**SENTIMENT ANALYSIS ON IMDB MOVIE REVIEWS**

GANESH KRISHNA LAKSHMISETTY (1002086481)

**IMPLEMENTATION PHASE**

**PRETRAINED WORD EMBEDDINGS USED :**

1.Word2Vec

2.Glove

**MODELS USED :**

1. LSTM
2. BIDIRECTIONAL LSTM
3. CNN\_LSTM

**INTRODUCTION TO WORD EMBEDDINGS – Word2vec :**

* Word2Vec is a technique for natural language processing (NLP) that aims to understand the semantic relationships between words.It was introduced by Tomas Mikolov and his team at Google in 2013.
* **Key Concepts:**
* **Word Embeddings**:Word2Vec converts words into numerical vectors in a continuous vector space.It represents words in such a way that similar words are close to each other in this space.
* **Skip-gram vs. Continuous Bag of Words (CBOW):**Word2Vec has two models: Skip-gram and CBOW.Skip-gram predicts the context words given a target word, while CBOW predicts a target word given its context.
* **Training Process:**Word2Vec is trained using a large corpus of text data.It learns to predict neighboring words based on the current word in the training data.
* **Applications:**Word2Vec has found applications in various NLP tasks like sentiment analysis, machine translation, and recommendation systems.

**INTRODUCTION TO WORD EMBEDDINGS – GLOVE :**

* GloVe is another word embedding technique introduced by researchers at Stanford University in 2014.It combines the advantages of both global matrix factorization techniques and local context window methods.
* **Key Concepts:**
* **Word Co-occurrence Matrix:** GloVe leverages global word co-occurrence statistics to learn word embeddings.It constructs a co-occurrence matrix which represents how often words co-occur in a given context.
* **Objective Function:**GloVe's training objective is to learn word vectors that capture the ratio of co-occurrence probabilities.
* **Advantages of GloVe**:It's capable of capturing global semantic relationships and local context features simultaneously. GloVe tends to perform well on tasks that require understanding both word-level and document-level semantics.
* **Applications:**GloVe embeddings have been used in a wide range of applications including sentiment analysis, document classification, and machine translation.

**Introduction to models used :**

**LONG SHORT-TERM MEMORY(LSTM) :**

* LSTMs, a type of RNN, capture order dependence, using prior output as current input.
* They address long-term dependency issues and they excel in processing time-series data, making accurate predictions while retaining long-term memory.
* LSTM unit comprises a cell, input gate, output gate, and forget gate, managing the flow of information and memory storage over time intervals.
* The LSTM algorithm solves the vanishing gradient problem RNN’s have and captures long term dependencies.

**BI-DIRECTIONAL LONG SHORT-TERM MEMORY(LSTM) :**

* Bidirectional LSTMs (BiLSTMs) are a valuable enhancement to traditional LSTMs for sequence modeling and prediction tasks.
* BiLSTMs employ two independent recurrent neural nets, processing training sequences in both forward and backward directions.
* This bidirectional approach equips BiLSTMs with a full contextual understanding of data points before and after each point in a sequence, eliminating the need to specify a time window or delay size.
* Conventional RNNs can only utilize past contexts, but BiLSTMs process data bidirectionally, using two hidden layers that contribute to the same output layer.

**CNN-LSTM :**

* Combining Convolutional Neural Networks (CNNs) with LSTM networks is a powerful approach for handling sequential data with spatial and temporal characteristics.
* CNNs are used as a front-end to extract relevant spatial features from the input data. These features are then passed to the LSTM for sequential analysis.LSTMs receive the output of the CNN as input and process it in a sequential manner, learning patterns and dependencies over time.
* Training CNN-LSTM models can be computationally intensive, and hyperparameter tuning is crucial to achieving optimal performance.

**IMPLEMENTATION :**

1. Using the saved Preprocessed dataset from the last phase to save time.
2. Adding the preprocessed\_review\_length column to dataset so that we can get idea to take maximum review length as input to the model.
3. Splitting the data into train, test.
4. Setting a max review length and truncating for larger reviews and padding for smaller reviews.
5. Next, Building a vocabulary of words from movie reviews using Tokenizer class from keras, and placing special token for out of Vocabulary words.
6. Mapping the reviews in Xtrain, Xtest to a sequence of integers using text\_to\_sequence and padding/truncating.
7. Next, using pretrained word embeddings like Word2Vec, glove that can understand relation between words.
8. Done by placing word vectors of Word2Vec, Glove in place of the index of the words in vocabulary that are matched with words in Word2Vec & not matched words will have zero vectors.

