……………………………………………...……………………………………………...……………………………………………...……

**Name:**  Ustad Munir Shafiyoddin.

**Email address:**   ustadmunir3232@gmail.com.

**Contact number:**  9503982353 .

**Anydesk address: desktop-e7gfk0j-munir@ad  .**

**Years of Work Experience:  0 years.**

**Date:    2nd Aug 2021**

……………………………………………...……………………………………………...……………………………………………...……

**Self Case Study -2:**

## Sarcasm Detection (Sentiment analysis in NLP)

……………………………………………...……………………………………………...……………………………………………...……

**Introduction :-**

Sarcasm is the use of words usually used to either mock or annoy someone, or for humorous purposes. (means negative response but in polite manner)

Sarcasm detection is sentiment analysis in NLP.

**Sentiment analysis also known as opinion mining.**

Sarcasm detection is an important component in many natural language processing (NLP) systems, directly relevant to natural language understanding, dialogue systems, and text mining. However, detecting sarcasm is difficult because it occurs infrequently and is difficult for even humans to discern.

In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention.

Humans are fairly intuitive when it comes to interpreting the tone of a piece of writing.

The human language is complex. Teaching a machine to analyse the various grammatical nuances, cultural variations, slang and misspellings that occur in online mentions is a difficult process. Teaching a machine to understand how context can affect tone is even more difficult.

Business problem:-

The applications of sentiment analysis are broad and powerful. The ability to extract insights from social data is a practice that is being widely adopted by organisations across the world.

Sentiment analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics.

Being able to quickly see the sentiment behind everything from forum posts to news articles means being better able to strategise and plan for the future.

It can also be an essential part of your market research and customer service approach. Not only can you see what people think of your own products or services, you can see what they think about your competitors too.  The overall customer experience of your users can be revealed quickly with sentiment analysis, but it can get far more granular too.

The ability to quickly understand consumer attitudes and react accordingly is something that any organization to take an advantage of when they noticed that there was a steady increase in negative feedback to their product.

**ML formulation of business problem:**

**Mapping the problem into machine learning problem:**

This is two class classification problem.

**Business constraints:**

No strict latency constraints.

Prediction of sarcasm should be correct.

**Data overview:**

This dataset contains 1.3 million Sarcastic comments from the Internet commentary website Reddit. The dataset contains comments, which contains the \s ( sarcasm) tag. This tag is often used by Redditors to indicate that their comment is in jest and not meant to be taken seriously, and is generally a reliable indicator of sarcastic comment content.

The data was gathered by: Mikhail Khodak and Nikunj Saunshi and Kiran Vodrahalli for their article "A Large Self-Annotated Corpus for Sarcasm".

Total datapoints : 1010826, Ranges from 0 to 1010825.

Data columns (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 label 1010826 non-null int64

1 comment 1010773 non-null object

2 author 1010826 non-null object

3 subreddit 1010826 non-null object

4 score 1010826 non-null int64

5 ups 1010826 non-null int64

6 downs 1010826 non-null int64

7 date 1010826 non-null object

8 created\_utc 1010826 non-null object

9 parent\_comment 1010826 non-null object

dtypes: int64(4), object(6)

memory usage: 77.1+ MB

**Columns and their description:-**

1. **Label**:- Sarcastic or not.
2. **Comment** :- Reply to a Parent Reddit comment.
3. **Author**:- Person who commented.
4. **Subreddit**:- Commented under which subreddit.
5. **Score**:- Number of upvotes -(minus) Number of downvotes.
6. **Ups**:-Number of upvotes.
7. **Downs**:-Number of downvotes.
8. **Date**:-Commented date.
9. **created\_utc**:-Commented time in the UTC Timezone.
10. **parent\_comment**:-The Parent Reddit comment to which sarcastic replies are made

This dataset is balanced dataset with 505413 sarcastic and 505413 non sarcastic comments.

**Metrics:-**

1. **accuracy** could be a good metric(KPI) as the dataset is balanced and both sarcastic and non sarcastic comments are equally important.
2. **Confusion matrix** will help us in understanding what is happening with predictions.

**Research:-**

1. **NLP(Natural Language Processing):-**

What is NLP?

Natural language processing is the ability of computer to understand human language as it is spoken and written (referred to Natural Language). It is part of Artificial Intelligence.

It has applications in a number of fields, including medical research, search engines and business intelligence.

