Project 2 - IMDB Review Classification

## Objectives:

* Task 1: Explore TfidfVectorizer
* Task 2: Explore Word2Vector Model glove-twitter-25
* Task 3: Explore distilbert-base-uncased
* Task 4: Compare the model’s classification abilities using IMDB dataset

## Data:

* IMDB: https://huggingface.co/datasets/imdb
* Large Movie Review Dataset. This is a dataset for binary sentiment classification containing substantially more data than previous benchmark datasets. We provide a set of 25,000 highly polar movie reviews for training, and 25,000 for testing. There is additional unlabeled data for use as well.

## Implementation:

### Libraries

* Torch - neural network optimization
* TfidfVectorizer- used for data parsing and processing
* gensim - Word2Vec
* Transformers - used to load pretrained models

### Hyperparameters

* Batch Size = 100 batches (250 entries per batch)
* Epoch = 10
* Test Size= 200 randomize entries
* Learning Rate= 0.01

### Algorithm and Code

The comparison code is divided into three main python files each containing the classifications model used for testing. In addition, there are additional utilizing functions for testing the models used across all of the testing.

## Results:

### Summary:

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Accuracy | Recall | F1 |
| Word2Vec | 0.56 | 0.0112 | 0.022 |
| Bert | 0.81 | 0.78 | 0.804 |
| Tfid | 0.735 | 0.72 | 0.746 |

### Task 1 Results:

### TfidfVectorizer

Vocab Size = 74849

ID for ”ambitious" = 2977

feature vector sum = 2.294023319093611

### Task 2 Results:

Word2Vector

Similarity between computer and laptop = 0.8352675

Similarity between computer and fruit = 0.45673344

Similarity between fruit and banana = 0.8357839

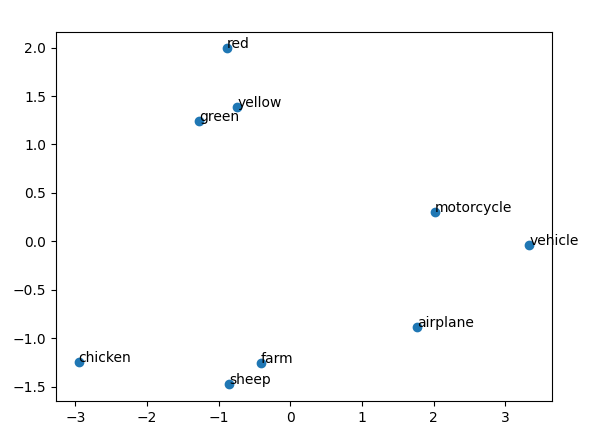
-------------------------------

Distance between france and paris 0.11305677890777588

Distance between canada and paris 0.28165972232818604

Distance between brazil and paris 0.3328576683998108

Closest words to ”boat” = ['cabin', 'truck', 'pool', 'plane', 'flying', 'balloon', 'roof', 'rides', 'backyard', 'cab']



### Task 3 Results:

Bert

Vocab Size = 30522

Hidden Size = 768

Bert uses a tokenizer to preprocess text inputs into a format that can be effectively processed by the mode. This allows the model to go beyond the syntax of the language and further break down meaning by breaking down words and its meaning. It also allows for a more predictable input handling.

['natural', 'language', 'processing', 'is', 'fun', '!']

[101, 3019, 2653, 6364, 2003, 4569, 999, 102]

What is input ID: Input IDs are numerical representations of tokenized input sequences in BERT, mapping each token to a unique integer ID through a pre-trained tokenizer. They form the basis of the input data, converting text into a format suitable for processing by the model.

Why Do we use an attention Mask? The attention mask in BERT is a binary tensor indicating which tokens should be attended to (with a value of 1) and which ones should be ignored (with a value of 0).

