T.R.

GEBZE TECHNICAL UNIVERSITY

FACULTY OF ENGINEERING

DEPARTMENT OF COMPUTER ENGINEERING

BERT ÇIZGE TRANSFORMER UYGULAMASININ KLASIK BERT MODELI ILE KARŞILAŞTIRILMASI

MERVE DUR

SUPERVISOR
DR.ÖĞR.ÜYESI BURCU YILMAZ

GEBZE
2022

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

BERT ÇIZGE TRANSFORMER UYGULAMASININ KLASIK BERT MODELI ILE KARŞILAŞTIRILMASI

MERVE DUR

SUPERVISOR
DR.ÖĞR.ÜYESI BURCU YILMAZ

2022
GEBZE

GRADUATION PROJECT
JURY APPROVAL FORM

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 14/01/2022 by the following jury.

JURY

Member
(Supervisor) : Dr.Öğr.Üyesi Burcu YILMAZ

Member : Prof. Dr. Hasari ÇELEBİ

ABSTRACT

The aim of this project is to compare the BERT graphic transformer application with the classical BERT model. These two models were compared by making text classification of the sentences in the appropriate dataset. BERT is based on a pre-trained, open source NPL (Natural Language Processing) model. While it examines the whole sentence to find the missing word in the search query, it differs from other models with this feature. BERT, which also uses machine learning algorithms, has a two-way language processing feature. It tries to understand the relationship of each word with another word. It uses a more complex masked language model as opposed to a superficial bidirectional language processing that goes right to left, left to right.

The classical Bert model and the Graph Bert model were used for this comparison. The two models were compared by performing text classification. Text classification is a machine learning technique of categorizing open-ended text into a predefined category. Text classifiers can be used to organize, structure and categorize almost any type of text from documents, medical studies and files, and the entire web. Text classification was run on these two models (datasets fitted to both models) and results were compared.

BERT and TRANSFORMER: In NLP, the dominant sequence transduction models are based on complex recurrent .However, the inherently sequential nature precludes parallelization within training examples. Graph Bert is a new network architecture, this transformer based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. In recent years, TRANSFORMER and BERT based learning approaches have been used extensively in various learning tasks.

While comparing these two models, train loss and accuracy values were compared with each other. Train loss graphs are drawn.

Keywords: Bert, Graph Bert,Text Classification.

ÖZET

Bu projenin amacı BERT grafik tranformatör uygulamasının klasik BERT modeli ile karşılaştırılmasıdır. Uygun verisetindeki cümlelerin metin sınıflandırması yapılarak bu iki model karşılaştırıldı. BERT, önceden eğitilmiş, açık kaynak kodlu bir NPL(Doğal Dil İşleme) modeline dayanır. Arama sorgusundaki eksik olan kelimeyi bulabilmek için tüm cümleyi incelerken, bu özelliği ile de diğer modellerden ayrılmaktadır.Makine öğrenimi algoritmalarını da kullanan BERT, iki yönlü bir dil işleme özelliğine sahiptir. Her kelimenin diğer bir kelimeyle ilişkisini anlamaya çalışır. Sağdan sola, soldan sağa giden yüzey- sel çift yönlü bir dil işlemesinin aksine daha karmaşık maskeli dil modeli kullanır.

Bu karşılaştırma için klasik Bert modeli ve Graph Bert modeli kullanıldı. Metin sınıflandırılması yapılarak iki model karşılaştırıldı.Metin sınıflandırması, açık uçlu metne önceden tanımlanmış bir kategorilere ayırma makine öğrenimi tekniğidir . Metin sınıflandırıcılar, belgelerden, tıbbi çalışmalardan ve dosyalardan ve tüm web'den hemen hemen her tür metni düzenlemek, yapılandırmak ve kategorilere ayırmak için kullanılabilir.Metin sınıflandırılması bu iki model üzerinde çalıştırılarak (datasetleri iki modele uygun hale getirildi) sonuçlar karşılaştırıldı.

