1 Setup

Data from IMDb review were imported from the CSV file and divided into train and test data. The sentiment variable in the data was converted from string to binary 0 and 1. To obtain our model and tokenizer, "BertTokenizer" and "BertForSequenceClassification" were then from Hugging Face Transformer library. To preprocess and cater the data to the BERT model, the tokenizer was used. Preprocess included tokenizing words, assigning masking tokens and matching them with corresponding label. To divide the data into batches, a dataloader was used.

2 Testing Configurations

Data was split into 75% for training purposes and 25% for testing purposes. Since the dataloader shuffles the data, there were no need to scramble the data set from the beginning. Testing was conducted for three different batch sizes: 10, 20 and 30. The different batch sizes will later be tested on two variations of the dataset, later referred to as configuration A and B. In configuration A, max length for reviews is set to 120 words. This resulted in 10494 reviews of which 52.81% were labelled with a positive sentiment. In configuration B the max length was set to 250 words which resulted in 35208 reviews of which 49.87% were labelled positive.

The reason for dividing the dataset in this manner was partly due to time and memory constraint. However, our project aims to investigate the sentiment of the average review, which means that a few outliers with unconventionally long reviews would create more noise for our BERT model. These configurations allow the project to investigate several interesting parameters over multiple batch sizes. Moreover, these test are conducted in order to ensure that any observed differences in the baseline implementation compared to the later implemented model are not due to the choice of batch size alone.

3 Evaluation Metrics

To evaluate the effectiveness of the model, and later do comparisons between the baseline and later implemented model, 4 performance evaluation metrics will be used.

1. Accuracy, which measures proportion of correctly classified instances by the model.
2. Precision, which measures proportion of predicted positives correctly classified by the model.
3. Recall, which measures proportion of true positives correctly classified by the model.
4. F1-score, which takes both precision and recall into account and gives a weighted avarage.

In this context, accuracy will be the most interesting metric to look at (especially since our dataset is balanced). However, in order to get additional insights and a full understanding of the performance of our model, precision, recall and f1-score will serve as a complementary to the accuracy.

4 Results

Training the baseline implementation for 1 epoch using each configuration as explained in section 2 yielded results as follows.

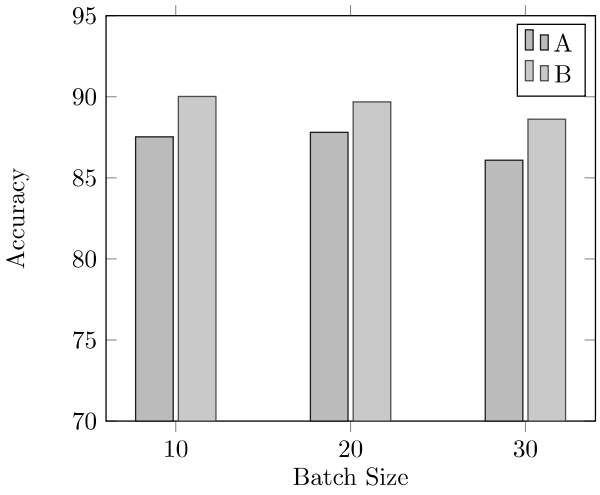

Figure 1: Accuracy on batch sizes and data set sizes

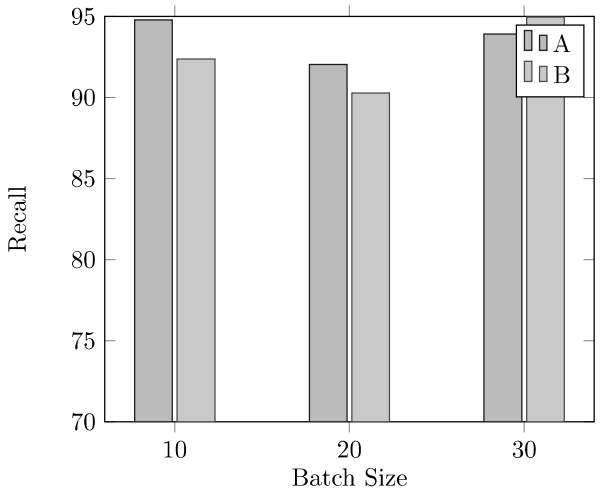

Figure 2: Recall on batch sizes and data set sizes

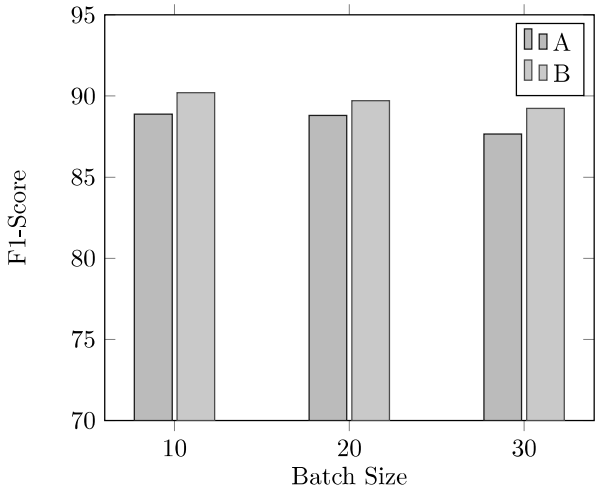

Figure 3: F1-score on batch sizes and data set sizes

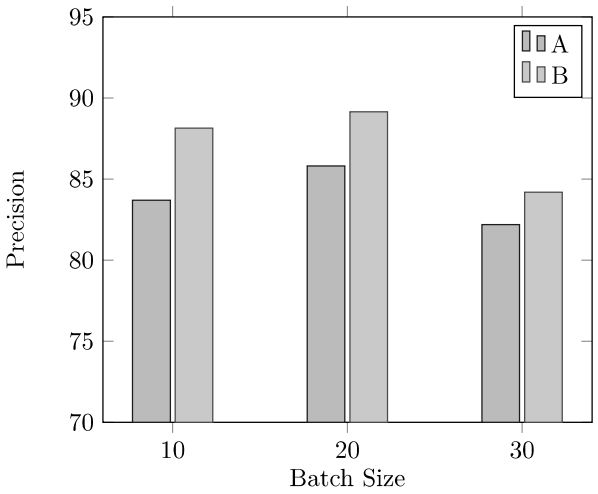

Figure 4: Precision on batch sizes and data set sizes

	Predicted negative	Predicted positive
True negative	989	255
True positive	72	1308

Figure 5: Confusion matrix of config A, batch size 10

	Predicted negative	Predicted positive
True negative	3879	544
True positive	334	4045

Figure 6: Confusion matrix of config B, batch size 10

	Predicted negative	Predicted positive
True negative	1034	210
True positive	110	1270

Figure 7: Confusion matrix of config A, batch size 20

	Predicted negative	Predicted positive
True negative	3942	481
True positive	426	3953

Figure 8: Confusion matrix of config B, batch size 20

	Predicted negative	Predicted positive
True negative	963	281
True positive	84	1296

Figure 9: Confusion matrix of config A, batch size 30

	Predicted negative	Predicted positive
True negative	3642	781
True positive	221	4158

Figure 10: Confusion matrix of config B, batch size 30