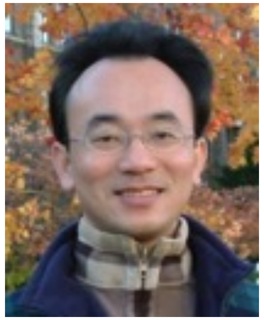
Artificial Intelligence for Text Analytics

Deep Learning, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics

1102AITA10
MBA, IM, NTPU (M5026) (Spring 2022)
Tue 2, 3, 4 (9:10-12:00) (B8F40)

https://meet.google.com/paj-zhhj-mya



Min-Yuh Day, Ph.D,
Associate Professor

Institute of Information Management, National Taipei University

https://web.ntpu.edu.tw/~myday

2022-05-24

Syllabus

Week	Date	Subject/Topics
1	2022/02/22	Introduction to Artificial Intelligence for Text Analytics
2	2022/03/01	Foundations of Text Analytics: Natural Language Processing (NLP)
3	2022/03/08	Python for Natural Language Processing
4	2022/03/15	Natural Language Processing with Transformers
5	2022/03/22	Case Study on Artificial Intelligence for Text Analytics I
6	2022/03/29	Text Classification and Sentiment Analysis

Syllabus

Week	Date	Subject/Topics
7	2022/04/05	Tomb-Sweeping Day (Holiday, No Classes)
8	2022/04/12	Midterm Project Report
9	2022/04/19	Multilingual Named Entity Recognition (NER), Text Similarity and Clustering
10	2022/04/26	Text Summarization and Topic Models
11	2022/05/03	Text Generation
12	2022/05/10	Case Study on Artificial Intelligence for Text Analytics II

Syllabus

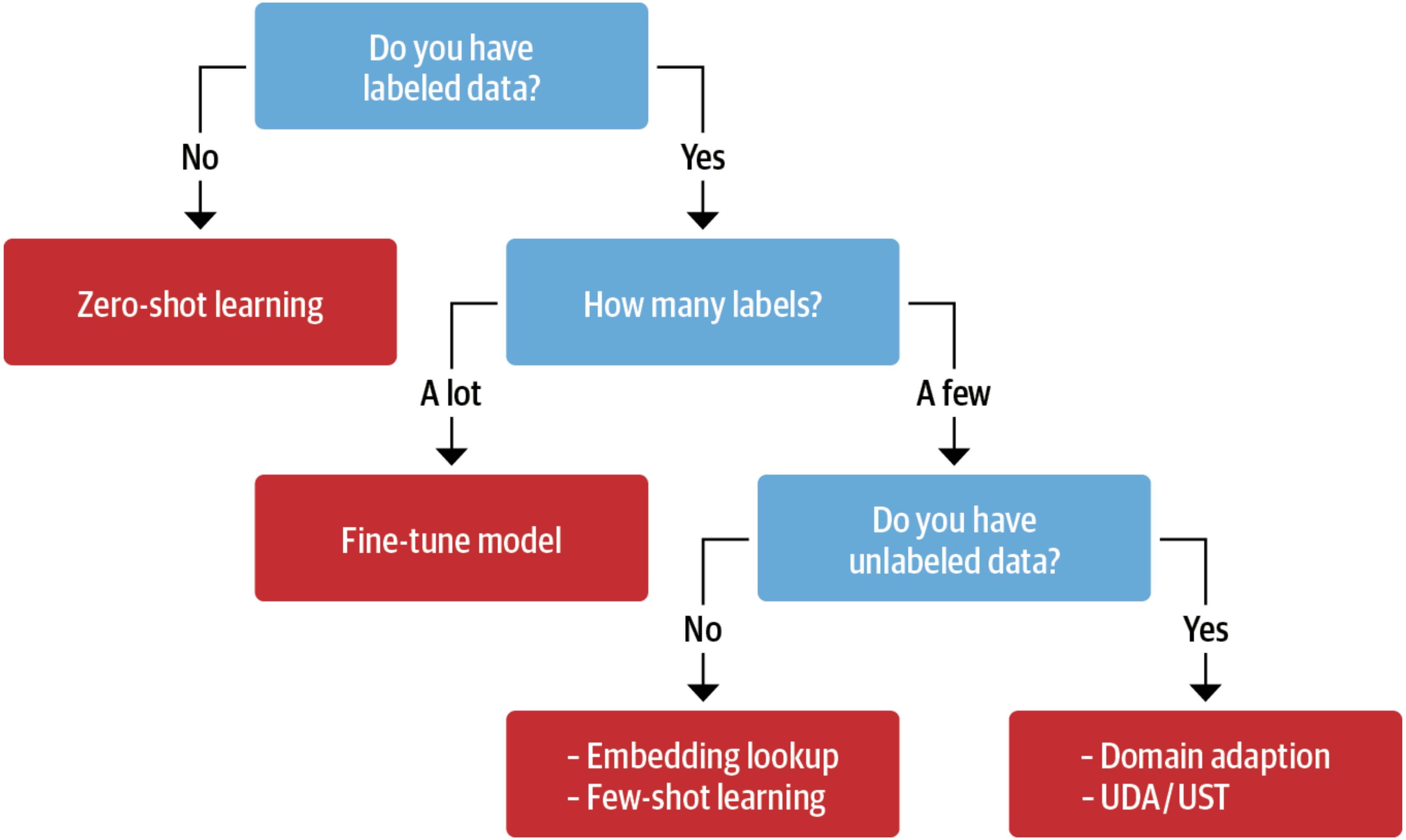
Week	Date	Subject/Topics
13	2022/05/17	Question Answering and Dialogue Systems
14	2022/05/24	Deep Learning, Transfer Learning, Zero-Shot, and Few-Shot Learning for Text Analytics
15	2022/05/31	Final Project Report I
16	2022/06/07	Final Project Report II
17	2022/06/14	Self-learning
18	2022/06/21	Self-learning

Deep Learning,
Transfer Learning,
Zero-Shot,
and Few-Shot Learning
for Text Analytics

Outline

- Deep Learning
  - Transfer Learning
    - Pre-training, Fine-Tuning (FT)
- Few-Shot Learning (FSL)
  - Meta Learning: Learn to Learn
- One-Shot Learning (1SL)
- Zero-Shot Learning (0SL)(ZSL)

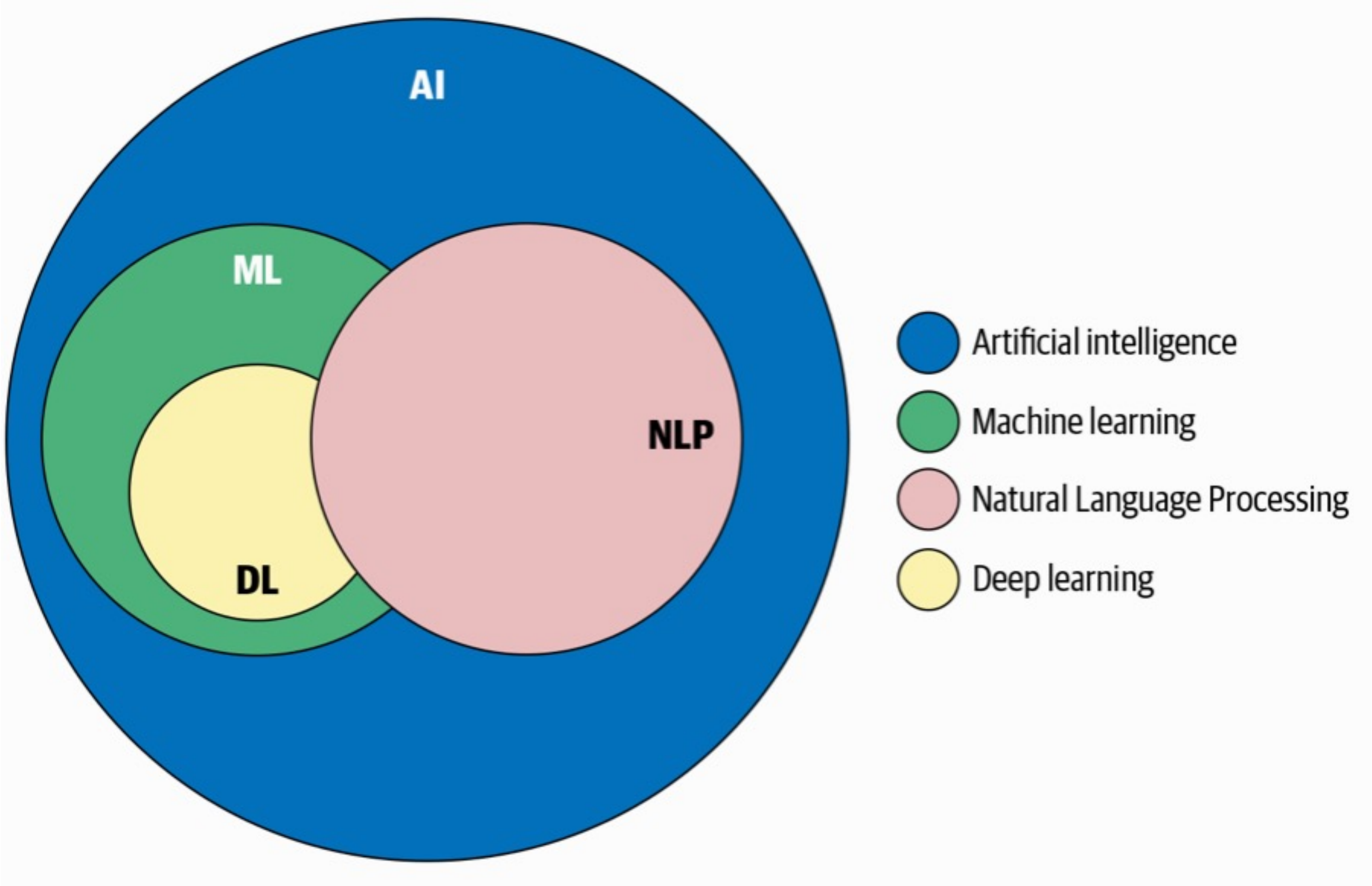
Transfer Learning, Fine-tuning, Few-shot learning



Unsupervised Data Augmentation (UDA)
Uncertainty-aware Self-Training (UST)

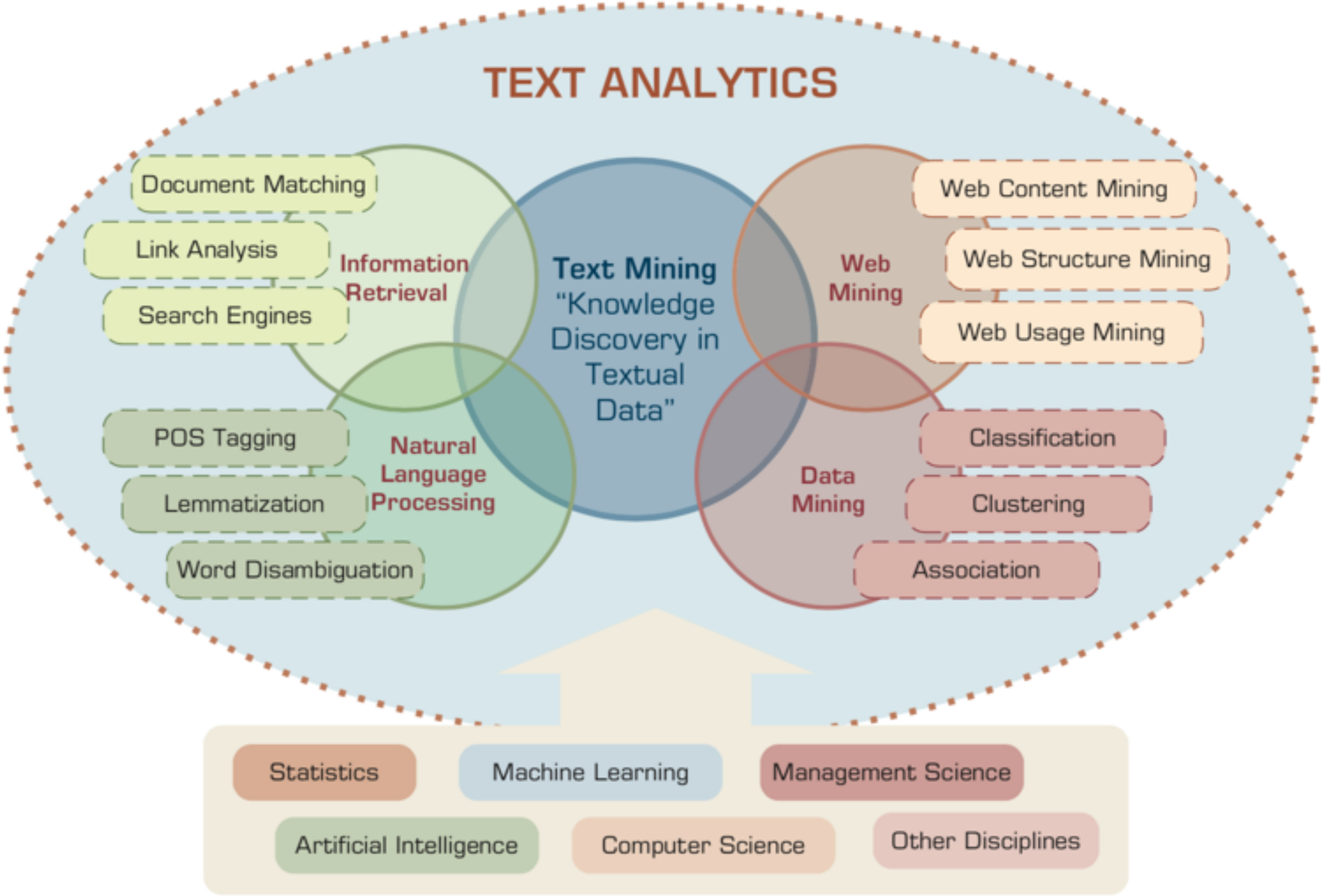
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

AI, NLP, ML, DL



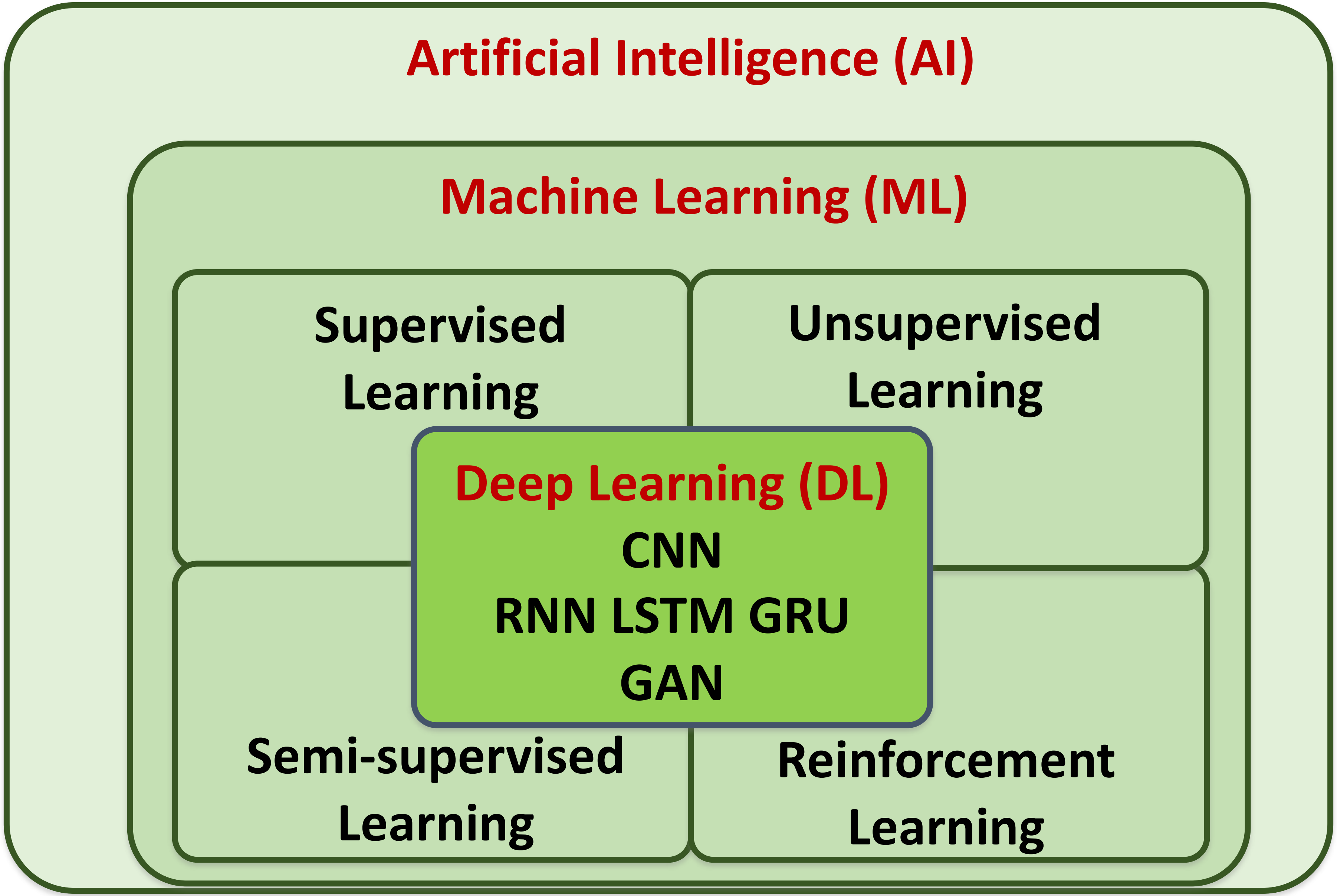
Source: Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.