Bu iki model karşılaştırılırken train loss vs accuracy değerleri birbirleriyle kıyaslandı. Train loss grafikleri çizildi.

Anahtar Kelimeler: Bert, Graph Bert,Metin Sınıflandırma

ACKNOWLEDGEMENT

I would like to express my sincere thanks to those who contributed to the preparation of this report, to my dear Professor Burcu YILMAZ, who guided the final version of the report, and to Gebze Technical University for supporting this study. In addition, I would like to express my respect and love to my family, who supported me in every way during my education, and to all my teachers who set an example for me with their lives.

Merve Dur

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or Abbreviation	:	**Explanation**
BERT	:	Bidirectional Encoder Representations from Transformers
NPL	:	Natural Language Processing model
MLM	:	Masked Language Model
GNN	:	Graph Neural Network
NSP	:	Next Sentence Prediction

CONTENTS

Abstract iv

Özet v

Acknowledgement vi

List of Symbols and Abbreviations vii

Contents viii

List of Figures ix

List of Tables x

1 Introduction 1

1.1 Bert Classical Model 2

1.1.1 How BERT Works? 2

1.1.2 Masked LM (MLM) 2

1.1.3 Next Sentence Prediction (NSP) 3

1.1.4 How to use Fine Tuning in BERT? 4

1.2 Graph Bert(Graph Transformer) 4

1.2.1 BERT and TRANSFORMER 5

2 PROJECT IMPLEMENTATION 7

2.1 Project Design 7

2.2 Preparation of the Datasets 7

2.2.1 Installation and Operation of Bert Models 8

3 Project Result 10

3.1 Results 10

4 CONCLUSIONS 12

Bibliography 13

LIST OF FIGURES

1.1 Bert Model 1
1.2 Bert Working Mechanism 3
1.3 Masking Bert Model 3
1.4 Bert Model 4
1.5 Graph Bert 6

2.1 Project Design 7
2.2 Bert Model Used 8
2.3 Google Colab 9
2.4 Simple Transformer 9

3.1 Graph Bert Model Result 10
3.2 Classical Bert Model Result 11
3.3 Classical Bert Model Loss Graph 11

LIST OF TABLES

1. INTRODUCTION

BERT is an open source machine learning model for natural language processing (NLP). BERT is designed to help computers understand the meaning of ambiguous language in text. The BERT framework is pre-trained using text from Wikipedia and can be fine-tuned with question-answer datasets.

BERT, which stands for Bidirectional Encoder Representations from Transformers, is based on Transformers, a deep learning model where each output element is linked to each input element and the weights between them are calculated dynamically based on their connections.

Previous language models could only read left-to-right or right-to-left text input, but not both at the same time. BERT is more advanced than other models as it is designed to read in both directions at the same time. This ability, provided with the release of Transformers, is known as bidirectionality. Using this dual capability, BERT is pre-trained on two different but related NLP tasks: Masked Language Modeling and Next Sentence Prediction. The goal of Masked Language Model (MLM) training is to hide a word in a sentence and then tell the program which word based on the context of the hidden word. is to make it guess that the word is hidden (masked). The purpose of the Next Sentence Prediction tutorial is to allow the program to predict whether two given sentences have a logical, sequential connection or whether their relationship is simply random.

GRAPH-BERT, on the other hand, is a Bert model based only on the attention mechanism, without any graphical convolution or addition operators. GRAPH-BERT has been proven to outperform existing GNNs in terms of both learning efficiency and efficiency. 1.1.

