Self-Supervised Representation Learning

Kedarnath P
MTech Computer Science & Automation
(SR NO: 20902)
kedarnathp@iisc.ac.in

Abstract

This project compares the performance of self-supervised representation learning (SSRL) models, specifically fine-tuned BERT, with traditional supervised learning models such as Naive Bayes, Linear SVC, and LSTMs on the sentiment classification task using the IMDb dataset. The dataset is divided into different training splits (10%, 40%, and 80%) to evaluate the model's performance under varying amounts of labeled data. The study demonstrates that BERT consistently achieves the highest test accuracy and lowest test loss across all splits due to its powerful pre-trained representations. Supervised learning models, such as Naive Bayes and Linear SVC, perform reasonably well, but are outperformed by BERT. The results emphasize the effectiveness of self-supervised representation learning techniques like BERT for natural language processing tasks.

1 Introduction

This project explores the performance of self-supervised representation learning (SSRL) using fine-tuned BERT models compared to traditional supervised learning approaches, with a focus on sentiment analysis using the IMDb dataset. The primary goal is to understand the benefits and limitations of both methodologies in different scenarios. SSRL leverages unsupervised learning techniques on large-scale unlabelled data, allowing models to be fine-tuned on smaller labeled datasets for specific tasks. In contrast, supervised learning relies on labeled data and typically requires more data for high accuracy. By comparing these approaches, the study provides valuable insights into their effectiveness, guiding the development of efficient learning algorithms and the selection of appropriate methodologies for different tasks and data availability scenarios.

2 Project Approach

This project compares self-supervised representation learning (SSRL) models, such as BERT, with supervised learning models like LSTM, Naive Bayes, and Linear SVC on the sentiment classification task using the IMDb dataset. The dataset is divided into different training splits (10%, 40%, and 80%) to evaluate the models' performance under varying amounts of labeled data. The remaining data is split between validation and test sets.

BERT, developed by Google AI, is a powerful pre-trained language model that captures deep language understanding. It is pre-trained on large-scale unsupervised text corpora using two tasks: Masked Language Modeling (MLM) and Next Sentence Prediction (NSP). As mentioned in the paper, Pretraining BERT from scratch is computationally expensive and challenging, making fine-tuning a more practical approach for adapting the model to specific tasks. Hence, in this project, we will be experimenting with the section 3.2 and section 4 of BERT paper, i-e we fine-tune the pre-trained BERT model for sentiment classification.

The project approach involves preprocessing the IMDb dataset, dividing it into train-validation-test splits, training the selected models on these splits, and evaluating their performance by comparing

test loss and test accuracy. This allows us to observe the effectiveness of SSRL models compared to supervised learning models in the context of sentiment classification under varying amounts of labeled data.

3 Literature survey Overview

The paper "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Devlin et al. (2019) introduces BERT (Bidirectional Encoder Representations from Transformers), a new pre-trained language model that has significantly advanced the state-of-the-art performance across various natural language processing (NLP) tasks.

BERT's architecture is based on the Transformer model, which was proposed by Vaswani et al. (2017) for machine translation. The key feature of BERT is its bidirectional nature, allowing the model to learn contextual representations by capturing information from both the left and right context of each token in a sentence. This bidirectional context learning is achieved using a technique called Masked Language Modeling (MLM), where a certain percentage of input tokens are randomly masked, and the model is trained to predict the masked tokens based on the context provided by the remaining unmasked tokens.

In addition to MLM, BERT is also pre-trained on a task called Next Sentence Prediction (NSP). In this task, the model is trained to predict whether a given pair of sentences are consecutive or not. NSP helps the model understand the relationship between sentences and improve its understanding of sentence-level context.

BERT is pre-trained on a large corpus of text data, specifically the BooksCorpus (800 million words) and English Wikipedia (2,500 million words). The pre-training process allows BERT to learn the structure of the language and understand context, which is then fine-tuned for specific NLP tasks using smaller labeled datasets. This transfer learning approach reduces the need for extensive labeled data and computational resources when adapting the model to various tasks.

