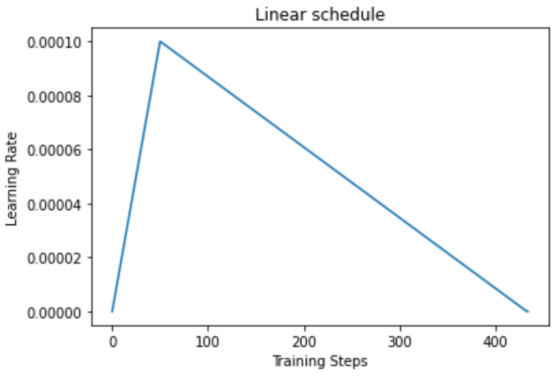



Figure 6: Learning Rate vs Training steps



BERT models with and without the contextual information and analysed the performances. Secondly, we tried to build towards a multilingual model that can support sarcasm detection in any language. Based on the results of the experiment, we concluded that modelling contextual incongruity is not as straightforward. The model easily learns the wrong cues and overfits on the data. As a next step, we plan to explore the role of weighing positional embeddings and adding NER features to the text with the intention to make the model learn well on sarcasm detection as a task and not just understand datasets.

## References


Wissam Antoun. Aubmindlab/bert-base-arabertv02-twitter · hugging face.

Wissam Antoun, Fady Baly, and Hazem M. Hajj. 2020. Arabert: Transformer-based model for arabic language understanding. *CoRR*, abs/2003.00104.

Taha Shangipour ataei, Soroush Javdan, and Behrouz Minaei-Bidgoli. 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. Association for Computational Linguistics.

Nastaran Babanejad, Heidar Davoudi, Aijun An, and Manos Papagelis. 2020. Affective and contextual embedding for sarcasm detection. pages 225–243.

Choudhary MA Bajwa IS. 2006. A rule based system for speech language context understanding.

Yitao Cai, Huiyu Cai, and Xiaojun Wan. 2019. Multi-modal sarcasm detection in Twitter with hierarchical fusion model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2506–2515, Florence, Italy. 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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Fatima zahra El-Alami, Said Ouatik El Alaoui, and Noureddine En Nahnahi. 2021. Contextual semantic embeddings based on fine-tuned arabert model for arabic text multi-class categorization. *Journal of King Saud University - Computer and Information Sciences*.

Rong Ge, Sham M. Kakade, Rahul Kidambi, and Praneeth Netrapalli. 2019. The step decay schedule: A near optimal, geometrically decaying learning rate procedure. *CoRR*, abs/1904.12838.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions.

Carman MJ Joshi A, Bhattacharyya P. 2017. Automatic sarcasm detection: a survey.

Dhiraj D. Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. 2019. A study of BFLOAT16 for deep learning training. *CoRR*, abs/1905.12322.

Pranav Khatri and Anand. 2020. Sarcasm detection in tweets with sbert and glove embeddings. arXiv preprint.

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates.

Kumar and Garg G. Garg (2019) Kumar A. 2019. Sarcm: sarcasm detection in typo-graphic memes. In *International Conference on Advances in Engineering Science Management & Technology (ICAESMT)-2019; Uttaranchal University, Dehradun, India*. Association for Computational Linguistics.

Avinash Kumar, Vishnu Teja Narapareddy, Aditya Srikanth Veerubhotla, Aruna Malapati, and Lalita Bhanu Murthy Neti. 2020. Sarcasm detection using multi-head attention based bidirectional lstm. *IEEE Access*, 8:6388–6397.

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.

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

Mandal and Mahto R Mahto (2019) Mandal PK. Deep cnn-lstm with word embeddings for news headline sarcasm detection.

Colin Raffel, Noam Shazeer, Katherine Lee Adam Roberts, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

U. Utterworks. 2019. Utterworks/fast-bert: Super easy library for bert based nlp models.

Yuehao Wu. 2020. Multilingual customized bert for zero shot sequence classification.

Yue Yu, Simiao Zuo, Haoming Jiang, Wendi Ren, Tuo Zhao, and Chao Zhang. 2020. Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach. *CoRR*, abs/2010.07835.