Assignment 4
Text and Sequence Data
Summary

Siva Sai Madhumitha Kotala
811257709

The Goal:

The objective of the IMDB dataset's binary classification task is to categorize movie reviews as either positive or negative. With 50,000 reviews in the dataset, only the top 10,000 words are considered. Training samples vary from 100 to 100,000, while validation involves 10,000 samples. Data preparation precedes embedding layer placement and the use of a pretrained embedding model. Multiple strategies are explored to gauge performance.

Data Preprocessing:

- As part of the dataset preparation procedure, each review undergoes a transformation into word embeddings, where every word is represented by a fixed-size vector. This meticulous process is bound by a limitation of 10,000 samples. Additionally, a numerical sequence is derived from the reviews, where individual numbers correspond to distinct words rather than complete strings of words. However, the neural network's input format does not readily accommodate this sequential list of numbers.

- To address this challenge, tensors must be constructed using the numerical sequence. The list of integers could potentially serve as the basis for generating a tensor with an integer data type and specific structure, organized as (samples, word indices). Nevertheless, achieving this requires ensuring that

each sample maintains a consistent length. This entails employing methods such as padding reviews with dummy words or numerical placeholders to standardize the length across all samples.

Procedure:

In this study, I explored two distinct methodologies for generating word embeddings in the context of the IMDB dataset.

1. Custom-trained embedding layer
2. pre-trained word embedding layer using the GloVe model

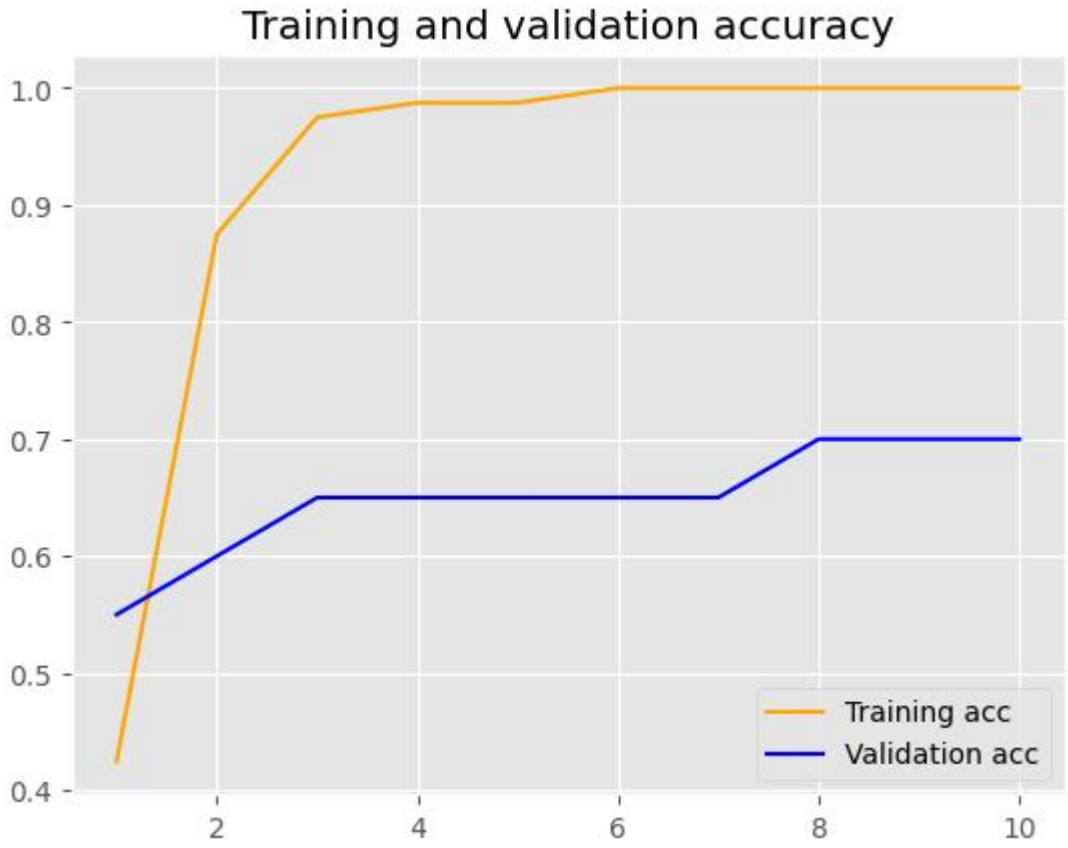
The GloVe model, a popular choice for word embeddings, which we employed in our research, is trained on extensive textual datasets.

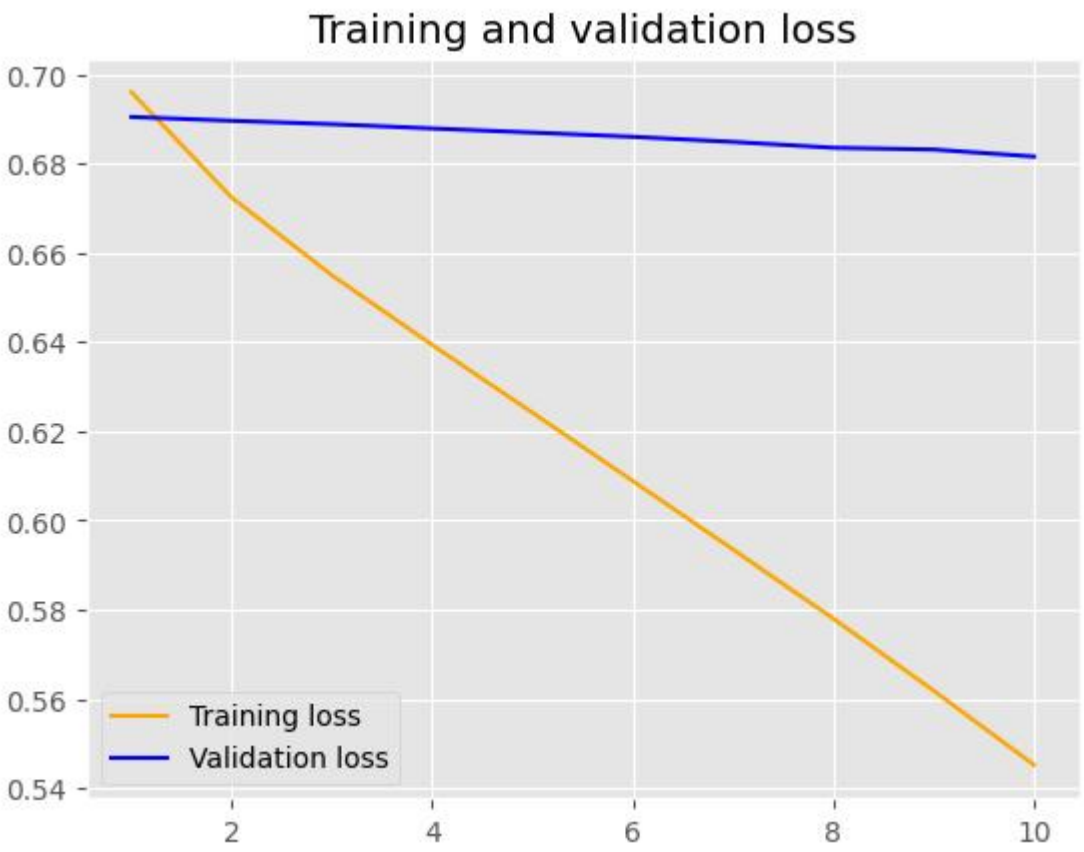
- For evaluating different embedding strategies, I employed two distinct embedding layers using the IMDB review dataset. One consisted of a custom-trained layer, while the other integrated a pre-trained word embedding layer. To gauge their effectiveness, I compared the accuracy of these two models across varying training sample sizes, including 100, 5000, 1000, and 10,000.

