Topic 8
Using pretrained models + classification continued

Notes

Freeze Layer

Print no of layers

require grad - True
\ False

Assign net 3 will be like the last part of Pretrain_Models

Note Book. Load Pretrained model

I have brought up pretrained models before

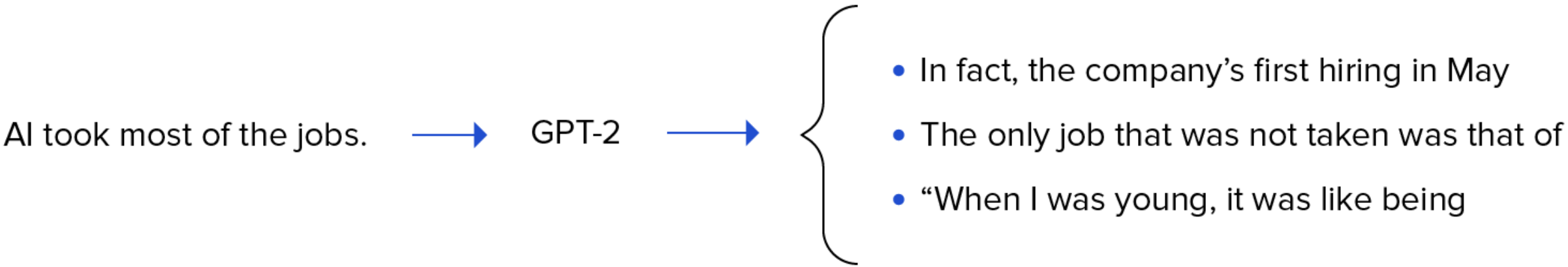
- But haven’t really explained what they are
- Any ideas?
- Can you name any pre-trained models? Ones we have covered in class?

We have already seen pre-trained models

- Word2Vec- word2vec-google-news-300, word2vec-ruscorpora-300
- GloVe- glove-wiki-gigaword-50, glove-wiki-gigaword-100
- SpaCy- en_core_web_sm (ner, parser, tagger, etc.)
- NLTK- PerceptronTagger, PunktSentenceTokenizer, etc.

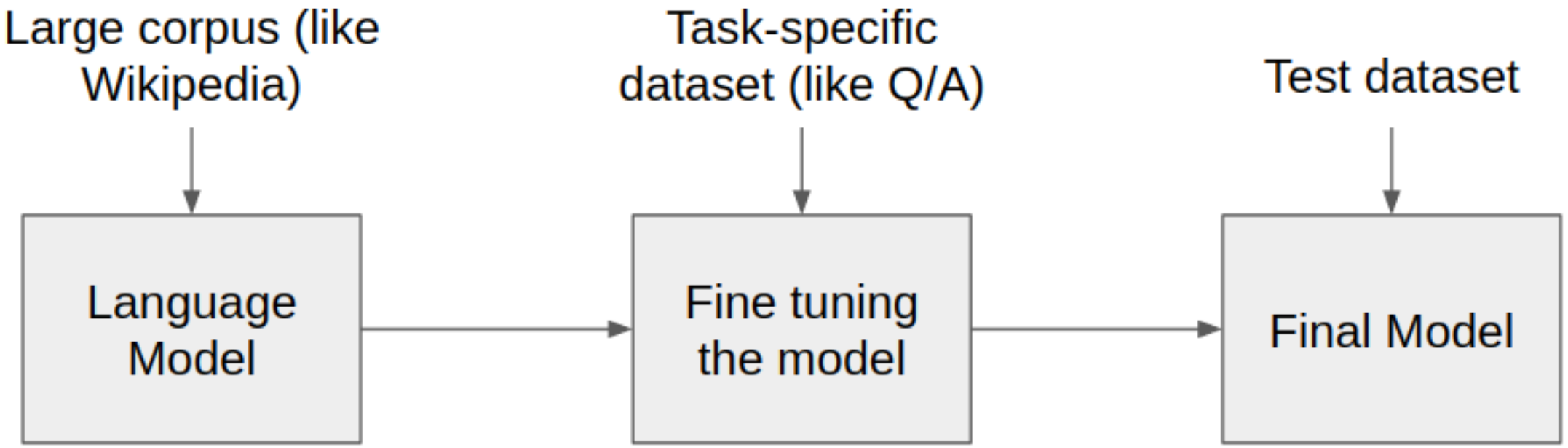
What if we want a Chatbot that handles customer complaints

- Isn’t it easier to start with something that sounds close to human to start off?