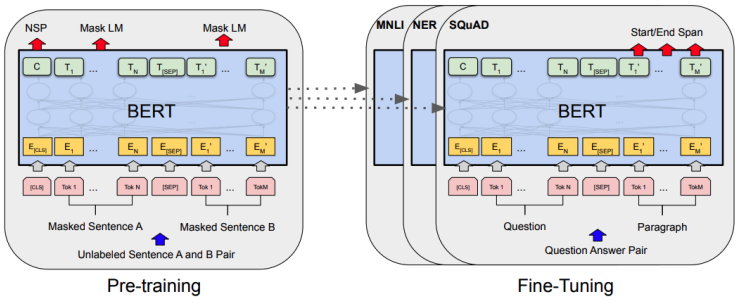

Figure 1.1: Bert Model

1.1. Bert Classical Model

BERT stands for "Bidirectional Encoder Representations from Transformers". It is designed to pre-train deep bidirectional representations from unlabeled text by co-conditioning in both left and right context. The pre-trained BERT model is designed to build state-of-the-art models for a wide variety of NLP tasks. It can be used anywhere by fine-tuning all datasets. BERT is based on transformer architecture. BERT is pre-trained on a large collection of unlabeled text, including the entire Wikipedia (2500 million words) and a collection of books (800 million words). Bidirectional means that BERT learns information from both the left and right sides of a token content during the training phase. The duality of a model is important to truly understanding the meaning of a language.

1.1.1. How BERT Works?

BERT uses transformer, an attention mechanism that learns the contextual relationships between words in a text. The Transformer includes two separate mechanisms: an encoder that reads the text input and a decoder that generates a prediction for the task. Since the purpose of BERT is to build a language model, only the encoder mechanism is required. Unlike directional models, which read text input sequentially (left to right or right to left), the transformer encoder reads the entire string of words at once. It is therefore considered bidirectional, but it would be more accurate to say that it is non-directional. This feature allows the model to learn the context of a word based on its entire circumference (left and right of the word). BERT uses two training strategies: 1-Masked LM (MLM) 2-Next Sentence Prediction (NSP)

1.1.2. Masked LM (MLM)

Masked LM (MLM) Before feeding the Masked LM (MLM) word strings to BERT it does the following:

1-Adds a classification layer on top of the encoder output.

2- Converts the output vectors to word size by multiplying them with the embedding matrix.

3-Calculates the probability of each word in the vocabulary with softmax.

The BERT loss function only considers the prediction of masked values and ignores the prediction of unmasked words. As a result, the model converges more slowly than the directional models; this is a feature that is balanced by increased context awareness.1.2.

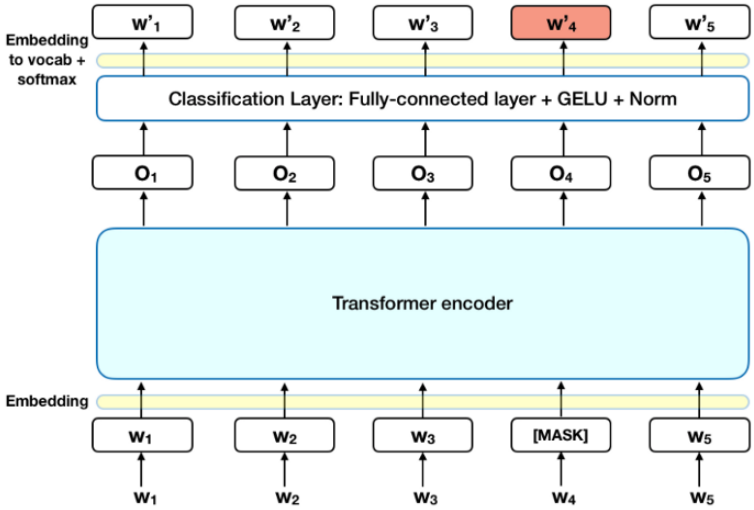

Figure 1.2: Bert Working Mechanism

1.1.3. Next Sentence Prediction (NSP)

In the BERT training process, the model takes pairs of sentences as input and learns to predict whether the second sentence in the pair is the next sentence in the original document. During training, 50

1-A [CLS] token is inserted at the beginning of the first sentence and a [SEP] token is inserted at the end of each sentence. 2-A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2. 3-A positional embedding is added to each token to indicate its position in the sequence. The concept and implementation of positional embedding are presented in the Transformer paper.

1.3.

