[**Working Colab File**](https://colab.research.google.com/drive/1jz09OFkApvrFpn4GWfnmMqdlzQzOqf_6?usp=sharing)

**Related articles**

geeksforgeeks.org/sentiment-classification-using-bert/

[Bert notebook](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07E%20-%20Text%20Classification%20with%20BERT%20Deep%20Transfer%20Learning.ipynb)

**Types of models**

* Baseline logistic
* [Neural network (with embedding)](https://towardsdatascience.com/nlp-spam-detection-in-sms-text-data-using-deep-learning-b8632db85cc8)
* [BERT](https://colab.research.google.com/github/dipanjanS/nlp_workshop_odsc19/blob/master/Module05%20-%20NLP%20Applications/Project07E%20-%20Text%20Classification%20with%20BERT%20Deep%20Transfer%20Learning.ipynb)

**Comments from Warren**

baseline with non-contextual embeddings, and then comparing the results from that to transformer-based embeddings → Neural network with embedding

think about how you're going to leverage BERT (embeddings generator vs. main model) → What does this mean exactly??

1. **All group member’s names**

Seung Hyun (Amy) Ryu

Joshua Toth

Joseph Poirier

* 1. **Breakdown of what each member will be responsible for:**

Data pre-processing: Joshua Toth

EDA: Seung Hyun (Amy) Ryu

Predictive modeling Pt Base: Joseph Poirier

Predictive modeling Pt Alternate: Joshua Toth

1. **Data source - if you are proposing your own dataset/ topic.**

We plan to use the following kaggle dataset comprised of comments scraped from Reddit and tagged for whether or not they express sarcasm based on usage of the \s tag by the commenter:

<https://www.kaggle.com/datasets/danofer/sarcasm>.

1. **Project Plan**
   1. **Explain what you intend to study with your project.**

We intend to use NLP to study the nature of sarcastic comments and their context. In addition to predicting whether a comment is sarcastic or not, we can also use this dataset to explore the impact of some other factors on the likelihood a comment is sarcastic, such as the time of day the comment was posted or whether or not the comment was in a reply or a main thread.

Potential questions:

Are sarcastic comments more popular than serious comments?

At what time of day/week/year are people most sarcastic/serious?

Are people getting more/less sarcastic over time?

Are reactions to sarcasm getting more/less positive over time?

* 1. **What is the ultimate objective?**

Our central goal will be building a high performing model to detect/predict whether a comment is sarcastic or not.

* 1. **What types of models are you considering?**

Logistic regression (baseline)

BERT

LSTM

Transformers

Attention

(We will probably end up picking logistic regression and one of the next three to compare against)

1. **Why is this project interesting?**

Sarcasm is a witty and fun form of talking. However, in writing the sigal of sarcasm can be muddled – particularly as we march ahead in a digital age that has transformed the way we communicate, with texting, emailing and online commentary replacing face-to-face chats or phone conversations.

Being able to accurately discern whether a comment is sarcastic or not will play a crucial role in moderating content on various online platforms including ecommerce, social media, rating platforms, and news. For example, a particular review of a product on ecommerce that was originally classified as positive e.g. “This hair product is SOOO good it will make you bald in no time” could accurately be tagged as negative. In addition, when doing sentiment analysis on a particular social or political topic that’s trending, researchers should be able to take into account sarcasm to get an accurate pulse on how people are feeling.

1. **What challenges and obstacles might you anticipate with this project?**

This is an extremely large dataset (1.3 M comments, 900K unique comments, 250K authors), so preprocessing could be a challenge. If we want to avoid using cloud computing, we may need to extract a subset of the data. A sequential algorithm like an LSTM may struggle with long compute times, especially for longer comments.

1. **TA name if you would like to request a TA** Warren Wang