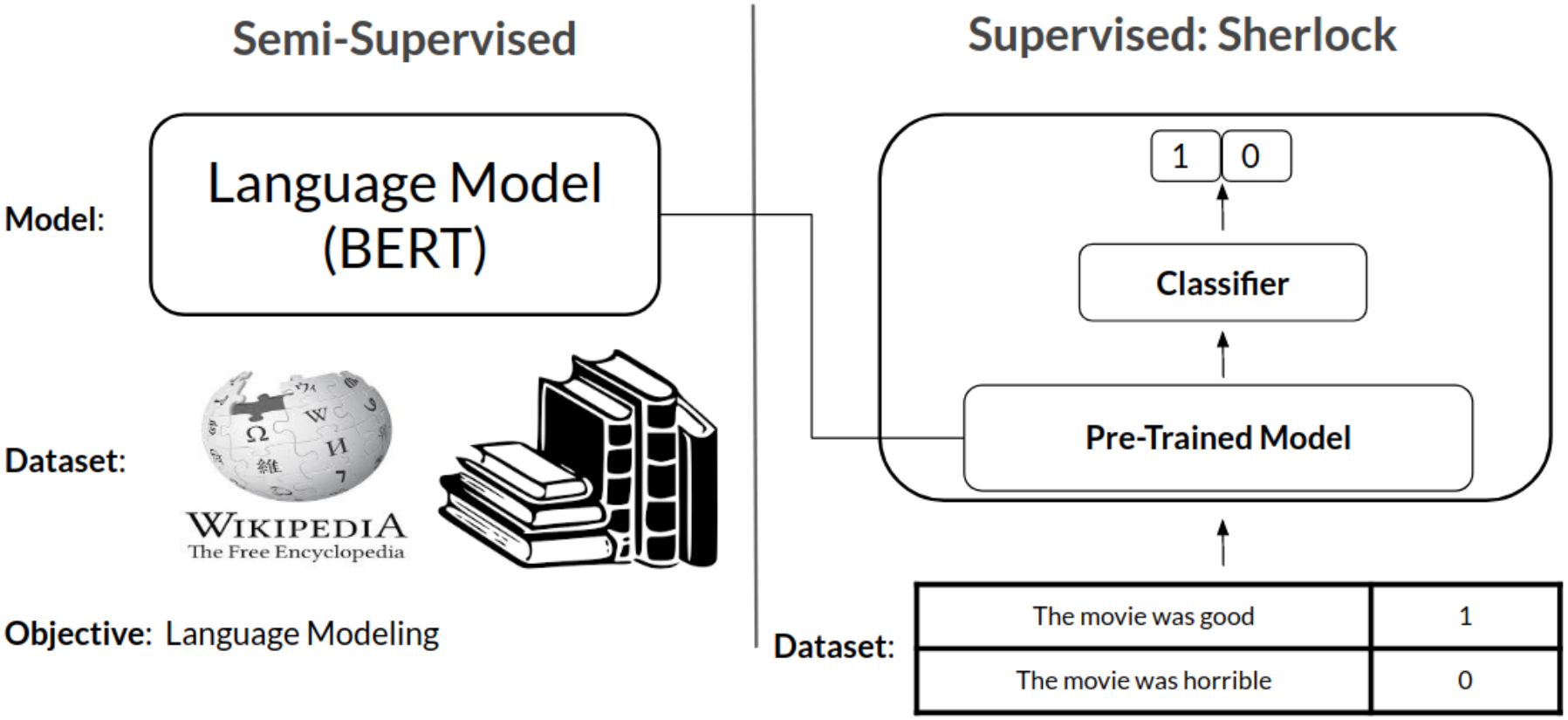
Using models that already exist

- Most will use these in an applied setting

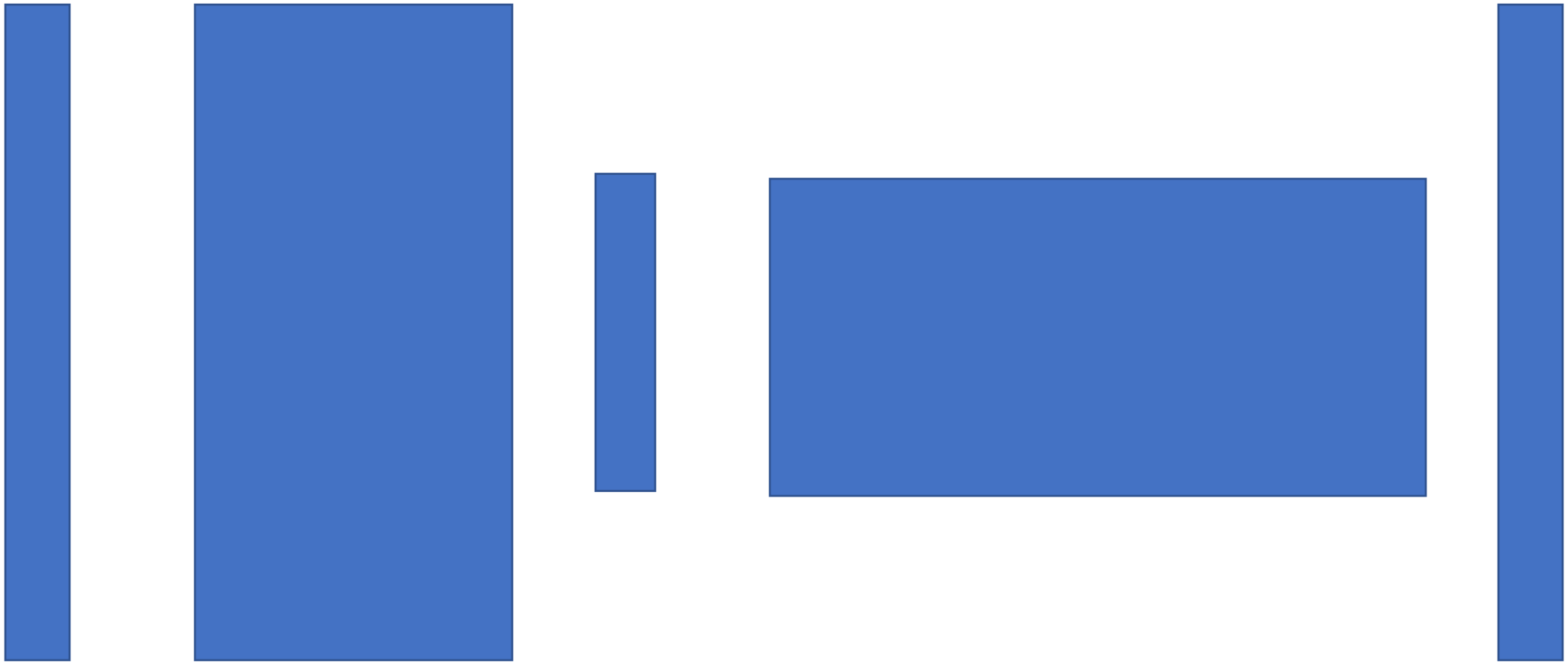


Adding our own training data to the model

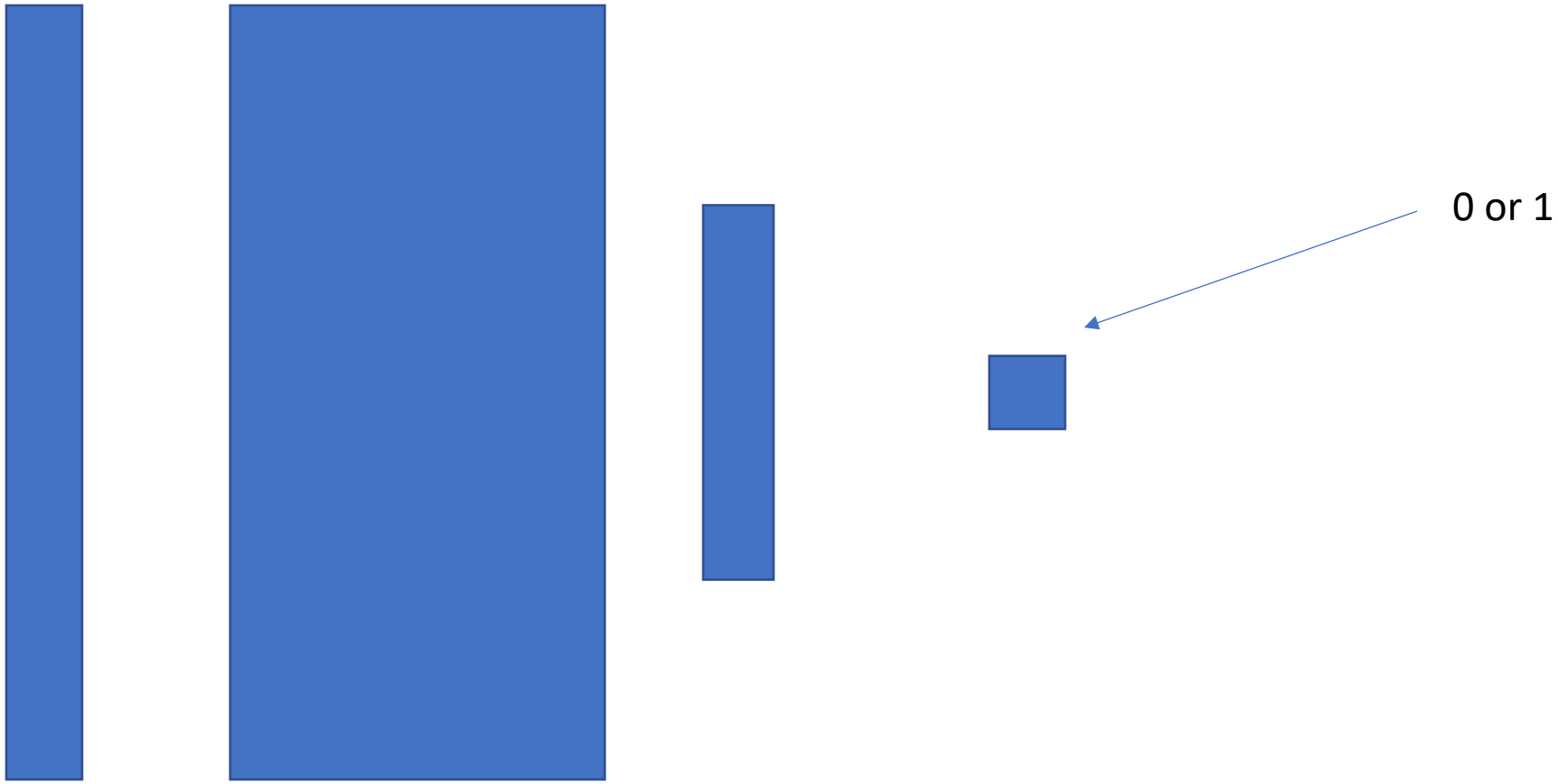
- What the data looks like will vary
- Depends on what the model accepts



Word2Vec Representation



Tack on a binary classifier



Why use pretrained models?

- The model creator(s) have already put in the effort to design a benchmark model for you!
- Instead of building a model from scratch to solve a similar NLP problem, we can use that pretrained model on our own NLP dataset
- A bit of fine-tuning will be required but it saves us a ton of time and computational resources
- Lack of data

Techniques using pre-trained models

- **Fine-tuning:** Update the parameters of a model (or part of it)
- **Transfer learning:** Using previously trained (model) and (optional) adding some new trainable layers. Also possible to fine-tune on your own dataset.

