
University of St.Gallen

Integrative Master Project (Fall 2023)

# **Mid-Term Progress Presentations:** Model Zoo - Language Transformers

Kaan Aydin, Sven Schnydrig

# Agenda

- Introduction
- Configuration Space
- Latent Generating Factors
- Trainings Protocol
- Status & Next Steps

# Agenda

- **Introduction**
- Configuration Space
- Latent Generating Factors
- Trainings Protocol
- Status & Next Steps

# Context & Introduction

- NN models evolve on **unique trajectories in weight space** during training - depending on hyperparameters and weight initialization - leading to **different model parameters** (i.e., weights & biases) & minima on the loss surface
- A population of NNs (or a model zoo) can **form structures in weight space** that contain information about the state of training and can **reveal latent properties** of individual models (e.g., accuracy)
- With model zoos, we can investigate novel approaches for a **variety of use cases**, such as model analysis, discovering learning dynamics & generative modelling of NN models
- The Blueprint paper[1] used CNNs of varying sizes and datasets to generate model zoos – for our project, we will look at **Language Transformers**, focusing on BERT


Model Zoos: A Dataset of Diverse Populations of Neural Network Models

Blueprint Paper[1]

1. Schürholt, K., Taskiran, D., Knyazev, B., Giró-i-Nieto, X., & Borth, D. (2022). Model Zoos: A Dataset of Diverse Populations of Neural Network Models.

# Unveiling the engine behind BERT's success

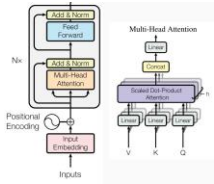
Introduced in 2019[1], BERT ignited **significant advancements** within the field of natural language processing

It employs a Transformer architecture[2], which uses self-attention mechanisms to enhance creation of **contextualized embeddings**

BERT only consists of **encoder layers**, thus processing text **bi-directionally** (i.e., from left and right)

## Model Details

- The base version of BERT consists of 12-encoder layers, each with 768 hidden units, 12 self-attention heads & 110M parameters
- BERT was trained with data from Wikipedia (2.5B words) and BookCorpus (800M words)




## Training Mechanism

BERT's success is attributed in part to its **two-step training approach**:

- BERT learns language representations from text through pre-text tasks
  - **Masked LM:** This involves randomly masking out words from the input and having the model predict those words
  - **Next Sentence Prediction:** This involves predicting whether two given sentences are consecutive or not
- BERT's robust representations can be used for downstream tasks, e.g., Sentiment Classification, Question & Answering, Machine Translation, Named Entity Recog.