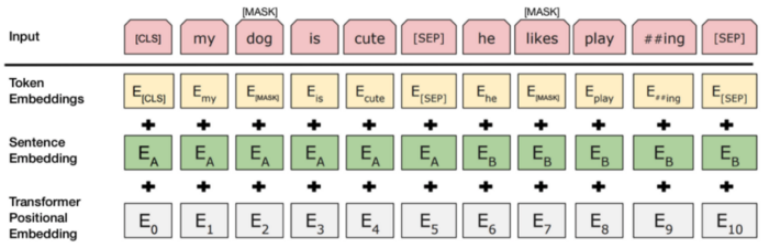

Figure 1.3: Masking Bert Model

To predict if the second sentence is indeed connected to the first, the following steps are performed:

1-The entire input sequence goes through the Transformer model.

2-The output of the token is transformed into a 2×1 shaped vector, using a simple classification layer (learned matrices of weights and biases).

3-Calculating the probability of IsNextSequence with softmax.

When training the BERT model, Masked LM and Next Sentence Prediction are trained together with the goal of minimizing the combined loss function of the two strategies.

1.1.4. How to use Fine Tuning in BERT?

1-Classification tasks such as sensitivity analysis are done similarly to the next sentence classification by adding a classification layer on top of the Transformer output for the CLS token.

2-In Question Answer tasks, the software receives a question about a string of text and must mark the answer in the string. Using BERT, a QA model can be trained by learning two extra vectors that mark the start and end of the answer.

3-In Named Entity Recognition (NER), the software takes a string of text and needs to mark various types of entities (Person, Organization, Date, etc.) appearing in the text. Using BERT, a NER model can be trained by feeding the output vector of each token into a classification layer that predicts the NER tag.

1.4.

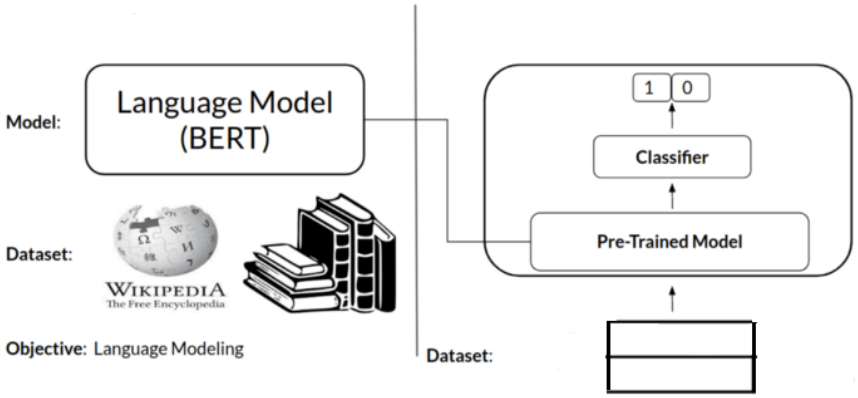

Figure 1.4: Bert Model

1.2. Graph Bert(Graph Transformer)

Graph provides a unified representation for many interconnected data in the real-world, which can model both the diverse attribute information of the node entities

and the extensive connections among these nodes.Traditional machine learning models can hardly be applied to the graph data directly, which usually take the feature vectors as the inputs. Viewed in such a perspective, learning the representations of the graph structured data is an important research task. In recent years, great efforts have been devoted to designing new graph neural networks (GNNs) for effective graph representation learning. Meanwhile, most of these existing graph representation learning models are still based on the graph structures, i.e., the links among the nodes. Via necessary neighborhood information aggregation or convolutional operators along the links, nodes' representations learned by such approaches can preserve the graph structure information. However, several serious learning performance problem, e.g., suspended animation problem and over-smoothing problem , with the existing GNN models have also been witnessed in recent years. According to [Zhang and Meng, 2019], for the GNNs based on the approximated graph convolutional operators, as the model architecture goes deeper and reaches certain limit, the model will not respond to the training data and suffers from the suspended animation problem. Meanwhile, the node representations obtained by such deep models tend to be over-smoothed and also become indistinguishable. Both of these two problems greatly hinder the applications of GNNs for deep graph representation learning tasks. What's more, the inherently interconnected nature precludes parallelization within the graph, which becomes critical for large-sized graph input, as memory constraints limit batching across the nodes.To prevent all these problems, a structure called GRAPHBERT has been developed. The GRAPH-BERT model is trained with nodes sampled from the input large graph data along with their context. Unlike current GNN models, GRAPH-BERT in the representation learning process does not use any connections in such samplings.

