

Identifying the Type of Sarcasm in Dravidian Languages using Deep-Learning Models

Ramya Sivakumar^{1,†}, C.Jerin Mahibha² and B.Monica Jenefer³

,2,3¹Meenakshi Sundararajan Engineering College Chennai-24

Abstract

Sarcasm is mainly described as a word that is a synonym for irony. However, it has a more specific meaning: It is commonly used in the context of mocking or conveying contempt. It is considered essential to detect sarcasm in any text on social media because it identifies and conveys the exact meaning of the word and the expected meaning. The Sentiment Analysis System automatically checks the polarity of content, but it doesn't take into account the impact of sarcastic statements. If the system gets it wrong, it won't work as well. So, if it can automatically recognize sarcastic statements from social network data, it can make the Sentiment Analysis System and other NLP-based apps better. In this shared task, we have used the ALBERT transformer model to detect and classify the given text as sarcasm and not sarcasm. Using this model to train and predict the data, we were able to achieve macro F1 scores of 0.48 and 0.34 for the Tamil dataset and Malayalam dataset, respectively.

Keywords

Sarcasm, ALBERT, Classification, Dravidian languages, Deep learning,

1. Introduction

Sarcasm is a word derived from the Greek verb "Sark'azein," which means to speak bitterly. These words are often used in a humorous way to mock people [1]. It is very easy to find sarcasm when one is having face-to-face communication. It can be identified either using facial expressions or tone of speech. However, this is not the case when we are involved in textual communication. For instance, among all the social media platforms, YouTube is one where a majority of people tend to share content on any topic as videos. At the same time, everyone has the access and privilege to comment on the videos that are being posted [2]. In general, detecting sarcasm from textual content itself is a more challenging task, and the freedom to comment in any preferred language and manner on YouTube makes the task even more challenging [3]. Basically, if something is said like, "Oh yes, you've been so helpful; thank you so much for your help" and then followed up with a smiley face, it's easy to tell it's not sarcastic. But if the message is accompanied by an angry or frustrated look, it's a sign that the person is trying to be sarcastic.

Amidst all the difficulties and challenges, researchers still try to come up with new ways to detect sarcasm for various reasons, like to improve communication, because sarcasm often

Forum for Information Retrieval Evaluation, December 15-18, 2023, India

*Ramya Sivakumar

✉ ramyacsemsec@gmail.com (R. Sivakumar); jerinmahibha@gmail.com (C.Jerin Mahibha);

monicamaheswaran@gmail.com (B.Monica Jenefer)



© 2023 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

 CEUR Workshop Proceedings (CEUR-WS.org)

leads to miscommunication as the intended meaning of that situation is different from the actual meaning of the word that is used. It also helps with sentiment analysis tasks that help MNCs and other big companies analyze their product reviews and work accordingly. With the increasing use of social media, these detection tasks also help in improving cyber security and wellness [4]. The importance of sarcasm detection also plays a major role in the development of AI, as it can improve the performance of AI by providing more contextually relevant content. With all these considerations the shared task on sarcasm detection has been introduced [5] as a part of FIRE 2023.