There are two main phases to natural language processing: data preprocessing and algorithm development.

Data preprocessing involves preparing and "cleaning" text data for machines to be able to analyze it. Preprocessing converts the data in a form with which algorithm can work. There are several ways this can be done, including:

* **Tokenization** **.** This is when text is broken down into smaller units to work with(broking the sentences to words).
* **Stop Word removal.** This is when common words are removed from text so unique words that offer the most information about the text remain(removing most frequent words).
* **Lemmatization and stemming. Stemming** is when words are reduced to their root forms to process and **Lemmatization** is the algorithmic process of determining lemma of word based on its intended meaning. Unlike stemming lemmatization depend upon the correctly identifying the intended part of speech and meaning of a word in a sentence as well as larger context surrounding that sentence such as neighbouring sentence or even document. Lemmatization is morphological analysis of word.

Once the data has been preprocessed(means removing symbols, stopwords, tokenizing, stemming), an algorithm is used to process it. There are many different natural language processing algorithms, but two main types are commonly used:

* **Rules-based system.** Using linguistic rules. This system uses carefully designed linguistic rules. This approach was used early on in the development of natural language processing, and is still used.
* **Machine learning-based system.** Using a combination of **machine learning, deep learning** and**neural networks,** natural language processing algorithms hone their own rules through repeated processing and learning.

### What is natural language processing used for?

Some of the main functions that natural language processing algorithms perform are:

* **Text classification.** This involves categorizing the text data used for sentiment analysis. For example, when brand A is mentioned in X number of texts, the algorithm can determine how many of those mentions were positive and how many were negative. It can also be useful for intent detection.
* **Text extraction.** This involves automatically summarizing text and finding important pieces of data. One example of this is keyword extraction, which can be useful for search engine optimization. For example, a tool might pull out the most frequently used words in the text. Another example is named entity recognition, which extracts the names of people, places and other entities from text.
* **Machine translation.** This is the process by which a computer translates text from one language, such as English, to another language, such as French, without human intervention.
* **Natural language generation.** This involves using natural language processing algorithms to analyze unstructured data and automatically produce content based on that data. One example of this is in language models such as GPT3, which are able to analyze an unstructured text and then generate believable articles based on the text.

### Benefits of natural language processing

The main benefit of NLP is that it improves the way humans and computers communicate with each other. The most direct way to manipulate a computer is through code -- the computer's language. By enabling computers to understand human language, interacting with computers becomes much more intuitive for humans.

### Challenges of natural language processing

There are a number of challenges of natural language processing and most of them boil down to the fact that natural language is ever-evolving and always somewhat ambiguous. They include:-

* **Precision.**Computers traditionally require humans to "speak" to them in a programming language that is precise, unambiguous and highly structured or through a limited number of clear voice commands.
* **Tone of voice and inflection.**Natural language processing has not yet been perfected. For example, semantic analysis can still be a challenge. Other difficulties include the fact that the abstract use of language is typically tricky for programs to understand. For instance, **natural language processing does not pick up sarcasm easily.** These topics usually require understanding the words being used and their context in a conversation. As another example, a sentence can change meaning depending on which word or syllable the speaker puts stress on. NLP algorithms may miss the subtle, but important, tone changes in a person's voice when performing speech recognition. The tone and inflection of speech may also vary between different accents, which can be challenging for an algorithm to parse.
* **Evolving use of language.** Natural language processing is also challenged by the fact that language and the way people use it is continually changing. Although there are rules to language, none are written in stone, and they are subject to change over time. Hard computational rules that work now may become obsolete as the characteristics of real-world language change over time.

Reference:- <https://searchenterpriseai.techtarget.com/definition/natural-language-processing-NLP>

**I will use this (NLP) to process the comment and parent comment features in the dataset as these two features are human comments which is natural language.**

1. **W2V:-**

Word2vec is a technique for natural language processing published in 2013. The word2vec algorithm uses a neural network model to learn word associations from a large corpus of text. Once trained, such a model can detect synonymous words or suggest additional words for a partial sentence. As the name implies, word2vec represents each distinct word with a vector. The vectors are chosen carefully such that a simple mathematical function (the cosine similarity between the vectors) indicates the level of semantic similarity  between the words represented by those vectors.