1.5.

1.2.1. BERT and TRANSFORMER

In NLP,the dominant sequence transduction models are based on complex recurrent.However, the inherently sequential nature precludes parallelization within training examples.Graph Bert is a new network architecture, this transformer based solely on attention mechanisms, dispensing with recurrence and convolutions entirely.In recent years, TRANSFORMER and BERT based learning approaches have been used extensively in various learning tasks.Officially, the clustering component used in GRAPH-BERT is KMeans, which takes raw feature vectors of nodes as input.Based on a batch of linkless subgraphs sampled from the original graph data, GRAPH-BERT can effectively learn the representations of the target node . GRAPH-BERT can serve as the graph representation learning component in graph learning pipeline. The pre-trained GRAPH-BERT can be transferred and applied to with necessary fine-tuning to different

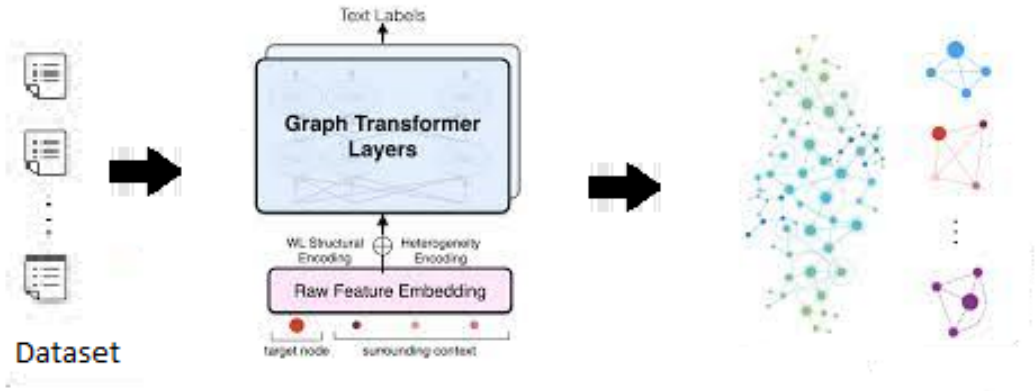

Figure 1.5: Graph Bert

dataset.

2. PROJECT IMPLEMENTATION

2.1. Project Design

2.1.