Text Analytics and Text Mining



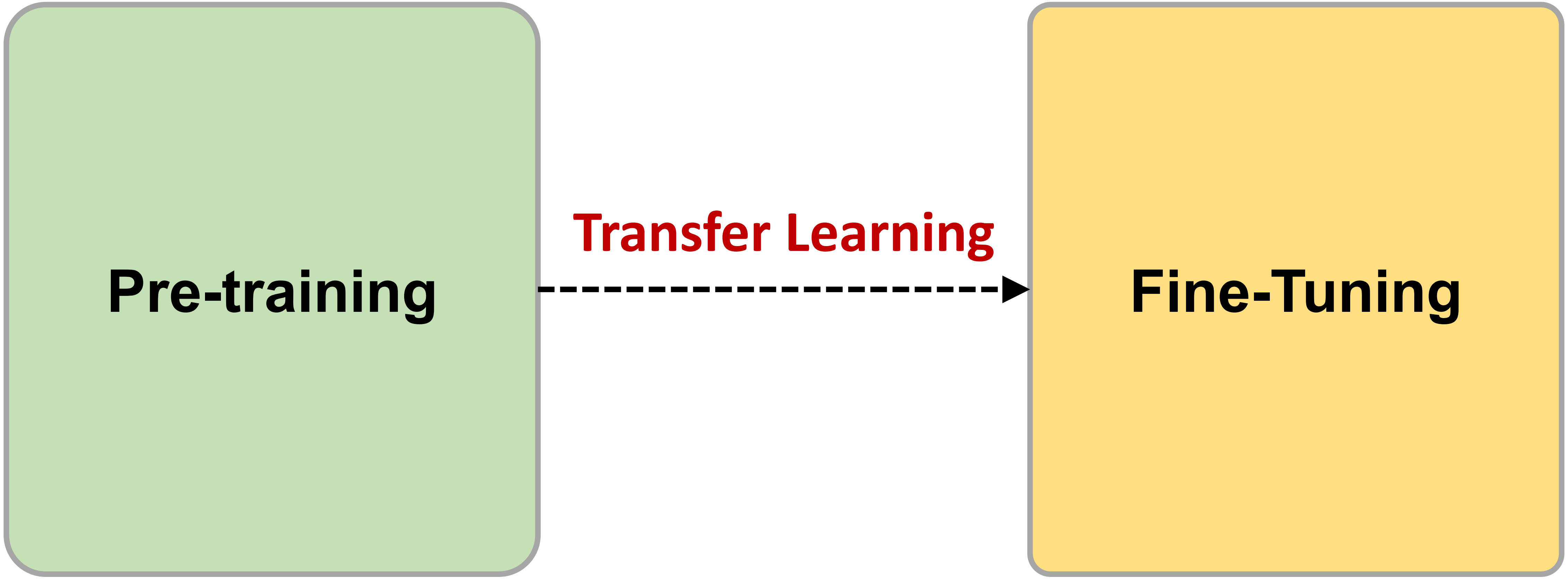
Source: Ramesh Sharda, Dursun Delen, and Efraim Turban (2017), Business Intelligence, Analytics, and Data Science: A Managerial Perspective, 4th Edition, Pearson

AI, ML, DL



Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/deep_learning.html

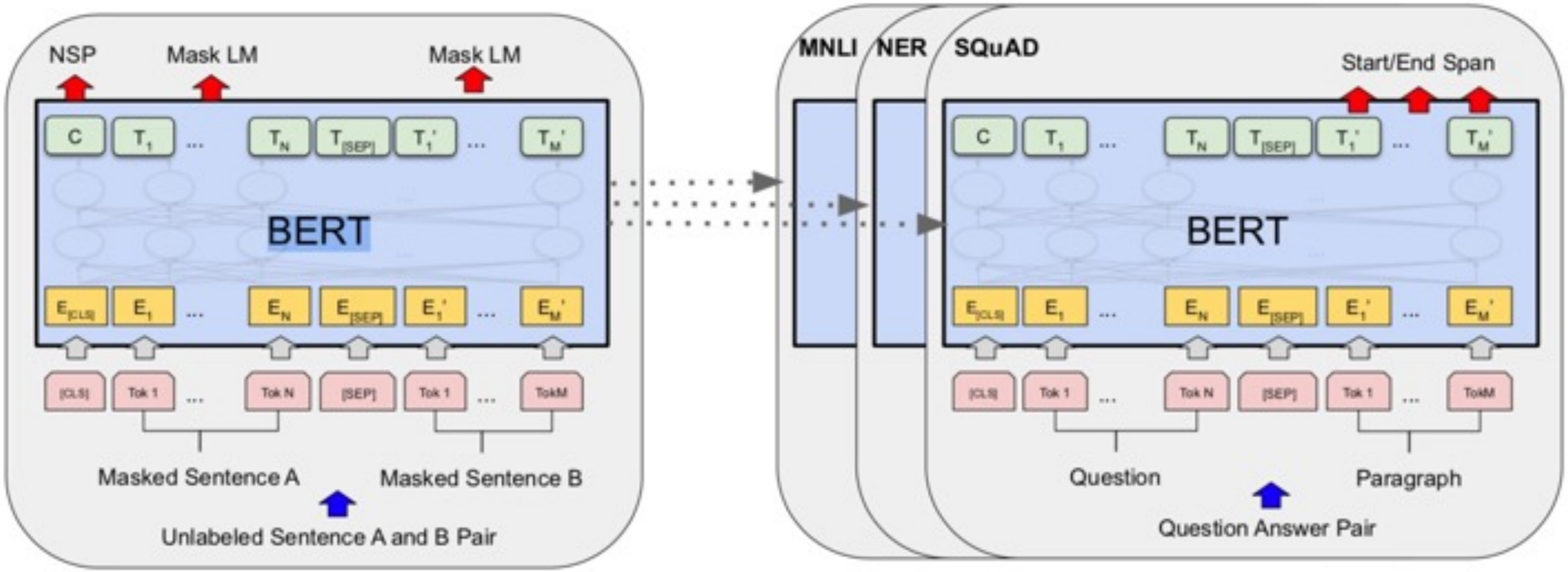
Transfer Learning



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT

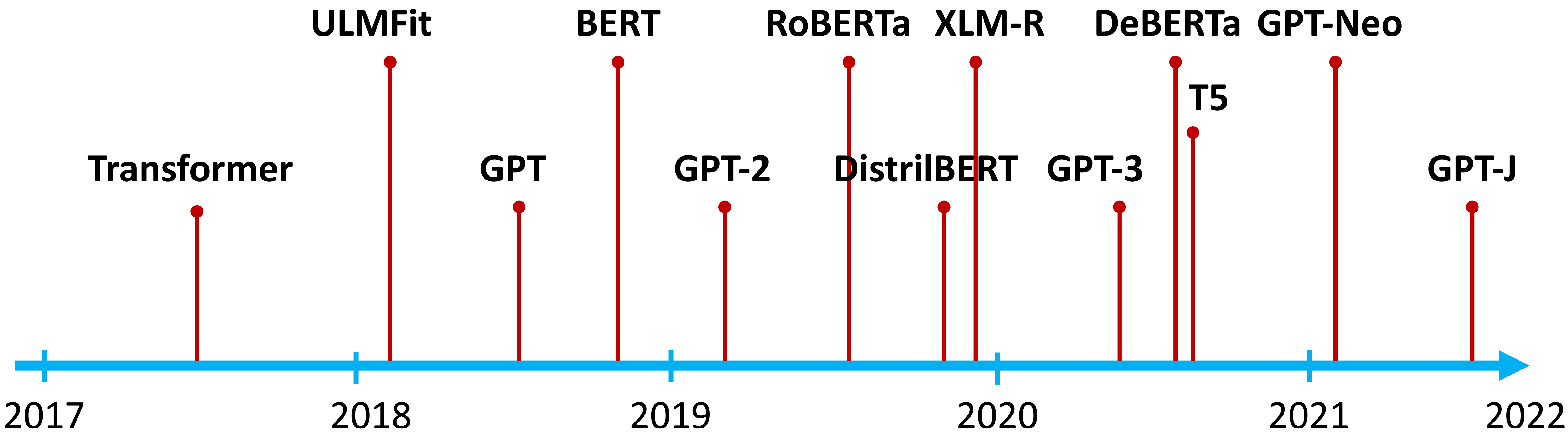


Pre-training

Fine-Tuning

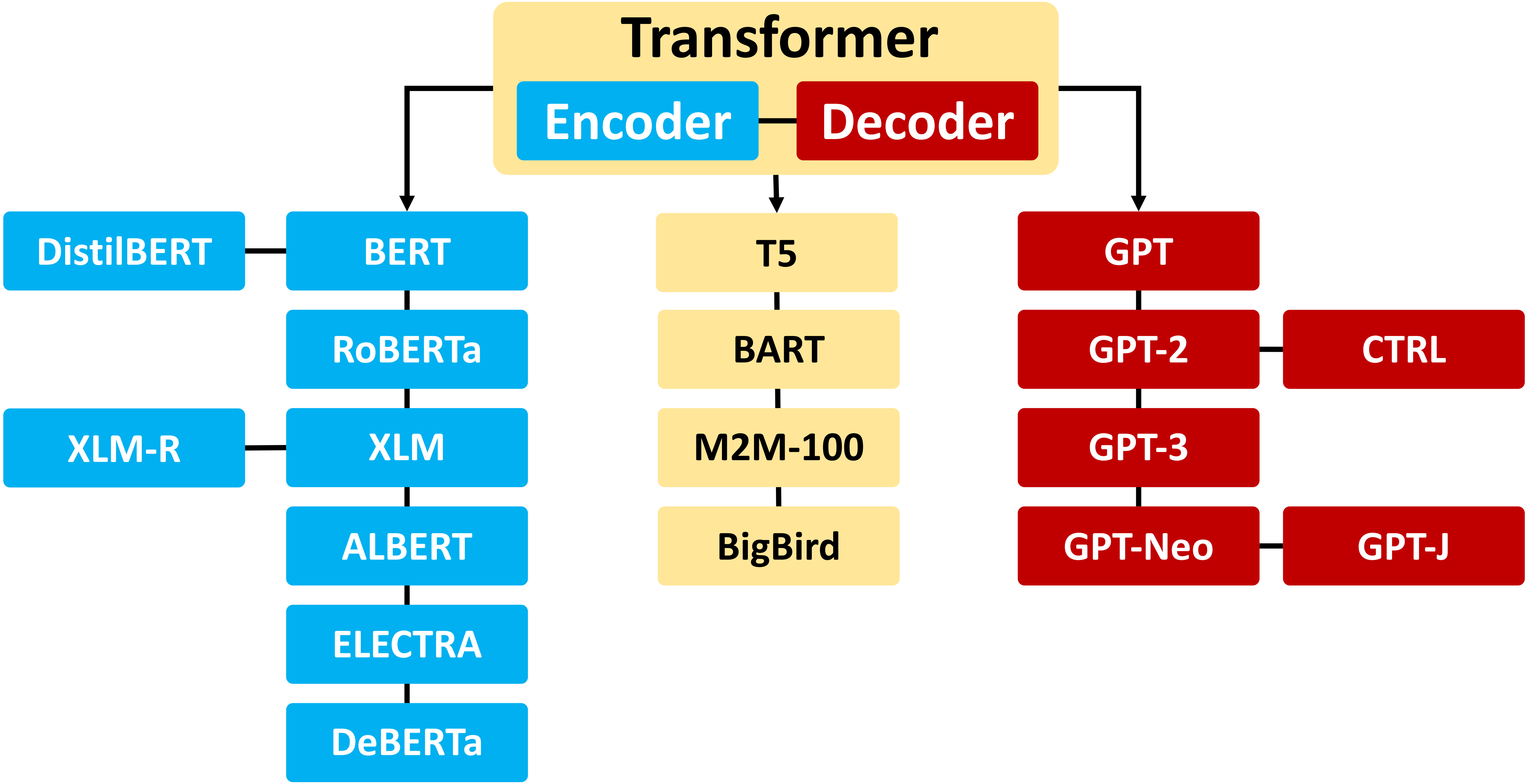
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

The Transformers Timeline



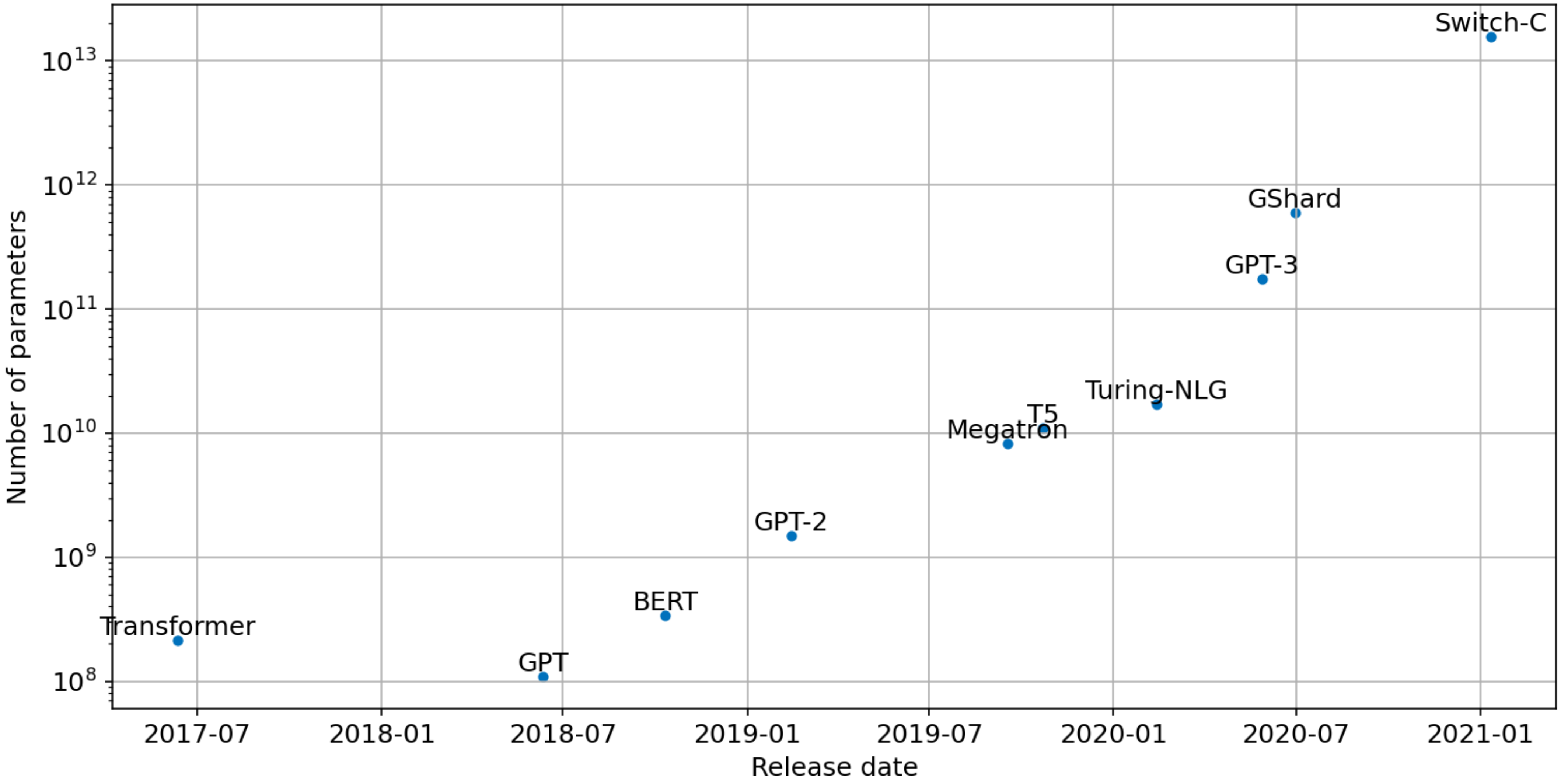
Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Transformer Models