The paper demonstrates that BERT significantly outperforms existing methods on a range of NLP benchmarks, including the General Language Understanding Evaluation (GLUE) benchmark, the Stanford Question Answering Dataset (SQuAD v1.1), and the CoNLL-2003 Named Entity Recognition (NER) task. BERT's performance can be attributed to its bidirectional nature, the use of the Transformer architecture, and its extensive pre-training on large-scale unsupervised text data

4 Methodology

An overview of the four different approaches for sentiment classification on the IMDB movie review dataset :

1. **Fine-tuned BERT:**
   The Implemented code fine-tunes a BERT model on the IMDb dataset to perform sentiment analysis. Let's break down the code into the main components.
   - *Imports:* Import the required libraries for the project.
   - *Load and preprocess the data:* Load the IMDb dataset, encode the sentiment labels, preprocess the reviews by removing HTML tags, non-ASCII characters, emojis, mentions, URLs, and punctuation marks. Then convert the text to lower case.
   - *Tokenize the text:* Tokenize the preprocessed text using BERT tokenizer, which returns input IDs and attention masks for each review.
   - *Create PyTorch dataset and dataloaders:* Define a custom TextDataset class, create train, validation, and test datasets and dataloaders using the preprocessed and tokenized data.
   - *Initialize the BERT model:* Load the pre-trained BERT model for sequence classification with two output labels (positive and negative sentiment). The BERT architecture for sequence classification includes the following layers:
     - BertEmbeddings: Combines token embeddings, position embeddings, and segment embeddings.

  - BertEncoder: Consists of multiple transformer layers to process the input sequence.
  - BertPooler: Applies a linear transformation followed by a Tanh activation to the first token's hidden state (the [CLS] token).
  - Dropout: Applies dropout regularization to the pooled output.
  - Linear: Fully connected layer to produce logits for the two output labels (positive and negative sentiment).
- *Set up training parameters:* Define the number of training epochs, optimizer, and learning rate scheduler.
- *Fine-tune the model:* Train the model on the IMDb dataset for the specified number of epochs. The training loop includes zeroing gradients, calculating loss, backpropagating, clipping gradients to avoid gradient explosion, updating model parameters, and updating the learning rate scheduler. After each epoch, evaluate the model on the validation dataset.
- *Visualize training and validation loss/accuracy:* Plot the training and validation loss and accuracy for each epoch.
- *Evaluate on the test dataset:* Use the fine-tuned model to make predictions on the test dataset and calculate the test loss and accuracy. Print the classification report and plot the confusion matrix.

2. **Naive Bayes:** Naive Bayes is a probabilistic machine learning model based on Bayes' theorem. It assumes that the features are conditionally independent, given the class label. In this approach, the IMDB dataset is preprocessed using TfidfVectorizer to create a Term Frequency-Inverse Document Frequency (TF-IDF) representation of the reviews. Then, the MultinomialNB class from the sklearn library is used to create a Naive Bayes model for sentiment classification. The model is trained on the training set and evaluated on the test set, with accuracy as the performance metric.
3. **Linear Support Vector Machine (SVC):** Support Vector Machines (SVM) are supervised learning models that can be used for classification tasks. Linear SVC is an SVM model with a linear kernel. In this approach, the IMDB dataset is preprocessed using TfidfVectorizer to create a Term Frequency-Inverse Document Frequency (TF-IDF) representation of the reviews. Then, the LinearSVC class from the sklearn library is used to create a Linear SVC model for sentiment classification. The model is trained on the training set and evaluated on the test set, with accuracy as the performance metric.
4. **LSTM:** LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) that is capable of learning long-term dependencies in sequences. In this approach, the IMDb dataset is preprocessed by tokenizing the reviews, building a vocabulary, and padding the sequences to have the same length. The model architecture consists of the following layers:
   - Embedding: Maps each token in the input sequence to a dense vector of fixed size (in this case, 256). The embedding layer learns these dense vector representations during training.
   - LSTM: A LSTM layer with 512 hidden units, batch_first set to True, and dropout rate of 0.25. This layer can capture the long-term dependencies in the input sequence. The model has two LSTM layers stacked on top of each other.
   - Dropout: Applies dropout regularization with a probability of 0.3 to help prevent overfitting.
   - Fully connected (Linear): A fully connected layer that maps the output of the LSTM layer to a single output unit. This layer is used for binary classification (positive or negative sentiment).
   - Sigmoid: A sigmoid activation function that squashes the output of the fully connected layer to a value between 0 and 1, representing the probability of the positive sentiment class.

   The model is trained using the Adam optimizer and BCELoss as the loss function. Early stopping is implemented based on the validation loss to prevent overfitting. Finally, the model is evaluated on the test set, and the classification report and confusion matrix are displayed.