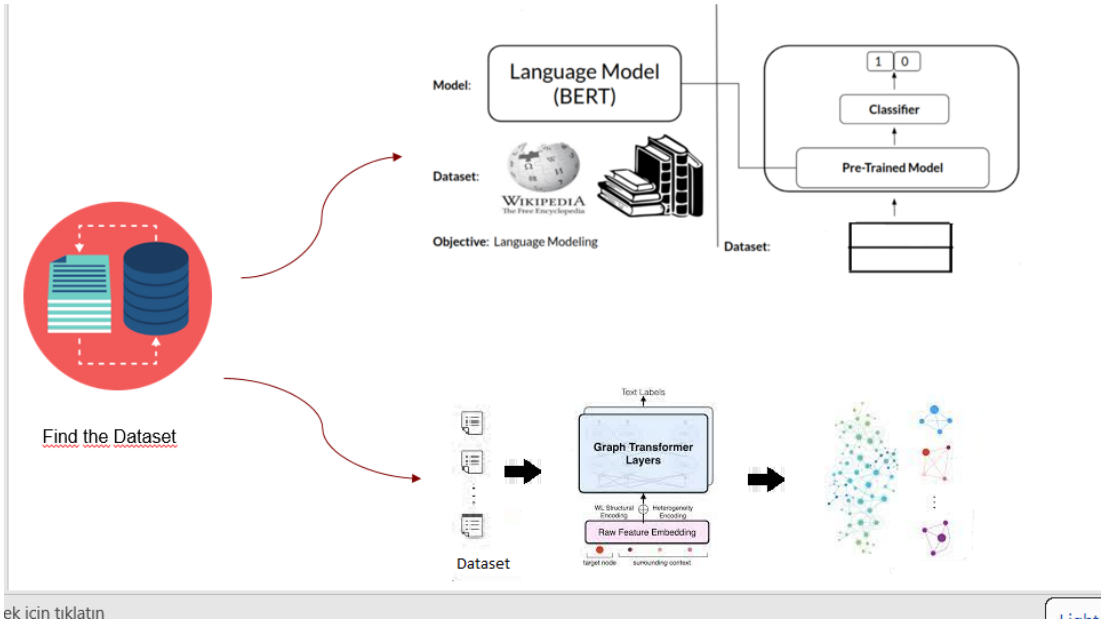

Figure 2.1: Project Design

2.2. Preparation of the Datasets

Firstly, I found a dataset suitable for text classification. The Cora dataset consists of machine learning articles. These articles are classified into one of the following seven classes:

1-Case Based
2-Genetic Algorithms
3-Neural Networks
4-Probabilistic Methods
5-Reinforcement Learning
6-Rule Learning
7-Theory

Articles have been selected so that each article is cited or cited from at least one other article in the final corpus. There are 2748 articles in the entire corpus. All words with document frequency less than 10 were removed. The articles and tags for the Classic Bert Model are saved in csv file.I divided the dataset into categories and assigned them to the labels. I split the model into two as training and testing, I set the training part to eighty percent and the test part to twenty percent. I did fine tüne the Bert model and used 15 epochs. This is how I trained the bert model.

Graph bert dataset consists of two files. The node file contains descriptions of the papers in the following format:

<paper id> <word attributes>+ <class label>

The first entry in each line contains the paper's unique string ID followed by binary values indicating whether or not each one is. the word in the vocabulary is present (indicated by 1) or absent (indicated by 0) on the paper. Finally, the last entry in the line contains the class label of the paper. The link file contains the excerpt graph of the collection. Each line defines a link in the following format: <Id of the cited article> <Id of the cited article> Each line has two paper IDs. The first entry is the ID of the cited article, and the second ID is the article containing the citation. The direction of the connection is from right to left. If a line is represented by "paper1 paper2", the link is "paper2->paper1".

2.2.1. Installation and Operation of Bert Models

Using Google colab, the bert model was downloaded and installed on hugging face. Added appropriate libraries to the project pandas and numpy. Added Simple transformer library. Simple transformer helps us train and evaluate transformer models quickly. The found dataset was saved in Google colaba and its data was read from the csv file.

Bert Model Used:

2.2.

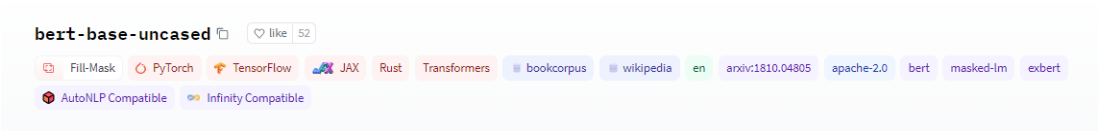

Figure 2.2: Bert Model Used

2.3.

2.4.

Figure 2.3: Google Colab

Figure 2.4: Simple Transformer

For the Graph bert model, the data resulting from clustering and vector embedding were run on the model.

[1]–[10]

3. PROJECT RESULT

3.1. Results

This loss epoch chart was created to compare. The success value was calculated. Graph bert success value was 0.80 and loss value was 0.68. The success value of classical bert was found to be 0.81 and loss value was 0.0 . 3.1.

