DISTIL-CODEGPT

DISTILLING CODE-GENERATION MODELS FOR LOCAL USE

What are the effects of compressing a CodeGPT model, regarding size, accuracy, and speed, through the application of in-training Knowledge Distillation?

INTRODUCTION

Using **large language models** for code completion has grown increasingly popular among developers.
Due to **size and performance** reasons, these models can **only** operate on **servers.** This limits **accessibility** and can raise **privacy** concerns.

This study explores **compressing** these models using **KD** to allow for **local usage**.

Previous research demonstrated that it is possible to **reduce** the size of **BERT** models for language tasks.

We **show** that it can also be applied to **CodeGPT**[1] models, albeit with a moderate **accuracy loss**. We explore why this loss is larger and how to **mitigate** it. Lastly, we give an indication that **pre-training** KD is preferred over **in-training** KD.

PRELIMINARIES

Knowledge Distillation (KD)

- The student model learns from the predictions of the teacher (**soft loss**) and the ground truth (**hard loss**) during the training.
- In **in-training** KD, the teacher also trains during the distillation. In **pre-training** KD it does not.

Other Compression Techniques

- **Pruning,** removing unnecessary weights or connections.
- **Quantization,** converting weights and activation tensors to have low-bit representations.

Transformer Models

An **ML** architecture revolutionizing language understanding and text generation

- **GPT**[2] predicts the next word based on the previous context.
- **BERT**[3], predicts a masked word in the middle of a sentence.

RELATED WORKS

Studies on Knowledge Distillation (KD)

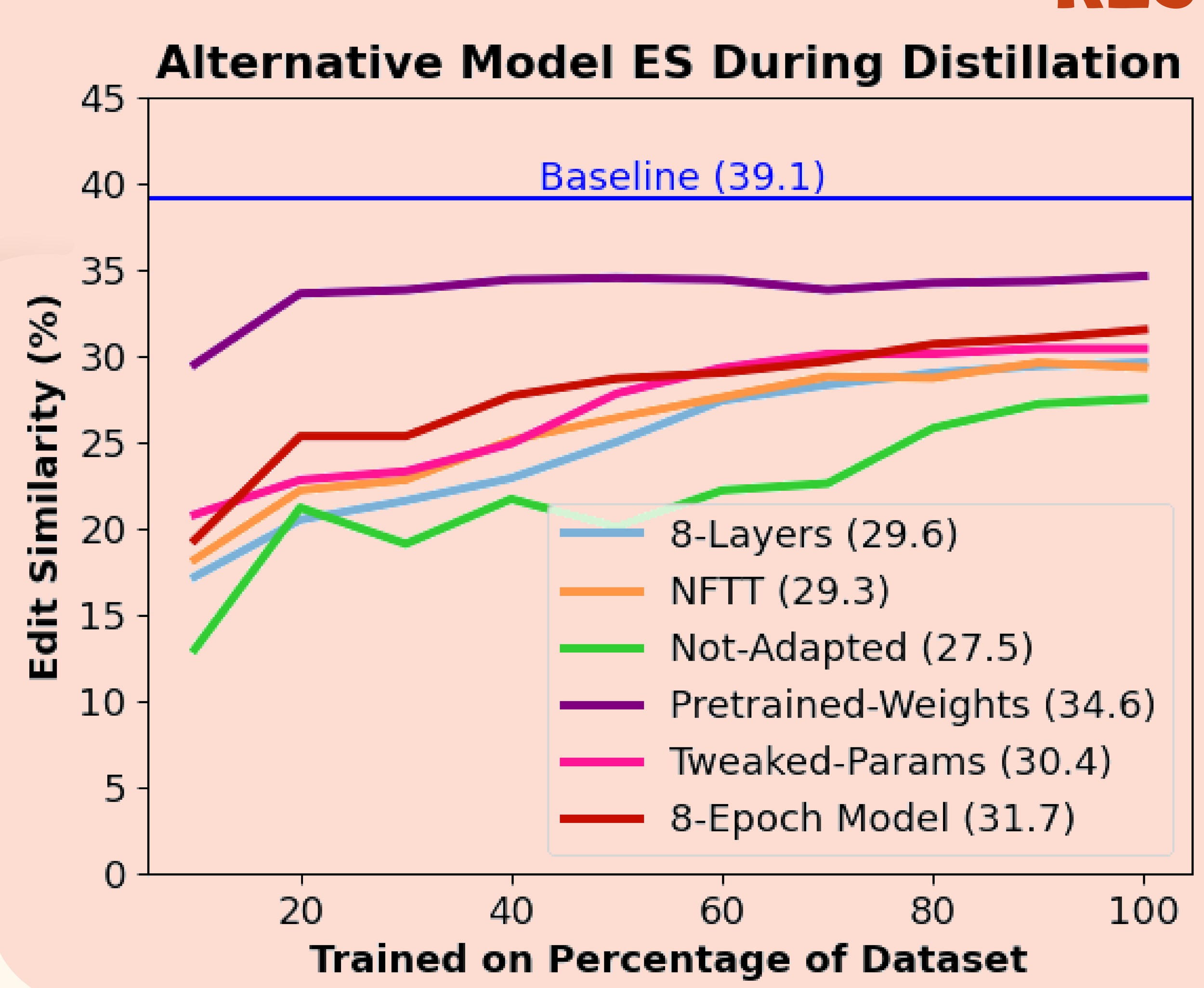
- **DistilBERT**[4]**,** Using pre-training KD to get a BERT model for language a **60% speedup** and a **40% size reduction** while retaining **97% accuracy**.
- **TinyBERT**[5]**,** Using other KD techniques to get the model a **60% speedup**, and **40% size reduction** while retaining **97% accuracy.**