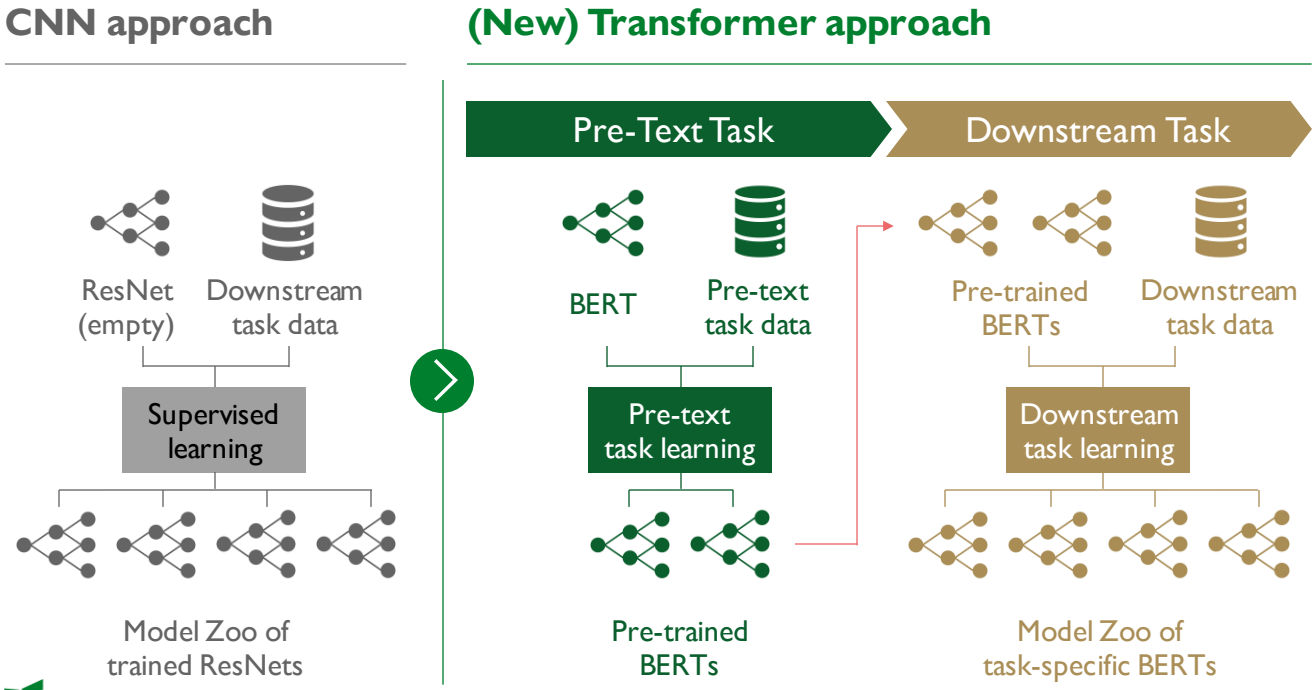
## BERT variations

Many different variations of BERT were developed **varying in size:**

- **BERT-Large:** 340M para., 24-layer, 1024-hidden, 16-heads
- **TinyBERT[3]:** 14.5M para., a smaller version distilled for maximum speed
- **DistillBert[4]:** 66M para., designed for speed by distilling knowledge from BERT

1. Devlin, J., Chang, M-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. 2. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention Is All You Need.3. Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding.4. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper & lighter.

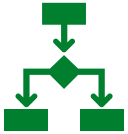
# We move from a supervised-only to a two-step training process




Transformers introduce an **additional layer of intricacy** in training (vs. CNNs) and require two training phases

Given the **fundamental changes** inherent to how transformers are trained, we need to **redefine our training setup**

We need to rethink along **three area**, addressing relevant questions for each



## Configuration Space

- How does our configuration space look like?
- How do we ensure model diversity given our resources?
- …

## Generating Factors

- What datasets do we use for the pre-text and downstream task?
- For which hyperparameters do we configure ranges?
- …

## Training Protocol

- For how many epochs do we train each model training trajectory?
- Are the splits for train, validation and test kept consistent?
- …

# Agenda

Introduction

**Configuration Space**

Latent Generating Factors

Trainings Protocol

Status & Next Steps

We outline our **configuration space** over two training steps in four sub-components

## Pre-Text Task

1. **Model selection**
   - BERT (base, pre-trained)
   - BERT (base, empty)
   - TinyBERT (empty)
   - …
2. **Training approach**
   - Pre-Training from scratch
   - Continued pre-training

## Down-stream Task

3. **Weights in TF blocks**
   - Unfreeze weights of TF blocks
   - Freeze weights of TF blocks
4. **Classification Head**
   - Sentiment Analysis
   - Question Answering
   - …

# We evaluated multiple options to determine the best configuration going forward

*Not exhaustive*

| 1 Model selection | 2 Training approach | 3 Weights in TF blocks | 4 Classification Head | Computational Requirements |
|---|---|---|---|---|
| **BERT-base** *(empty)* | PT from scratch | Unfrozen weights | Sentiment | Very high |
| | … | … | … | … |
| **BERT-base** *(pre-trained)* | Continued PT | Unfrozen weights | Sentiment | Medium |
| | | | Q&A | |
| | | Frozen weights | Sentiment | |
| | | | Q&A | |
| | … | … | … | |
| **TinyBERT** *(empty)* | PT from scratch | Unfrozen weights | Q&A | Very high |
| | … | … | … | … |

