Journal of Computer Science
https://dergipark.org.tr/en/pub/bbd
http://doi.org/10.53070/bbd.990890

Journal of Computer Science
ISSN: 2548-1304
Volume: IDAP-2021, Issue: Special , pp: 126-131
@Anatolian Science

Sarcastic Text Detection Using Keras

Abdullah Erhan Akkaya*[1]

[1]Department of Computer Engineering, Inonu University, Malatya, Turkey

(erhan.akkaya@inonu.edu.tr)

Received: Sep.3, 2021 Accepted: Sep.20, 2021 Published: Oct.20, 2021

Abstract— Sarcastic text is a type of text that contains a kind of irony, in which negative expressions are expressed as positive by attributing meanings to words that contradict their real meanings during communication. During face-to-face communication, changes in tone of voice, body language, eye contact or word stress make it easier for the other person to detect the sarcastic expression. However, it is challenging to detect sarcastic expressions only through text in machine learning-based systems since human-centred qualities cannot be transferred. Newspapers often use sarcastic expressions in their headlines to attract people's attention. However, many people cannot fully understand whether these expressions are sarcastic or not without reading the content of the text. As a result, they can transmit false information to the people around them through direct communication or social media. In this study, to prevent such misinformation, newspaper headlines with and without sarcasm are tried to be classified on three different GPUs with deep learning methods. As a result, the developed model can successfully detect news headlines containing sarcasm.

Keywords: *Sarcastic text detection, Deep learning, Sentiment analysis.*

1. Introduction

People are in constant interaction through communication to meet their daily needs, solve problems, and present their ideas. While communicating, many positive or negative expressions are conveyed to the other person or people with body language, looks, tone of voice and word stress. As a result of this transfer, the listener can determine whether the opposite expression is positive or negative and whether it is contradicted in a real mean based on human characteristics. The use of an expression that contradicts its true meaning is called sarcasm (Tay et al., 2018). In the dictionary of the Turkish Language Association, sarcasm is defined as “a word that tells what is thought indirectly”, “an implicit, reproachful, poignant word”, and “the art of using a word beyond its real meaning” (Dictionary of Turkish Language Association, n.d.). In the Oxford Lexico dictionary, sarcasm is defined as: “the use of irony to mock or convey contempt.” (SARCASM | Definition of SARCASM by Oxford Dictionary on Lexico.Com Also Meaning of SARCASM, n.d.).

In recent years, the use of social media platforms such as Twitter and Facebook has increased rapidly. Through such social media platforms, people have the freedom and privilege to express their spiritual and mental states, perspectives on events through text and images, anytime and anywhere. As a result, millions of messages are sent in just one day. Natural language processing techniques are applied to analyse these social media messages sent in different languages from different locations of the world.

In many virtual communication platforms (such as Twitter, blogs, and news sites), it is important to determine what people really imply, the subject title and the harmony in its content to analyse what the masses think. Concerning the increasing comments, especially in social media, the detection of sarcastic messages has become increasingly important. The vast majority of studies in this area focused on sarcasm in the texts.

It is difficult to detect sarcasm through text alone without humanoid features. Only text-based analyzes can produce false output. From this point of view, sarcasm is related to the context and other humanoid features. As an example of sarcasm, the bored reader reading this article might say, “It is a wonderful feeling to spend my precious hours reading this article.” The word "wonderful", which is actually positive here, is sarcastically emphasized by the reader, causing a sarcastic sentence that people can easily perceive. The goal of this paper is to propose a system to detect sarcasm in the text.

2. Related Work

In recent years, deep learning is used for sentiment analysis, especially sarcasm detection. Poria et al. (Poria et al., 2016) used different datasets to extract feature sets using convolutional neural networks (CNN). Then, these feature sets were combined and classified by a supported vector machine (SVM) classifier. (Das & Clark, 2018), built a CNN based model that utilizes captions and visual contents of images. (Ilić et al., 2018) used a deep learning model based on character-level word representations derived from Embeddings from Language Models (ELMo). (Maynard & Greenwood, 2014) achieved 50% success with their work on sentiment analysis on tweets. (Mandal & Mahto, 2019) studied another detector built using Deep CNN-LSTM. They used a news headline dataset to achieve 86.16% performance. The news headline dataset has minimal grammatical errors, unique sentences and very clean data. They used Natural Language Toolkit (NLTK) for tokenization, lemmatization and pre-processing. The news headline dataset consisted of 26,709 news headlines. A total of 11,725 were sarcastic headlines. The first layer of the proposed model is the embedding layer and then CNN-LSTM. In this study, The proposed method in this study has 86.44% performance without using LSTM.