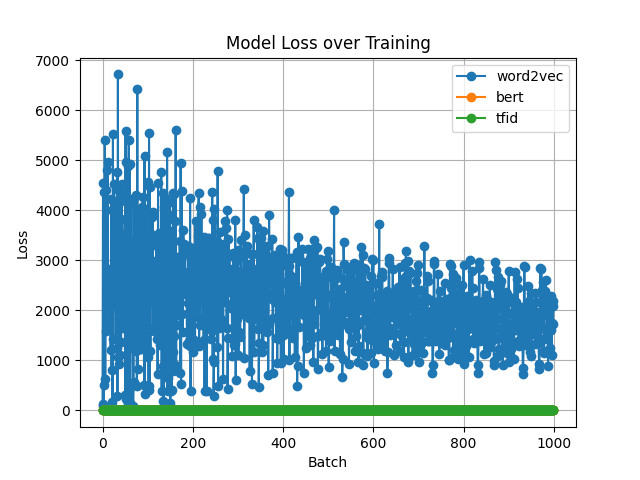
Task 4 Results:

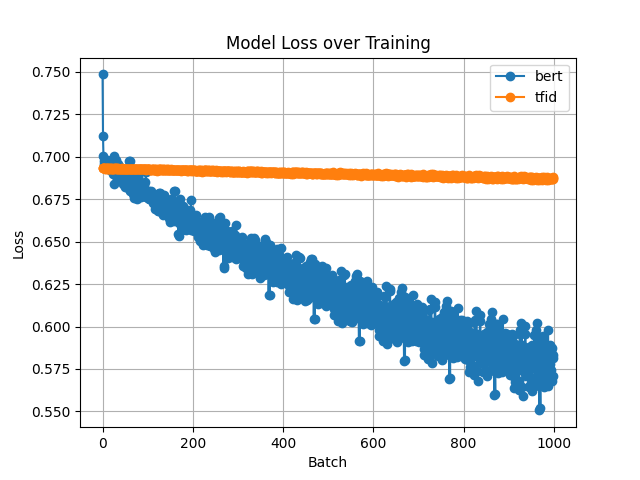
Model Classification for “The Dark Knight was a masterpiece! The plot, cast, and everything were absolutely sick!”

Bert = 1

Word2Vec = 1

Tfid = 1





IMDB Dataset Classification Testing

Epoch [1/10] [Batch 1/100], Loss: 126.4317

Epoch [1/10] [Batch 51/100], Loss: 303.9315

Epoch [1/10] [Batch 100/100], Loss: 2457.9600

Epoch [2/10] [Batch 1/100], Loss: 1063.0090

Epoch [2/10] [Batch 51/100], Loss: 158.5443

Epoch [2/10] [Batch 100/100], Loss: 3251.4592

Epoch [3/10] [Batch 1/100], Loss: 3205.6118

Epoch [3/10] [Batch 51/100], Loss: 2592.6526

Epoch [3/10] [Batch 100/100], Loss: 3203.9482

Epoch [4/10] [Batch 1/100], Loss: 3109.8796

Epoch [4/10] [Batch 51/100], Loss: 1921.8265

Epoch [4/10] [Batch 100/100], Loss: 3086.6624

Epoch [5/10] [Batch 1/100], Loss: 3060.0234

Epoch [5/10] [Batch 51/100], Loss: 1256.3806

Epoch [5/10] [Batch 100/100], Loss: 2674.9924

Epoch [6/10] [Batch 1/100], Loss: 3182.2952

Epoch [6/10] [Batch 51/100], Loss: 1371.8215

Epoch [6/10] [Batch 100/100], Loss: 2558.4595

Epoch [7/10] [Batch 1/100], Loss: 3014.4380

Epoch [7/10] [Batch 51/100], Loss: 1304.4070

Epoch [7/10] [Batch 100/100], Loss: 2402.4958

Epoch [8/10] [Batch 1/100], Loss: 2914.2891

Epoch [8/10] [Batch 51/100], Loss: 1341.7588

Epoch [8/10] [Batch 100/100], Loss: 2219.8572

Epoch [9/10] [Batch 1/100], Loss: 2917.6953

Epoch [9/10] [Batch 51/100], Loss: 1149.2812

Epoch [9/10] [Batch 100/100], Loss: 2177.4436

Epoch [10/10] [Batch 1/100], Loss: 2779.9236

Epoch [10/10] [Batch 51/100], Loss: 1082.4510

Epoch [10/10] [Batch 100/100], Loss: 2097.5608

Sum= 1

Accuracy 0.56

Precision: (1.0,)

Recall: 0.011235955056179775

F1 Score: 0.022222222222222223

We strongly recommend passing in an `attention\_mask` since your input\_ids may be padded. See https://huggingface.co/docs/transformers/troubleshooting#incorrect-output-when-padding-tokens-arent-masked.