- Initially, we crafted a custom-trained embedding layer utilizing the IMDB review dataset. Subsequently, each model underwent training across a range of dataset samples, followed by an assessment of its accuracy using a dedicated testing set. Following this evaluation, we juxtaposed these precision results with those obtained from a model that underwent similar testing across varied sample sizes but incorporated a pre-trained word embedding layer.

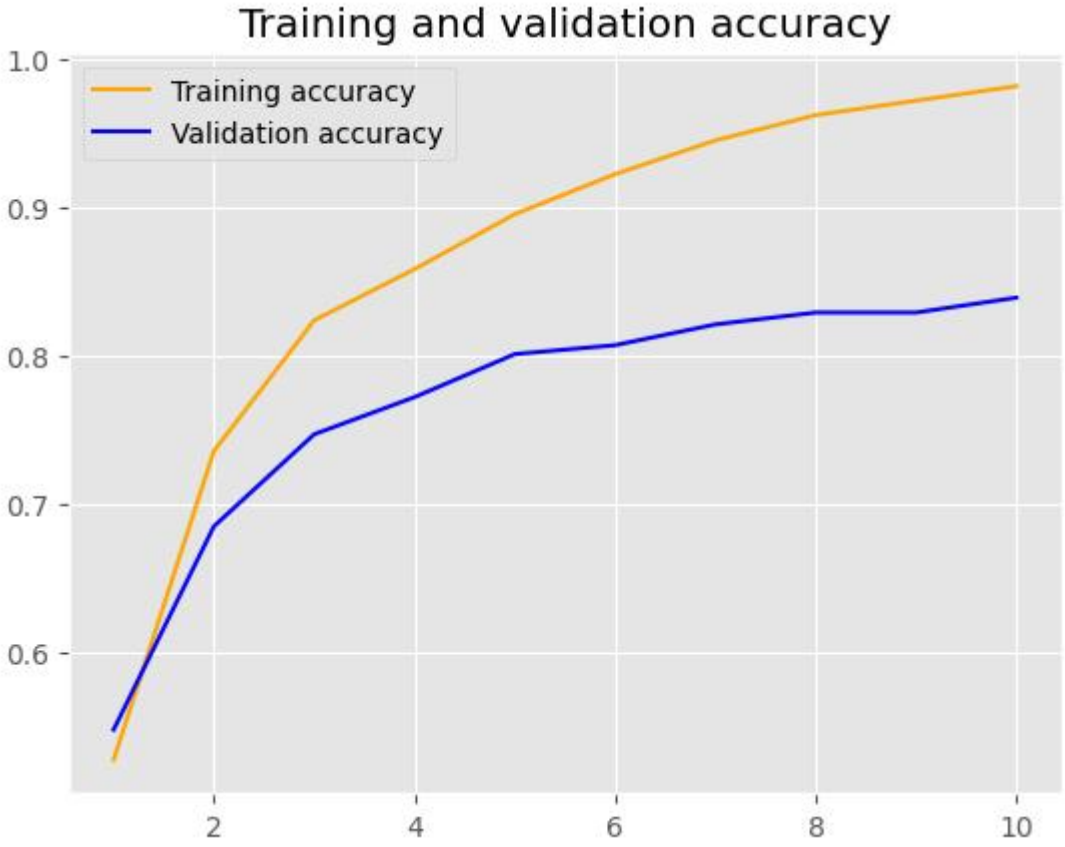
CUSTOM-TRAINED EMBEDDING LAYER

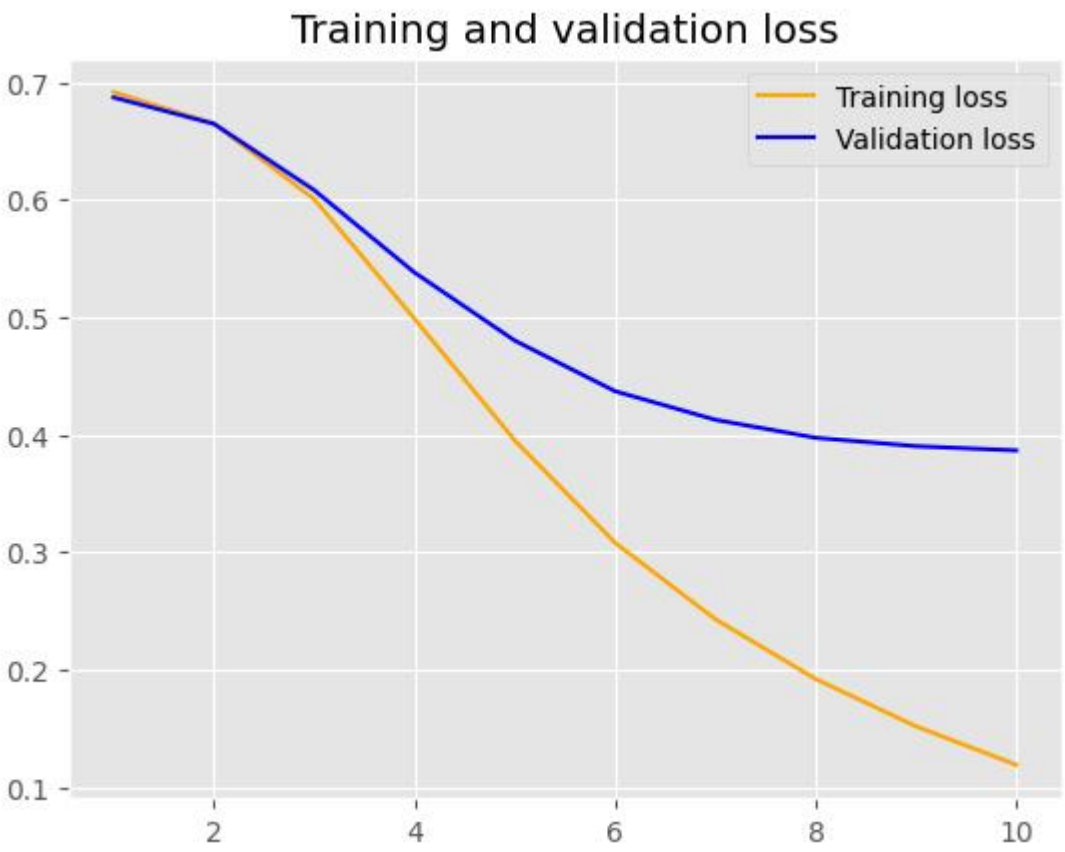
1. Custom-trained embedding layer with training sample size =100