Preparing Text Data for Neural Networks

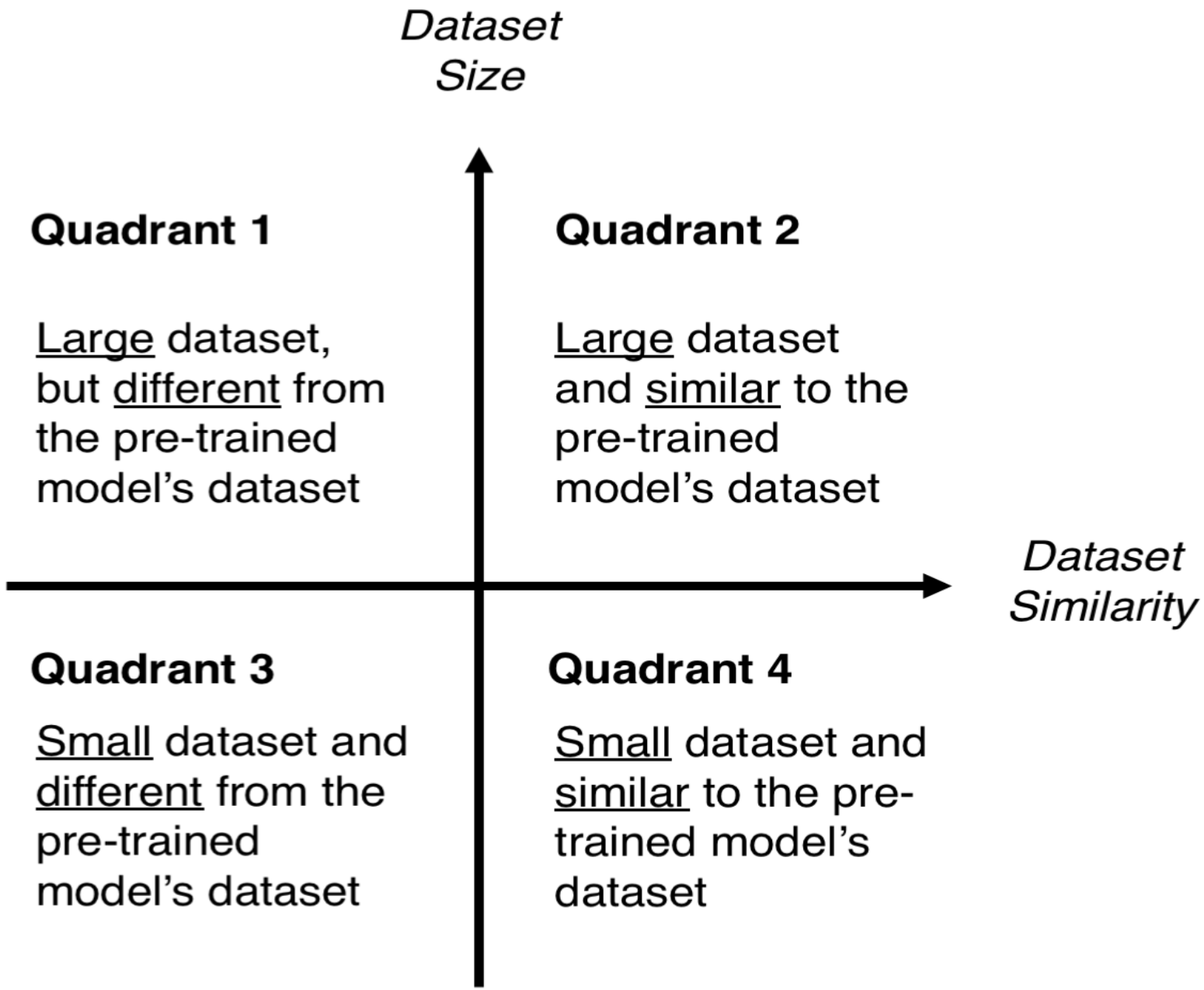
- Tokenization and conversion to word index vectors
- Pad text sequences so that all vectors are the same length
- Map every word to an embedding vector
- Use the output above as input to the neural network

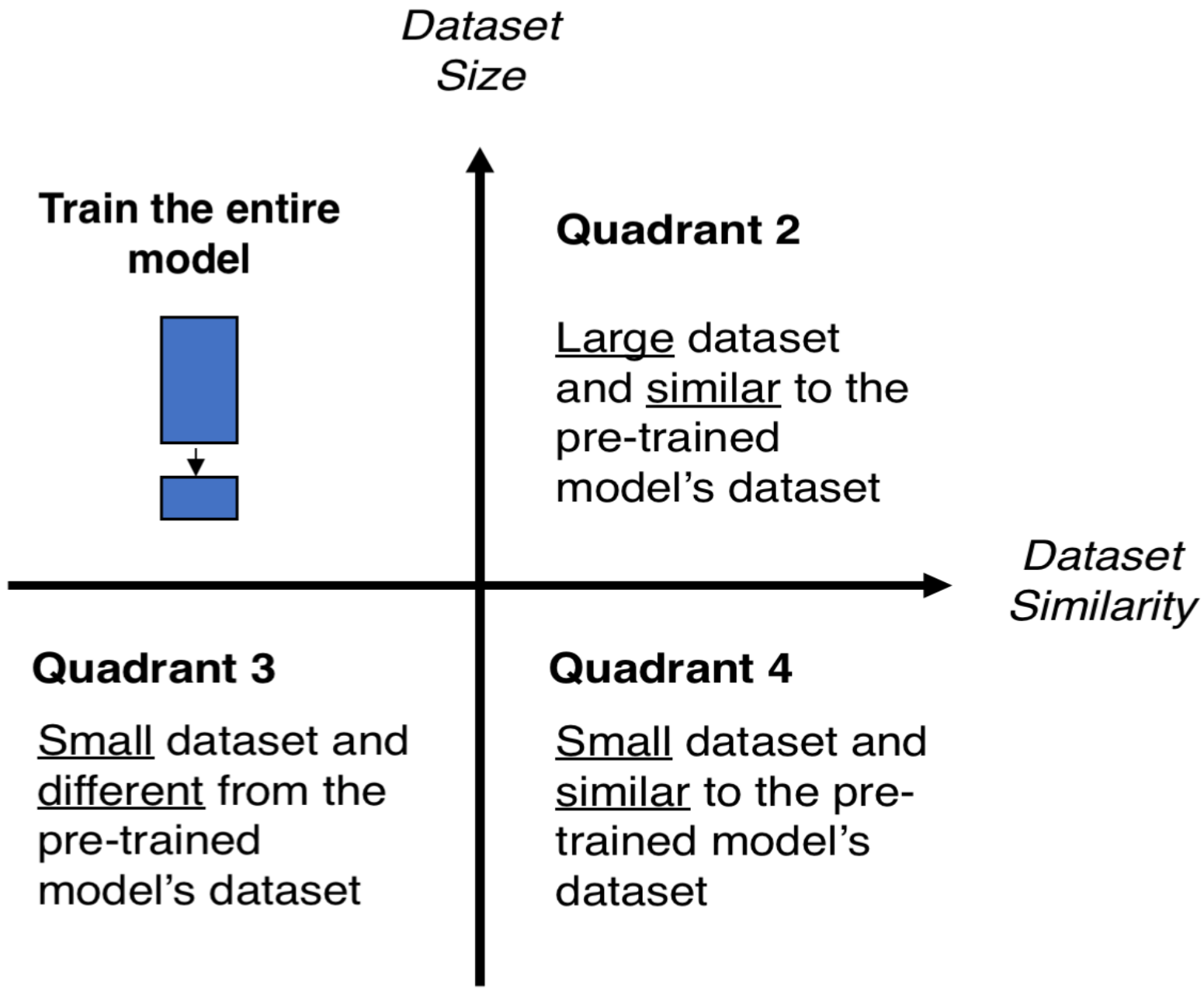
What does it mean to freeze layers?