3. Proposed System

In this study, the TensorFlow (Abadi et al., 2016) Keras application is performed to detect sarcastic headlines on GPU using Python language in the Anaconda environment. The calculations on the GPU are performed via the Nvidia CUDA library. First, CUDA acceleration package and cudacnn are installed, and then some related TensorFlow, Scikit-learn and Keras packages are installed. After the installations were completed, each title in the dataset was converted into a string of a maximum of 300 words using the Keras feature Tokenizer. After the titles were created, the extra spaces at the end of the sentences were filled via padding, a feature of the Keras tokenizer.

The News Headlines dataset consists of 26,709 lines with a total of 29,656 unique words. All words are given to the input of the deep learning network regardless of their frequency. In the next step, the Embedding Matrix is created. Word embeddings are created and placed in this matrix for each unique word selected by the tokenizer in the previous step. The output of the embedding matrix is given to the second layer called the global average pooling layer. The third and fourth layers are dense.

The Rectified Linear Unit (ReLU) (Glorot et al., 2011) is one of the activation functions used in deep neural networks. The ReLU function is used in conjunction with other layers. ReLU activation function has become the most commonly used in deep learning networks and more popular than logistic sigmoid and hyperbolic tangent functions (Andrej, n.d.). In this study, the ReLU function is used as an activation function. ReLU function truncates all negative values input to zero. The Sigmoid activation function is used in the last layer. The proposed network is trained independently on the Nvidia GeForce GTX 1070, Nvidia GeForce RTX 3060 and Nvidia GeForce RTX 3060Ti graphic cards. Table 1 shows hardware information about these GPUs. Even RTX 3060 has fewer CUDA cores than 3060 Ti, and it shows the best result.

Table 1. GPU features

	GeForce GTX 1070	GeForce RTX 3060	GeForce RTX 3060 Ti
CUDA cores	1920	3584	4864
Memory	8 GB GDDR5	12 GB GDDR6	8 GB GDDR6
Maximum power	150 W	170 W	200 W
Memory bus	256-bit	192-bit	256-bit

4. Performance Evaluation

Adam optimization (Kingma & Ba, 2014) is chosen as the optimization algorithm. The learning rate of the Adam algorithm is set to 0.001. The network model trained 30 epochs for each split ratio on each GPU. The model checkpoint class is used to select optimal model weights based on the validation loss value. The left side of Figure 1 shows the training loss and validation loss. The right side of Figure 1 shows the training accuracy and validation accuracy curves. The Nvidia GeForce RTX 3060 GPU has the best performance on a 90:10 split ratio for 30 epochs.