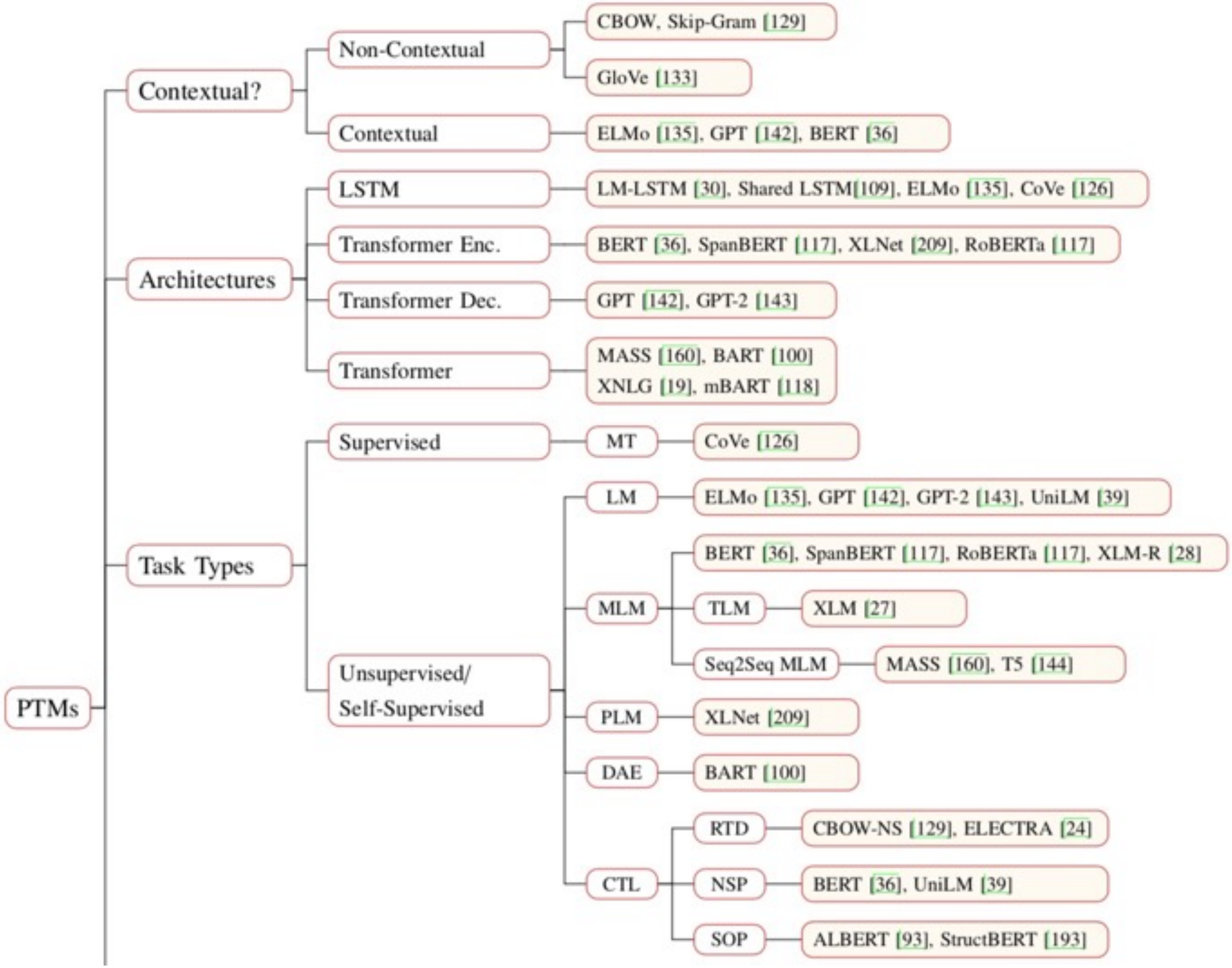


Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Scaling Transformers

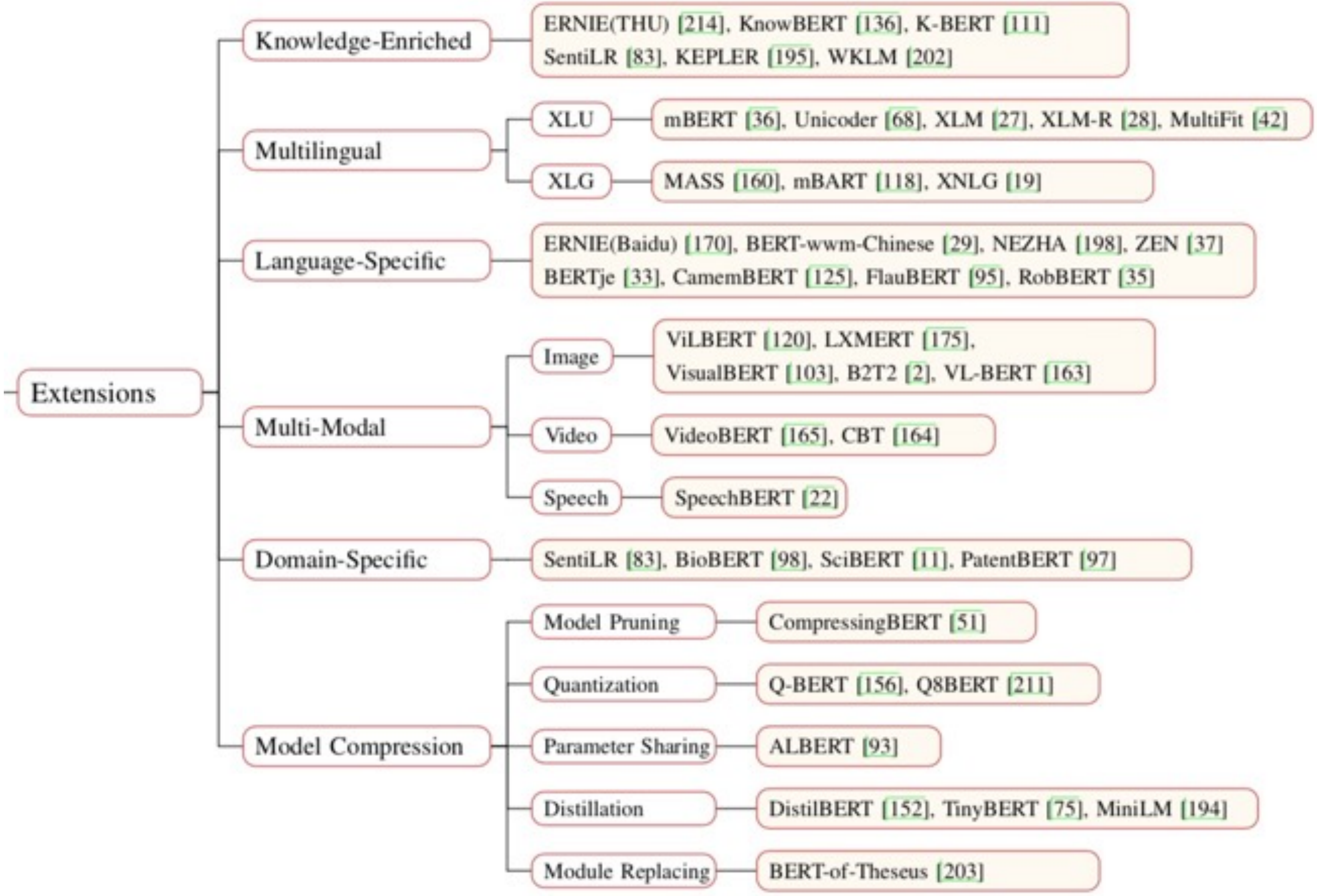


Pre-trained Models (PTM)



Source: Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained Models for Natural Language Processing: A Survey." arXiv preprint arXiv:2003.08271 (2020).

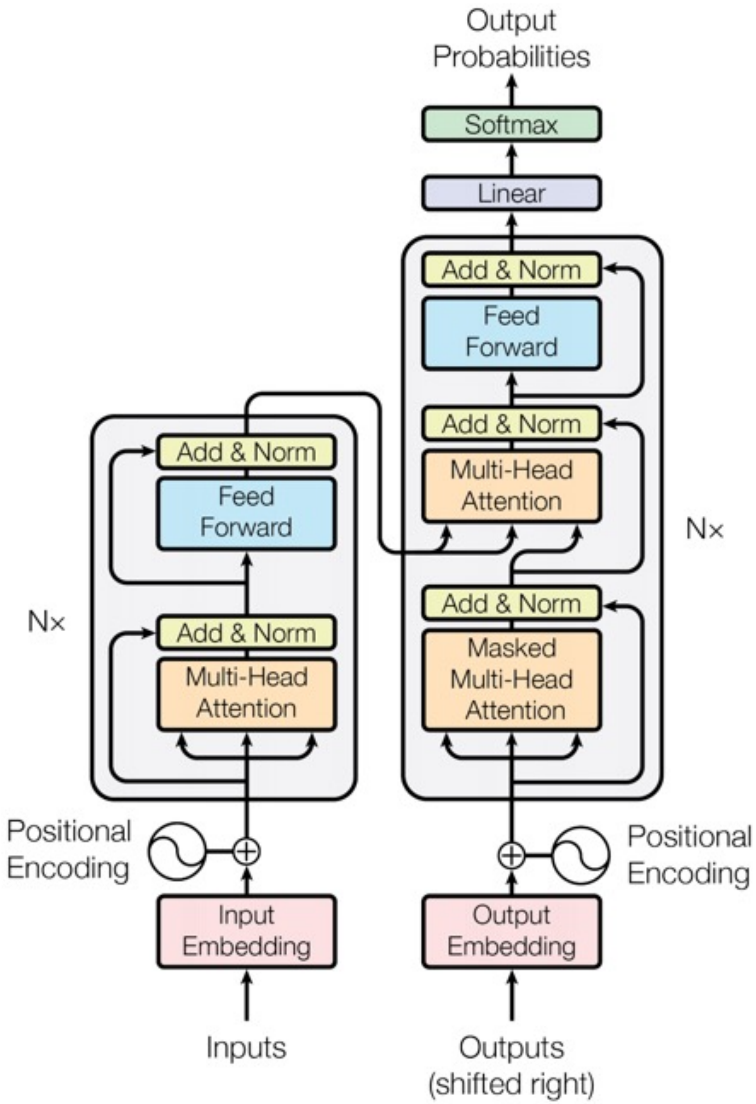
Pre-trained Models (PTM)



Source: Qiu, Xipeng, Tianxiang Sun, Yige Xu, Yunfan Shao, Ning Dai, and Xuanjing Huang. "Pre-trained Models for Natural Language Processing: A Survey." arXiv preprint arXiv:2003.08271 (2020).