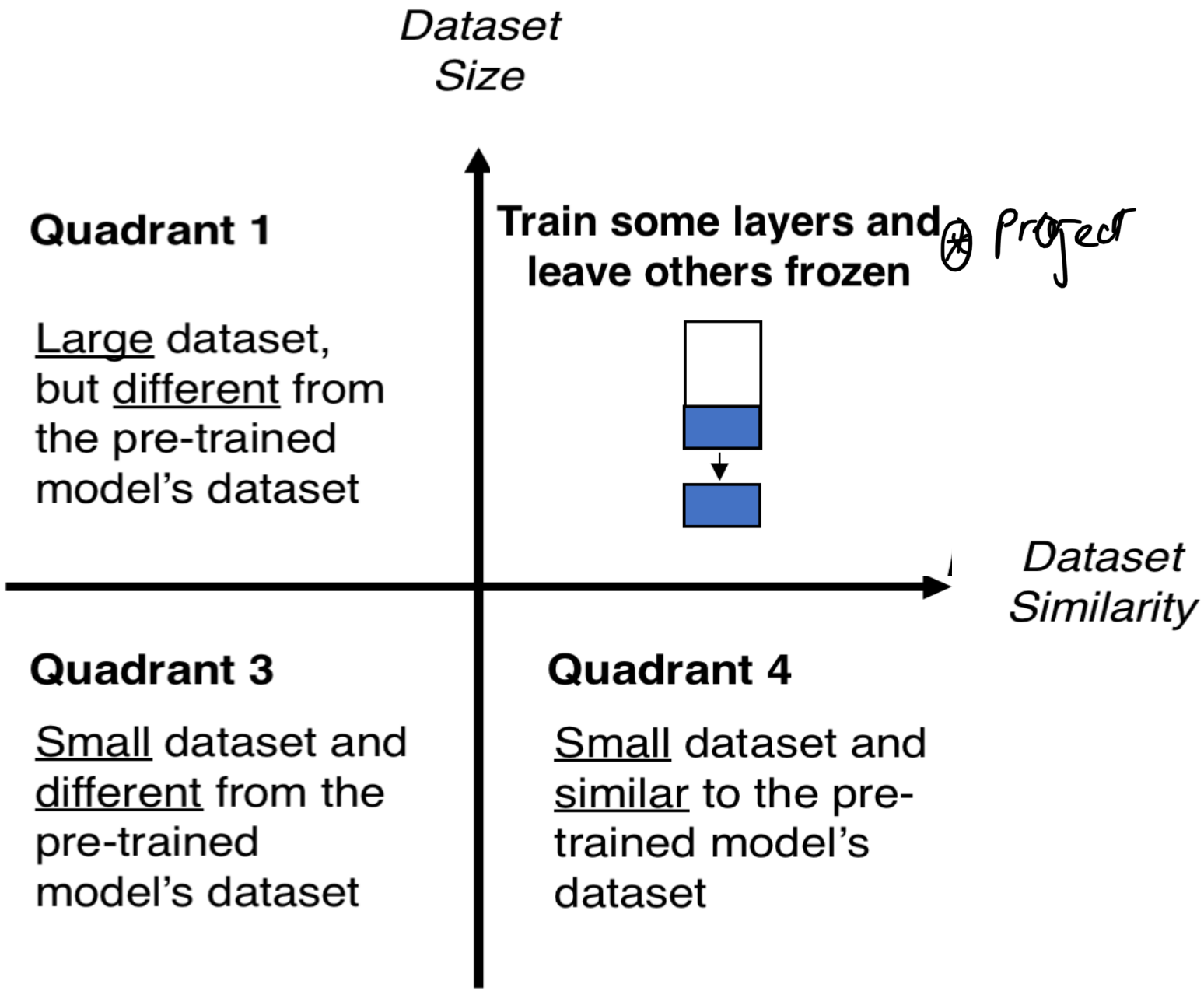
- Disabling gradient computation and backpropagation for the weights of these layers

