Università della Svizzera italiana

Experimenting with Curriculum Learning in Software Engineering with CodeBERT

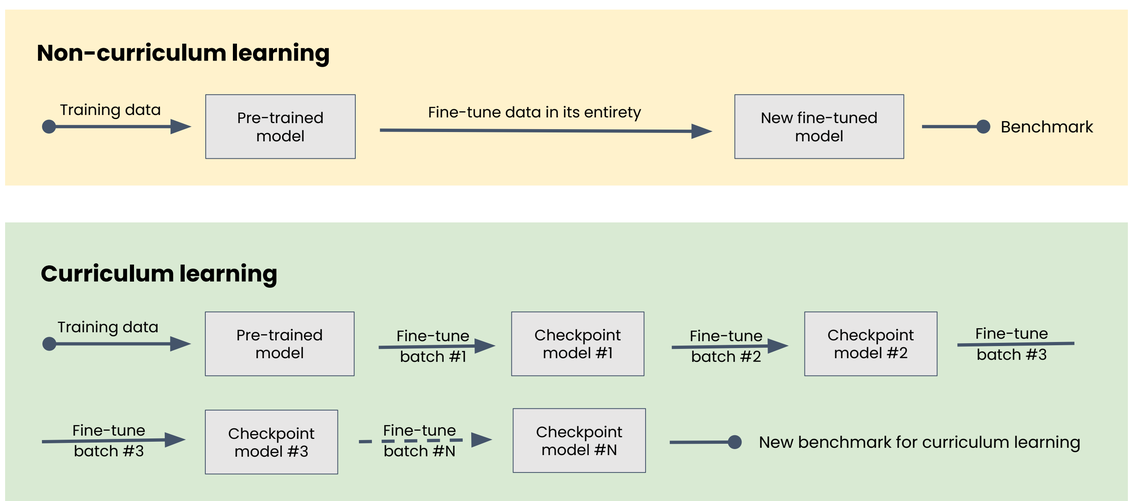
Student: Mak Fazlić Advisor: Prof. Dr. Bavota Gabriele Co-advisor: MSc. Mastropaolo Antonio

Motivation

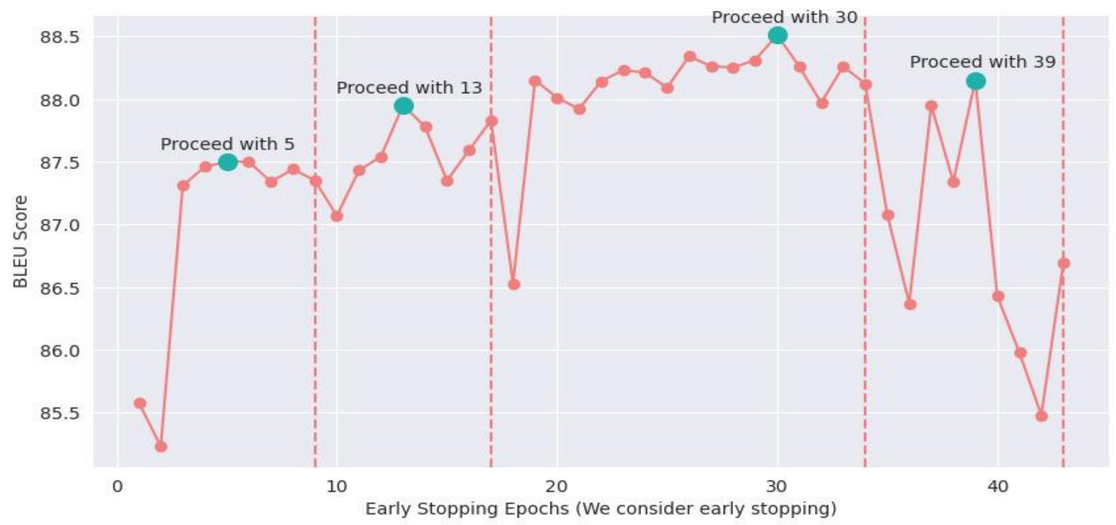
Several deep learning models exist that specialize in code-based tasks such as bug-fixing or writing documentation, however the performance is still limited. To this aim, one concept that has been successfully applied in other domains (other than software engineering) is that of curriculum learning. The methodology is similar to the way humans and animals learn by tackling simpler problems before graduating to more complex. We applied curriculum learning to the mentioned code-based tasks and compared the performance against the CodeBERT model, developed at Microsoft, as a benchmark. We evaluated the performance of curriculum learning on supervised Java code instances.