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

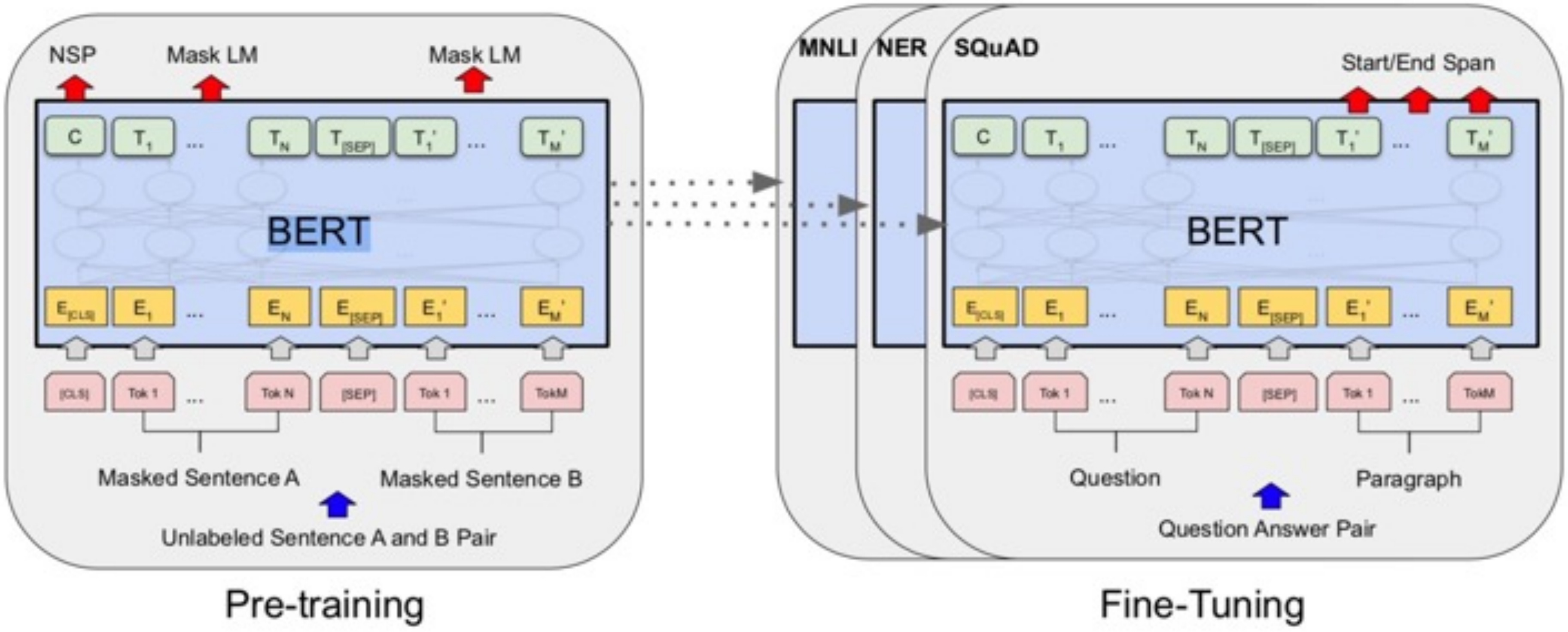


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

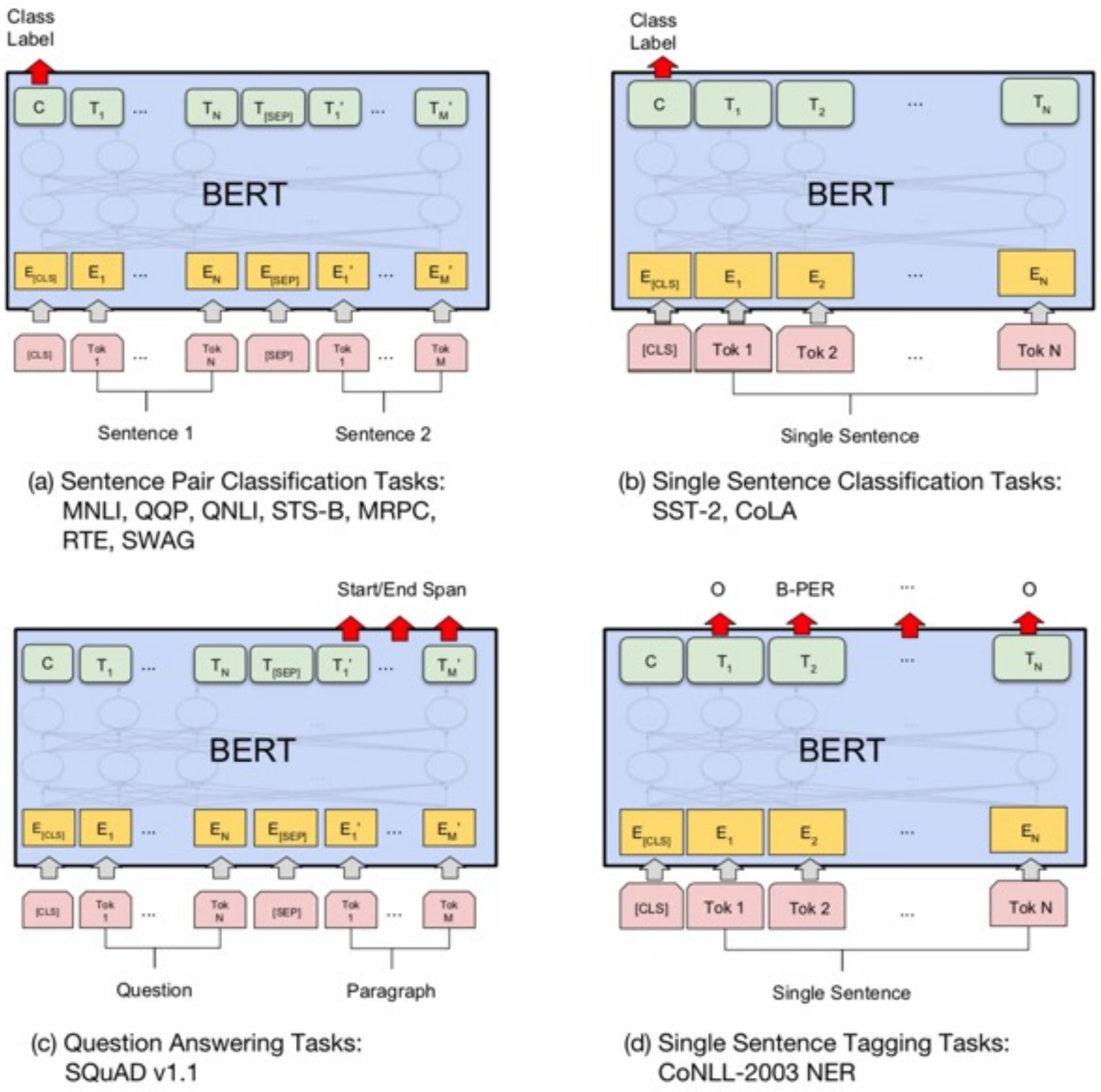
BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT



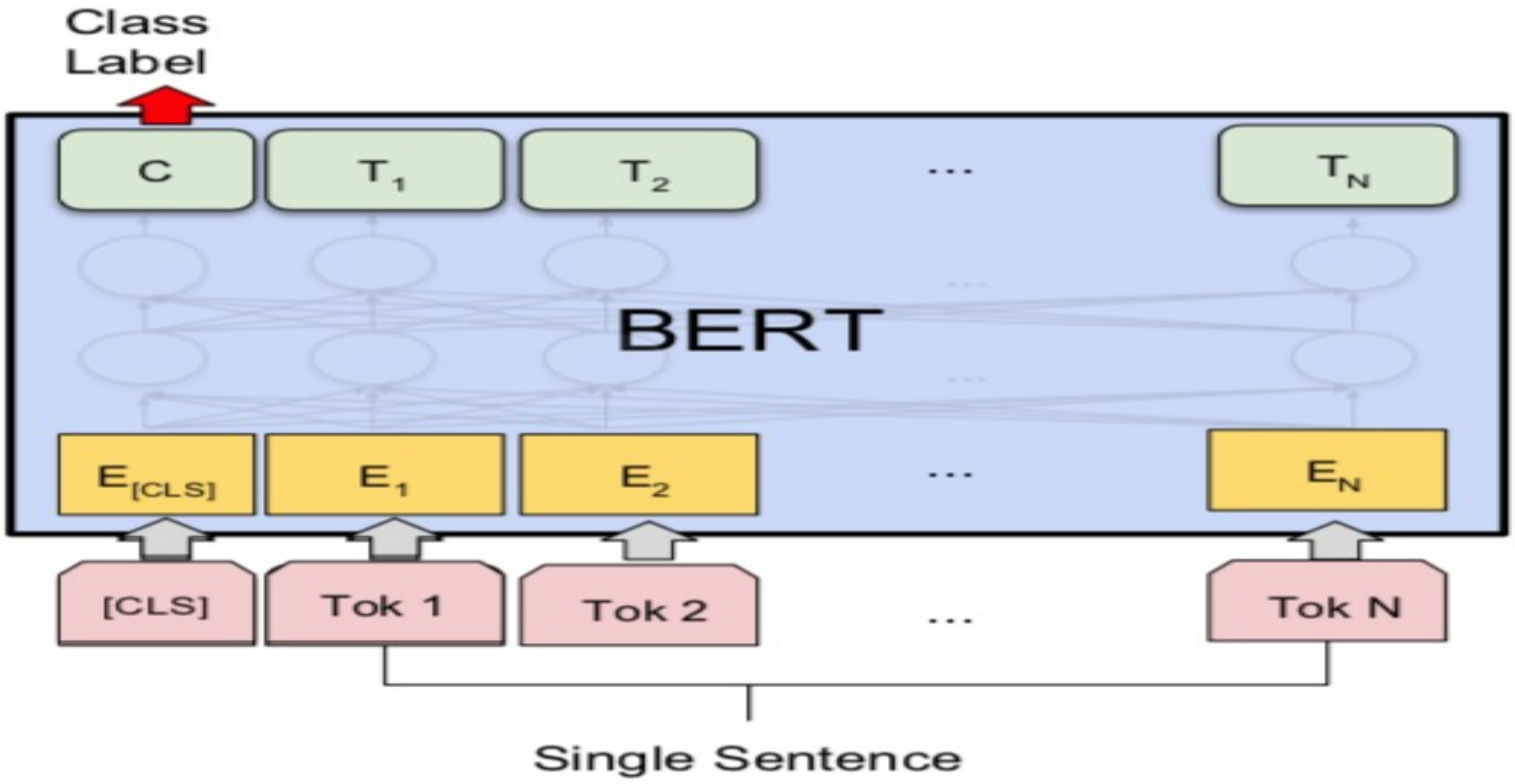
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Different Tasks



Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018).
"Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

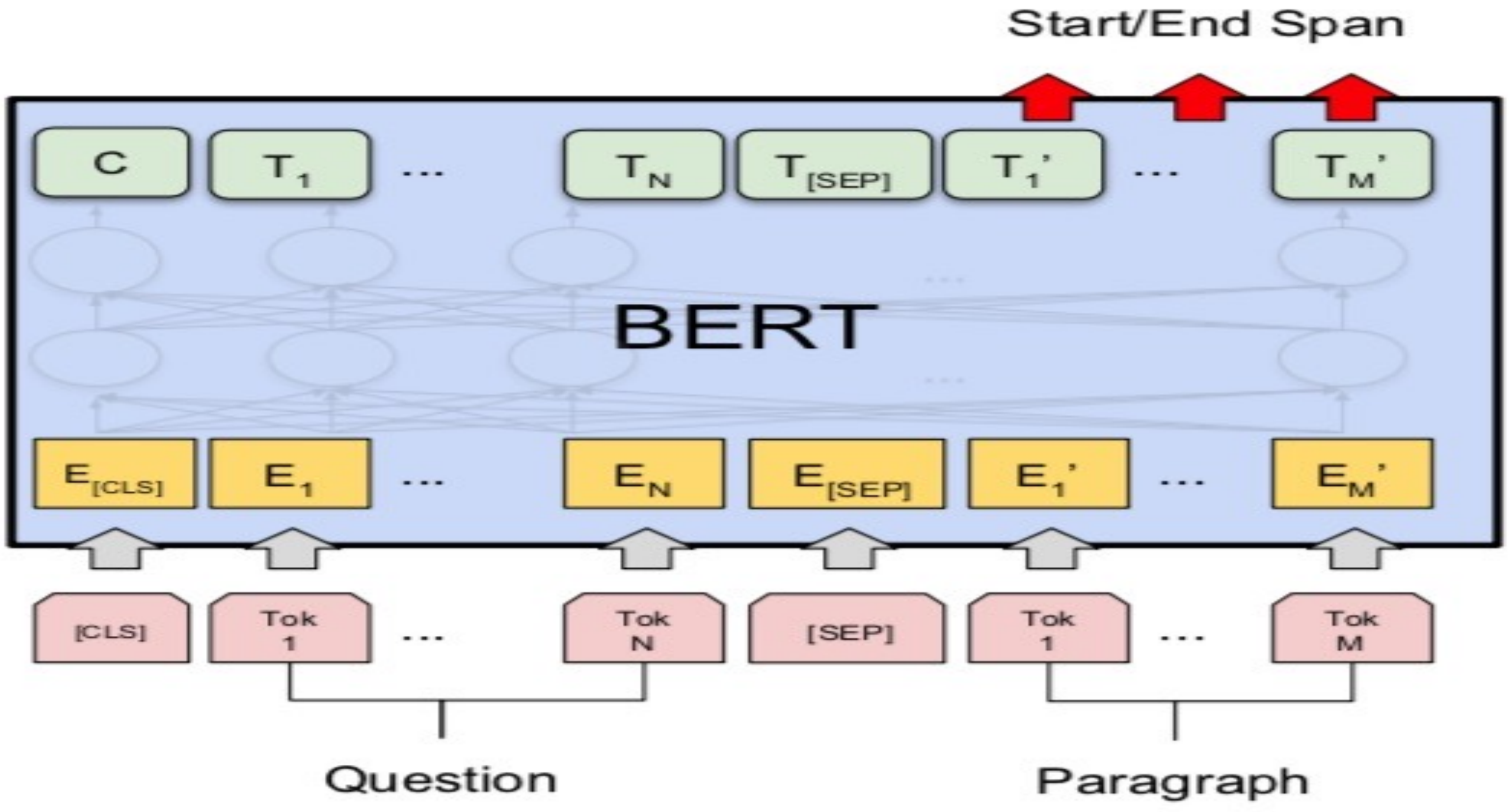
Sentiment Analysis: Single Sentence Classification



(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805

Fine-tuning BERT on Question Answering (QA)

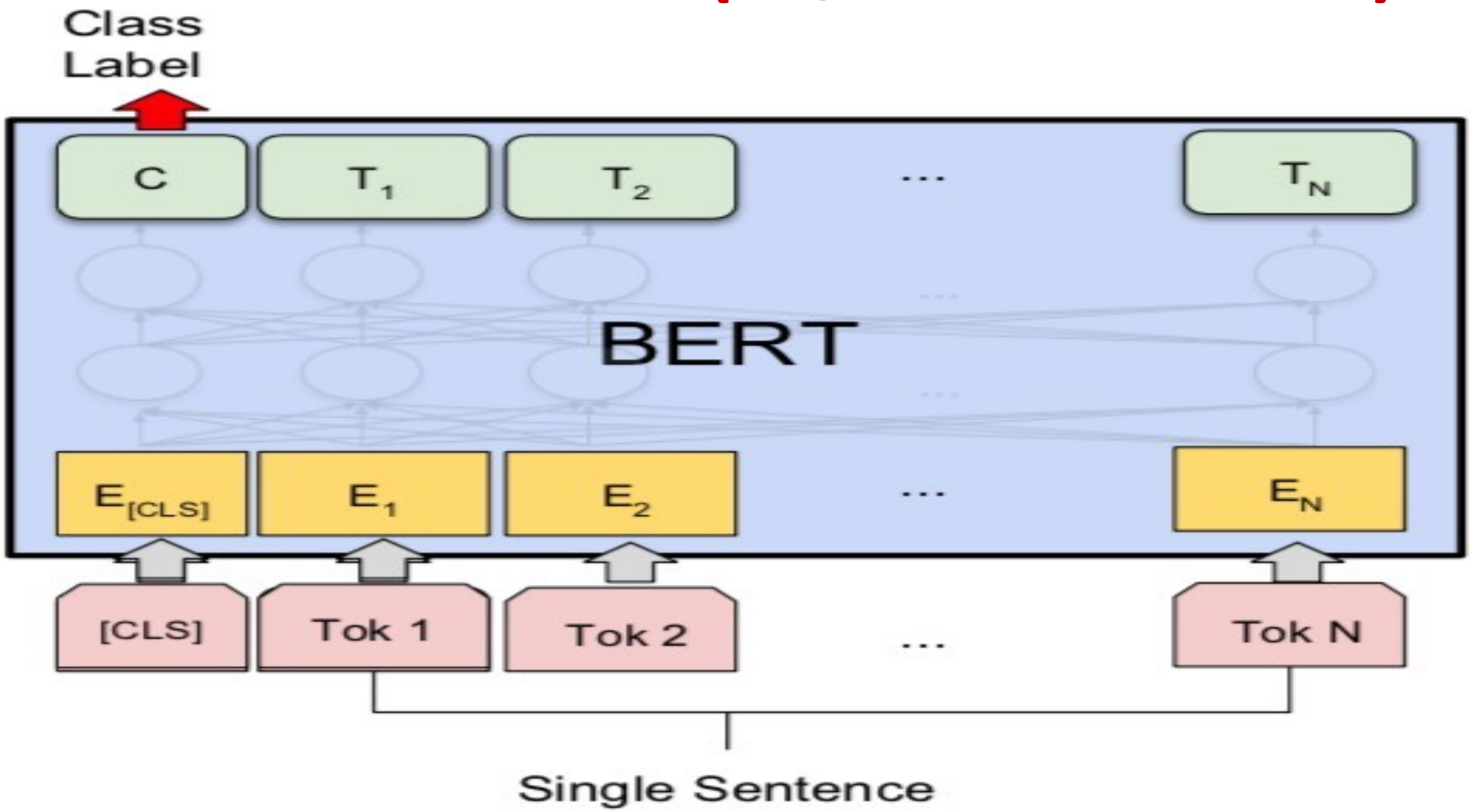


(c) Question Answering Tasks: SQuAD v1.1

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Intent Detection (ID; Classification)