5 Observations

The Implementation can be found here: code link

We analyze the performance of four models (BERT, Naive Bayes, Linear SVC, and LSTM) on the IMDb dataset with different training, validation, and test splits. The key observations are as follows:

1. BERT consistently achieves the highest test accuracy and lowest test loss across all splits due to its powerful pre-trained representations. As the percentage of training data increases, BERT's performance further improves.
2. Baseline models like Naive Bayes and Linear SVC show competitive results, especially with smaller training datasets. However, they do not surpass BERT's performance even with more training data.
3. LSTM underperforms compared to other models, but its performance improves with more training data. It requires a larger dataset to achieve comparable results. As generally, LSTMs are more prone to overfitting when trained with small datasets.

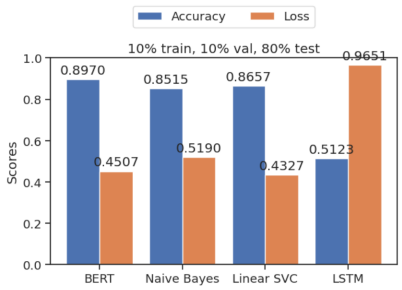

Figure 1: With 10% training, 10% validation, and 80% testing data.

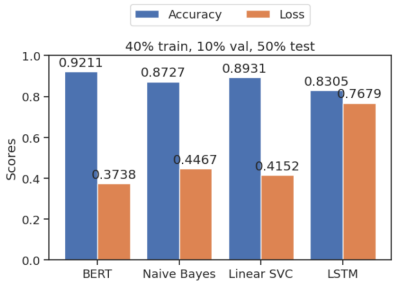

Figure 2: With 40% training, 10% validation, and 50% testing data.

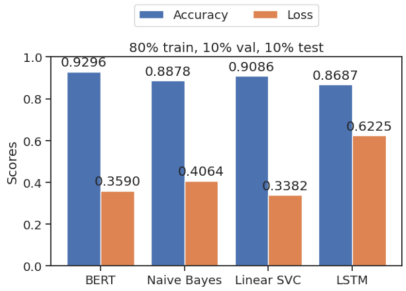

Figure 3: With 80% training, 10% validation, and 10% testing data.

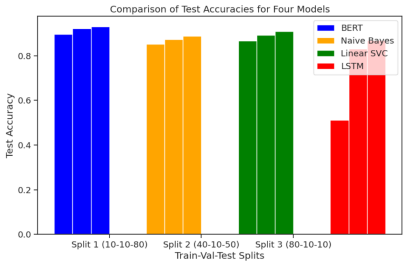

Figure 4: Performance of all four models across the three splits.

6 Results

The results show that BERT consistently achieves the highest test accuracy in all data splits, demonstrating the effectiveness of SSRL. Supervised learning models, such as Naive Bayes and Linear SVC also perform reasonably well, with Linear SVC slightly outperforming Naive Bayes, but both are outperformed by BERT. LSTM, on the other hand, shows varying performance, with improvements as the training data size increases.

7 Conclusions

The project demonstrates the effectiveness of self-supervised representation learning techniques in the context of natural language processing tasks. BERT emerges as the most effective model in this study, followed by Linear SVC and Naive Bayes, while LSTM shows relatively lower performance.

The comparison between these models on various dataset splits showcases the robustness of BERT, consistently outperforming other models in terms of both loss and accuracy. The performance advantage of BERT can be attributed to its pre-trained representations and self-supervised learning mechanism, which enable it to capture complex language patterns more effectively.

On the other hand, the Linear SVC and Naive Bayes classifiers show reasonable performance, but they do not benefit from the self-supervised learning mechanism. As a result, their performance is not as impressive as BERT.

The LSTM model's performance is lower than the other models, with higher loss and lower accuracy. The reason behind this could be the architecture's limitations in capturing long-range dependencies and inability to leverage pre-trained representations. Additionally, hyperparameter tuning may be required to optimize the LSTM model for this task.

In conclusion, this project highlights the power of self-supervised representation learning techniques like BERT for natural language processing tasks. The experiments conducted in this study show that BERT consistently outperforms traditional machine learning models and LSTM-based approaches. This underlines the importance of leveraging pre-trained representations and self-supervised learning mechanisms for achieving high performance in natural language processing tasks.

8 Future Work

Further experiments can be conducted to explore the potential of other self-supervised learning models, such as RoBERTa, or GPT, and how they compare to BERT in similar tasks. Additionally, investigating other NLP tasks and fine-tuning the models for domain-specific applications could provide insights into the generalizability of self-supervised representation learning techniques.

References

[1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

[2] Reference for Implementations:
Hugging Face
scikit-learn