

# Sarcasm Detection in Headlines

Md. Mahmudul Haq	Moni Kishore Dhar
2016-2-60-031	2016-2-60-099
rajon789@gmail.com	kishor.hridoy1996@gmail.com
Reaz Ahmed	Ahmed Omar
2016-1-63-029	2016-2-60-093
reaz63029@gmail.com	omeruday@gmail.com

May 15, 2020

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Objective . . . . .	2
1.2	Motivation . . . . .	2
1.3	Existing Works . . . . .	2
1.4	Necessity . . . . .	2
<b>2</b>	<b>Methodology</b>	<b>2</b>
<b>3</b>	<b>Implementation</b>	<b>3</b>
3.1	Data Collection . . . . .	3
3.2	Data Processing . . . . .	3
3.3	Model . . . . .	3
3.4	Results . . . . .	4
<b>4</b>	<b>Conclusion</b>	<b>4</b>
4.1	Challenges . . . . .	4
4.2	Limitations . . . . .	4
4.3	Future Prospect . . . . .	4

## **Abstract**

Sarcasm involves some common understanding between speaker and audience. It is a phenomenon of profound contextual arguments. Sometimes sarcasm can be seen in news headlines or any headlines. In this project, a machine learning model was developed to deploy a system which can detect sarcasm of headlines. Here, different types of model were developed and tested with machine learning techniques. The best model we developed using neural network. The model is good enough to recall, that's why it works just fine.

# **1 Introduction**

Sarcasm is defined by the use of remarks that clearly mean, the opposite of what they say, made to hurt someone's feelings or to criticize something humorously. In our project, sarcasm detection refers to the task of accurately labeling a headline as sarcastic or non-sarcastic. A machine can't comprehend the motive of a headline as it only contains some words in a sentence or sentences with no intonation and facial expressions even if it contains so. However, a person can spot a sarcastic sentiment in a text and reason about what makes it so. It is also one of the many critical tasks of NLP to analyze sarcasm in a text to avoid misinterpretation of sarcastic remarks as literal statements. Accuracy and robustness of NLP pave the way to build such models.

## **1.1 Objective**

Main goal of our project is to predict whether a news headline is sarcastic or not.

## **1.2 Motivation**

Let us start with an example, a sentence like "So thrilled to be on call for work the entire weekend!" could be classified merely a sentence with a positive sentiment. However, it is actually the opposite, that is cleverly implied through sarcasm. The use of sarcasm is prevalent across all social media, micro-blogging and e-commerce platforms. Sarcasm detection is imperative for accurate sentiment analysis and opinion mining. It can contribute toward enhancing automated feedback systems in the context of customer-based sites.

## **1.3 Existing Works**

At first sarcasm was defined with the help of six tuples in 2003 that consisted of Speaker, Listener, Context, Utterance, Literal, and Intended Prepositions. There are several works with related of sarcasm detection and many authors have already addressed the problem of identifying sarcasm and tried to solve it using various methods. Some authors tried to build a system that was capable enough to identify sarcasm in the text [1]. The approach proposed by them was based on the behavioral and psychological sciences. In another approach one author have tried to predict sarcasm in the reddit comments using the sub-reddit mentions [2]. In another approach by Rockwell, the vocals and tones of the sarcastic speech were evaluated [3]. In a study, authors have tried to identify sarcasm in a tweet posted on Twitter using the contextual features of the post itself [4].

## **1.4 Necessity**

Sometimes news headlines are so complex to identify whether it is sarcastic or not. On that time this model will help to detect the sarcasm. Again detection of sarcasm is of great importance and beneficial to many NLP applications, such as sentiment analysis, opinion mining and advertising.

# **2 Methodology**

The following pseudo-code describes how the goal was achieved.