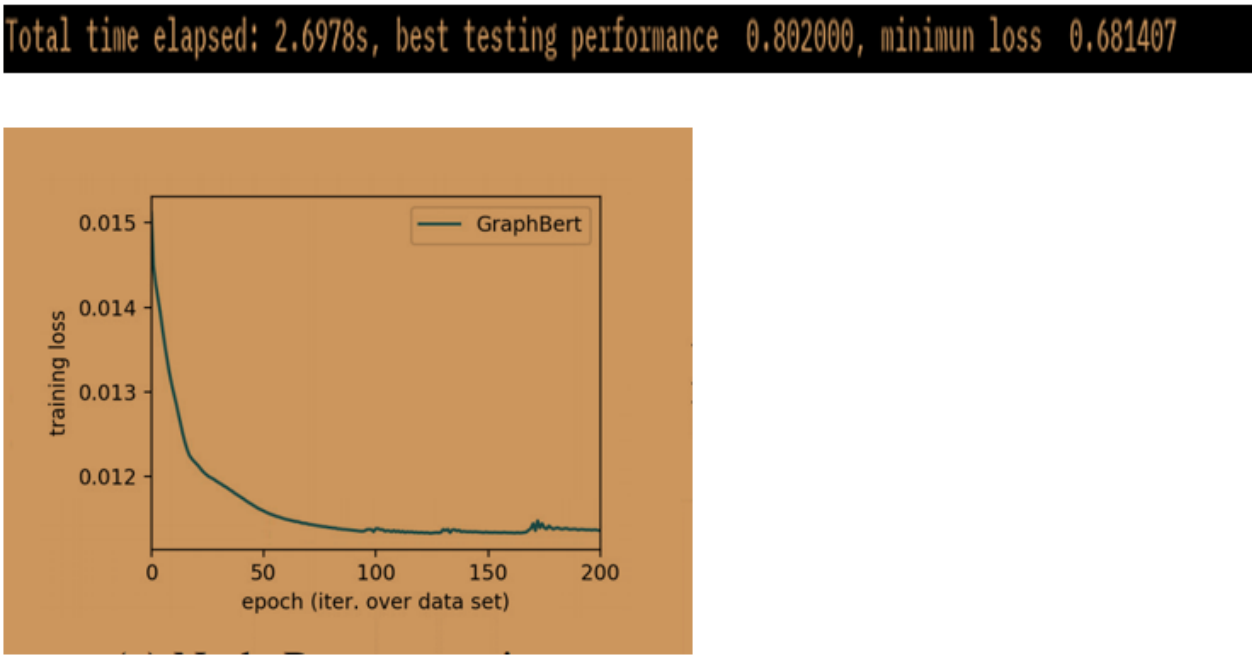

Figure 3.1: Graph Bert Model Result

Graph Bert Model Result**??** [1]

3.2.

3.3.

```
F-Score:  0.7807750847261562
Recall:  0.7701633555680081
Precision:  0.8028880757115575
```

	Case_Based	Genetic_Algorithms	Neural_Networks	Probabilistic_Methods	Reinforcement_Learning	Rule_Learning	Theory	accuracy	macro avg	weighted avg
precision	0.761905	0.853933	0.838150	0.797872	0.862069	0.709677	0.796610	0.814126	0.802888	0.814151
recall	0.827586	0.915663	0.884146	0.882353	0.581395	0.628571	0.671429	0.814126	0.770163	0.814126
f1-score	0.793388	0.883721	0.860534	0.837989	0.694444	0.666667	0.728682	0.814126	0.780775	0.810268
support	58.000000	83.000000	164.000000	85.000000	43.000000	35.000000	70.000000	0.814126	538.000000	538.000000