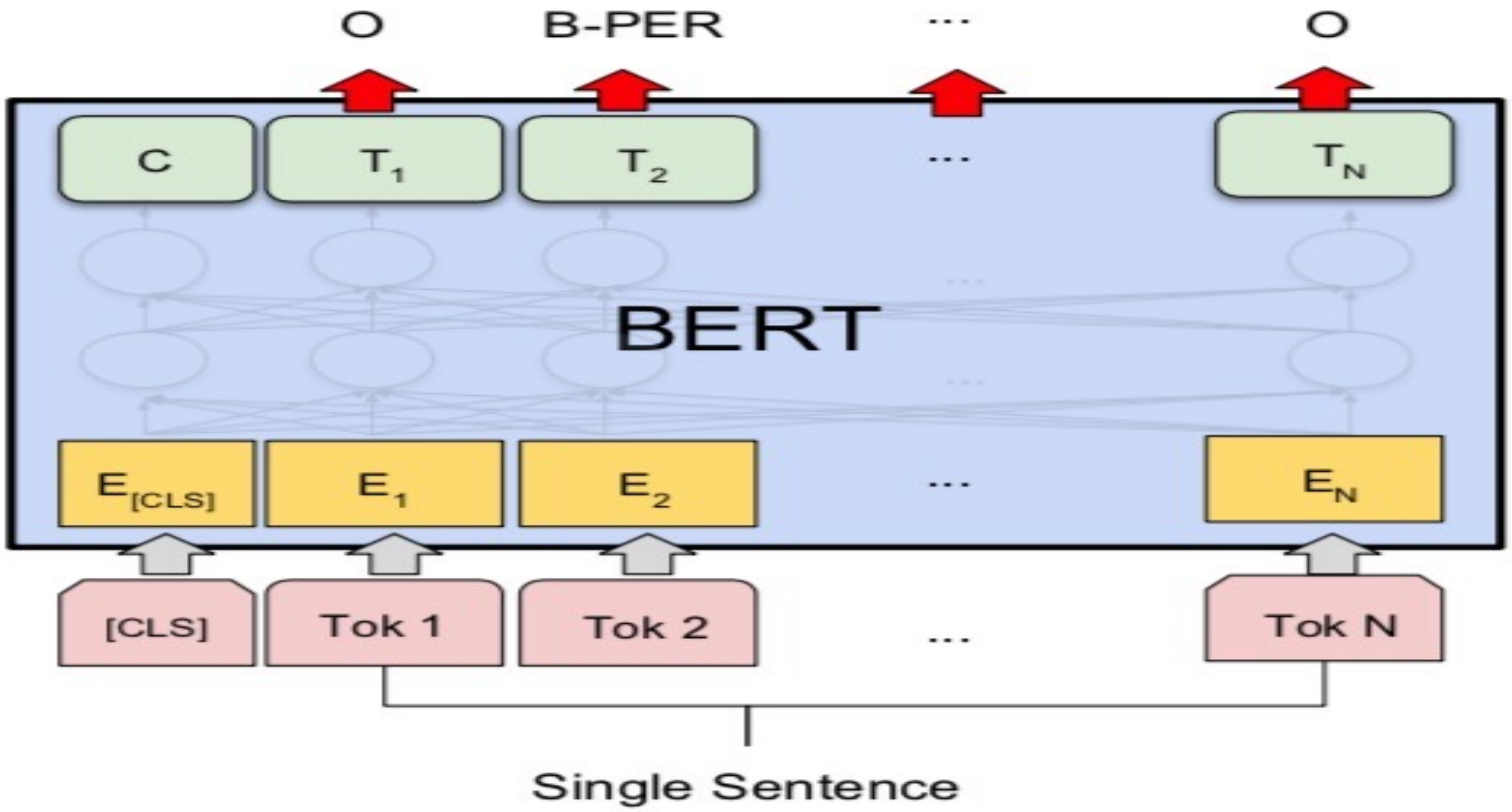


(b) Single Sentence Classification Tasks: SST-2, CoLA

Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

Fine-tuning BERT on Dialogue

Slot Filling (SF)

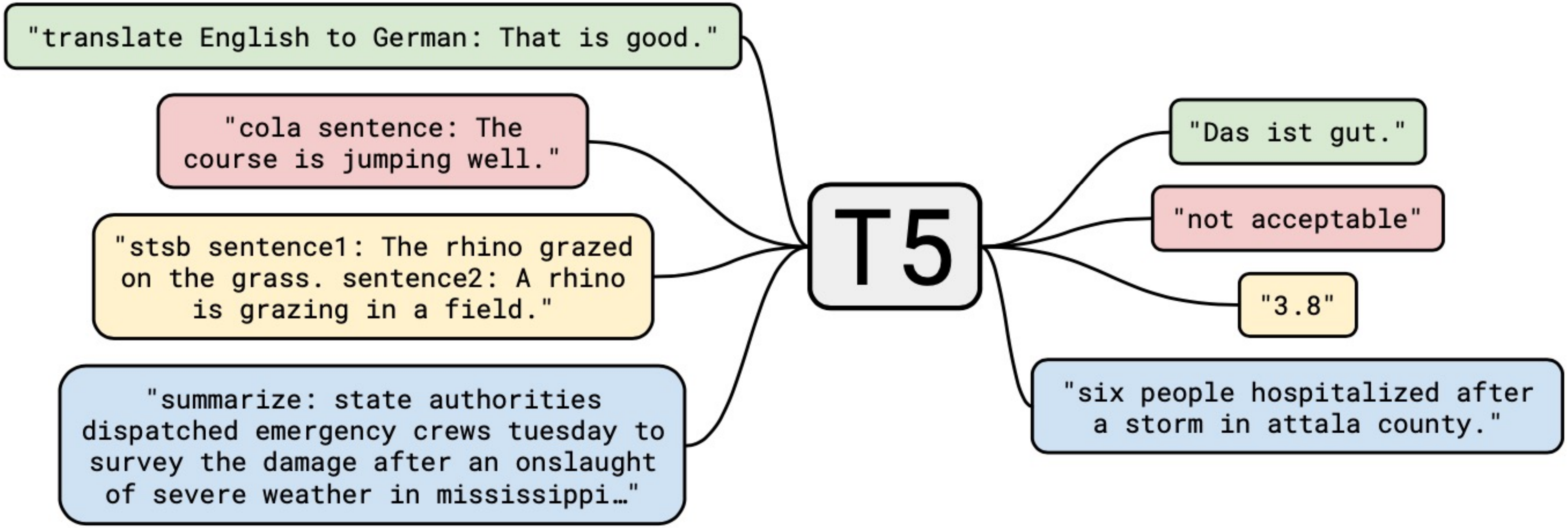


(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

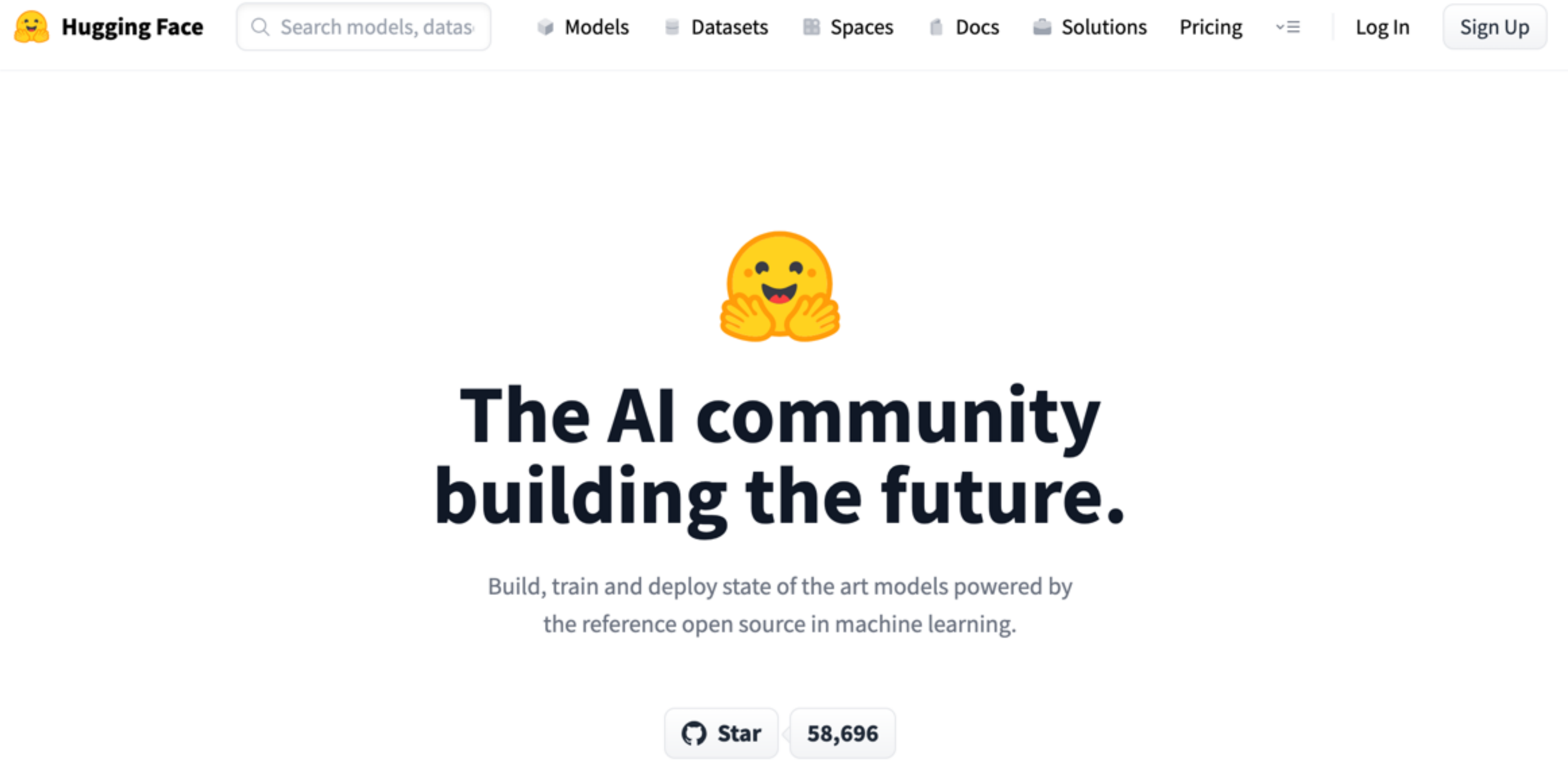
Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