2. Related Works

Sharma et al. [6] has proposed a hybrid model for the detection of sarcasm in a given dataset. This hybrid model mainly comprises three subordinates, namely: BERT, USE, and Autoencoder. This hybrid algorithm has been implemented on SARC, Twitter, and Headlines datasets and has achieved an average accuracy of around 90 percent. Meriem et al. [7] had come up with a fuzzy approach to solve the task. This approach focuses on predicting the right label based on a measure known as the Sarcasm Score Measure, which calculates the measure of sarcasm based on which the prediction is made. This model has been implemented on two datasets: one is SemEval2014, and the other is the Bamman et al. dataset. This resulted in an F1-score of 75.9 and 74.8 percent, respectively. Both binary and multi-class classifications were performed in this task. Sundararajan and Palanisamy [8] had come up with a probabilistic model that helped in the prediction of sarcastic texts. The whole model had worked with the help of both the probabilistic model and the CNN (convolutional neural network). The confidence level that is obtained as an output from the probabilistic model was later fed into the CNN for the actual prediction. Tweets collected from the Tweet API were used as the data for implementation. This had an accuracy of 97.25 percent. Vinoth and Prabhavathy [9] had presented a model named IMLB-SDC, which is the intelligence machine learning sarcasm detection and classification. This proposed data model, besides the text processing and feature extraction methods, also used the SVM (Support Vector Machine) and penalty factor to enhance its performance. Govindan and Balakrishnan [10] had created a data model called the hyperbole-based Sarcasm detection model (HbSD). Here, the paper examined negative sentiment tweets that contain hyperbole for sarcasm detection tasks. This data model has been implemented on the Streaming Twitter API. 78.74% accuracy and 0.71 F1 score were achieved when the HbSD model was used. Kalaivani and Thenmozhi [11] had performed sentiment analysis on the Dravidian-CodeMix-FIRE2021 dataset, where comments in 3 languages were handled: Tamil, Malayalam, and Kannada. They had used the pre-defined BERT model with the ktrain library to perform this task. The main idea was to work with and analyze comments from YouTube. As a result, they were able to achieve macro F1 scores of 0.47, 0.64, and 0.48, respectively. The task of humor detection had been carried out by training the dataset with different transformer models like Multilingual BERT, Multilingual DistilBERT, and XLM-RoBERTa, and all the results were compared by Bellamkonda et al. [12]. Among these, XLM-RoBERTa was found to perform best with an F1-score of 0.82 and 81.5% accuracy. The model experimentation dataset had been formed by scrapping tweets from Twitter and filtering specific tags. Traditional machine learning models had been used to detect sarcasm

in the Ben-Sarc corpus [13]. Models used in this task include Logistic Regression, Decision Tree, Random Forest, Multinomial Naive Bayes, K-Nearest Neighbors, Linear Support Vector Machine, and Kernel SVM. At the end of this task, the BERT model had attained the maximum accuracy of 75.05 percent and the second highest accuracy by the LSTM model of 72.48 percent, followed by 72.36 percent. The use of a bidirectional dual encoder with Additive Margin Softmax to perform offensive language classification tasks had been proposed by Mahibha et al. [14], which resulted in an F1 score of 0.865.

3. Dataset

The task of sarcasm detection was implemented based on two Dravidian languages, namely Tamil and Malayalam. Separate datasets in code-mixed Tamil-English and Malayalam-English were provided by the task organizers for carrying out the task of sarcasm detection. The text in the dataset was represented in Roman and native scripts. Text and label information were provided for each of the instances in the dataset. Text is the actual comment that was posted on social media, and labels define the two main sub-categories in which the comments are grouped, which are sarcastic and non-sarcastic. The training dataset is first fed to the proposed deep learning model. The model uses the data to learn so that it can be used for the purpose of prediction. Later, the model is fed with the validation dataset for further training. This is commonly known as the development dataset. After the training process, the model is fed with instances of the development data, using which it fine-tunes the parameters to increase the accuracy of the results. The last phase of the task involves the use of a test dataset that contains only the text for which the corresponding labels have to be predicted using the trained model. The number of instances under each category of the different datasets is shown in Table 1. The training dataset for Tamil and Malayalam languages had 27036 and 12057

Table 1
Data Distribution

Tasks	Labels	Training Dataset	Validation Dataset
Tamil Dataset	Sarcastic	7170	1820
	Non-Sarcastic	19866	4939
Malayalam Dataset	Sarcastic	2259	588
	Non-Sarcastic	9798	2427

samples, respectively. Similarly, the validation dataset had 6759 and 3015 instances in Tamil and Malayalam, respectively, and the test dataset of the Malayalam language contained 3768 comments and the Tamil language contained 8449 comments, for which the labels had to be predicted.

4. System Description

Given a dataset, text classification and prediction are implemented in a sequence of steps. Initially, the training and validation datasets are fed into the system for pre-processing. Various

pre-processing techniques, including tokenization, stemming, lemmatization, and the removal of stop words, are implemented, which help in gaining a more accurate result.

The next process is data encoding. The transformer model accepts data in numerical format. Hence, in order to feed the data into the model, the cleaned data is further encoded into numerical data. These encoded data are mapped to the existing words and index values in the model's vocabulary.