Pre-training BERT **from scratch is expensive** (~800 days with current resources / model) vs. continued pre-training

Training the classification head is **rather "cheap"** compared to pre-training

By unfreezing weights in the TF blocks, we can generate **more diversity in the BERT layers**

We focus on continued pre-training and 2 DS tasks to balance **model diversity & resource availability**

# Agenda

Introduction

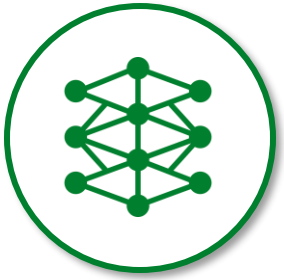
Configuration Space

**Latent Generating Factors**

Trainings Protocol

Status & Next Steps

# A model zoo's generating factors is defined by three components



**Architecture**

**Dataset**

**Hyperparameters**

We need to define the generating factors for the:

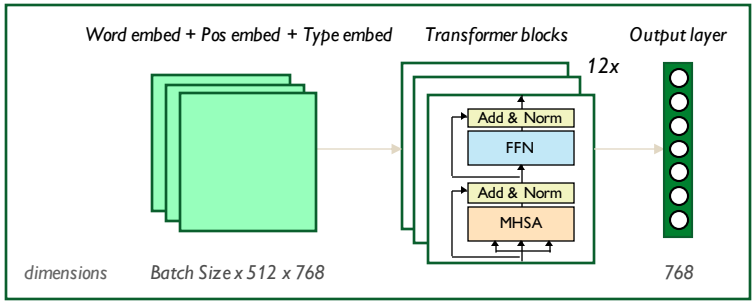
i. Pre-text task

ii. Downstream task

# Architecture for the Pre-Text and Downstream Task

## Pre-Text Task

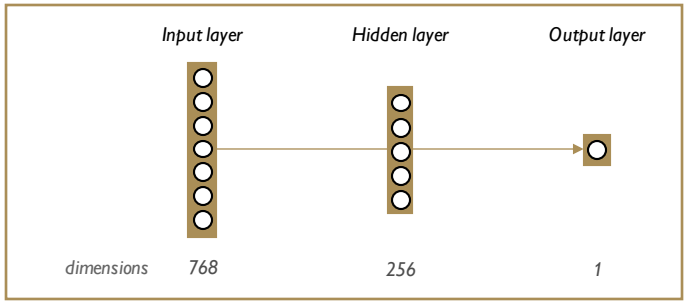
- BERT (base) consists of 12 Transformer encoders that are stacked on top of each other, returning a 768-dimensional embedding as the final layer (i.e., the [CLS] token's output)
- The embedding layer (lowest layer) contains three types of embeddings which are added and passed as input to the first encoder stack




*BERT architecture*

## Downstream Task

- In typical BERT classification, the [CLS] token's output is directly (without any hidden layers) used for classification due to its comprehensive representation of the input
- However, we introduce 1 additional (hidden) layers in the classification head to generate more weights which can be used for further processing (e.g., model analysis)




*MLP classification head*

# Datasets for the Pre-Text and Downstream Task

## Pre-Text Task

- We decided to use the **OpenWebText** dataset[1] for the continued pretraining
- Openwebtext is an open-source replication of the WebText dataset from OpenAI that was **used to train GPT-2**

| text<br>string |
| --- |
| Port-au-Prince, Haiti (CNN) -- Earthquake victims, writhing in pain and grasping at life, field hospital Friday night after a Belgian medical team evacuated the area, saying it was |
| Former secretary of state Hillary Clinton meets voters at a campaign rally in St. Louis or Democratic front-runner Hillary Clinton was ahead by a slim margin in Missouri on Wednesda |

*OpenWebText dataset*

## Downstream Task

### Sentiment Analysis

- We will use the Stanford Sentiment Treebank (SST-2)[2], which contains ~70k observations
- Evaluation metric: accuracy