T5

Text-to-Text Transfer Transformer

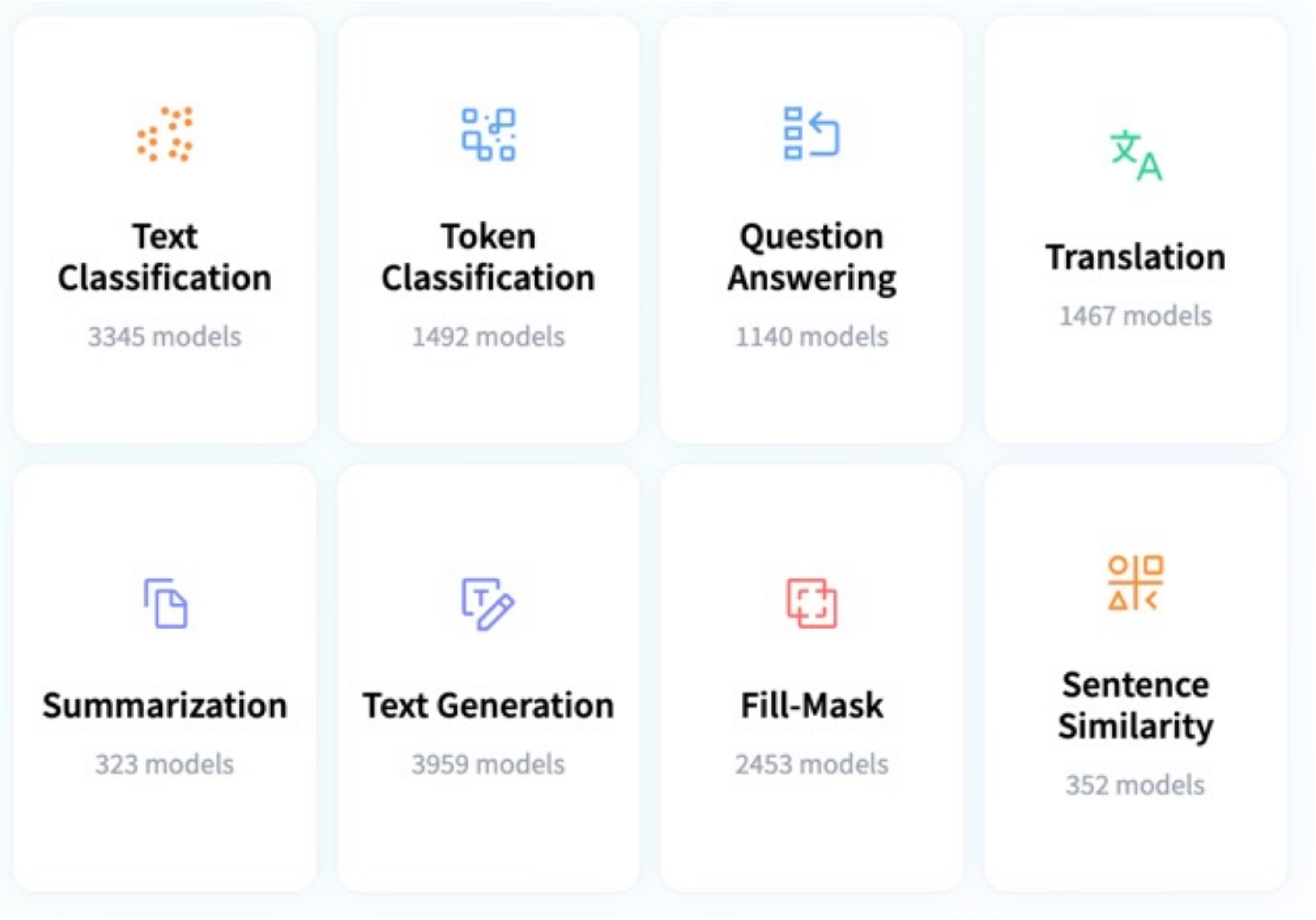


Hugging Face



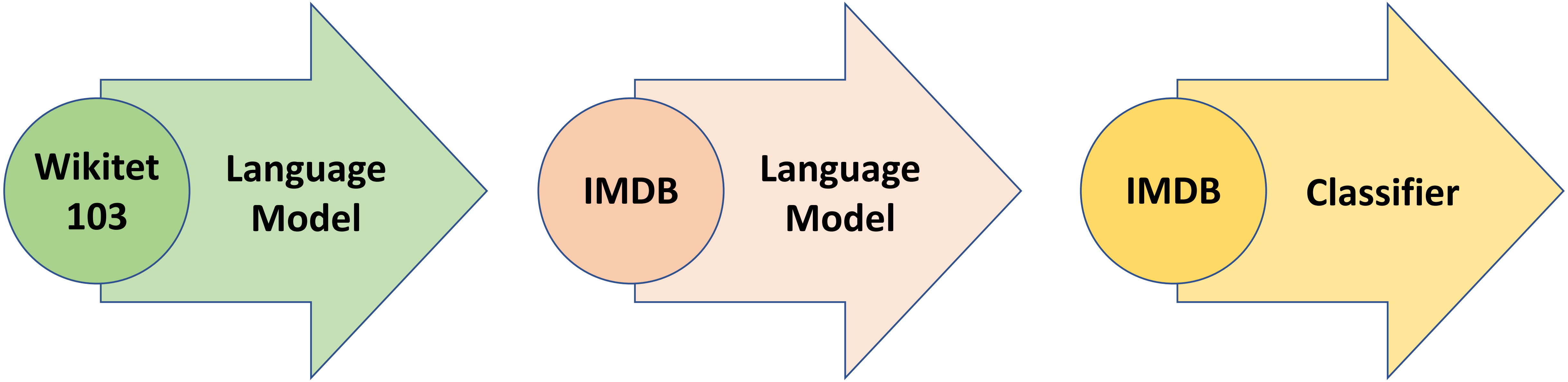
https://huggingface.co/

Hugging Face Tasks
Natural Language Processing



https://huggingface.co/tasks

ULMFiT: 3 Steps
Transfer Learning in NLP



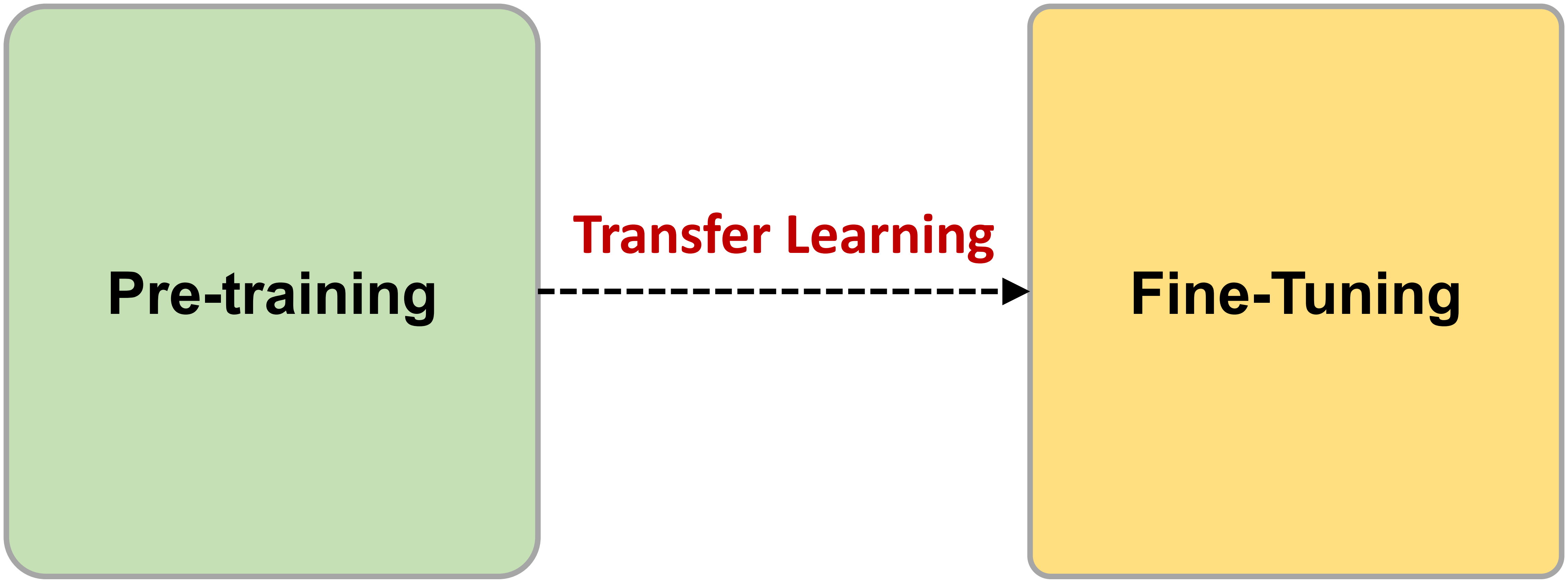
1. Pretraining

2. Domain adaptation

3. Fine-tuning

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

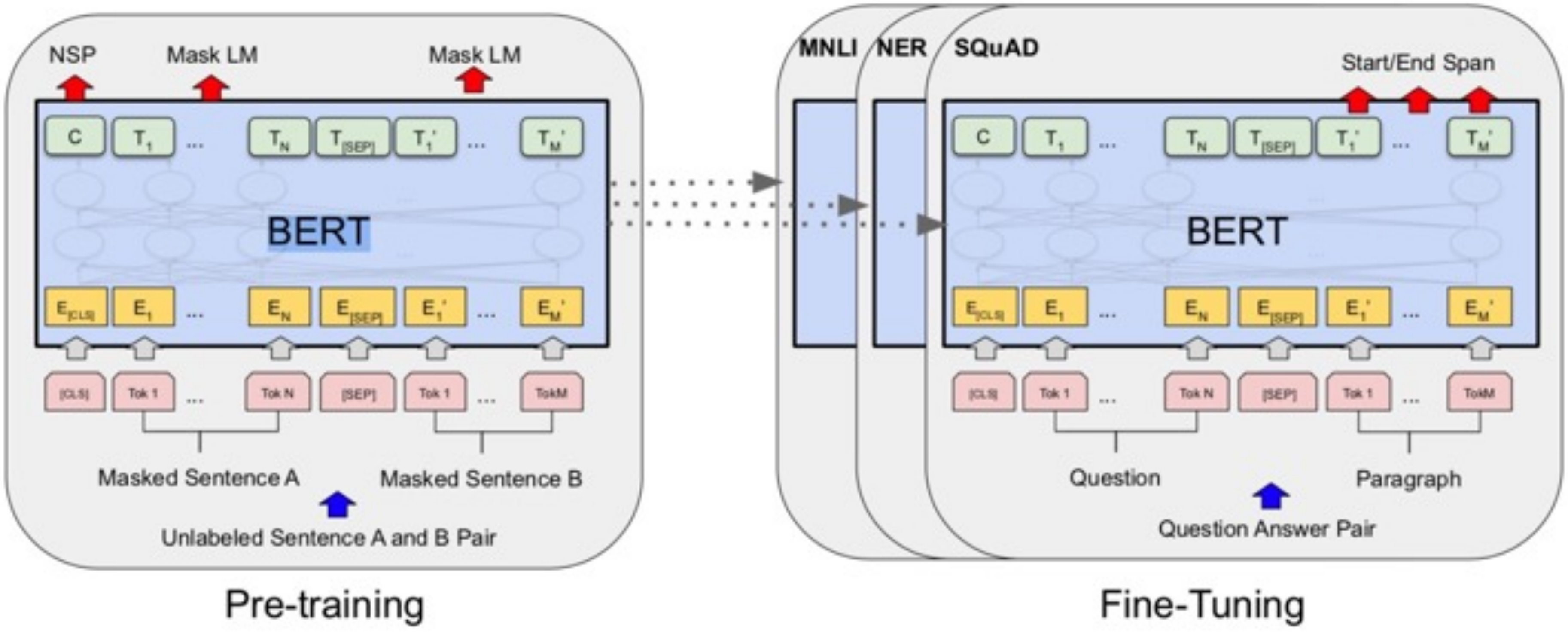
Transfer Learning



BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT



Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

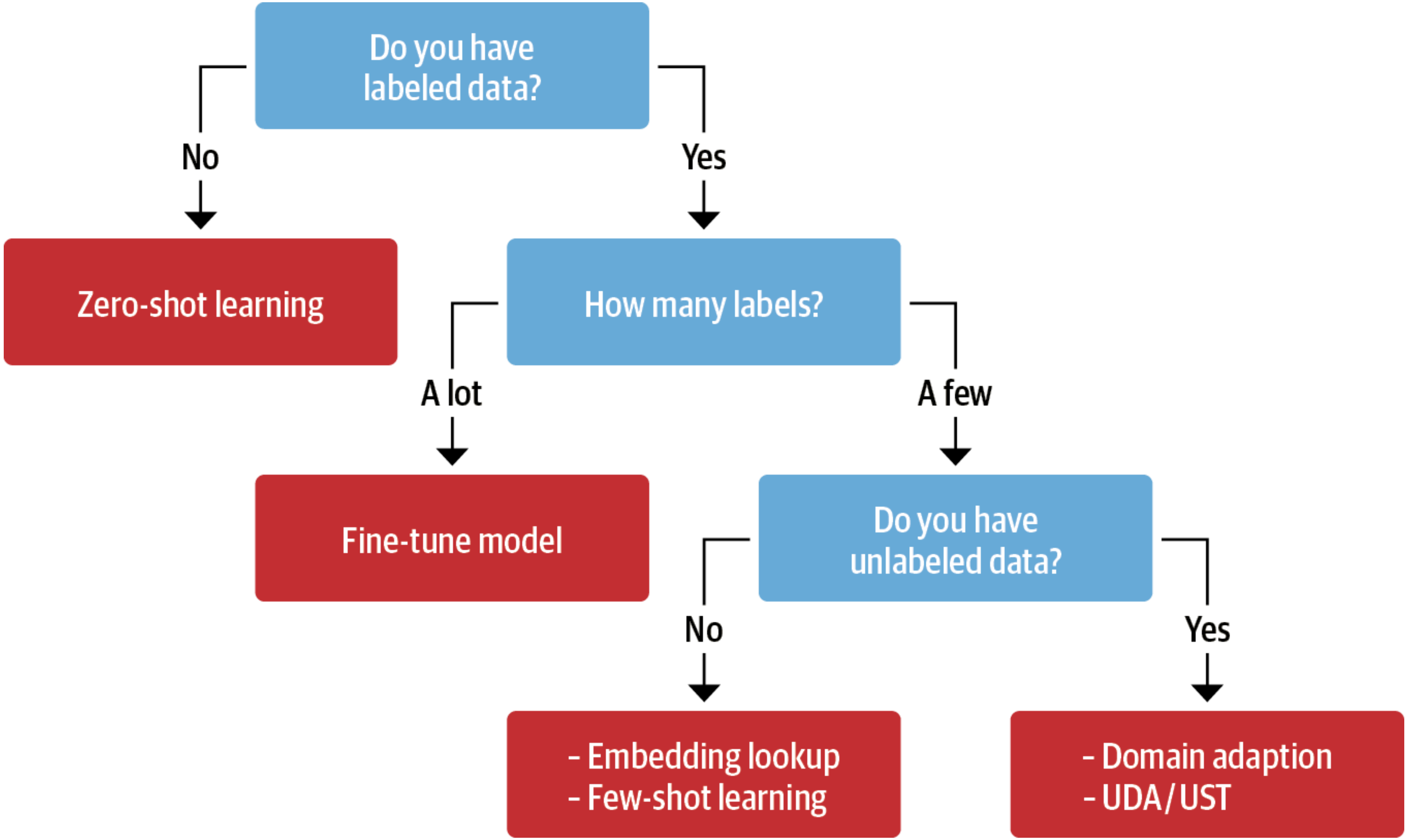
A typical pipeline for training transformer models

with the Datasets, Tokenizers, and Transformers libraries

Datasets	Tokenizers	Transformers	Datasets
Load and process datasets	Tokenize input texts	Load models, train and infer	Load metrics evaluate models

Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
https://github.com/nlp-with-transformers/notebooks

Transfer Learning, Fine-tuning, Few-shot learning



Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

Few-Shot Learning (FSL)
Typical Scenarios

- **Acting as a test bed for learning like human**
- **Learning for rare cases**
- **Reducing data gathering effort and computational cost**

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

- **Few-Shot Learning (FSL) is a sub-area in machine learning.**
- **Machine Learning Definition**
  - **A computer program is said to learn from experience E with respect to some classes of task T and performance measure P if its performance can improve with E on T measured by P.**
    - Example: Image classification task (T), a machine learning program can improve its classification accuracy (P) through E obtained by training on a large number of labeled images (e.g., the ImageNet data set).

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Machine Learning

task T	experience E	performance P
image classification [73]	large-scale labeled images for each class	classification accuracy
the ancient game of Go [120]	a database containing around 30 million recorded moves of human experts and self-play records	winning rate

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

- **Few-shot Learning (FSL) is a type of machine learning problems (specified by E, T, and P), where E contains only a limited number of examples with supervised information for the target T.**
  - **Existing FSL problems are mainly supervised learning problems.**
  - **Few-shot classification learns classifiers given only a few labeled examples of each class.**
    - image classification
    - sentiment classification from short text
    - object recognition

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

- **Few-shot classification learns a classifier h, which predicts label y_i for each input x_i.**
- **Usually, one considers the *N-way-K-shot* classification, in which D_{train} contains $I = KN$ examples from N classes each with K examples**

Few-Shot Learning (FSL)

- **Few-Shot Learning (FSL)**
  - $K = 10 \sim 100$ **examples**
- **One-Shot Learning (1SL)**
  - $K = 1$ **example**
- **Zero-Shot Learning (0SL)(ZSL)**
  - $K = 0$

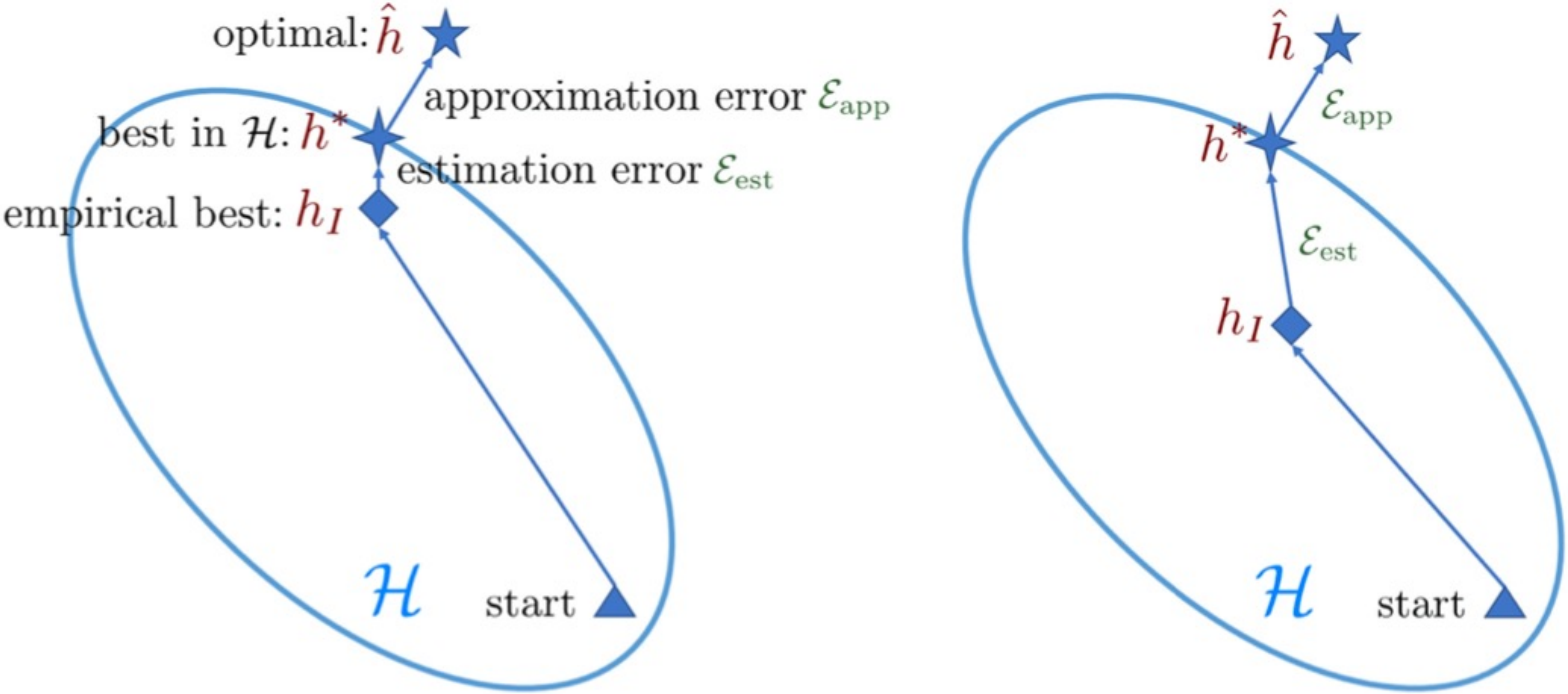
Few-Shot Learning (FSL)

task T	experience E		performance P
	supervised information	prior knowledge	
character generation [76]	a few examples of new character	pre-learned knowledge of parts and relations	pass rate of visual Turing test
drug toxicity discovery [4]	new molecule's limited assay	similar molecules' assays	classification accuracy
image classification [70]	a few labeled images for each class of the target T	raw images of other classes, or pre-trained models	classification accuracy

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Comparison of learning with sufficient and few training samples

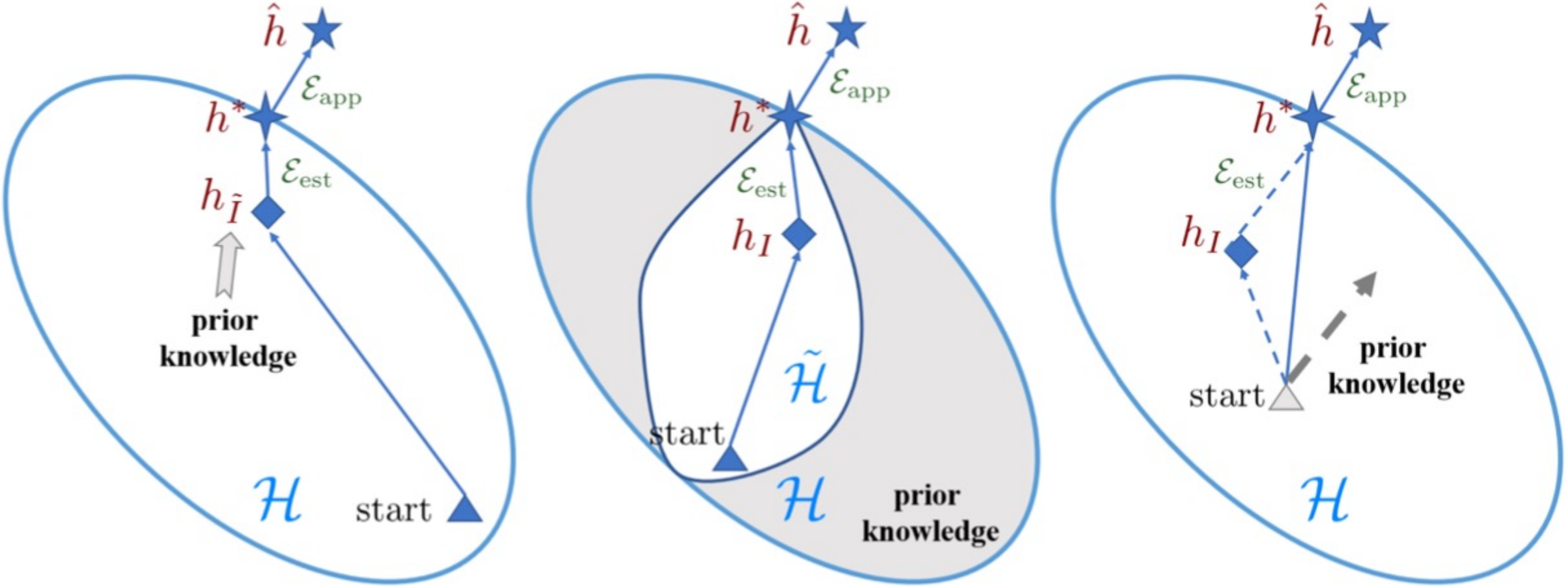


(a) Large I (b) Small I

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Different perspectives on how FSL methods solve the few-shot problem