**CBOW AND SKIP-GRAMS** are two types of models used to obtain **WORD2VEC**. Word2vec can utilize either of two model architectures to produce a distributed representation of words: continuous bag-of-words (CBOW) or continuous skip-gram. In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). In the continuous skip-gram architecture, the model uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. According to the authors' note, CBOW is faster while skip-gram does a better job for infrequent words.

Word2Vec captures semantic meaning (e.g. the ability to tell if words are similar, or opposites, eg:- “Stockholm” and “Sweden” have the same relationship as “Cairo” and “Egypt” have) as well as syntactic, or grammar-based, relationships (e.g. the relationship between “had” and “has” is the same as that between “was” and “is”).

It is a great idea to use embeddings that were pre-trained on vast amounts of text data instead of training them. We can download embeddings generated by pre-training with Word2Vec or GloVe.

**I will use Google’s pre-trained w2v model to get vector for every word.  It includes word vectors for a vocabulary of 3 million words. The vector length is 300 features.**

1. **TFIDF**

**TF-IDF** stands for **“Term Frequency — Inverse Document Frequency”.**

Term frequency = this measures frequency of word in document.

Document frequency = this measures frequency of word in corpus.

Tf-idf=term frequency\*log( inverse document freaquency )

If word is more frequent in document and less frequent in corpus then more importance will be given to that word or tf-idf value of that word will be high.

Log is used to reduce the dominance of idf in equation for less frequent word in corpus.

**I will use this to get important word for sarcasm and non sarcasm.**

**Tf-idf word2vec is technique to convert word to vec to sentence to vec. I will use this to get a vector representation for entire comment and a vector representation for entire parent comment.**

1. **Transfer learning:-**

Transfer learning (TL) is a research problem in machine learning that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. For example, knowledge gained while learning to recognize cars could apply when trying to recognize trucks

There are four cases where we can use transfer learning

To understand these four cases we will define some terminology it will be easy to understand these four cases.

**D**- dataset on which model(pretrained) is trained.

**d**- our new data for which we want to find performance metric.

**Case1**:- d is small and similar to D, then we can use pretrained model for feature engineering .

**Case2**:-d is large and similar to D, then we can fine tune the whole pretrained model and use it to compute the best performance metric for our dataset d.

**Case3**:-d is medium sized and similar to D, then we can fine tune last layers of pretrained model . and can compute best performance metric.

**Case4**:-d is small and not similar to D, then we can obtain features from pre trained model using new dataset d using initial layers of pre trained model as feature engineering model.

**Case5**:-d is large and not similar to D, then initialize the pre trained model and tune the complete model using small learning rate.

Reference:- <https://en.wikipedia.org/wiki/Transfer_learning>

**I will use this to use pre trained BIRT model to get the feature representation for comment and parent comment feature.**

1. **Birt (Bidirectional Encoder Representation from transformer):-**

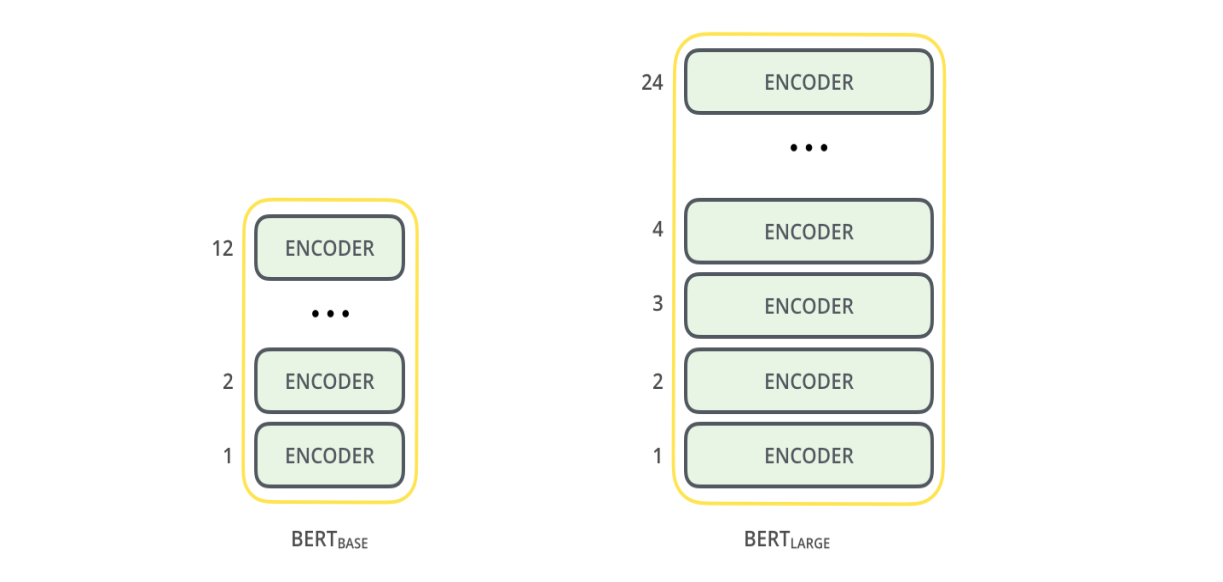
The results achieved by transformer on machine translation made them like a replacement to LSTMs, this due the fact that transformers deals with long term dependencies better than LSTMs.