### Question Answering

- We will use the Stanford Question Answering Dataset (SQuAD) dataset[3] which contains over 100,000 Q&A pairs
- Evaluation metric: F1 score

| | |
| --- | --- |
| the greatest musicians | 1 positive |
| cold movie | 0 negative |
| with his usual intelligence and subtlety | 1 positive |
| redundant concept | 0 negative |
| swimming is above all about a young woman ' actress whose face projects that woman 's c | 1 positive |

*SST-2*

| | | |
| --- | --- | --- |
| Architecturally, the school has a Catholic… | To whom did the Virgin Mary allegedly appear in… | { "text": [ "Saint Bernadette Soubirous"… |
| Architecturally, the school has a Catholic… | What is in front of the Notre Dame Main Building? | { "text": [ "a copper statue of Christ" ],… |
| Architecturally, the school has a Catholic… | The Basilica of the Sacred heart at Notre… | { "text": [ "the Main Building" ],… |
| Architecturally, the school has a Catholic… | What is the Grotto at Notre Dame? | { "text": [ "a Marian place of prayer and… |

*SQuAD*



1. Gokaslan, A., Cohen, V., Ellie, P., & Tellex, S. (2019). Openwebtext corpus. 2. Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. 3. Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for Machine Comprehension of Text.

# Hyperparameters for the Pre-Text and Downstream Task

| | Pre-Text Task | Downstream Task |
|---|---|---|
| Learning Rate | 1e-04, 3e-04, 5e-04 | 2e-05, 3e-05, 5e-05 |
| Weight Decay | 0.01, 0.001 | 0.1, 0,01 |
| Number of Epochs | Constantly for 10 epochs | Constantly for 5 epochs |
| Batch Size | Depending on HW (16 or 32) | |
| Optimizer | Adam | AdamW |
| Dropout Rate | 0.1 | 0.1 |
| Learning Rate Scheduler | No warmup steps to keep learning rate constant | |
| Learning Rate Decay | No learning rate decay to keep learning rate constant | |
| Activation Function | GeLU | ReLU / SoftMax |
| Masking Fraction | 15% (same as original BERT) | not applicable |
| Max Sequence Len | Keep fixed at 512 | |
| Initialization | not applicable | Random initialization |

# Hyperparameters for the Pre-Text and Downstream Task

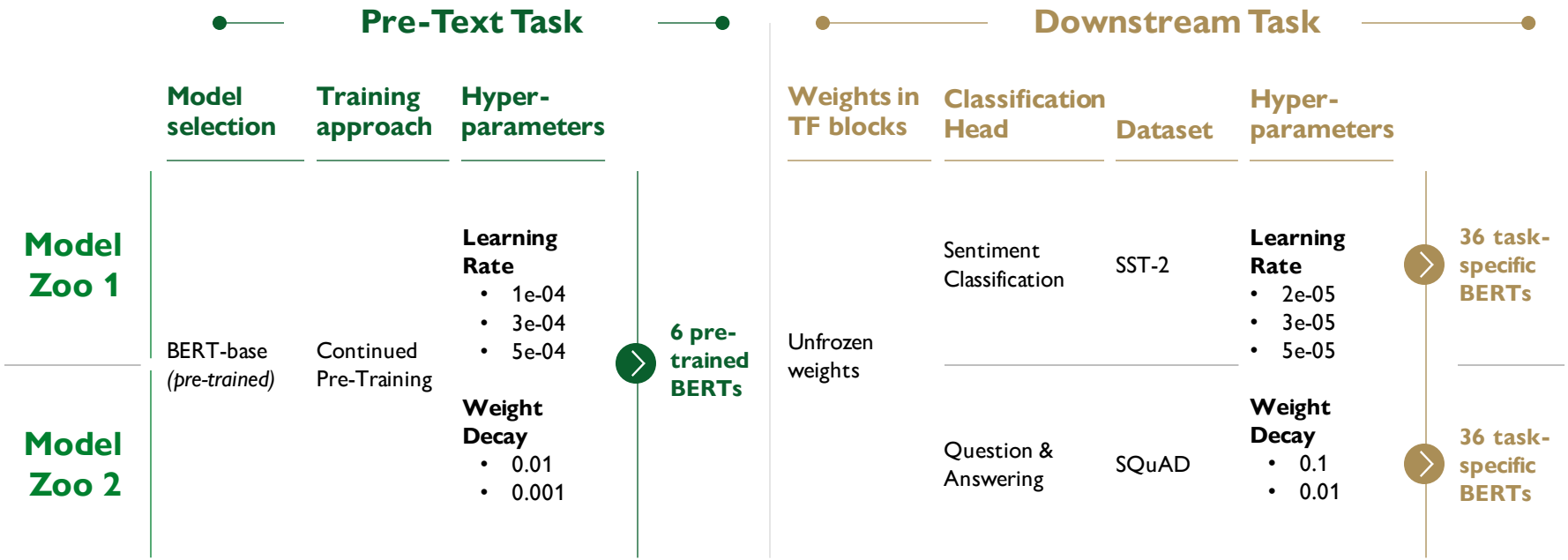
| | Pre-Text Task | Downstream Task |
|---|---|---|
| **Learning Rate** | **1e-04, 3e-04, 5e-04** | **2e-05, 3e-05, 5e-05** |
| **Weight Decay** | **0.01, 0.001** | **0.1, 0,01** |
| Number of Epochs | Constantly for 10 epochs | Constantly for 5 epochs |
| Batch Size | Depending on HW (16 or 32) | |
| Optimizer | Adam | AdamW |
| Dropout Rate | 0.1 | 0.1 |
| Learning Rate Scheduler | No warmup steps to keep learning rate constant | |
| Learning Rate Decay | No learning rate decay to keep learning rate constant | |
| Activation Function | GeLU | ReLU / SoftMax |
| Masking Fraction | 15% (same as original BERT) | not applicable |
| Max Sequence Len | Keep fixed at 512 | |
| Initialization | not applicable | Random initialization |