* **Implementing DeepLearning Models : LSTM, BI-Directional LSTM, CNN\_LSTM.**
* **-For LSTM:**
* Embedding layer, which contains wights of pretrained word vectors from Word2Vec, Glove.
* A LSTM layer with no of lstm units, and dropout value.
* A Dense layer as output with sigmoid activation.
* **-For BI-Directional LSTM :**
* Embedding layer, which contains wights of pretrained word vectors from Word2Vec, Glove.
* 2 BiDirectional LSTM layers with no of lstm units, and dropout value.
* A Dense layer as output with sigmoid activation.
* **-Finally for CNN\_LSTM :**
* Embedding layer, which contains wights of pretrained word vectors from Word2Vec, Glove.
* A CNN layer with filters, padding, and activation.
* A LSTM layer with no of lstm units, and dropout value.
* A Dense layer as output with sigmoid activation.

**1.Setting the max length for reviews:**

Checking the distribution of the review lengths after preprocessing to get an idea for setting max input length for a review.

A graph of a number of different sizes

Description automatically generated with medium confidence

So, the mean of reviews is around 120, so value below that has not be taken because information will be lost. We tried experimenting various lengths and set value to 250.

**2.Building vocabulary:**

With the help of **Tokenizer** class from Keras, we build the vocabulary of distinct words from all the movie reviews and set the **special token**  for words out of the vocabulary.

A screenshot of a computer

Description automatically generated

**3.MAPPING REVIEWS TO SEQUENCE OF INTEGERS:**

A movie review **before** mapping to sequence of integers.

A close up of text

Description automatically generated

A movie review **after** mapping by text\_to\_sequence and **padded** as max length of review is set to 250

A close up of a number

Description automatically generated

**4.USING PRE-TRAINED WORD EMBEDDINGS- “WORD2VEC” :**

These are the words in Word2Vec, and representation of words in vectors.

A screenshot of a computer

Description automatically generated

The **words in our vocabulary are replaced with the word vectors** of Word2Vec Embeddings if they are matched. Not matched words are placed with zeroes.

The total of matched words with Word2Vec are **55414**

A screenshot of a computer

Description automatically generated

The total of matched words with Word2Vec are **55414**

A white background with black text

Description automatically generated

**4. USING PRE-TRAINED WORD EMBEDDINGS- “GLOVE” :**

These are the words in Glove, and representation of words in vectors.

A screen shot of a computer

Description automatically generated

The **words in our vocabulary are replaced with the word vectors** of Glove Embeddings if they are matched.Not matched words are placed with zeroes.

A screenshot of a computer

Description automatically generated

The total of matched words with Word2Vec are **69691 .**

A black text on a white background

Description automatically generated

**5. BUILDING MODELS : LSTM, BIDIRECTIONAL LSTM, CNNLSTM:**

**LSTM:**

Here the LSTM model we used contains three layers: embedding layer, which takes weights from either **Word2Vec,** or **Glove** . LSTM layer and a single Dense layer with sigmoid activation for output.

A screenshot of a computer program

Description automatically generated

**BI-DIRECTIONAL LSTM :**

Here the BI-LSTM model we used contains four layers: embedding layer, which takes weights from either **Word2Vec,** or **Glove** . 2 Biderctional LSTM layers and a single Dense layer with sigmoid activation for output.

A screenshot of a computer

Description automatically generated

**CNN\_LSTM**

Here the CNN-LSTM model we used contains five layers: embedding layer, which takes weights from either **Word2Vec,** or **Glove** . CNN and MaxPooling layers, LSTM layer, and a single Dense layer with sigmoid activation for output.

A screenshot of a computer

Description automatically generated

**TRAINING:**

We have fit the models we build with 47 no of epochs, and used loss as cross entropy and optimizer is adam.

You can see the accuracy for training as well as accuracy for the testing , i.e for validation in the image below.

