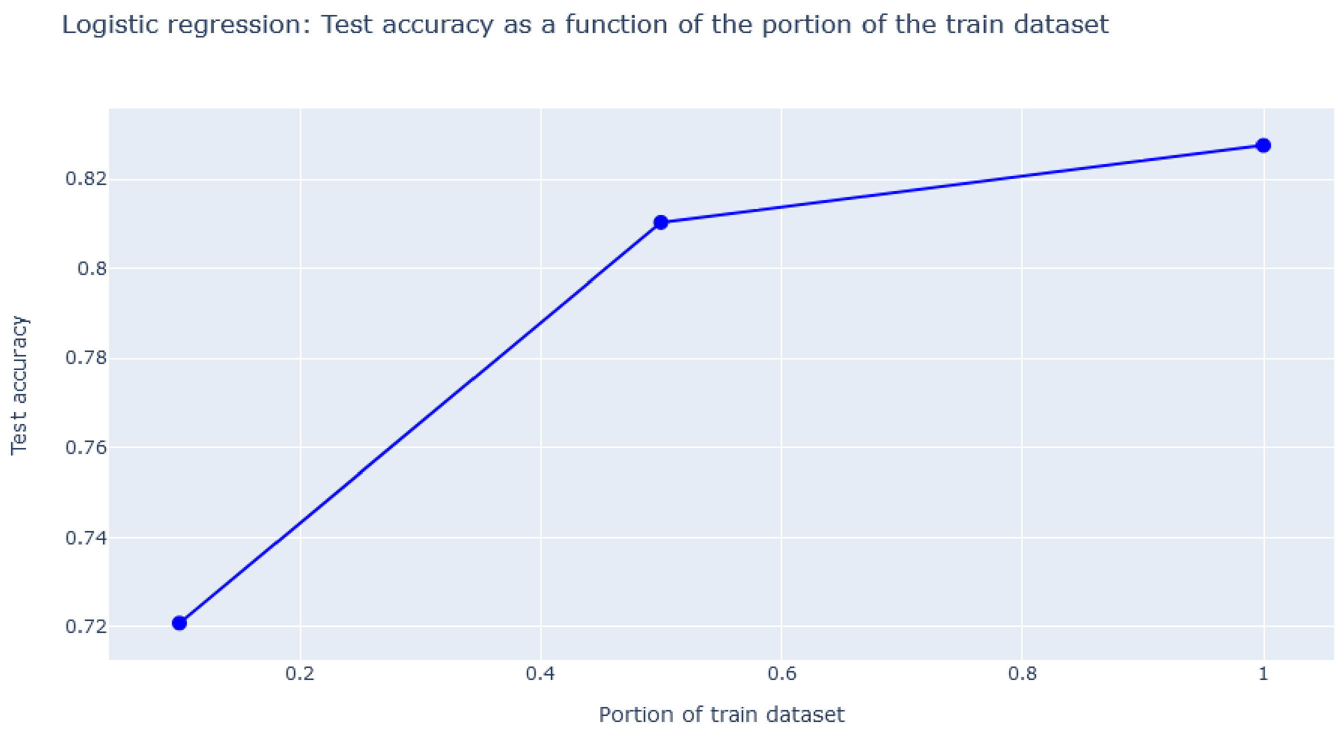
NLP Course – Exercise 4 – Bar Rousso 203765698

1) Test accuracy results of Logistic Regression model:

Training Data Portion	Test Accuracy
0.1	0.7208
0.5	0.8103
1	0.8276

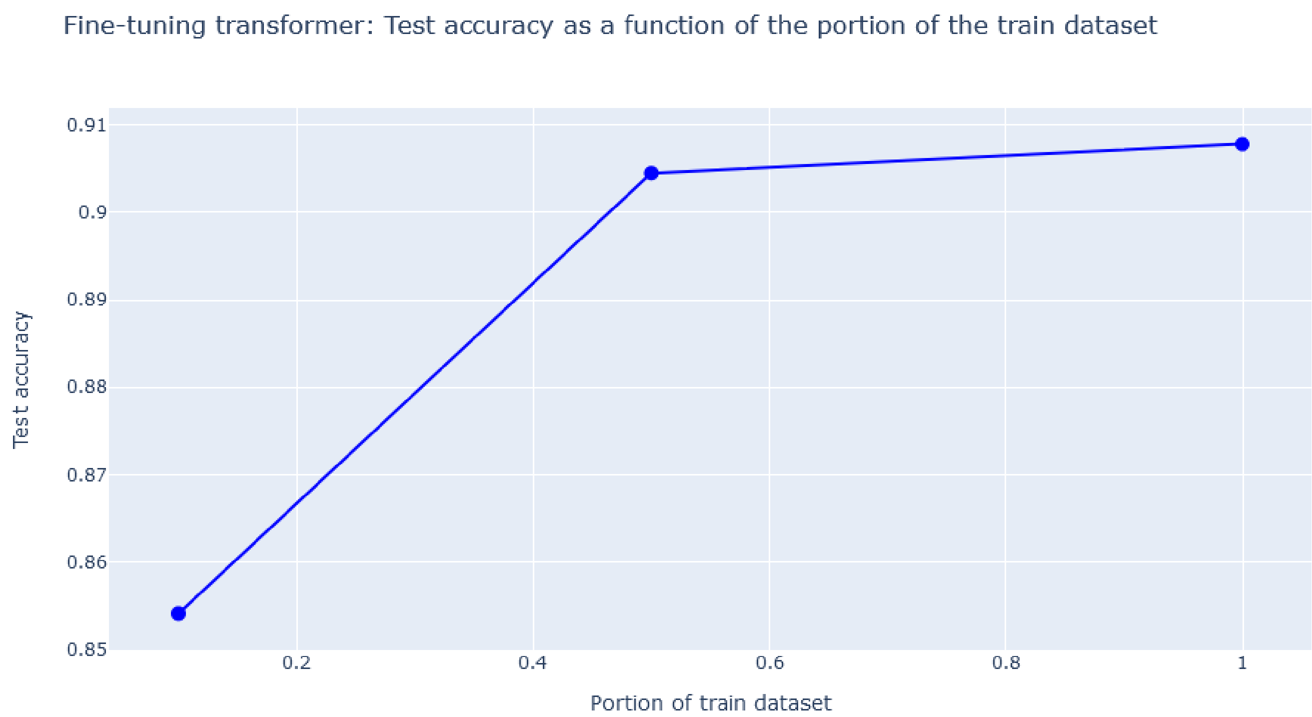
A plot of the test accuracy as a function of training data portion:



2) Test accuracy results of fine-tuned transformer model:

Training Data Portion	Test Accuracy
0.1	0.8541
0.5	0.9045
1	0.9078

A plot of the test accuracy as a function of training data portion:



Training dataset portion 0.1 - Average training losses and validation accuracy

Epoch Number	Average Training Loss	Validation Accuracy
1	1.2534	0.7918
2	0.5204	0.8475
3	0.2897	0.8541

Training dataset portion 0.5 - Average training losses and validation accuracy

Epoch Number	Average Training Loss	Validation Accuracy
1	0.5920	0.8879
2	0.1960	0.8866
3	0.0772	0.9045

Training dataset portion 1.0 - Average training losses and validation accuracy

Epoch Number	Average Training Loss	Validation Accuracy
1	0.4336	0.8939
2	0.1803	0.9038
3	0.0883	0.9078

3) Test accuracy result of zero-shot model: 0.7712 (with portion = 1.0)

4) Models' comparison:
   a. Question: Which model had the highest accuracy?
      Answer: The **fine-tuned transformer** model has the highest accuracy - 0.9078

      Explanation:
      Better accuracy than the Logistic Regression model:

      The fine-tuned transformer model comprises pre-train LLM (like BERT or GTP) and a simple log-linear model (i.e. Logistic Regression).

      Conversely, the regular Logistic Regression model is a simple log-linear model that relies on TFIDF embedding vectors.

      Hence the main difference between the models relies on the embedding vectors:
      Since the pre-train LLM already learns to generalize and understand the language from a large corpus, It will output a vector that better captures the context of the document compared to the TFIDF vector.

      As a result, the training of the fine-tuned transformer model (i.e. the fine-tuning part on the training dataset) will eventually produce a better accuracy on the test dataset.

      Better accuracy than the Zero-shot model:
      Since the fine-tuned transformer model was also trained (fine-tuning) to the particular classification task based on the training dataset, we expect this model to have a higher accuracy on the test set compared to the zero-shot module.

   b. Which model was the most sensitive to the size of the training set?
      Answer: The **Logistic Regression** model was the most sensitive to the size of the training set

      Explanation: Based on the results above, both fine-tuned transformer and Logistic Regression models show greater improvement in their accuracy when moving from the training portion of 0.1 to the training portion of 0.5, compared to moving from the training portion of 0.5 to the training portion of 1.0.
      This is makes sense as the portion's ratio is bigger in the first portion move:
      0.5/0.1 = 5 > 3 = 1.0/0.5.

      But a closer inspection will also show that:

Portions move	Logistic Regression accuracy improvement	fine-tunned transformer accuracy improvement
0.1 to 0.5	12.4%	5.9%
0.5 to 1.0	2.1%	0.36%

      Thus, we deduce that the **Logistic Regression** model is more sensitive to the size of the training set.

Note: The zero-shot model accuracy is not affected by the training data, since we only inference the pre-train model. Thus, it didn't take into account.

c. Two pros of zero-shot model:

   i. A Zero-Shot model doesn't require any additional examples or training data specific to the task at hand. The model uses its pre-trained knowledge to understand and perform the task based on the instructions in the prompt. This is particularly useful for tasks where collecting a labeled dataset is very hard.

   ii. A Zero-shot model can handle tasks and classify data into categories it has never explicitly seen during training. This flexibility allows it handling multiple tasks without the need for additional training data or fine-tuning for each specific task.

   Two cons of zero-shot model:

   i. A Zero-shot model may underperform compared to a fine-tuned model on specific tasks. This is because a fine-tuned model leverage task-specific training data to learn the nuances of a particular problem.

   ii. The effectiveness of a zero-shot model is heavily dependent on quality of its pre-training. If the pre-training data does not cover the knowledge domain of a particular task well, (for example not cover documents about baseball) the model may struggle to make accurate predictions.