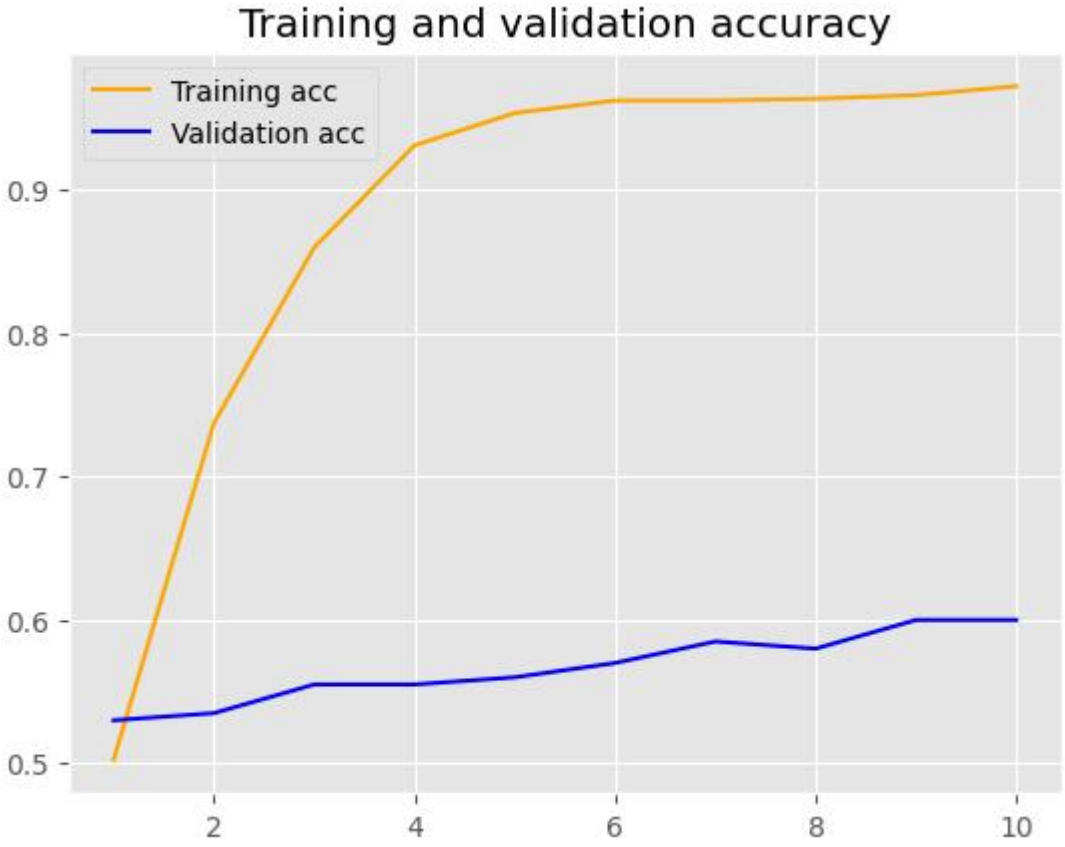


2. Custom-trained embedding layer with training sample size = 5000

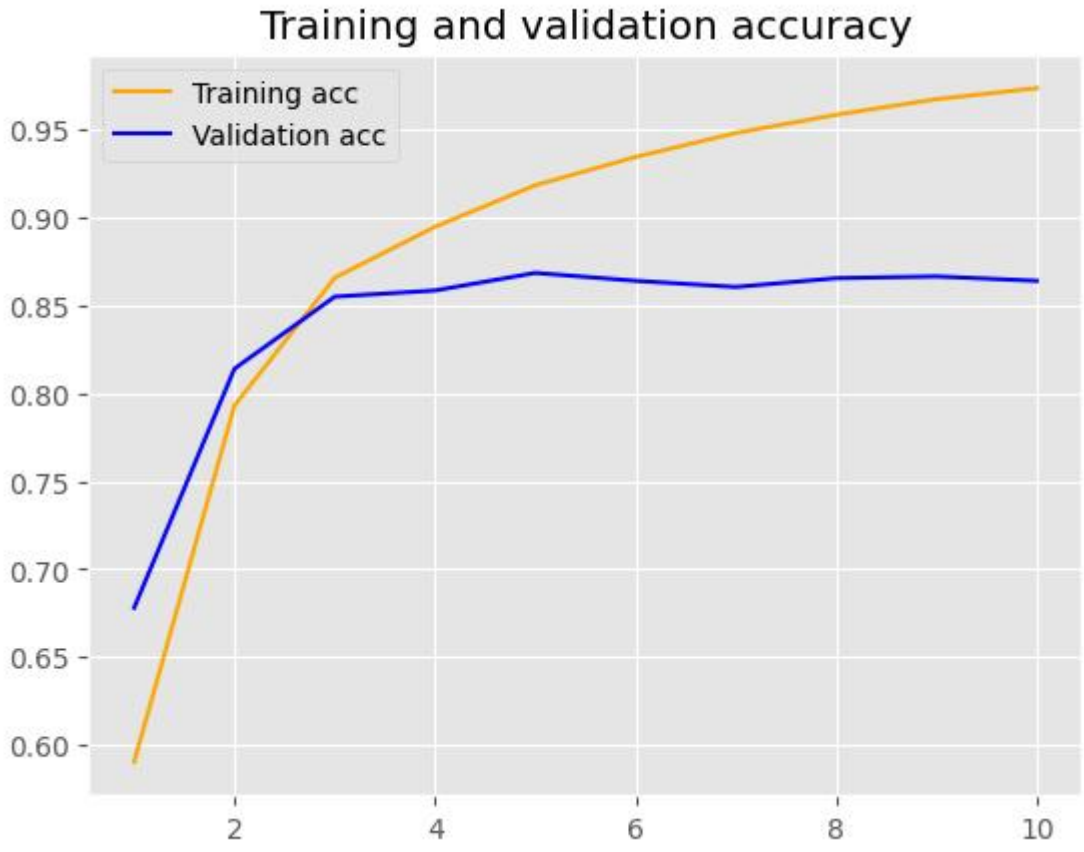


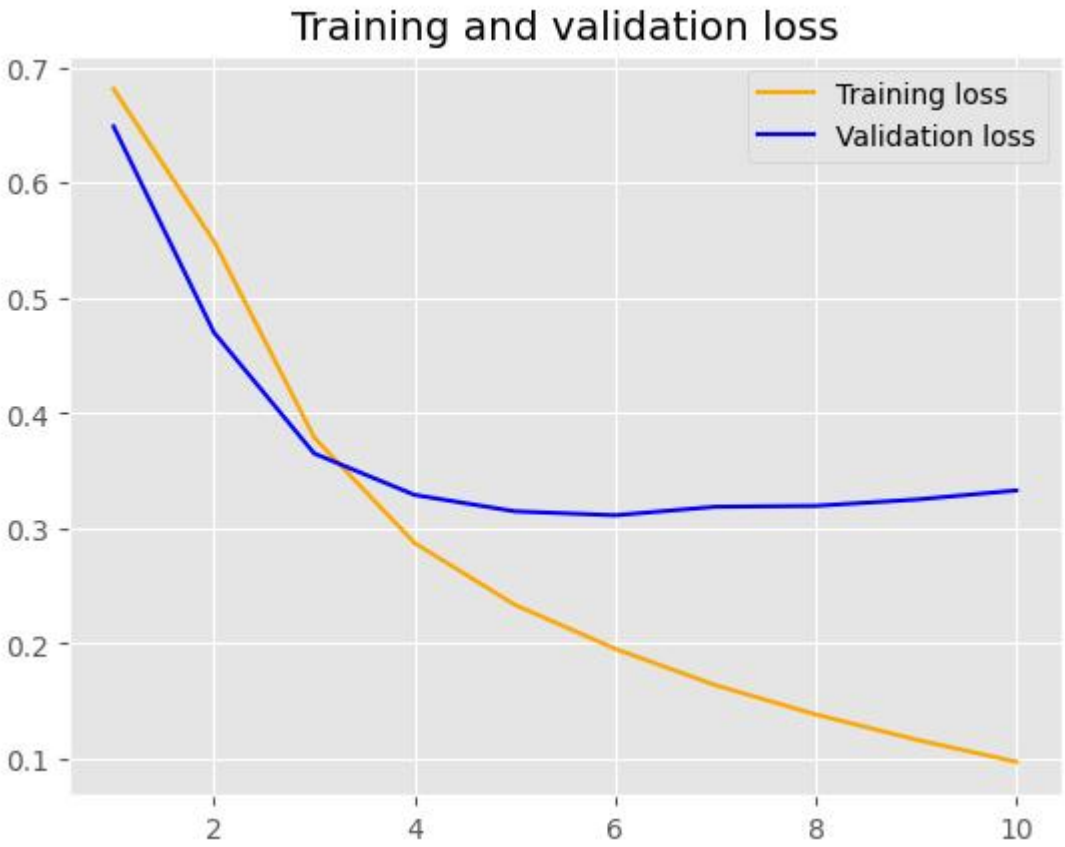