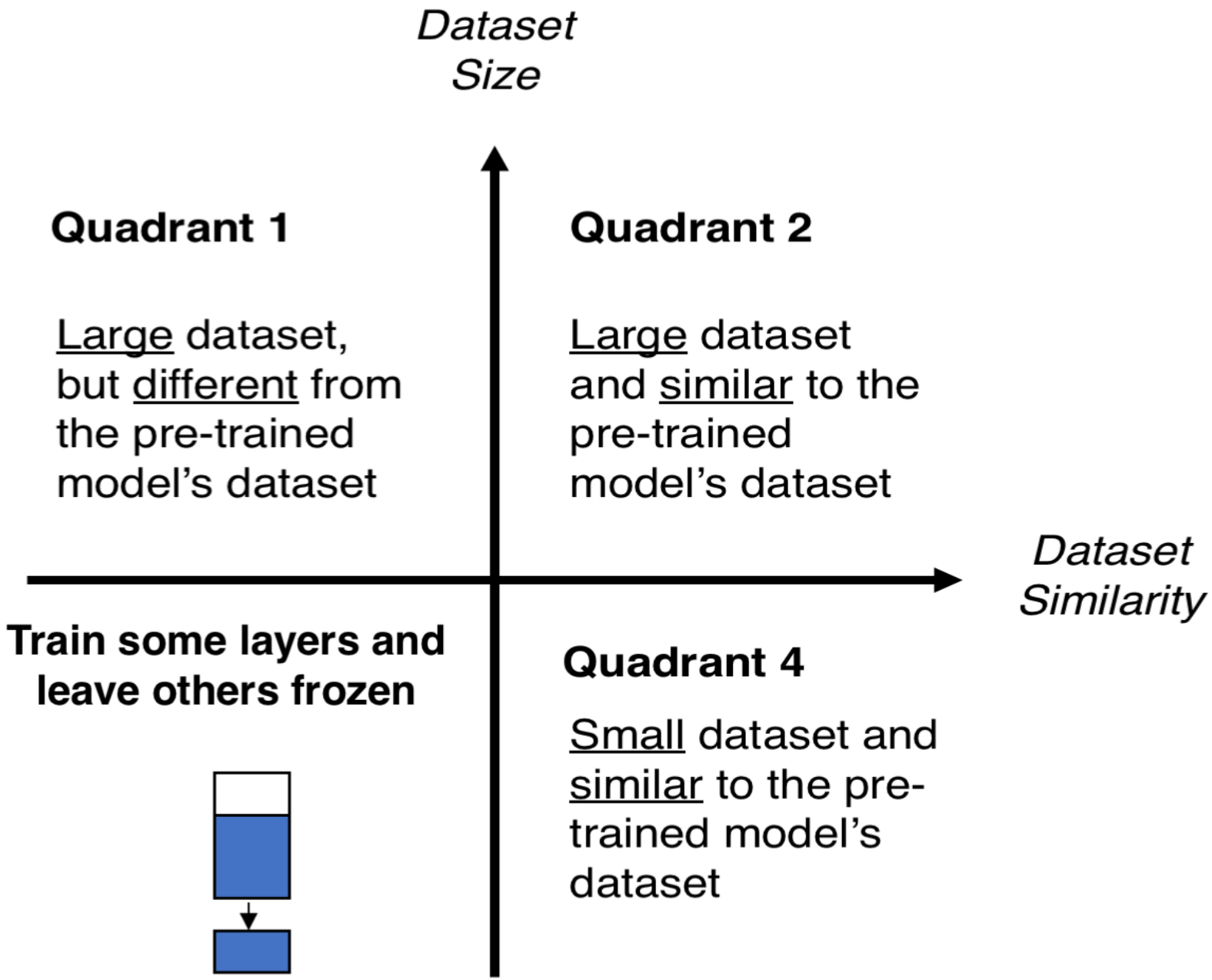
- Within each layer, there are parameters (or weights), which can be obtained using .param() on any children (i.e. layer)
- All models have a function that returns it's layers
- Every parameter has an attribute called **requires_grad** which is by default True → this means that the weight can be updated

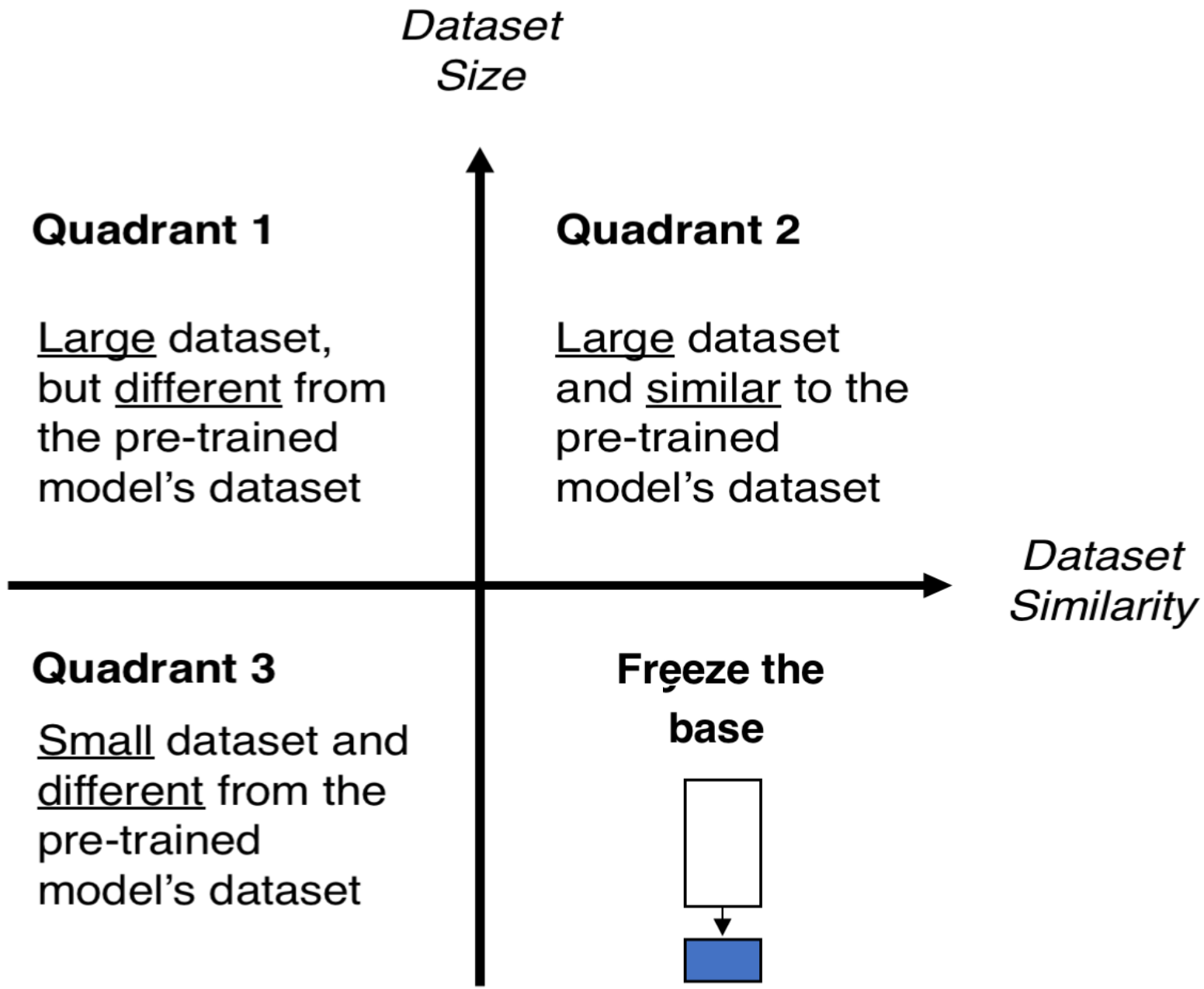
Freeze everything but last layer


Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset


Dataset Size
Train the entire model
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset


Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Train some layers and leave others frozen
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Quadrant 4
Small dataset and similar to the pre-trained model's dataset


Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Train some layers and leave others frozen
Quadrant 4
Small dataset and similar to the pre-trained model's dataset


Dataset Size
Quadrant 1
Large dataset, but different from the pre-trained model's dataset
Quadrant 2
Large dataset and similar to the pre-trained model's dataset
Dataset Similarity
Quadrant 3
Small dataset and different from the pre-trained model's dataset
Freeze the base

Pytorch Code Examples

Freeze all the layers:

```
for param in model.parameters():
        param.requires_grad = False
```

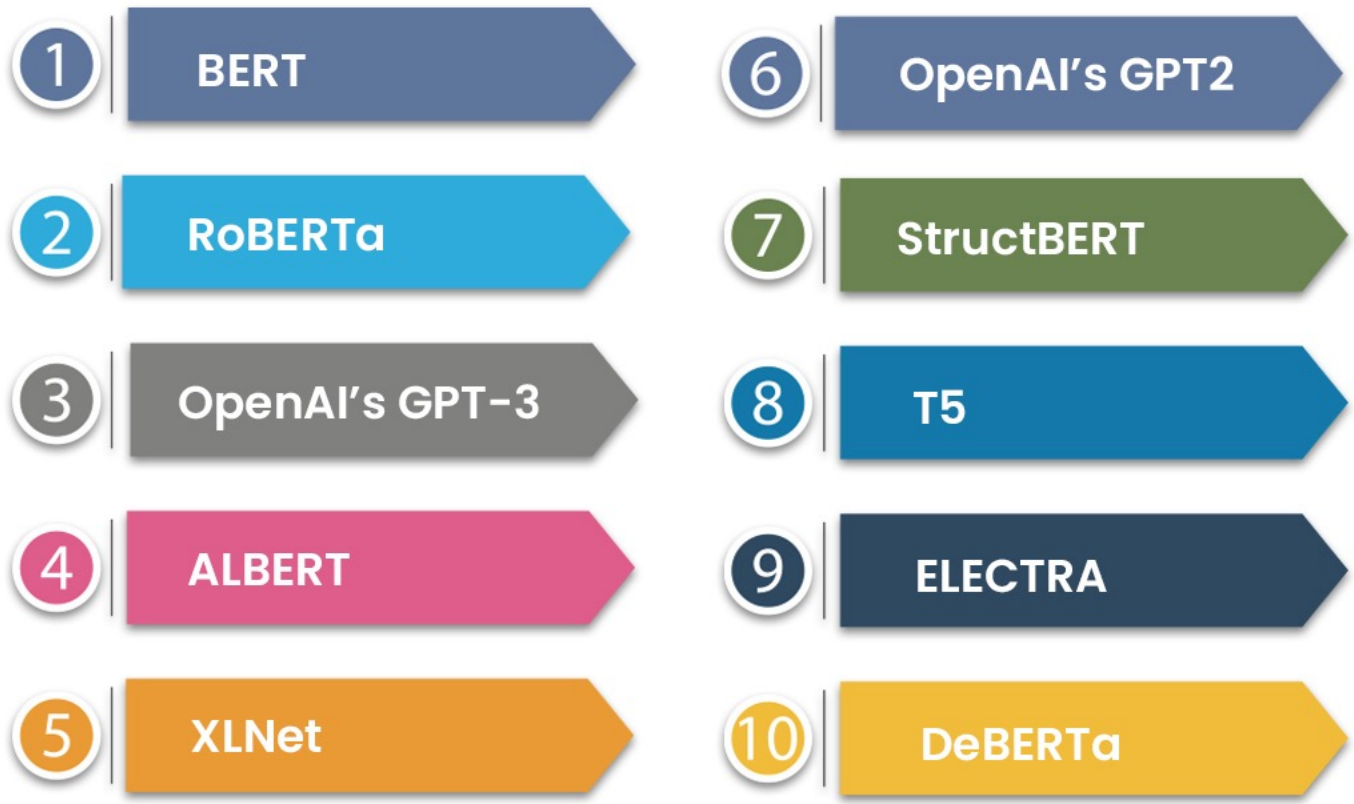
Freeze all but the last layer:

```
for param in model.parameters():
        param.requires_grad = False
model.fc.requires_grad = True
```

Which layers should we re-train?

- Depends on the domain
- Start by re-training the last layers
  - Work backwards if performance is not satisfactory