---

**Algorithm 1:** Pseudo-code

---

**Result:** Write here the result

1. Read the dataset
  2. import stopwords from nltk.corpus
  3. import PorterStemmer from nltk.stem
  4. Set a list variable corpus = []
  5. **while** *headline* != *EOF* **do**
    - headline* = re.sub('[^a-zA-Z]', ' ', dataset['headline'][i])
    - headline* = *headline*.lower()
    - headline* = *headline*.split()
    - ps = PorterStemmer()
    - headline* = [ps.stem(word) for word in *headline* if not word in set(stopwords.words('english'))]
    - headline* = ' '.join(*headline*)
    - corpus.append(*headline*)
  - end**
  6. Vectorize the words
  7. fit corpus using CountVectorizer from sklearn.feature\_extraction.text
  8. Select training and testing set
  9. Train the model using ML model
  10. Show confusing matrix
  11. Print the accuracy
- 

### 3 Implementation

After importing the data set, first task was to preprocess the data set using NLTK. Here, the stop words were removed from the data set then all the verb that were not in their root form, taken to their root form. Having this step done, the words were projected by integer or floating points and the data set was then prepared for fitting data to the required models.

Initially Naive Bayes and Decision Tree were used. Even-though they showed accuracy more than 70 percent but they were very lackey at recall. To overcome this issue, Sequential classifier is used with 10 epochs.

#### 3.1 Data Collection

The dataset generated by collecting news from two major websites. the Onion website that provided the sarcastic version of various events happened around the globe and HuffPost provided the non-sarcastic version of the same news so as to maintain the balance in the dataset.

#### 3.2 Data Processing

The data came somewhat preprocessed as there was no missing value but some adjustments had to be made to make the data compatible with the classifiers using NLTK. Removing stopwords and tokenizing.

#### 3.3 Model

To accomplish the goal, so far, we have used a machine learning model called naive\_bayes from sklearn library. We have also used MultinomialNB from the same library. From the dataset, 80 percent of the data is used to train the model, and the rest 20 percent is for testing the model. Python is used for coding.

### 3.4 Results

The final model showed promising results with an accuracy of 77 percent.

We tested the model with headlines of 10/05/2020 from The daily Star and Al Jazeera as well as from the data-set.

The model outputs the followings. 1 being sarcastic and 0 not sarcastic.

input : "Shutdown being gradually relaxed to help people make a living: PM"

output : 0

input : "Obama slams Trump response to coronavirus as 'chaotic disaster'"

output : 0

input : "nuclear bomb detonates during rehearsal for 'spider-man' musical"

output : 1

input : "mom starting to fear son's web series closest thing she will have to grandchild"

output : 1

## 4 Conclusion

The final model is pretty much convenient. It is good enough to detect whether a headline is sarcastic or not. Some remarks on this project titled as "Sarcasm Detection in Headlines" are given below.

### 4.1 Challenges

It was somewhat hard for us at some point because the model we built using Naive Bayes did not show good result. Later on, we had to try few other techniques such as Sequential and dense from keras.

### 4.2 Limitations

It can sense sarcasm in headlines but it may fail to understand sometime if the headline contains proverbs or particular phrases or figurative expressions.

### 4.3 Future Prospect

It can be used in AI driven chat bots to improve responses while dealing with customers or users.

## References

- [1] A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm detection on twitter: A behavioral modeling approach," in *Proceedings of the eighth ACM international conference on web search and data mining*, pp. 97–106, 2015.
- [2] B. C. Wallace, E. Charniak, *et al.*, "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)*, pp. 1035–1044, 2015.
- [3] P. Rockwell, "Lower, slower, louder: Vocal cues of sarcasm," *Journal of Psycholinguistic research*, vol. 29, no. 5, pp. 483–495, 2000.
- [4] M. Zhang, Y. Zhang, and G. Fu, "Tweet sarcasm detection using deep neural network," in *Proceedings of COLING 2016, The 26th International Conference on Computational Linguistics: Technical Papers*, pp. 2449–2460, 2016.