3. Custom-trained embedding layer with training sample size = 1000



4. Custom-trained embedding layer with training sample size = 10000

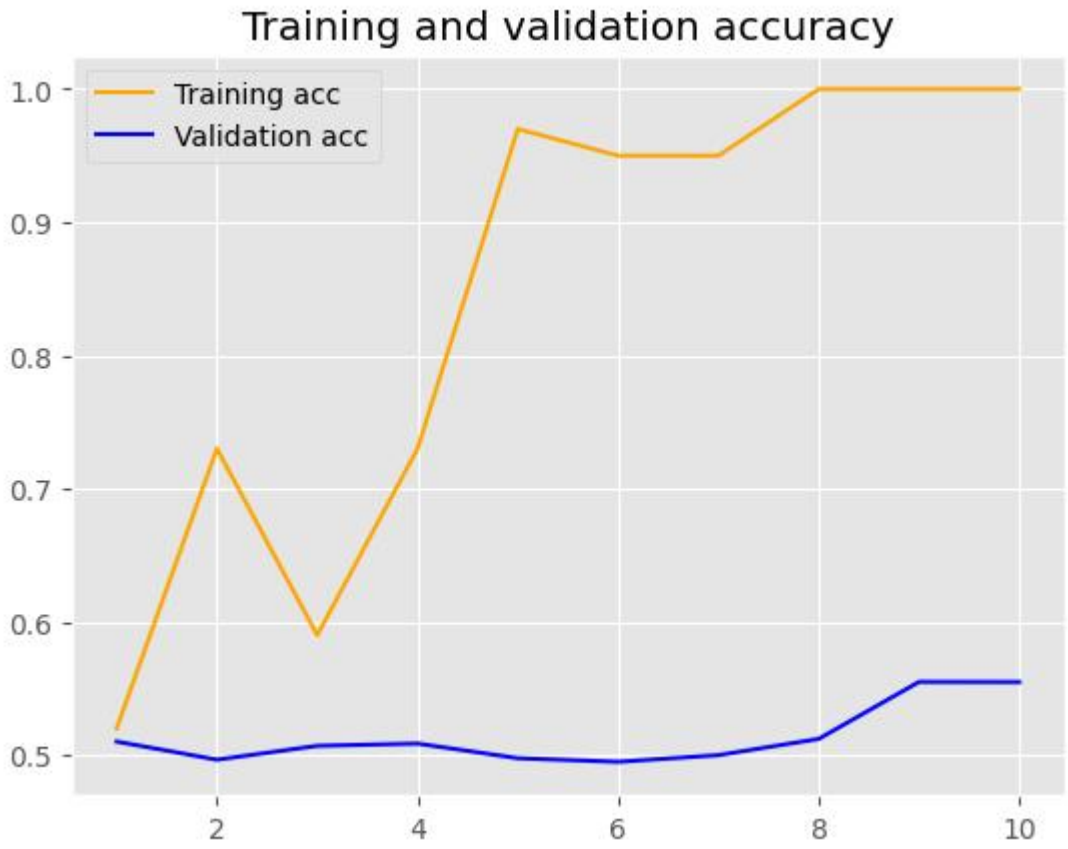


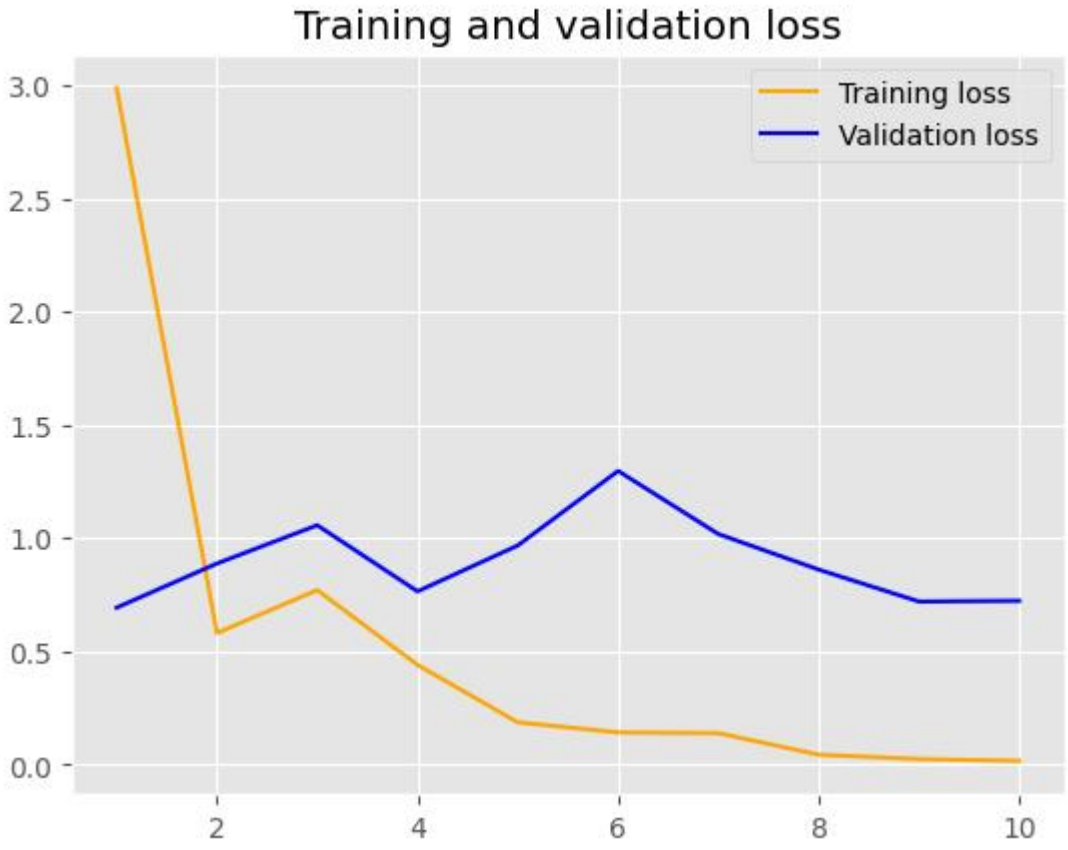


Using the custom-trained embedding layer, the accuracy varied from 97.25% to 100%, contingent upon the size of the training sample. The highest accuracy, reaching 100%, was achieved with a training sample size of 100.