Following this, model selection is done, where the suitable version of the model is chosen to implement the process of classification. The proposed system uses the ALBERT [15] model for the process of implementation.

The process of tokenizing data is carried out by the proposed model to satisfy its requirements. Some of the main categories of classifications include segregating the text as classifiers, separators, etc. It is important to ensure that all the tokens generated are of the same length; padding of data needs to be done to rectify the same. Input formatting is also done on the input data. ALBERT models accept the input data to be in the format of segment ID, followed by the attention masks. Segment ID is responsible for differentiating between the sentence pairs, and attention masks indicate to the model the set of tokens that need attention. Hence, it is necessary that the input data be in this format. Fine-tuning helps the model make predictions based on the encoded input data, which is followed by optimization. The ALBERT model reduces loss and optimizes the output using SOP (sentence order prediction), which reduces loss by avoiding topic prediction. The next significant step in the process is that the model is trained using the dataset, and evaluation is done. Now the model is trained using the development dataset, and the model's performance is noted. Based on the inference, changes to hyperparameters can be made to achieve better results. As a result of the training process, the model is now made to predict the labels for the instances of the validation dataset, and the comparison of output is done. The architecture of the proposed model is represented by Figure 1.

Finally, the test data which is the new unseen data is fed into the model and labels are generated for the data. Compared to other BERT models we have used the ALBERT model for classification purposes as it supports Indian languages and classifies text in an efficient way by parameter sharing which reduces overfitting and computation is done faster. This model is also highly scalable in nature which makes it versatile.

4.1. ALBERT

ALBERT [15] stands for "A Lite BERT" as it is extracted from the BERT model. BERT (Bi-directional Encoder Representations from Transformers) is a transformer model that uses transformer encoders to process the given input data. Both BERT and ALBERT models use the same backbone architecture represented by Figure 2. The advantages of using ALBERT over BERT are that its computational speed is fast, and it is also stated that ALBERT performs better even with a smaller number of parameters, unlike BERT. The number of parameters is reduced by the parameter sharing method and the factorization of the embedding matrix. Using this method, the embeddings generated are split into two matrices. Input-level embeddings will have all the embeddings that will process the context-independent learning. Similarly, high-level embeddings are responsible for context-dependent learning. ALBERT is a supervised learning model, meaning it learns from the given input dataset and trains the model based on

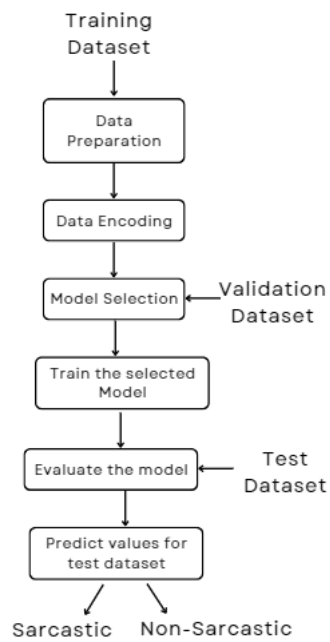


Figure 1: Architecture of Proposed Model

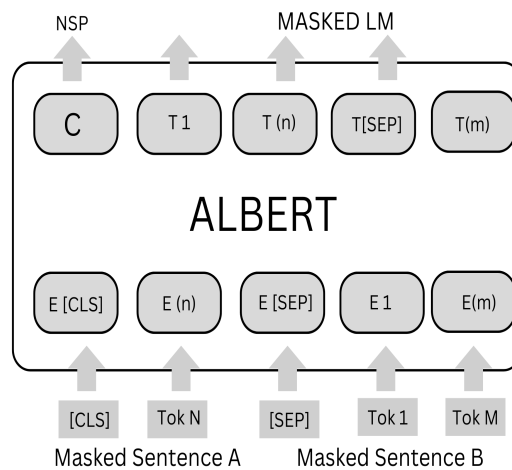


Figure 2: Albert Model Architecture

its learning. Albert uses masked language models to train the data. This model makes use of the self-supervised sentence order prediction loss to find out the inter-sentence relations in the given input data.

ALBERT is a pre-trained model, and hence performing operations is made easy using the TensorFlow hub.