Figure 3.2: Classical Bert Model Result

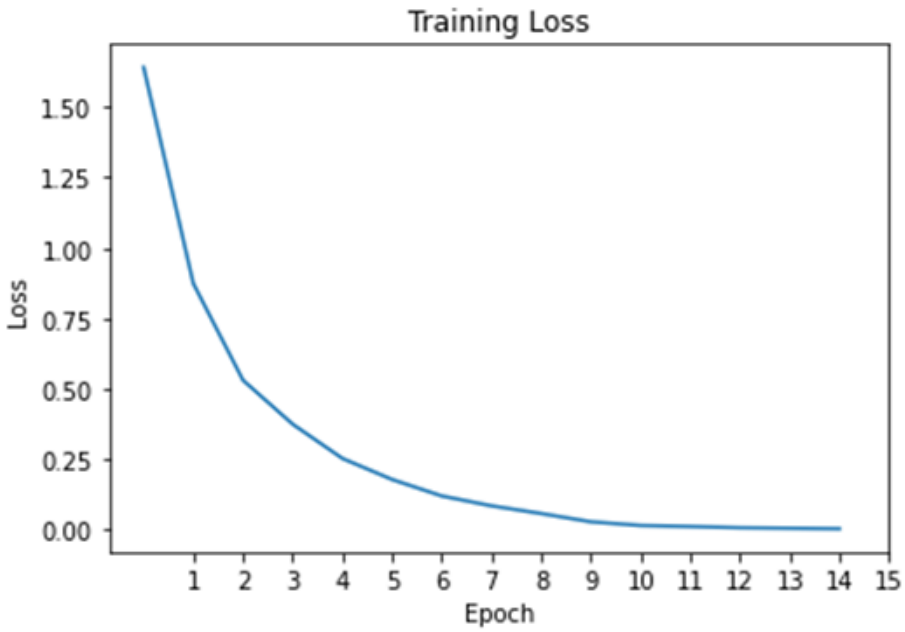

Figure 3.3: Classical Bert Model Loss Graph

4. CONCLUSIONS

Loss can be seen as a distance between the true values of the problem and the values predicted by the model. Greater the loss is, more huge is the errors you made on the data. Accuracy can be seen as the number of errors you make on the data. That means:

- a low accuracy and huge loss means you made huge errors on a lot of data
- a low accuracy but low loss means you made little errors on a lot of data
- a great accuracy with low loss means you made low errors on a few data (best case)

As a result, the classical bert model gave better results because it has high accuracy value and low loss value.

BIBLIOGRAPHY

[1] J. Zhang, “Graph-bert: Only attention is needed for learning graph representations (CTAN),” 2001.

[2] ". Z. " “Graph-bert: Only attention is needed for learning graph representations.” (2001), [Online]. Available: https://arxiv.org/abs/2001.05140.

[3] J. Zhang, “G5: A universal graph-bert for graph-to-graph transfer and apocalypse learning,” *TUGBoat*, vol. 14, no. 3, pp. 342–351, 1993.

[4] (), [Online]. Available: https://www.analyticsvidhya.com/blog/2019/09/demystifying-bert-groundbreaking-nlp-framework/#:~:text=%E2%80%9CBERT%20stands%20for%20Bidirectional%20Encoder,both%20left%20and%20right%20context..

[5] (), [Online]. Available: https://www.geeksforgeeks.org/explanation-of-bert-model-nlp/.

[6] (), [Online]. Available: https://medium.com/@toprakucar/bert-modeli-ile-t%C3%BCrk%C3%A7e-metinlerde-s%C4%B1n%C4%B1fland%C4%B1rma-yapmak-260f15a65611.

[7] (), [Online]. Available: https://www.researchgate.net/publication/338620970_Graph-Bert_Only_Attention_is_Needed_for_Learning_Graph_Representations.

[8] (), [Online]. Available: https://www.tensorflow.org/text/tutorials/classify_text_with_bert.

[9] (), [Online]. Available: https://paperswithcode.com/paper/graph-bert-only-attention-is-needed-for/review/.

[10] (), [Online]. Available: https://www.techtarget.com/searchenterpriseai/definition/BERT-language-model.