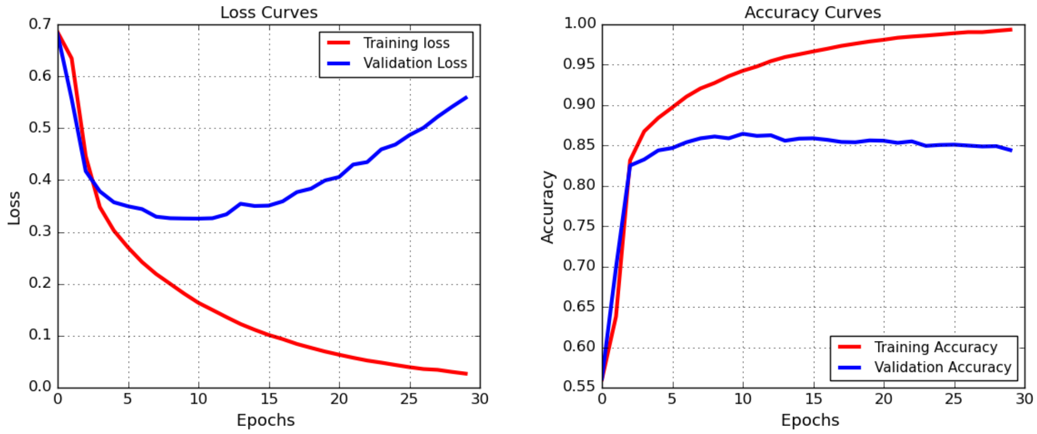

Figure 1. Loss and accuracy curves on RTX 3060 GPU.

Figure 2 shows the confusion matrix. For evaluating the deep learning network, it is considered that the class containing the non-sarcastic texts as the positive class. Therefore, the accuracy, sensitivity, specificity and precision were calculated concerning it. Thus, P is the total number of headlines as positive class (0) non-sarcastic (TP+FN). N is the number of headlines as the negative class (1) sarcastic (FP+TN).

True Positive (TP) is the number of non-sarcastic headlines correctly classified; True Negative (TN) is the number of sarcastic headlines correctly classified; False Positive (FP) is the number of sarcastic headlines incorrectly classified as non-sarcastic; False Negative (FN) is the number of non-sarcastic headlines incorrectly classified as sarcastic.

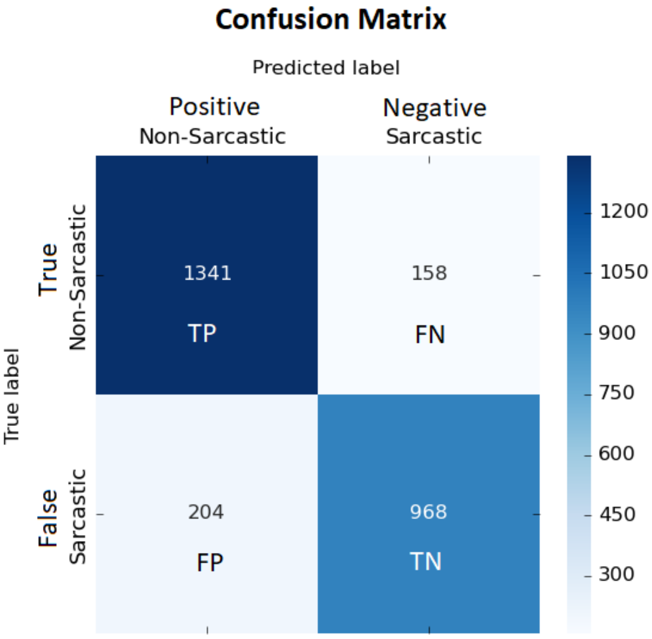

Figure 2. Confusion matrix for 90:10 split rate on RTX 3060 GPU.

Accuracy and error rate can be obtained from the confusion matrix. The error rate is calculated by dividing the number of all false estimates by the total number of the validation data set. The worst error rate is 1, the best is 0. The accuracy is calculated by dividing the number of all correct estimates by the total number of the validation data set. The worst accuracy is 0, the best is 1. It can also calculated as (1–error rate).

$$Accuracy, ACC = \frac{TP + TN}{P + N} = 1 - error$$

Sensitivity is calculated by dividing the number of true positive predictions by the total number of positives. This is also called a *recall* (REC), *hit rate* or *true positive rate* (TPR). The worst sensitivity is 0, the best is 1.

$$Sensitivity, SN = \frac{TP}{TP + FN} = \frac{TP}{P}$$

Specificity is calculated by dividing the number of true negative predictions by the total number of negatives. This is also called *selectivity* or *true negative rate* (TNR). The worst specificity is 0, the best is 1.

$$Specificity, SP = \frac{TN}{TN + FP} = \frac{TN}{N}$$

Precision is calculated by dividing the number of true positive predictions by the total number of positive predictions. This is called a *positive predictive value* (PPV). The worst precision is 0, the best is 1.

$$Precision, PREC = \frac{TP}{TP + FP}$$

The false-positive rate is calculated by dividing the number of false-positive predictions by the total number of negatives. The worst false positive rate is 1, the best is 0. It can also calculated as (1–specificity).

$$False\ positive\ rate, FPR = \frac{FP}{TN + FP} = \frac{FP}{N} = 1 - specificity$$

Table 2 shows the discrimination ability of the proposed deep neural network over validation dataset. Whole experiments are shown in the table in terms of accuracy, sensitivity, specificity and precision.