Studies on Compressing Code Models

Three studies were conducted simultaneously on compressing code models, namely:

- **CodeGPT on XTC**[6] used **KD** and **quantization** to reduce model size **15x** while preserving a **fair amount** of accuracy.
- **MP and PEG PTQ on CodeGPT**[7] used **quantization** to compress the model **4x** while maintaining **nearly all** accuracy.
- **CodeGPT on Intel**[8] used **quantization** and **pruning** to compress the model down **60%** and maintained an **acceptable level** of accuracy

RESULTS



- The results were slightly **worse** than DistilBERT.
- Compared to baseline, the best model (**Pretrained-Weights**) maintained **90%** accuracy while being **20%** faster and **25%** smaller.
- The 12-layer model, same size as the baseline, had **9-points** worse ES. Also, 3 alternative models had higher ES than it. This indicates that flaws in the **method**, rather than the smaller **size**, holds back performance

Model	Params	Size	Inf.	ES	EM
Baseline	124	510	26	39.1	14.5
12 Layers	124	510	24	30.3	6.4
10 Layers	110	450	27	29.5	6.1
8 Layers	96	390	31	29.6	6.3
6 Layers	82	340	36	28.7	6.4
4 Layers	68	280	43	27.6	5.9

METHOD AND SETUP

We adapted **DistilBERT** to be an in-training KD algorithm for **code** models.
We benchmarked:

- A **standard model** for layer counts 4,6,8,10, and 12.
- 6 **alternative models** with slightly different settings, all with 8 layers. One example is the **Pretrained-Weights** model where the student had pre-trained weights for predicting code before the distillation.

Evaluation

- **Accuracy,** Edit similarity (ES). It measures similarity in strings.
- **Size,** The model size on GPU
- **Speed,** samples predicted per second. Seen in the table as inf.

DISCUSSION

Improvements

- **The student model,** using one which is pre-trained on code yields better results
- **Different Parameter Selections,** such as a lower temperature or more epochs, could benefit KD on code models
- **Pre-training KD,** it uses less GPU during training and results seem better

Efficacy of Compressing Code Models

It shows potential for two reasons:

- This study, which was a first attempt, got **decent results** and showed which settings could be changed to **improve** it further.
- **MP and PEG PTQ on CodeGPT** had an equally good performance as studies compressing language models

Threats to Validity

- **The benchmarks,** ES might is not a perfect metric
- **Generalizability,** All models are not expected to react the same to KD

CONCLUSION

- We adapted **DistilBERT** to do **in-training** KD for a **CodeGPT** model
- The study showed **potential** as it enabled **significant** compression albeit with a **slight** reduction in language understanding.
- Different settings could **improve** results
- Future work might use **different base models**, instead of DistilBERT, or experiment with other **settings**.

REFERENCES

1. Lu et al., Codexglue: A machine learning benchmark dataset for code understanding and generation. CoRR, abs/2102.04664, 2021.
2. Radford et al., Improving language understanding by generative pre-training. 2018.
3. Devlin et al., Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.
4. Sanh et al., DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. October 2019. doi: 10.48550/arXiv.1910.01108.
5. Jiao et al., TinyBERT: Distilling BERT for Natural Language Understanding. In Findings of the Association for Computational Linguistics: EMNLP 2020, pages 4163–4174, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020. findings-emnlp.372.
6. Aral de Moor. CodeGPT on XTC. 2023.
7. Mauro Storti. Leveraging efficient transformer quantization for CodeGPT: A post-training analysis. 2023.
8. Dan Sochirca. Compressing code generation language models on CPUs. 2023.

CONTRIBUTORS

AUTHOR
Emil Malmsten
e.l.malmsten@student.tudelft.nl

INSTITUTE
Delft University of Technology

SUPERVISORS
Dr. Maliheh Izadi
ir. Ali Al-Kaswan

RESPONSIBLE PROFESSOR
Prof. Dr. Arie van Deursen

EXAMINER
Prof. Avishek Anand