Epoch [1/10] [Batch 1/100], Loss: 0.7485

Epoch [1/10] [Batch 51/100], Loss: 0.6834

Epoch [1/10] [Batch 100/100], Loss: 0.6799

Epoch [2/10] [Batch 1/100], Loss: 0.6762

Epoch [2/10] [Batch 51/100], Loss: 0.6628

Epoch [2/10] [Batch 100/100], Loss: 0.6640

Epoch [3/10] [Batch 1/100], Loss: 0.6582

Epoch [3/10] [Batch 51/100], Loss: 0.6464

Epoch [3/10] [Batch 100/100], Loss: 0.6498

Epoch [4/10] [Batch 1/100], Loss: 0.6425

Epoch [4/10] [Batch 51/100], Loss: 0.6325

Epoch [4/10] [Batch 100/100], Loss: 0.6370

Epoch [5/10] [Batch 1/100], Loss: 0.6285

Epoch [5/10] [Batch 51/100], Loss: 0.6204

Epoch [5/10] [Batch 100/100], Loss: 0.6255

Epoch [6/10] [Batch 1/100], Loss: 0.6159

Epoch [6/10] [Batch 51/100], Loss: 0.6097

Epoch [6/10] [Batch 100/100], Loss: 0.6150

Epoch [7/10] [Batch 1/100], Loss: 0.6044

Epoch [7/10] [Batch 51/100], Loss: 0.6001

Epoch [7/10] [Batch 100/100], Loss: 0.6054

Epoch [8/10] [Batch 1/100], Loss: 0.5939

Epoch [8/10] [Batch 51/100], Loss: 0.5915

Epoch [8/10] [Batch 100/100], Loss: 0.5967

Epoch [9/10] [Batch 1/100], Loss: 0.5843

Epoch [9/10] [Batch 51/100], Loss: 0.5836

Epoch [9/10] [Batch 100/100], Loss: 0.5887

Epoch [10/10] [Batch 1/100], Loss: 0.5754

Epoch [10/10] [Batch 51/100], Loss: 0.5764

Epoch [10/10] [Batch 100/100], Loss: 0.5813

Sum= 94

Accuracy 0.81

Precision: (0.8297872340425532,)

Recall: 0.78

F1 Score: 0.8041237113402062

Epoch [1/10] [Batch 1/100], Loss: 0.6931

Epoch [1/10] [Batch 51/100], Loss: 0.6928

Epoch [1/10] [Batch 100/100], Loss: 0.6926

Epoch [2/10] [Batch 1/100], Loss: 0.6925

Epoch [2/10] [Batch 51/100], Loss: 0.6920

Epoch [2/10] [Batch 100/100], Loss: 0.6919

Epoch [3/10] [Batch 1/100], Loss: 0.6918

Epoch [3/10] [Batch 51/100], Loss: 0.6913

Epoch [3/10] [Batch 100/100], Loss: 0.6912

Epoch [4/10] [Batch 1/100], Loss: 0.6912

Epoch [4/10] [Batch 51/100], Loss: 0.6906

Epoch [4/10] [Batch 100/100], Loss: 0.6906

Epoch [5/10] [Batch 1/100], Loss: 0.6906

Epoch [5/10] [Batch 51/100], Loss: 0.6899

Epoch [5/10] [Batch 100/100], Loss: 0.6899

Epoch [6/10] [Batch 1/100], Loss: 0.6899

Epoch [6/10] [Batch 51/100], Loss: 0.6892

Epoch [6/10] [Batch 100/100], Loss: 0.6893

Epoch [7/10] [Batch 1/100], Loss: 0.6893

Epoch [7/10] [Batch 51/100], Loss: 0.6885

Epoch [7/10] [Batch 100/100], Loss: 0.6886

Epoch [8/10] [Batch 1/100], Loss: 0.6887

Epoch [8/10] [Batch 51/100], Loss: 0.6878

Epoch [8/10] [Batch 100/100], Loss: 0.6879

Epoch [9/10] [Batch 1/100], Loss: 0.6881

Epoch [9/10] [Batch 51/100], Loss: 0.6871

Epoch [9/10] [Batch 100/100], Loss: 0.6873

Epoch [10/10] [Batch 1/100], Loss: 0.6874

Epoch [10/10] [Batch 51/100], Loss: 0.6864

Epoch [10/10] [Batch 100/100], Loss: 0.6866

Sum= 108

Accuracy 0.735

Precision: (0.7222222222222222,)

Recall: 0.7722772277227723

F1 Score: 0.7464114832535885