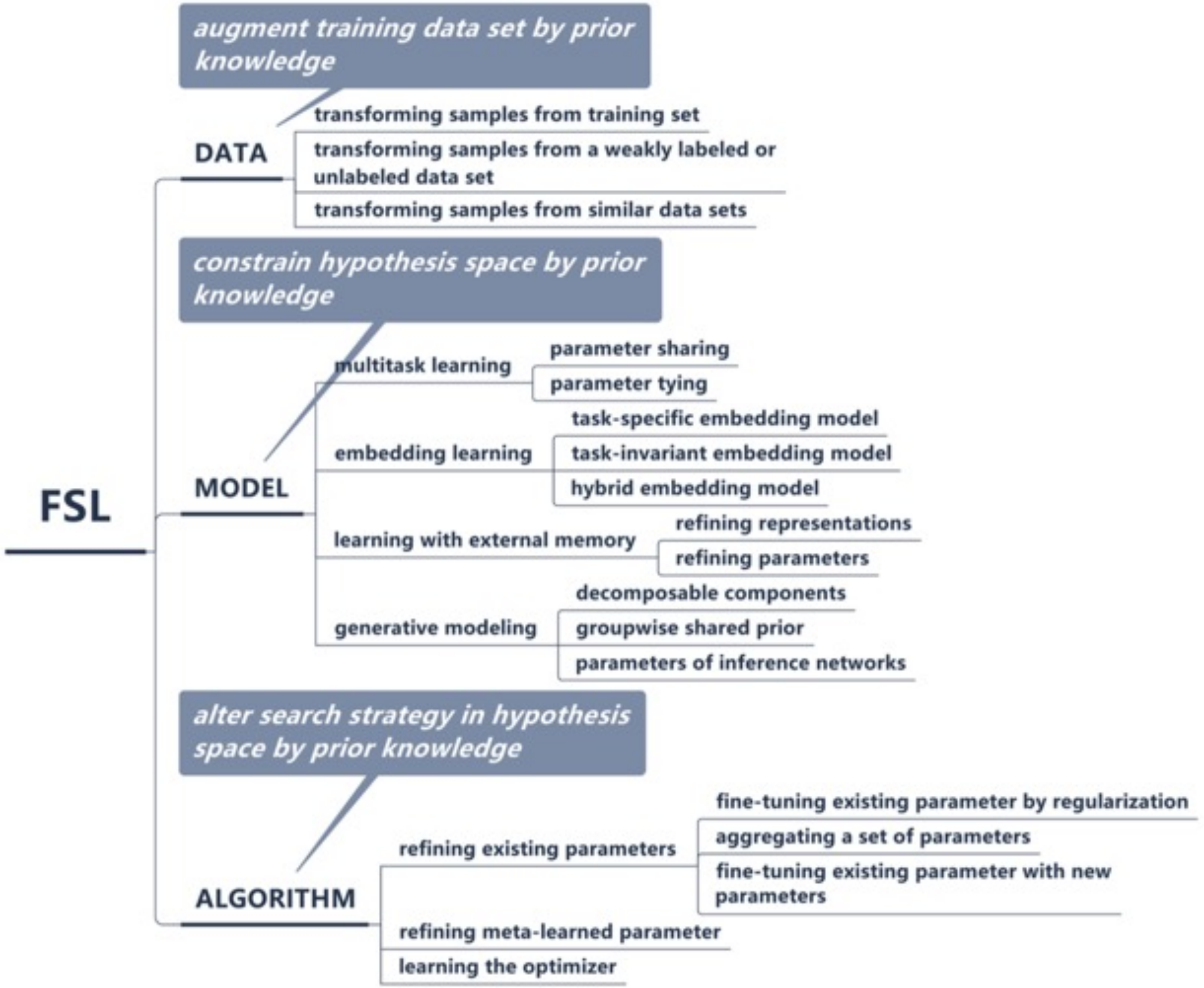


(a) Data. (b) Model. (c) Algorithm.

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

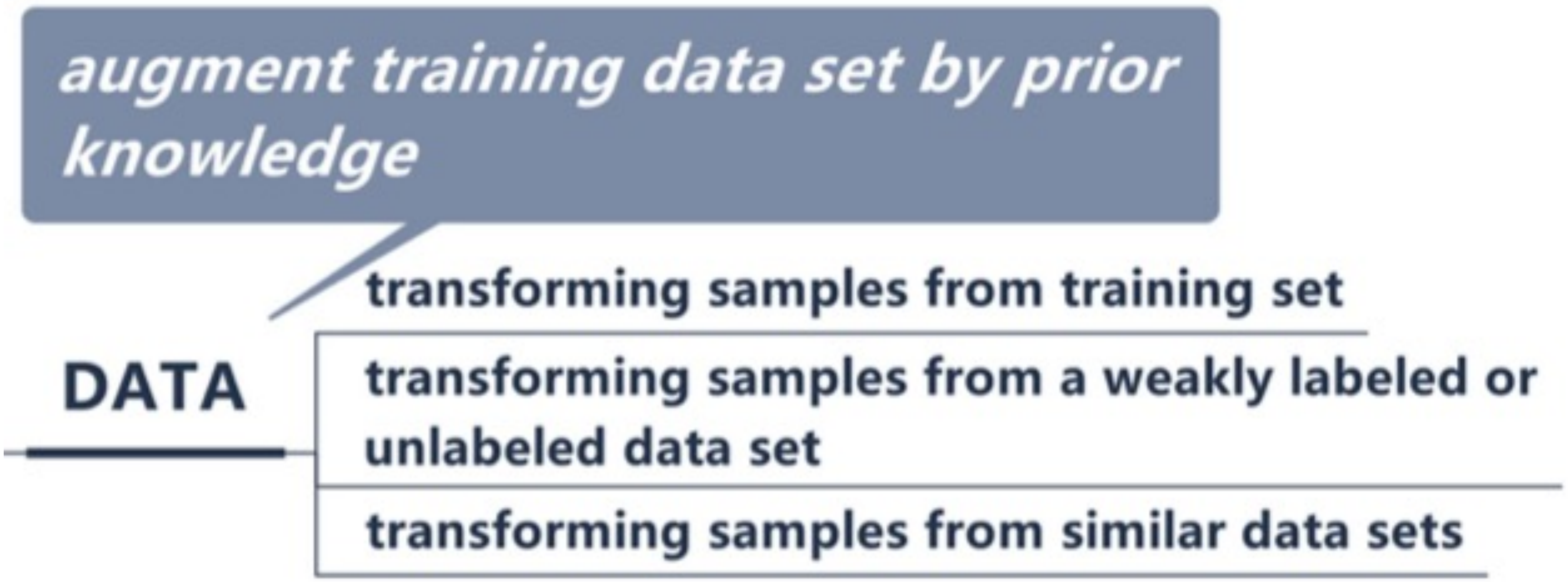
Few-Shot Learning (FSL)

A taxonomy of FSL methods



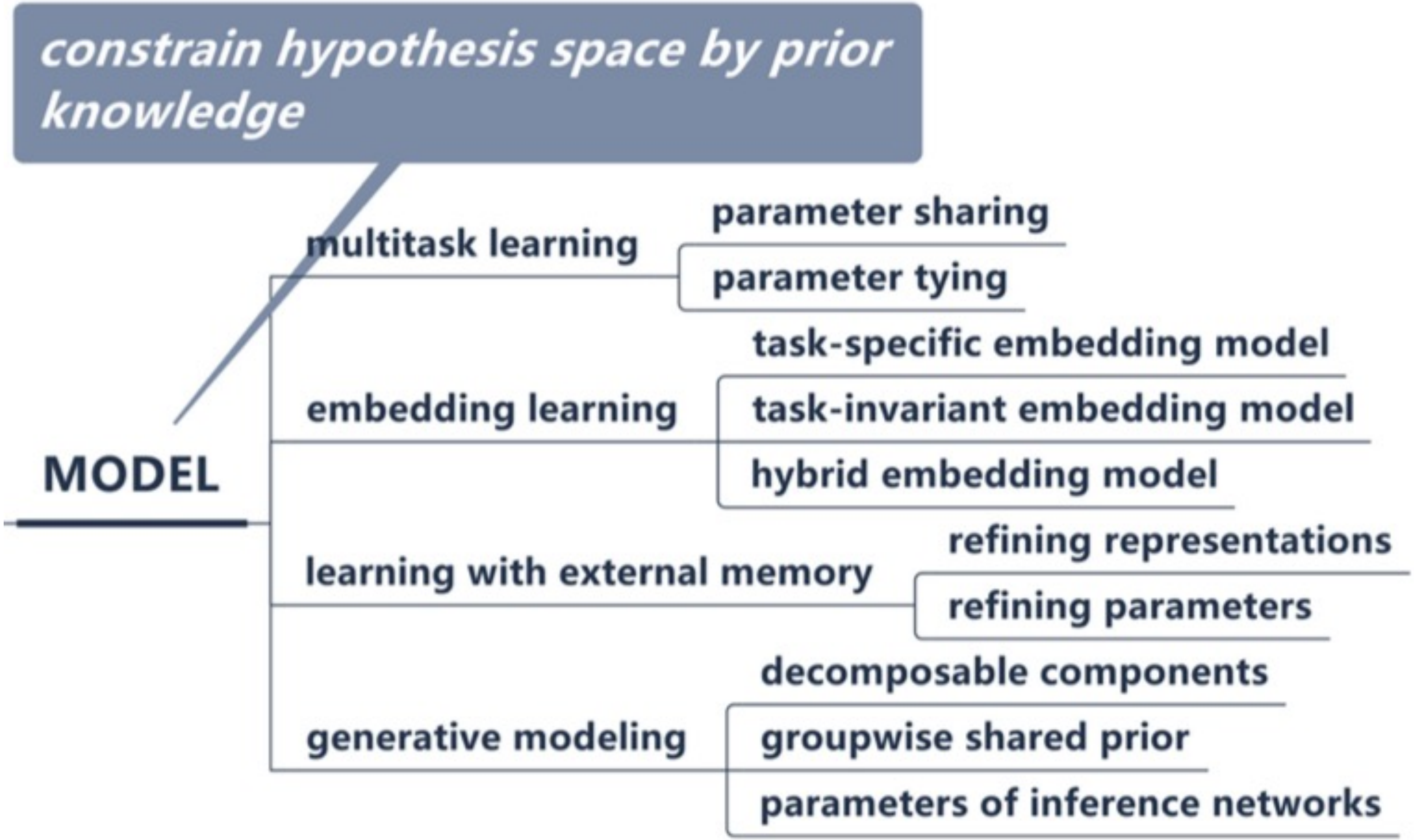
Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)



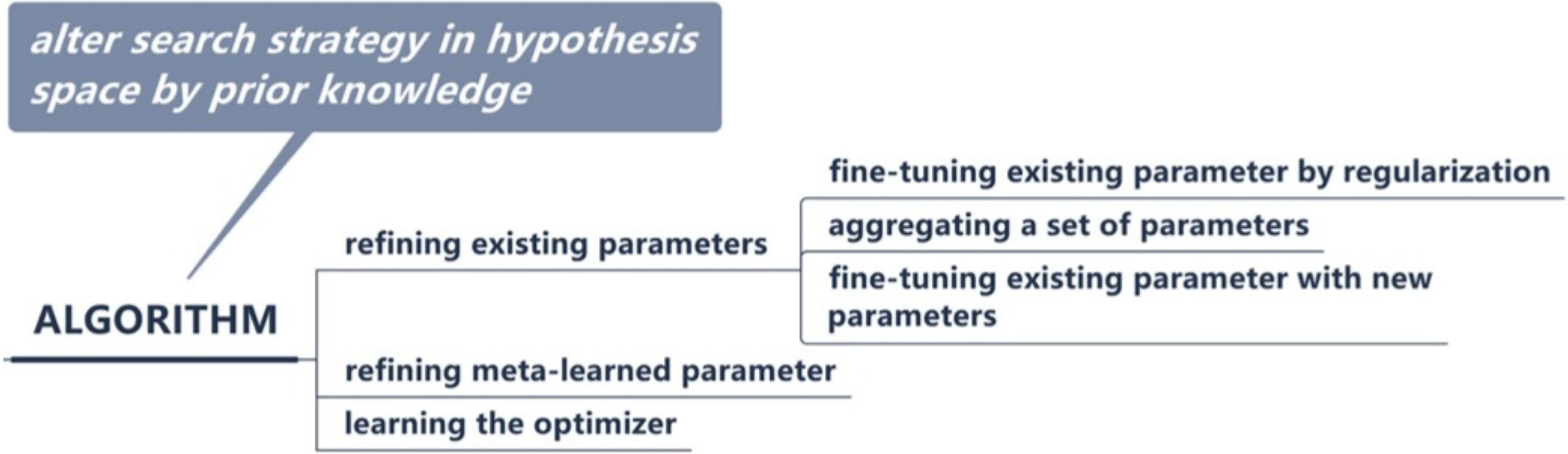
Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

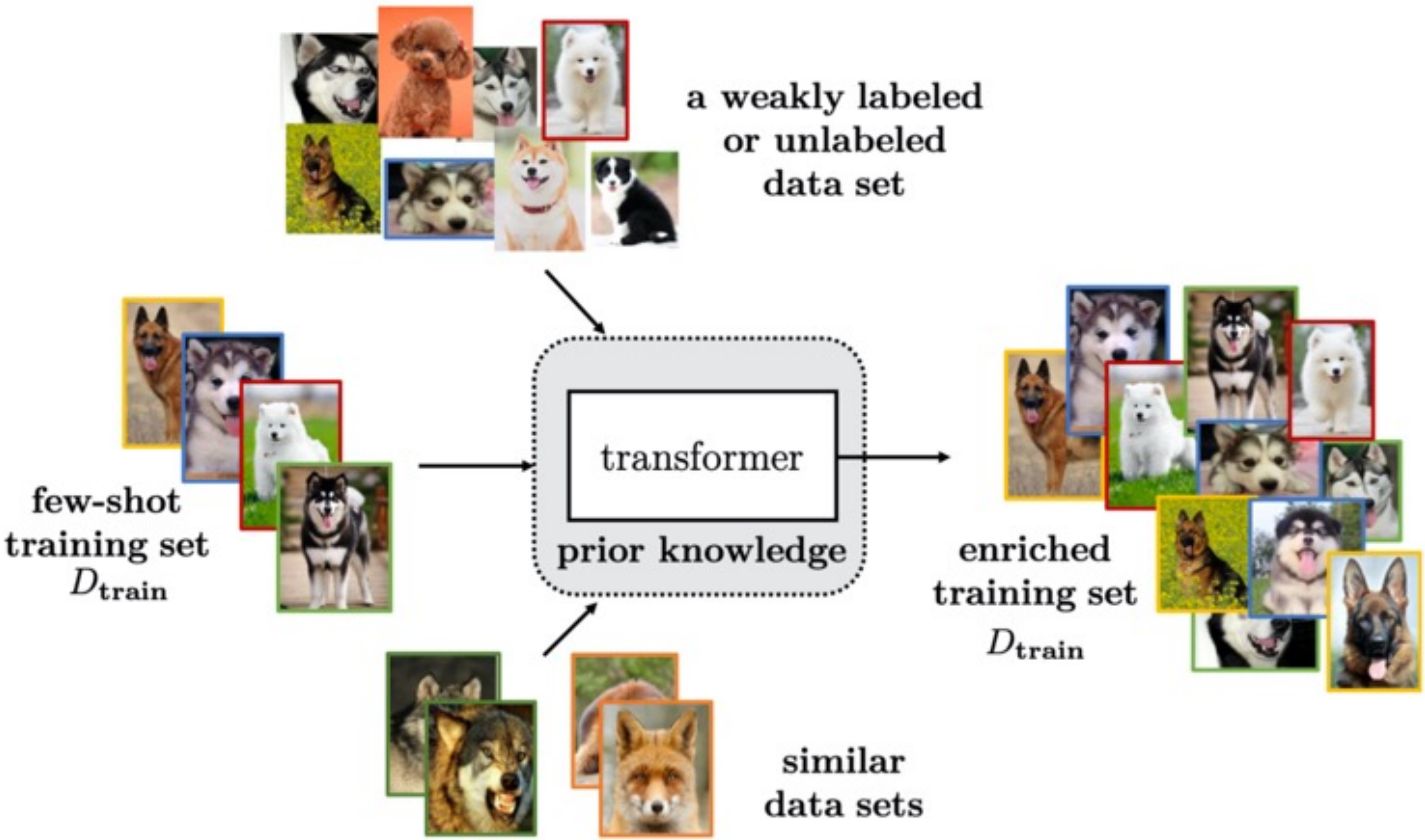
Few-Shot Learning (FSL)



