Logo, company name

Description automatically generated

**CSE-6363 Spring 2023 002: Machine Learning Phase 3**

**NLP- SENTIMENT ANALYSIS FOR IMDB MOVIE REVIEWS**

# **Team Members:**

1. Sriram Manoj Arava-1002025610
2. Aravind Yalla -1002038759
3. Abhishek Wadhwani - 1002035719

# Data modelling using word2vec:

* After performing data preprocessing and cleaning, we implemented the Word2Vec Model to detect positive and negative words on trained samples.
* Word2Vec model extracts the notion of relatedness across words or products.
* A Word2Vec model learns meaningful relations and encodes the relatedness into vector similarity.
* It helps in identifying semantic relatedness, synonym detection, concept categorization, selection preferences, and analogy.

Graphical user interface, text, application, email

Description automatically generated

# Word2Vec Training:

# build\_vocab method is used to construct the vocabulary from the corpus of text

# The epochs attribute of a Word2Vec model in gensim represents the number of times the model has iterated over the entire corpus during training.

# The corpus\_count attribute represents the number of documents in the corpus used to train the model.

Graphical user interface, text, application, Word

Description automatically generated

Graphical user interface, text, application

Description automatically generated

# **Similarity:**

* The similarity is measured based on the cosine similarity between the word vectors. The output is a list of (word, similarity score) pairs, sorted by similarity score in descending  order.
* Calculates the cosine similarity between the two words given as input. In this case, it calculates the cosine similarity between the word "awful" and "poor". The value returned indicates how similar the two words are to each other in terms of their contexts and distributions within the training corpus of all the text files along with labels.

Text

Description automatically generated

Graphical user interface, application, Word

Description automatically generated

# **BERT Model:**

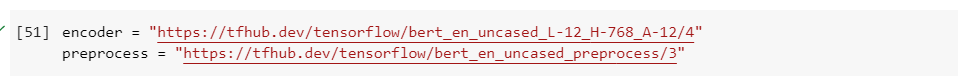
* BERT (Bidirectional Encoder Representations from Transformers) is a Natural Language Processing Model.
* It is built to condition both left and right context in all layers to pre-train deep bidirectional representations from unlabeled text.
* As a result, using just one additional output layer, the pre-trained BERT model may be fine-tuned to generate state-of-the-art models for a variety of tasks.

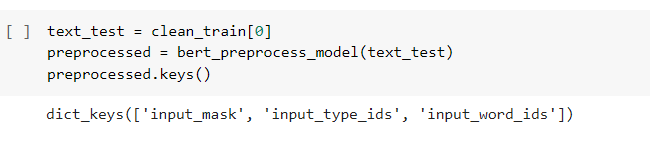
# **BERT Pre-processing:**

* TensorFlow Hub KerasLayer that performs the preprocessing required to the BERT model. It tokenizes the input text into sub words, creates input segments and masks, and standardizes the input format. This layer takes in raw text input and outputs a dictionary containing the preprocessed input for the BERT mode.
* BERT preprocess model has various features to preprocess the text data such as tokenization, masking, and padding. When we pass the text data through this preprocess model, it returns a dictionary containing various keys. These keys contain the preprocessed data in the form of token IDs, mask IDs, and type IDs which are further used in the BERT model to generate the embeddings for the text data.

A picture containing chart

Description automatically generated





# **BERT Pre-processing:**

* input\_word\_ids:
  + Represents the tokenized input sequence, where each word in the sequence is mapped to a unique integer ID. These integer IDs are generated based on the BERT tokenizer, which has a fixed vocabulary size. Each input sequence is padded to a maximum length, which is specified during the preprocessing step.
* Input\_mask
  + Represents the tokenized input sequence, where each word in the sequence is mapped to a unique integer ID. These integer IDs are generated based on the BERT tokenizer, which has a fixed vocabulary size. Each input sequence is padded to a maximum length, which is specified during the preprocessing step.
* Input\_type\_ids
  + Input\_type\_ids tensor is used to distinguish between the different segments in the input sequence. It has the same shape as input\_word\_ids and input\_mask, and consists of 0s and 1s, where 0s indicate the first segment and 1s indicate the second segment. In the case of a single-segment input, all values in input\_type\_ids will be 0.
  + [CLS]: A special token added at the beginning of the input sequence, which represents the classification task.
  + [SEP]: A special token added between two sentences in the input sequence. Regular tokens from the input sequence