A screenshot of a computer

Description automatically generated

**6. RESULTS :**

**WORD2VEC EMBEDDING**

|  |  |
| --- | --- |
| **MODEL** | **ACCURACY** |
| **LSTM** | 89.200% |
| **BIDIRECTIONAL LSTM** | **89.856%** |
| **CNN\_LSTM** | 89.039% |

For all these models , **binary cross entropy** is taken as loss, optimizer is **adam**, and no of epochs taken are 47.

So, from the results the best performed model with **Word2Vec** Embeddings is **BiDirectional LSTM** with accuracy **89.86%**

A screenshot of a computer program

Description automatically generated

ROC curve with AUC: **AUC=0.959 [BIDIRECTIONAL\_LSTM - WORD2VEC]**

**A blue line with orange lines

Description automatically generated**

**GLOVE EMBEDDING**

|  |  |
| --- | --- |
| **MODEL** | **ACCURACY** |
| **LSTM** | **89.039%** |
| **BIDIRECTIONAL LSTM** | **88.444%** |
| **CNN\_LSTM** | **88.424%** |

For all these models , binary cross entropy is taken as loss, optimizer is adam, and no of epochs taken are 47.

So, from the results the best performed model with **GLOVE** Embeddings is  **LSTM** with accuracy **89%**

A screenshot of a computer screen

Description automatically generated

ROC curve with AUC: **AUC=0.953**

A graph of a curve

Description automatically generated

**Comparison with the references :**

In the reference paper we have considered, authors have implemented Word2Vec with LSTM, MLP, CNN\_LSTM. They have got the maximum accuracy of **89.2%** with multiple tunings for hybrid CNN\_LSTM model.

A table with text and numbers

Description automatically generated

In our implementation we have considered, using two different pretrained Word Embeddings - Word2Vec, and Glove with LSTM, BI-Directional LSTM, CNN\_LSTM. For **Glove** Embedding we have got the maximum accuracy for LSTM of **89%** and for **Word2Vec** Embeddings we have got the maximum accuracy for BI-Directional LSTM of **89.86%.**

|  |  |
| --- | --- |
| MODEL | ACCURACY |
| LSTM | 89.200% |
| BIDIRECTIONAL LSTM | 89.856% |
| CNN\_LSTM | 89.039% |

**EVALUATION OF MODEL:**

Evaluating the model on a 8 star movie review(2023 movie) taken from IMDB to see whether it classifies it as a positive sentiment or a negative sentiment.

A screenshot of a computer

Description automatically generated

Model correctly predicted the sentiment positive for a 8 star movie review.

Evaluating the model on a 2 star movie review(2023 movie) taken from IMDB to see whether it classifies it as a positive sentiment or a negative sentiment.

A screenshot of a computer

Description automatically generated

Model correctly predicted the sentiment negative for a 2 star movie review.

**REFERENCES:**

•Ali, Nehal & Hamid, Marwa & Youssif, Aliaa. (2019). SENTIMENT ANALYSIS FOR MOVIES REVIEWS DATASET USING DEEP LEARNING MODELS. International Journal of Data Mining & Knowledge Management Process. 09. 19-27. 10.5121/ijdkp.2019.9302. [(PDF) SENTIMENT ANALYSIS FOR MOVIES REVIEWS DATASET USING DEEP LEARNING MODELS (researchgate.net)](https://www.researchgate.net/publication/333607586_SENTIMENT_ANALYSIS_FOR_MOVIES_REVIEWS_DATASET_USING_DEEP_LEARNING_MODELS)

•

•Y. E. Rizk and W. M. Asal, "Sentiment Analysis using Machine Learning and Deep Learning Models on Movies Reviews," 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2021, pp. 129-132, doi: 10.1109/NILES53778.2021.9600548.  [Sentiment Analysis using Machine Learning and Deep Learning Models on Movies Reviews | IEEE Conference Publication | IEEE Xplore](https://ieeexplore.ieee.org/document/9600548/figures)

•

•Tarimer, İlhan & Çoban, Adil & Kocaman, Arif. (2019). Sentiment Analysis on IMDB Movie Comments and Twitter Data by Machine Learning and Vector Space Techniques. <https://www.researchgate.net/publication/332070477_Sentiment_Analysis_on_IMDB_Movie_Comments_and_Twitter_Data_by_Machine_Learning_and_Vector_Space_Techniques>