We believe that by changing the **learning rate & weight decay**, we have the most impact on weights (thus generate more diverse models)

Values are based on what was typically used during training and **slightly deviated to generate ranges**

We may need to adjust **the hyperparameters** during the next weeks depending on impact on the weights

# Given configuration setup & generating factors, we can define our model zoos and number of models



# Agenda

- Introduction
- Configuration Space
- Latent Generating Factors
- **Trainings Protocol**
- Status & Next Steps

# Training Protocol

- Every model in our model zoos is trained **according to the same protocol**
- We keep the **same train (80%), validation (10%) and test (10%) splits** for each zoo on the downstream tasks
- We will train during
  - **Pre-Text** for 10 epochs
  - **Downstream Task** for 5 epochs
- At every epoch, we will **record the model checkpoints** as well as accuracy and loss of all splits
- Ultimately, we aim to have **~500 model states** (~55B parameters)
  - **Pre-Text Task:** ~60 model states
  - **Downstream Task:** ~220 model states per model zoo

# Agenda

Introduction

Configuration Space

Latent Generating Factors

Trainings Protocol

**Status & Next Steps**

## Progress So Far

- ✓ Conduct literature research
- ✓ Decide on configuration setup
- ✓ Explore latent generating factors
- ✓ Build code for pre-training

## Next steps

- ~ Build trainings pipeline
- ~ Train our model zoos
- ~ Investigate results *(e.g., agreement score, weight space)*
- ~ Apply sparsification

✓ *Done*
~ *Open*

University of St.Gallen


# Any questions?

# References

# References


Devlin, J., Chang, M-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.

Gokaslan, A., Cohen, V., Ellie, P., & Tellex, S. (2019). Openwebtext corpus.

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). SQuAD: 100,000+ questions for Machine Comprehension of Text.

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2020). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter.

Schürholt, K., Taskiran, D., Knyazev, B., Giró-i-Nieto, X., & Borth, D. (2022). Model Zoos: A Dataset of Diverse Populations of Neural Network Models.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C., Ng, A., & Potts, C. (2013). Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2023). Attention Is All You Need.

Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., Wang, F., & Liu, Q. (2020). TinyBERT: Distilling BERT for Natural Language Understanding.