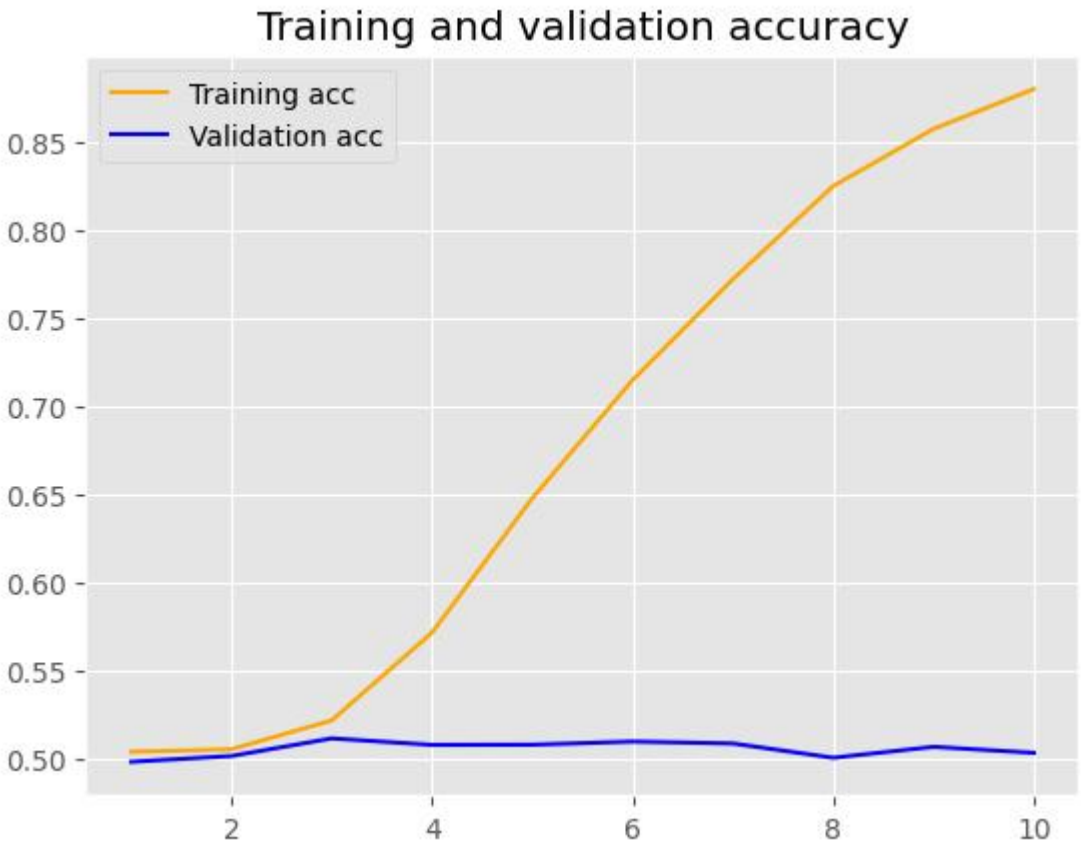
PRETRAINED WORD EMBEDDING LAYER

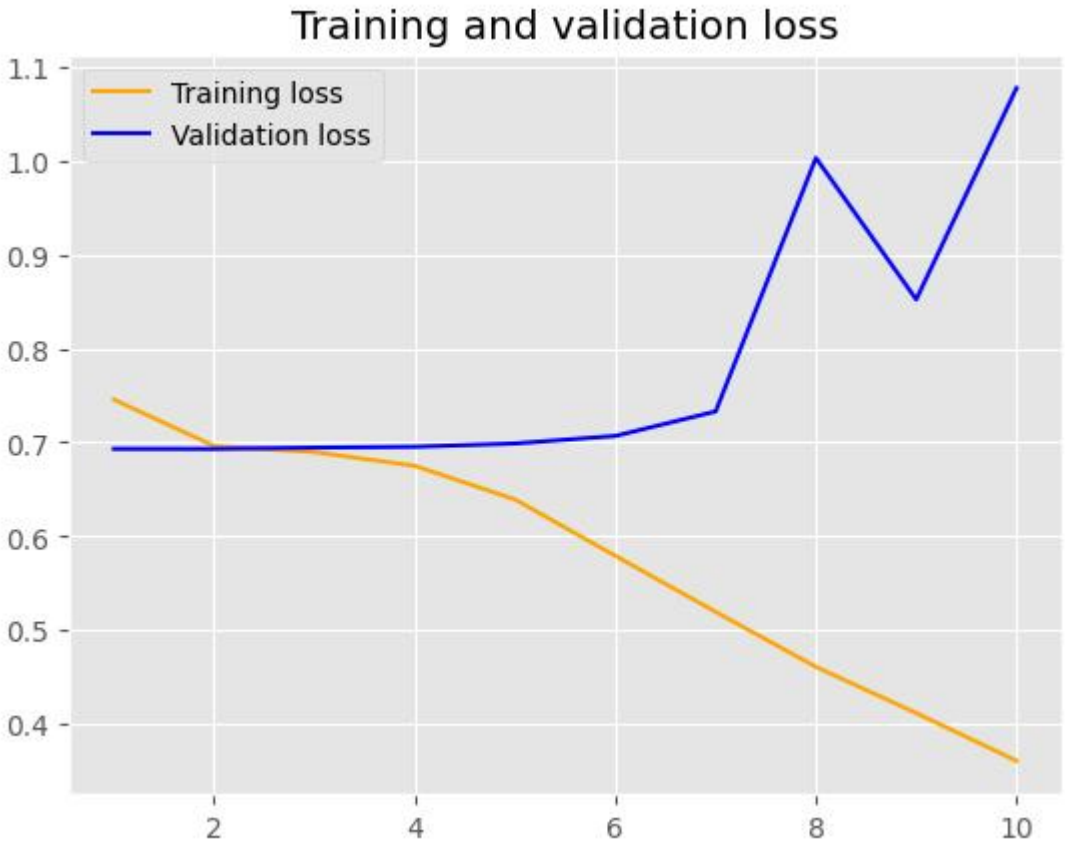
1. Pretrained word embedding layer with training sample size = 100