A picture containing table

Description automatically generated

A picture containing text

Description automatically generatedText

Description automatically generated

# Bert encoding:

* BERT encoder model from TensorFlow Hub to encode the preprocessed input text. It first initializes the BERT model using the hub.KerasLayer() function and passing the URL of the pre-trained BERT encoder model. Then, it passes the preprocessed text through this model using the bert\_model() function and storing the output in bert\_op. The output is a dictionary containing various embeddings and other features.

Graphical user interface, text, application

Description automatically generated

# Graphical user interface, text, application Description automatically generated

# Training BERT:

* To Train BERT a learner instance needs to be created using ktrain library with which we can train the BERT model

Text

Description automatically generated

Graphical user interface, text

Description automatically generated

text.texts\_from\_df() function is used to preprocess text data for a BERT model by taking in the training and validation dataframes, specifying the text and label columns, setting the maximum length of the sequences to 500, and returning the preprocessed data.

**Accuracy:**

* The "fit\_onecycle" method is called on a learner object to train a model using a one-cycle policy, which involves gradually increasing and then decreasing the learning rate over the course of a single epoch. The "lr" parameter specifies the maximum learning rate to be used, and the "epochs" parameter specifies the number of epochs to train the model for.

Graphical user interface, text, application

Description automatically generated

**Goal achieved:**

* To improve the data preprocessing and cleaning using best possible techniques.
* Worked on implementing BERT model getting an accuracy of 88.70%.

**Goal needs to be achieved:**

* Also, we will compare the accuracy of the same using different Model(ALBERT) in the upcoming Milestone

# **Timeline:**

* Topic Selection – 01/21/2023 – 02/04/2023(2 weeks)
* Data preprocessing – 02/04/2023 - 02/17/2023 (2 to 3 weeks)
* Algorithm implementation – 02/17/2023 - 03/09/2023 (3rd week to 5th week)
* Accuracy Evaluation – 03/09/2023 - 03/25/2023(6th week to 8th week)
* Paper Submission – 03/25/2023- 04/10/2023 (9th week to 11th week)

# **References:**

* Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. (2011). "Learning Word Vectors for Sentiment Analysis." The 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011). <https://ai.stanford.edu/~ang/papers/acl11-WordVectorsSentimentAnalysis.pdf>
* Systematic reviews in sentiment analysis: a tertiary study Alexander Ligthart, Cagatay Catal, Bedir Tekinerdogan

<https://link.springer.com/article/10.1007/s10462-021-09973-3>

* Sentiment Analysis for movies reviews dataset using Deep Learning models

<https://www.researchgate.net/publication/333607586_SENTIMENT_ANALYSIS_FOR_MOVIES_REVIEWS_DATASET_USING_DEEP_LEARNING_MODELS>

* “Machine Learning Approach to Sentiment Analysis of Users Movie Reviews.” - Adetunmbi, Adebayo, Oluwafemi A. Sarumi, and O. Boyinbode. <https://www.researchgate.net/publication/327477969_Machine_Learning_Approach_to_Sentiment_Analysis_of_Users_Movie_Reviews>
* "Sentiment Analysis of IMDB Movie Reviews Using BERT-Based Models" by A. K. Mishra, published in the Journal of Information and Communication Technology.
* C. Lin and Y. He. 2009. Joint sentiment/topic model for sentiment analysis. In Proceeding of the 18th ACM Conference on Information and Knowledge Management