Curriculum learning

The input stream to the deep learning system is sorted based on a predefined complexity/simplicity metric and separated in N batches. The model is then fine-tuned on each of the N batches consecutively.

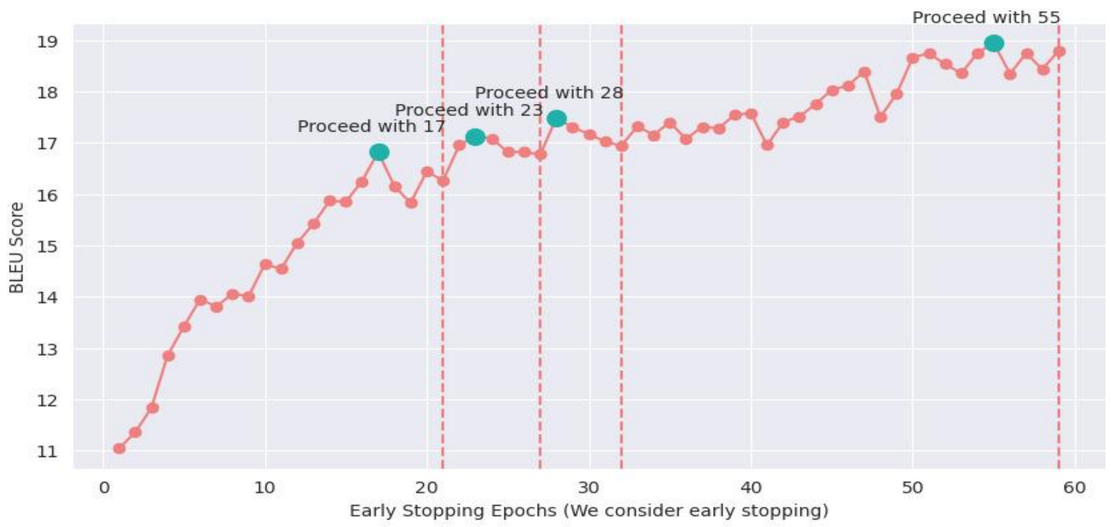


Bug-fixing



Evolution of the training process for bug-fixing. Considering we have four batches of supervised instances of increasing complexity, we denote checkpoint model produced for each batch in gree, while the changing of batches with a red line.

Code summarization



Evolution of the training process for code summarization. Considering we have four batches of supervised instances of increasing complexity, we once again use same indicators for the checkpoint models and the employment of a new batch.

Metrics

For the evaluation of the fine-tuning process on the bug-fixing task we used **BLEU-4** and **Accuracy** scores. For code summarization we used the same two scores with the addition of **ROUGE-L** and **METEOR**. Notice metrics definitions.s

$$\text{BLEU-4} = \text{BP} \times \exp\left(\frac{1}{4}\sum_{n=1}^{4}\log\left(\text{precision}_n\right)\right)$$

$$\text{Accuracy} = \frac{\text{Number of correctly predicted samples}}{\text{Total number of samples}}$$

$$\text{ROUGE-L} = \frac{\text{Longest Common Subsequence (LCS)(Reference, Candidate)}}{\text{Reference Length}}$$

$$\text{METEOR} = (1 - \text{Penalty}) \times \left(\frac{\text{Precision} \times \text{Recall}}{(1 - \text{Weight}) \times \text{Precision} + \text{Weight} \times \text{Recall}}\right)$$

Results

The BLEU scores for traditional learning are taking the lead, while the Accuracy for curriculum learning is significantly higher. This indicates that the curriculum learned model is more accurate when evaluating the inference predictions and therefore should be more robust and reliable.

Methodology	Bleu Score	Accuracy Score
Traditional Learning	88.41	6.58
Curriculum Learning	87.86	8.21

For code summarization, the traditional learning approach was consistently better across all metrics when we evaluate both models. This was not an expected outcome and further research or detailed assessment of threats to validity need to be considered.

Methodology	Bleu Score	Rouge Score	Meteor Score	ExMatch Score
Traditional Learning	18.75	0.17	0.29	1.57
Curriculum Learning	18.26	0.17	0.28	1.15

Reproduction package and report

30/06/2023
Bachelor's thesis defense