BIRT is a transformer and release of BIRT is been referred to as NLP’s ImageNet moment, referencing how years ago similar developments accelerated the development of machine learning in Computer Vision tasks

BERT is a model that broke several records for how well models can handle language-based tasks.

BERT is basically a trained Transformer Encoder stack.

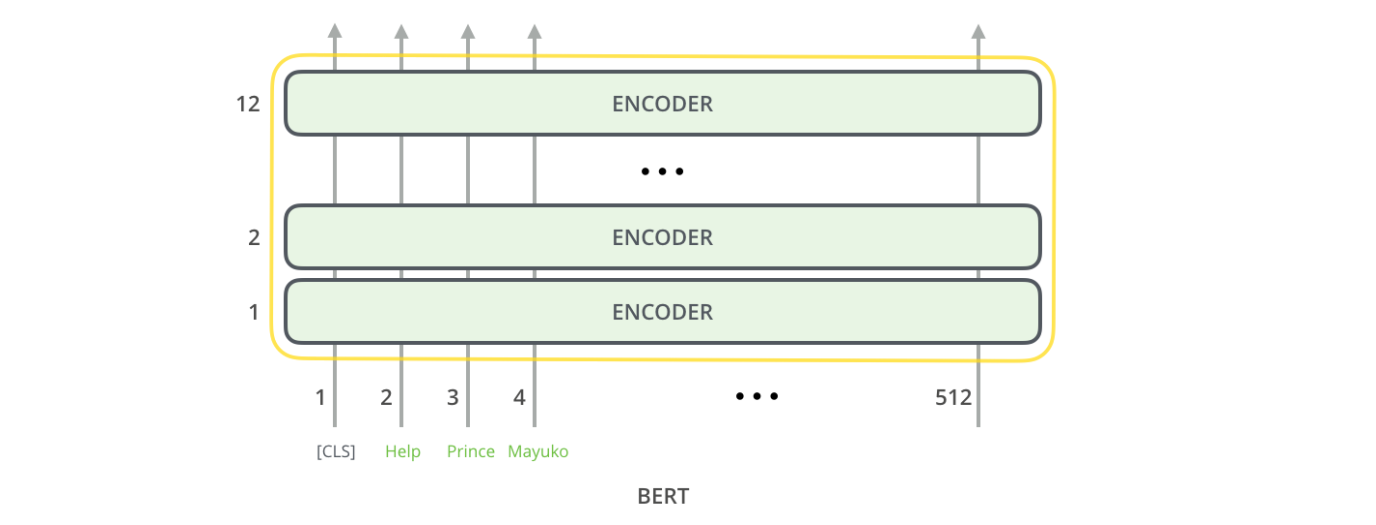
**Architecture of BIRT**

****

**Reference:-**[**https://jalammar.github.io/images/bert-base-bert-large-encoders.png**](https://jalammar.github.io/images/bert-base-bert-large-encoders.png)

Both BERT model sizes have a large number of encoder layers (which the paper calls Transformer Blocks) – twelve for the Base version, and twenty four for the Large version. These also have larger feedforward-networks (768 and 1024 hidden units respectively), and more attention heads (12 and 16 respectively) than the default configuration in the Transformer (6 encoder layers, 512 hidden units, and 8 attention heads).

**MODEL INPUT:-** The first input token is supplied with a special [CLS] token for reasons that vector representation of CLS is used for several applications like using output of CLS token we can classify the sentence. CLS here stands for Classification.