Top 10 Pre-Trained NLP Language Models for AI Application Building



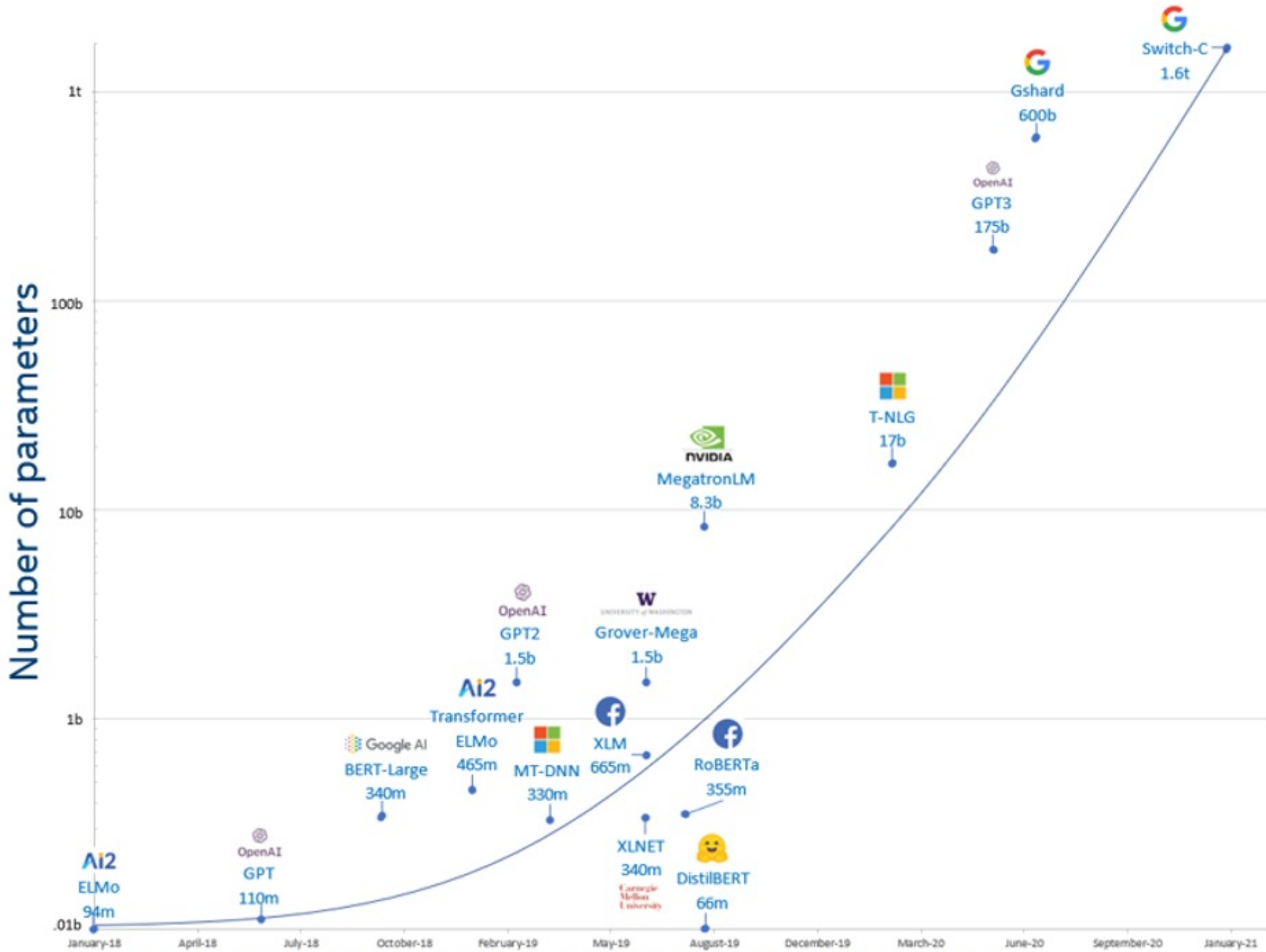
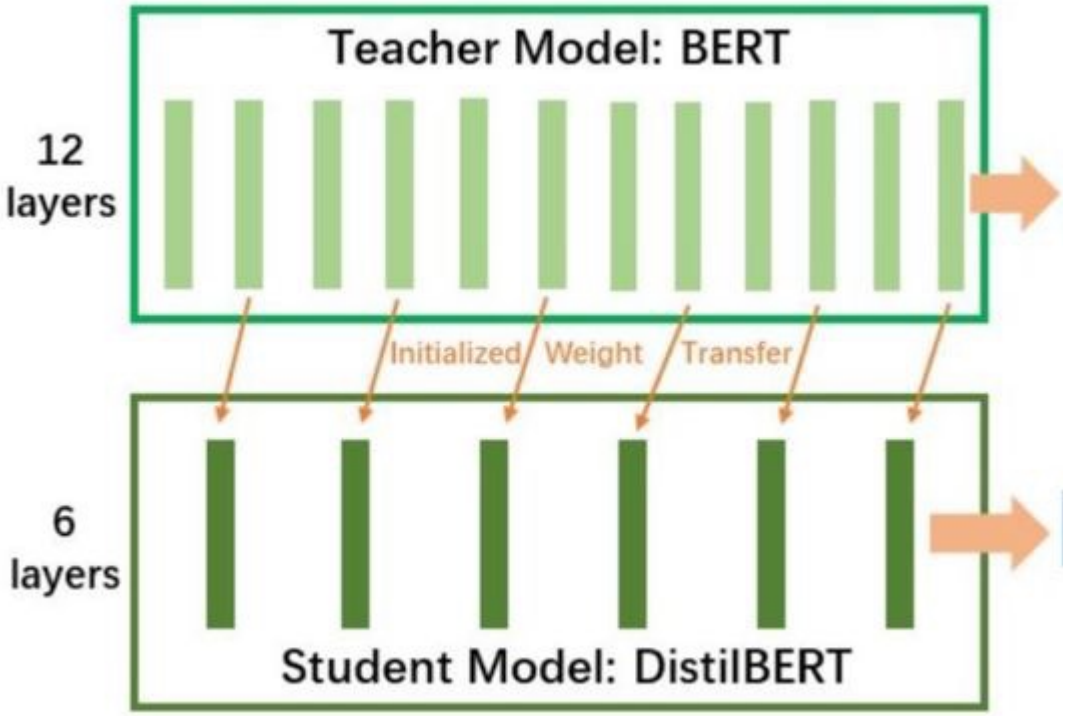


Figure 1: Exponential growth of number of parameters in DL models

Some Feasible Options



- DistilBert by Hugging Face
- ALBERT by Google
- ELMo by AllenNLP
- Flair by Zalando Research
- RoBERTa-tiny by Hugging Face
- These models can handle NLP tasks such as sentiment analysis, named entity recognition, and text classification. They can also be fine-tuned on smaller datasets.

Planning to use pre-trained models?

- Find out about your models early if you want to use something pretrained:
  - Can you load the model into your environment?
  - How many layers does it have? What are their names?
  - What should the input look like to use this model? Can you get your data in the right format?
  - Can you freeze all but the last layer and train on a small dataset?

Next time

- More on classification with deep learning and pre-trained models