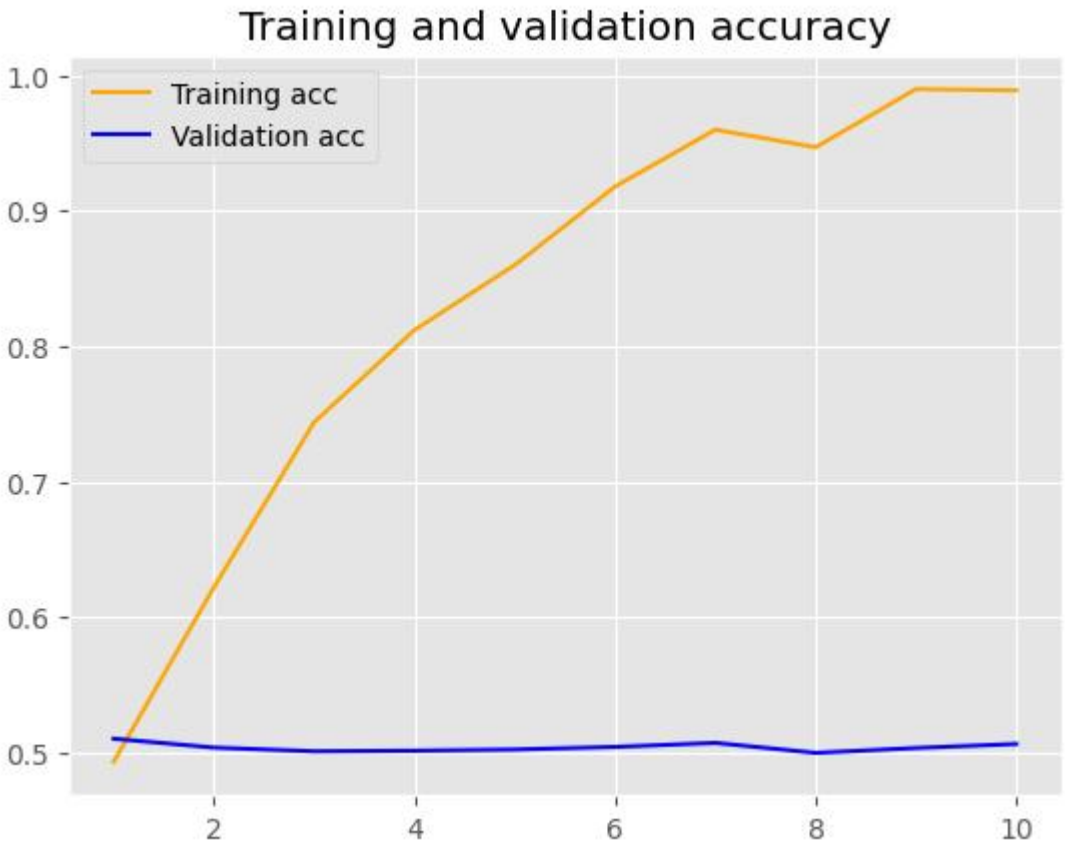


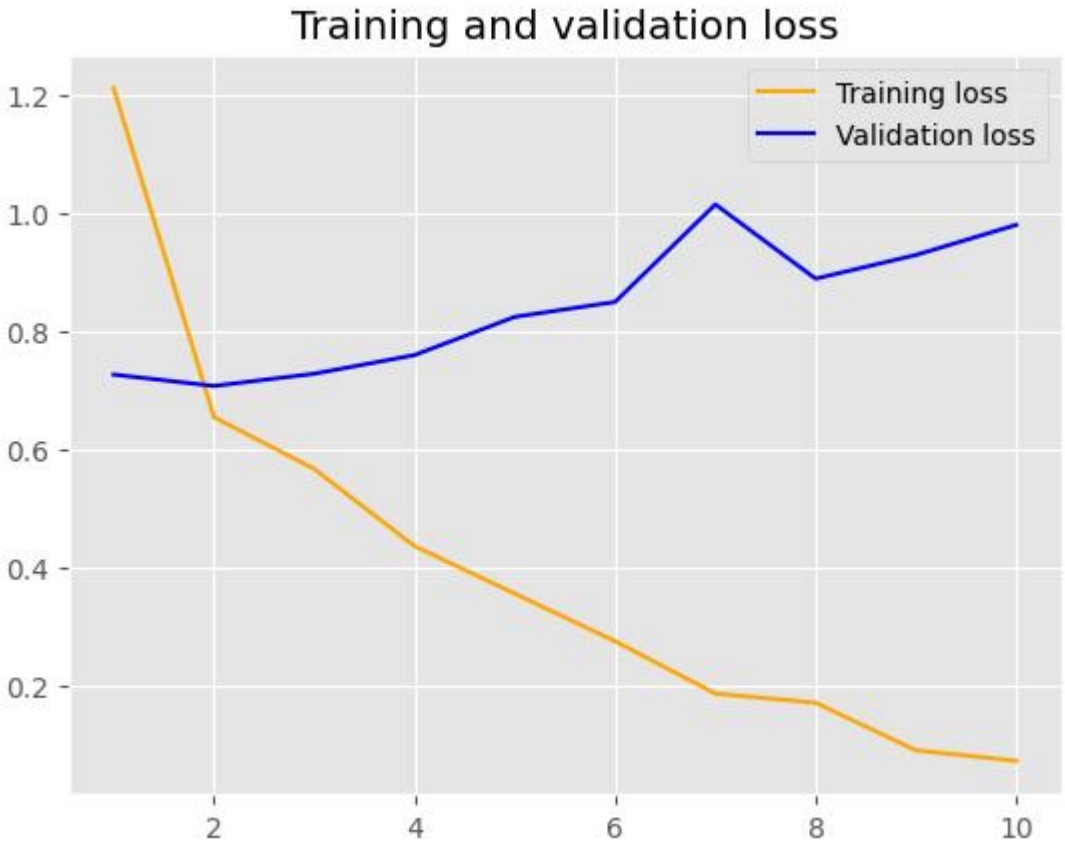
2. Pretrained word embedding layer with training sample size = 5000