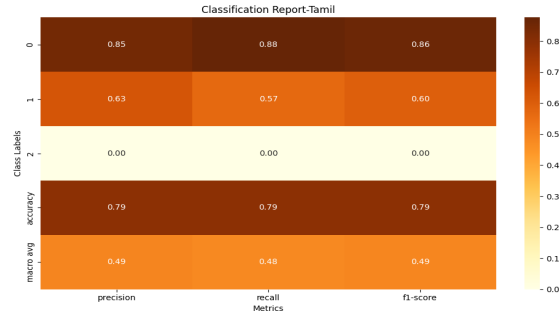


Figure 3: Classification Report- Tamil

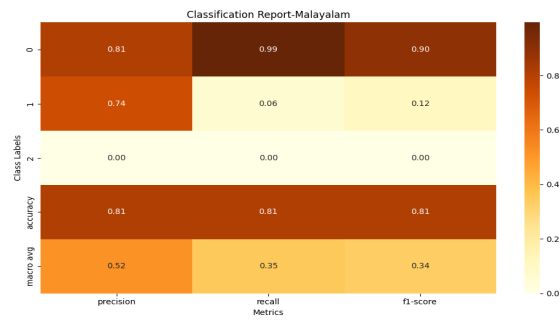


Figure 4: Classification Report- Malayalam

5. Results

Table 2
Performance Metrics

Tasks	Model	F1-Score	Accuracy	Weighted Average
Tamil Dataset	ALBERT	0.48	0.79	0.79
Malayalam Dataset	ALBERT	0.34	0.81	0.74

Table 2 shows the results of the Sarcasm detection task that was carried out. Using the proposed model, we were able to predict the labels for the comments given in the dataset. It yielded an accuracy of 0.81 and 0.79 for the Tamil and Malayalam datasets, respectively. It could be seen that out of 8452 comments, 1883 comments are sarcastic and 6567 comments are non-sarcastic in the Tamil dataset. Similarly, in the Malayalam dataset, out of 3768 comments, 69 are sarcastic and 3699 are non-sarcastic. Macro-F1 scores of 0.48 and 0.34 were also achieved in the Tamil and Malayalam datasets, respectively. The classification report obtained for Tamil and Malayalam are represented by heatmaps in the Figure 3 and Figure 4

6. Error Analysis

While comparing the predicted labels obtained using the proposed model and the actual labels for each instance of the dataset, it was found that there are both false positive and false negative values. This can be further witnessed with the F1-score that is obtained during the process.

The reasons for the error in the predictions could be due to the absence of any sarcastic word; thus, it is classified as "non-sarcastic" instead of the appropriate label of "sarcastic". Another reason could be that few texts do not have words in them but rather just symbols; hence, it is difficult to predict the correct label. Hence, such texts are classified as non-sarcastic instead of the actual label "sarcastic". All the example texts demonstrate sarcasm and play a significant role in the classification process. Sample text instances that are misclassified are represented in tables 3 and 4

Table 3
Sample Predictions-Tamil

S.No	Text	Predicted Label	Actual Label
1	Thala Thala tha tamil in beggast flim	Non-sarcastic	Sarcastic
2	one name... Vetri Maaran all kerala vetrimaaran fans like here...	Non-sarcastic	Sarcastic
3	Oruthar mela neenga viswasam kata.. Inoruthara neenga asingam paduthuringa.... Deep	Sarcastic	Non-sarcastic
4	Final la rajini ya vachi senjitanga	Non-Sarcastic	Sarcastic
5	August 8 tamilnadu fulla terika vidrom.....	Non-Sarcastic	Sarcastic

Table 4
Sample Predictions-Malayalam

S.No	Text	Predicted Label	Actual Label
1	Atom bomb, tsunami, volcano eruption oke athi jeevichadallejapan.Last dialogue seriyayilla	Non-sarcastic	Sarcastic
2	Nalla oombiya pattu... Ini padam irangumbo mammottiye kayaril ketti pokkunnathum kanam...	Sarcastic	Non-sarcastic
3	njan oru katta ettan fan pakshe ikkayude ee unda pollikkum	Non-sarcastic	Sarcastic
4	Kottayam to paala daily kettuuu vattanu e songgg my addicted song	Non-Sarcastic	Sarcastic
5	Dislike full odiyan kanji fans,	Non-Sarcastic	Sarcastic