Table 2. Discriminating ability of the classifier

		GTX 1070	**RTX 3060**	**RTX 3060 Ti**
Split ratio 67:33	Accuracy	0.856819	**0.864096**	0.856592
	Sensitivity	0.863903	0.889927	0.861274
	Specificity	0.847764	0.831058	0.850607
	Precision	0.829750	0.855136	0.827508
Split ratio 75:25	Accuracy	**0.861935**	0.860737	**0.861935**
	Sensitivity	0.868428	0.875634	0.865759
	Specificity	0.853634	0.841692	0.857045
	Precision	0.835392	0.841118	0.833167
Split ratio 90:10	Accuracy	0.864096	**0.864470**	0.863721
	Sensitivity	0.889927	0.894596	0.880587
	Specificity	0.831058	0.825939	0.842150
	Precision	0.855136	0.867961	0.846484

5. Conclusion

Sarcasm detection studies have increased in recent years both due to the increasing number of messages on social media and in order to detect the behaviour of the masses. In this study, sarcasm detection was carried out on the data set obtained from two news sites in English. The deep learning network developed for the detection process was run 30 epochs on three different GPUs with three different split ratios. Nvidia GTX 3060 graphics card showed the highest performance with 86.44%. In future work, it is planned to develop a study on a humanoid robot that can engage in dialogue with human and detect sarcastic speech.

6. References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Zheng, X. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. https://arxiv.org/abs/1603.04467v2

Andrej, K. (n.d.). *Commonly used activation functions*. Retrieved July 26, 2021, from http://cs231n.github.io/linear-classify/#loss

Das, D., & Clark, A. J. (2018). Sarcasm detection on Flickr using a CNN. *ACM International Conference Proceeding Series*, 56–61. https://doi.org/10.1145/3277104.3277118

Dictionary of Turkish Language Association. (n.d.). Retrieved August 8, 2021, from https://sozluk.gov.tr/

Glorot, X., Bordes, A., & Bengio, Y. (2011). *Deep Sparse Rectifier Neural Networks* (pp. 315–323). JMLR Workshop and Conference Proceedings. http://proceedings.mlr.press/v15/glorot11a.html

Ilić, S., Marrese-Taylor, E., Balazs, J. A., & Matsuo, Y. (2018). *Deep contextualized word representations for detecting sarcasm and irony*. 2–7. https://arxiv.org/abs/1809.09795v1

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1412.6980v9

Mandal, P. K., & Mahto, R. (2019). Deep CNN-LSTM with Word Embeddings for News Headline Sarcasm Detection. *Advances in Intelligent Systems and Computing*, *800 Part F1*, 495–498. https://doi.org/10.1007/978-3-030-14070-0_69

Maynard, D., & Greenwood, M. A. (2014). Who cares about sarcastic tweets? Investigating the impact of sarcasm on sentiment analysis. *Proceedings of the 9th International Conference on Language Resources and Evaluation, LREC 2014*, 4238–4243.

Poria, S., Cambria, E., Hazarika, D., & Vij, P. (2016). A Deeper Look into Sarcastic Tweets Using Deep Convolutional Neural Networks. *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers*, 1601–1612. https://arxiv.org/abs/1610.08815v2

SARCASM | Definition of SARCASM by Oxford Dictionary on Lexico.com also meaning of SARCASM. (n.d.). Retrieved August 8, 2021, from https://www.lexico.com/definition/sarcasm

Tay, Y., Luu, A. T., Hui, S. C., & Su, J. (2018). Reasoning with Sarcasm by Reading In-Between. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*, 1010–1020. https://doi.org/10.18653/V1/P18-1093