REFERENCE:- <https://jalammar.github.io/images/bert-encoders-input.png>

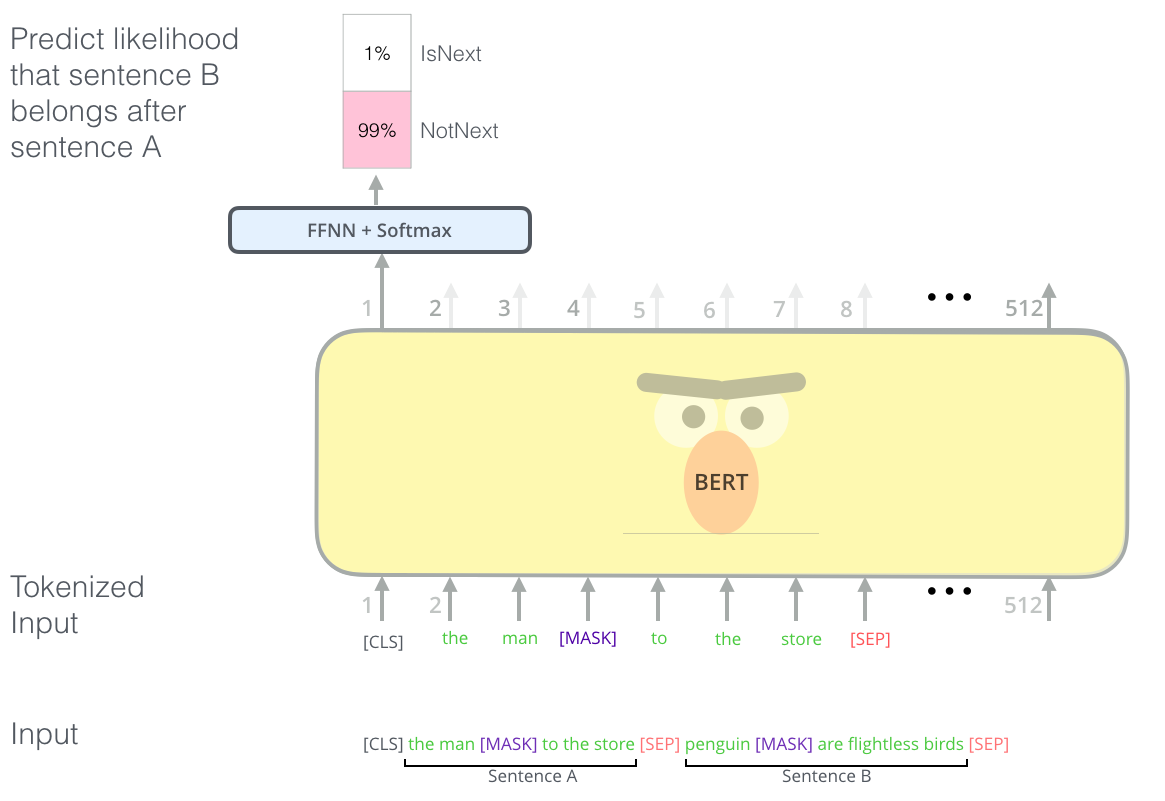
Just like the encoder of the transformer, BERT takes a sequence of words as input in first encoder applies self-attention, and passes its results through a feed-forward network, and then pass it to the next encoder.

MODEL OUTPUT:-

For each token input model return a vector representation of size 768 in BIRT base.

And for classification we can use vector corrosponding to CLS as it stores the information of all sentence , as in the creation of vector for CLS all words have been considered in self attention layer.

**HOW I will use it :-**

****

**Passing two sentences to BIRT using a SEP as seperating token for two sentences I can get vector represention for all tokens and using only one token representation I can classify sarcasm or not.**

**Using above architecture I want to get the vector represention for CLS token using parent comment as sentence A and comment as sentence B and classify the comment as sarcasm or not.**

**Reference:-**[**https://jalammar.github.io/images/bert-next-sentence-prediction.png**](https://jalammar.github.io/images/bert-next-sentence-prediction.png)

References:-

1. <https://en.wikipedia.org/wiki/Sarcasm>
2. <https://www.brandwatch.com/blog/understanding-sentiment-analysis/>
3. "A Large Self-Annotated Corpus for Sarcasm" :- <https://arxiv.org/pdf/1704.05579.pdf>