7. Conclusion

The way people communicate online is getting more and more complicated. So traditional methods like feature-based or machine-learning-based methods won't work if you're trying to detect sarcasm. It's important to differentiate between sarcastic and non-sarcastic text when it

comes to online content. Trying to detect sarcasm by looking at things like language, sentiment, and syntax can give people the wrong idea. Context and semantic information are key when it comes to spotting sarcasm. We want to make our work better in the future by using a bigger dataset for training. Plus, emojis and emoticons are really important for showing what a comment means on social media, so we'll think about adding them to the text.

References

- [1] A. Reyes, P. Rosso, T. Veale, A multidimensional approach for detecting irony in twitter, *Language resources and evaluation* 47 (2013) 239–268.
- [2] M. Birjali, M. Kasri, A. Beni-Hssane, A comprehensive survey on sentiment analysis: Approaches, challenges and trends, *Knowledge-Based Systems* 226 (2021) 107134.
- [3] C. J. Mahibha, S. Kayalvizhi, D. Thenmozhi, Sentiment analysis using cross lingual word embedding model (2021).
- [4] T. P. Nagarhalli, V. Vaze, N. Rana, Impact of machine learning in natural language processing: A review, in: *2021 third international conference on intelligent communication technologies and virtual mobile networks (ICICV)*, IEEE, 2021, pp. 1529–1534.
- [5] B. R. Chakravarthi, N. Sripriya, B. Bharathi, K. Nandhini, S. Chinnaudayar Navaneethakrishnan, T. Durairaj, R. Ponnusamy, P. K. Kumaresan, K. K. Ponnusamy, C. Rajkumar, Overview of the shared task on sarcasm identification of Dravidian languages (Malayalam and Tamil) in DravidianCodeMix, in: *Forum of Information Retrieval and Evaluation FIRE - 2023*, 2023.
- [6] D. K. Sharma, B. Singh, S. Agarwal, H. Kim, R. Sharma, Sarcasm detection over social media platforms using hybrid auto-encoder-based model, *Electronics* 11 (2022) 2844.
- [7] A. B. Meriem, L. Hlaoua, L. B. Romdhane, A fuzzy approach for sarcasm detection in social networks, *Procedia Computer Science* 192 (2021) 602–611.
- [8] K. Sundararajan, A. Palanisamy, Probabilistic model based context augmented deep learning approach for sarcasm detection in social media, *Int. J. Adv. Sci. Technol* 29 (2020) 8461–79.
- [9] D. Vinoth, P. Prabhavathy, An intelligent machine learning-based sarcasm detection and classification model on social networks, *The Journal of Supercomputing* 78 (2022) 10575–10594.
- [10] V. Govindan, V. Balakrishnan, A machine learning approach in analysing the effect of hyperboles using negative sentiment tweets for sarcasm detection, *Journal of King Saud University-Computer and Information Sciences* 34 (2022) 5110–5120.
- [11] A. Kalaivani, D. Thenmozhi, Multilingual sentiment analysis in tamil malayalam and kannada code-mixed social media posts using mbert., in: *FIRE (Working Notes)*, 2021, pp. 1020–1028.
- [12] S. Bellamkonda, M. Lohakare, S. Patel, A dataset for detecting humor in Telugu social media text, in: *Proceedings of the Second Workshop on Speech and Language Technologies for Dravidian Languages*, Association for Computational Linguistics, Dublin, Ireland, 2022, pp. 9–14. URL: <https://aclanthology.org/2022.dravidianlangtech-1.2>. doi:10.18653/v1/2022.dravidianlangtech-1.2.

- [13] S. K. Lora, G. Shahariar, T. Nazmin, N. N. Rahman, R. Rahman, M. Bhuiyan, et al., Ben-sarc: A corpus for sarcasm detection from bengali social media comments and its baseline evaluation (2022).
- [14] J. Mahibha, S. Kayalvizhi, D. Thenmozhi, Offensive language identification using machine learning and deep learning techniques (2021).
- [15] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: A lite bert for self-supervised learning of language representations, arXiv preprint arXiv:1909.11942 (2019).