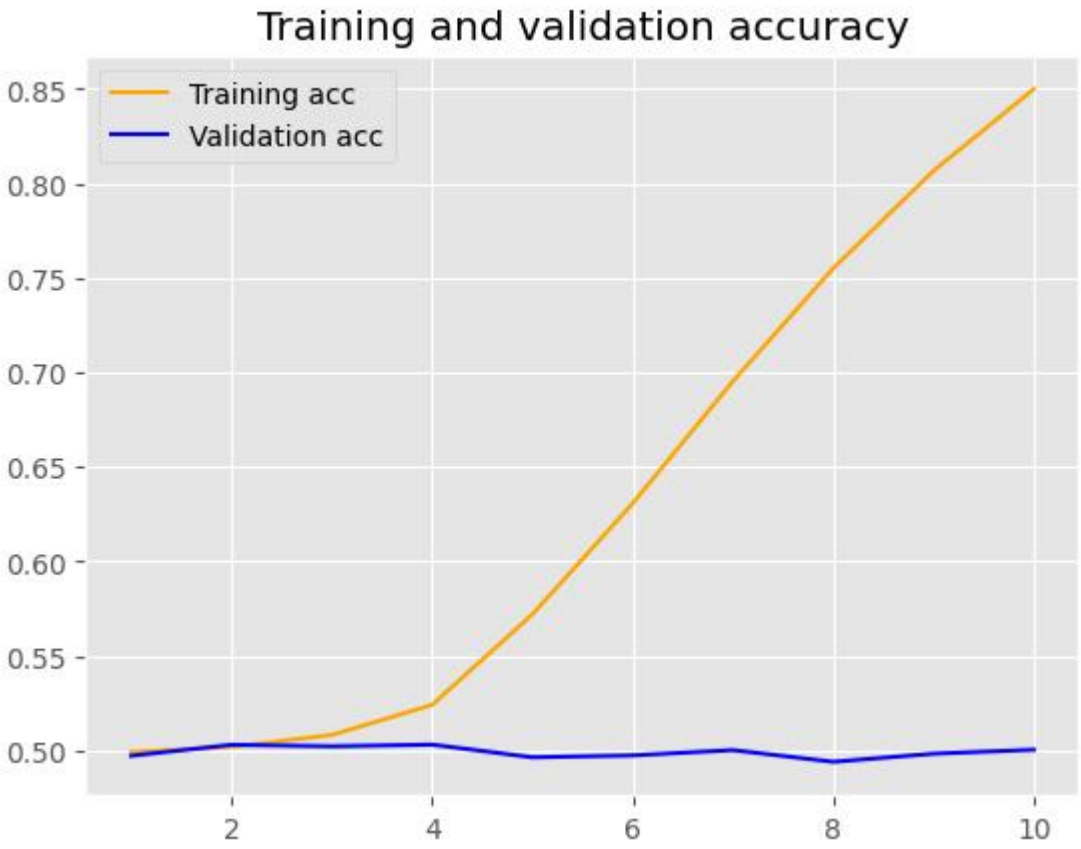


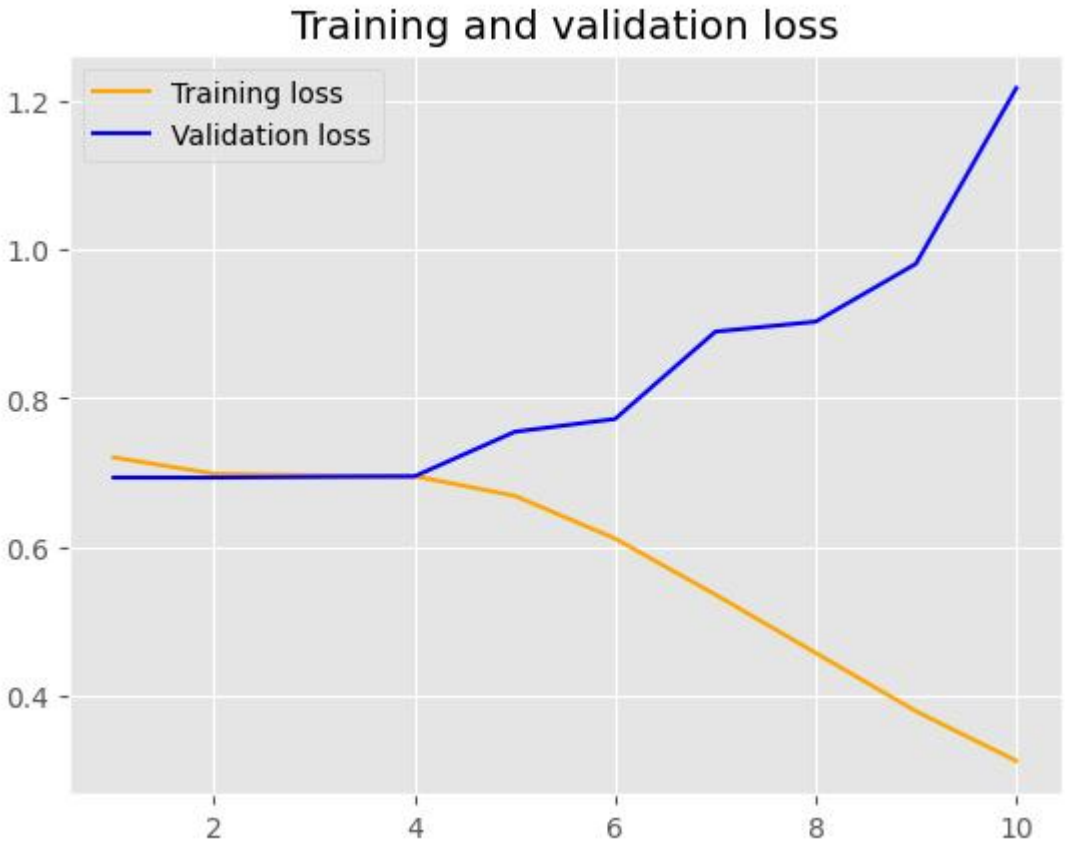
3. Pretrained word embedding layer with training sample size = 1000





4. Pretrained word embedding layer with training sample size = 10000





The Pretrained word embedding layer (GloVe) demonstrated accuracy levels ranging from 85% to 100%, influenced by the size

of the training sample. The highest accuracy, reaching 100%, was observed with a training sample size of 100. However, as the training sample sizes increased, the model utilizing pretrained embeddings tended to overfit more rapidly, resulting in reduced accuracy. These findings underscore the challenge of confidently determining the optimal approach, as it heavily relies on the specific requirements and limitations of the given task.

Results:

Embedding Technique	Training Sample Size	Training Accuracy (%)	Test loss
Custom-trained embedding layer	100	100	0.69
Custom-trained embedding layer	5000	98.15	0.37
Custom-trained embedding layer	1000	97.25	0.68
Custom-trained embedding layer	10000	97.36	0.33
Pretrained word embedding (GloVe)	100	100	0.77
Pretrained word embedding (GloVe)	5000	87.96	1.06
Pretrained word embedding (GloVe)	1000	98.90	0.98
Pretrained word embedding (GloVe)	10000	85	1.22

Conclusion:

In this experiment, the custom-trained embedding layer consistently outperformed the pretrained word embedding layer, especially when training with larger sample sizes. However, it's worth noting that the pretrained word embedding layer could still be considered a "better choice" in certain scenarios, particularly

when computational resources are limited and a small training sample size is required, despite the risk of overfitting.