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Solving the FSL problem by data augmentation



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Characteristics for FSL Methods Focusing on the Data Perspective

category	input (x, y)	transformer t	output $(\tilde{x}, \tilde{y})$
transforming samples from D_{train}	original (x_i, y_i)	learned transformation function on x_i	$(t(x_i), y_i)$
transforming samples from a weakly labeled or unlabeled data set	weakly labeled or unlabeled $(\bar{x}, -)$	a predictor trained from D_{train}	$(\bar{x}, t(\bar{x}))$
transforming samples from similar data sets	samples $\{(\hat{x}_j, \hat{y}_j)\}$ from similar data sets	an aggregator to combine $\{(\hat{x}_j, \hat{y}_j)\}$	$(t(\{\hat{x}_j\}), t(\{\hat{y}_j\}))$

The transformer $t(\cdot)$ takes input (x, y) and returns synthesized sample $(\tilde{x}, \tilde{y})$ to augment the few-shot D_{train}.

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

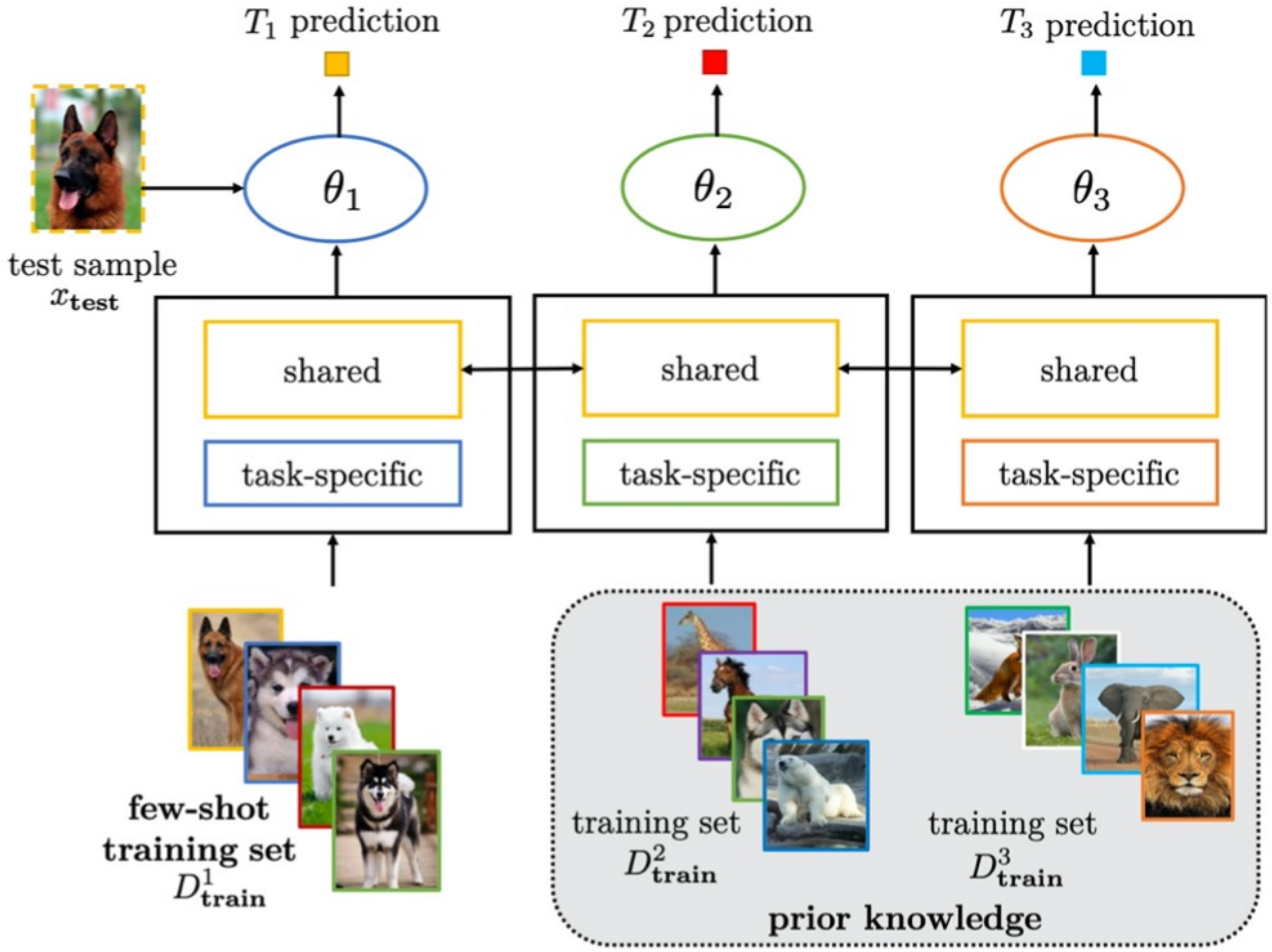
Characteristics for FSL Methods Focusing on the Model Perspective

strategy	prior knowledge	how to constrain $\mathcal{H}$
multitask learning	other T's with their data sets D's	share/tie parameter
embedding learning	embedding learned from/together with other T's	project samples to a smaller embedding space in which similar and dissimilar samples can be easily discriminated
learning with external memory	embedding learned from other T's to interact with memory	refine samples using key-value pairs stored in memory
generative modeling	prior model learned from other T's	restrict the form of distribution

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

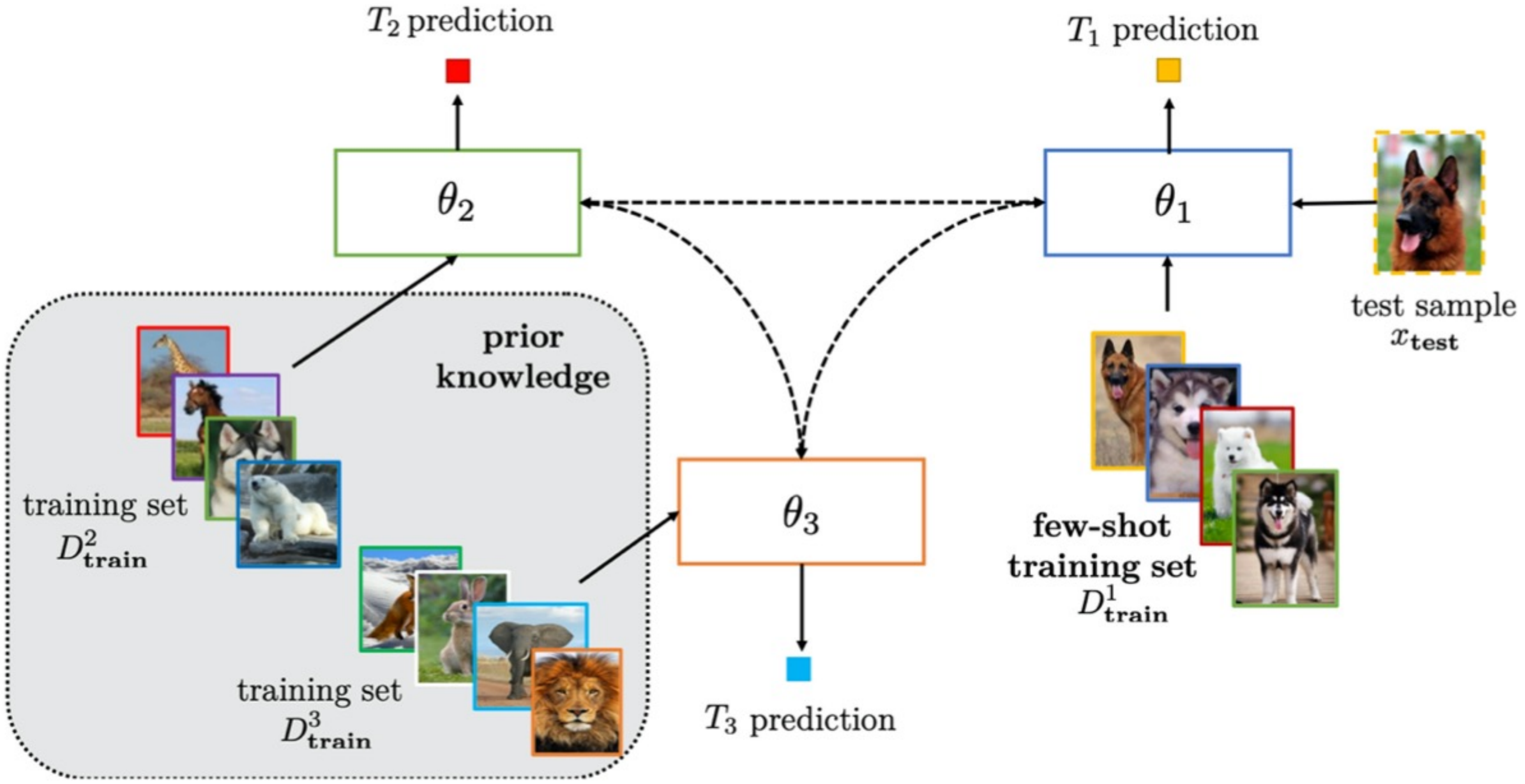
Few-Shot Learning (FSL)

Solving the FSL problem by multitask learning with parameter sharing



Few-Shot Learning (FSL)

Solving the FSL problem by multitask learning with parameter tying



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

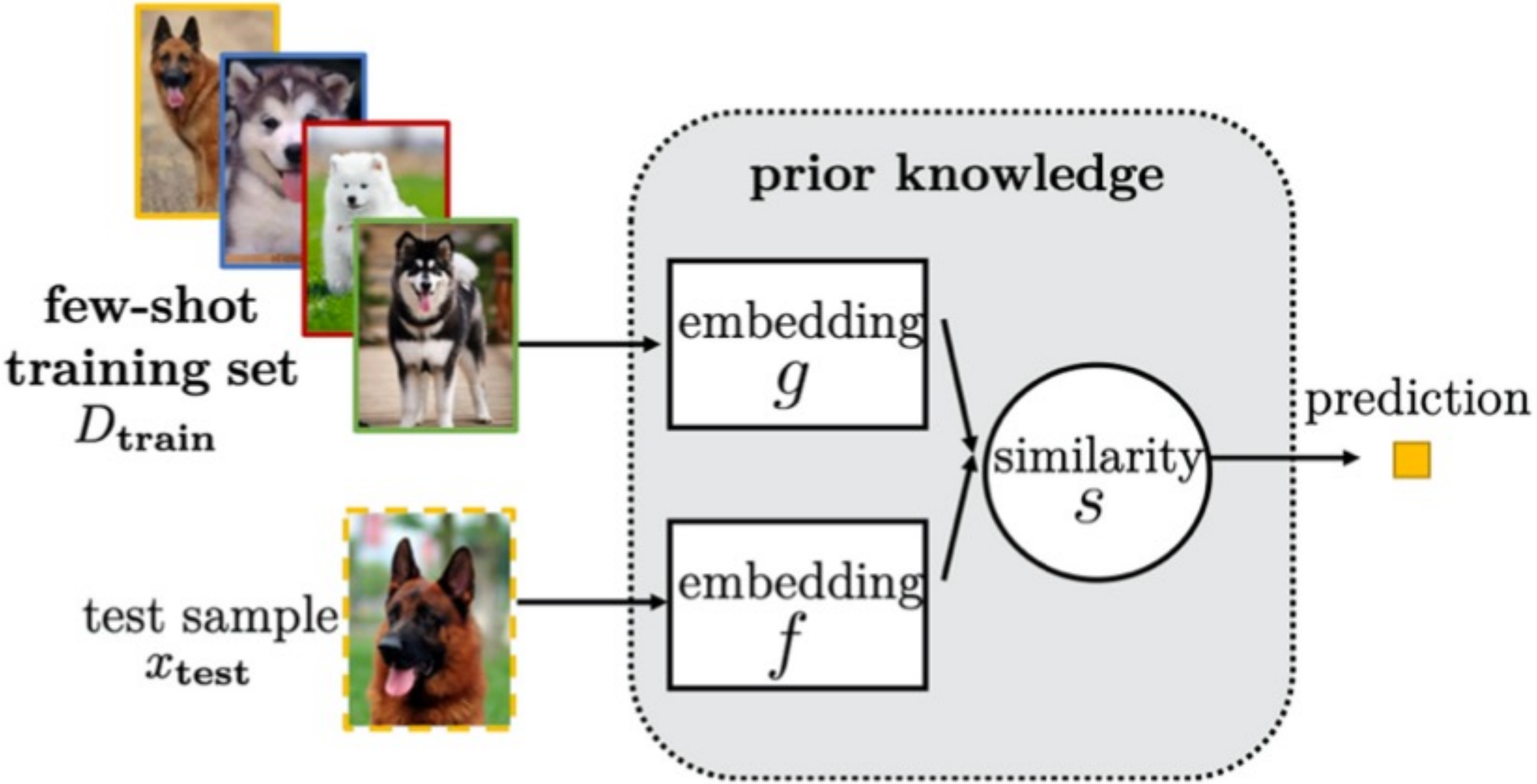
Characteristics of Embedding Learning Methods

category	method	embedding function f for x_{test}	embedding function g for D_{train}	similarity measure s
task-specific	mAP-DLM/SSVM[130]	CNN	the same as f	cosine similarity
task-invariant	class relevance pseudo-metric [36]	kernel	the same as f	squared ℓ_2 distance
	convolutional siamese net [70]	CNN	the same as f	weighted ℓ_1 distance
	Micro-Set[127]	logistic projection	the same as f	ℓ_2 distance
	Matching Nets [138]	CNN, LSTM	CNN, biLSTM	cosine similarity
	resLSTM [4]	GNN, LSTM	GNN, LSTM	cosine similarity
	Active MN [8]	CNN	biLSTM	cosine similarity
	SSMN [24]	CNN	another CNN	learned distance
	ProtoNet [121]	CNN	the same as f	squared ℓ_2 distance
	semi-supervised ProtoNet[108]	CNN	the same as f	squared ℓ_2 distance
	PMN [141]	CNN, LSTM	CNN, biLSTM	cosine similarity
	ARC [119]	LSTM, biLSTM	the same as f	-
	Relation Net [126]	CNN	the same as f	-
	GNN [115]	CNN, GNN	the same as f	learned distance
	TPN [84]	CNN	the same as f	Gaussian similarity
	SNAIL [91]	CNN	the same as f	-
hybrid	Learnet [14]	adaptive CNN	CNN	weighted ℓ_1 distance
	DCCN [162]	adaptive CNN	CNN	-
	R2-D2 [13]	adaptive CNN	CNN	-
	TADAM [100]	adaptive CNN	the same as f	squared ℓ_2 distance

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

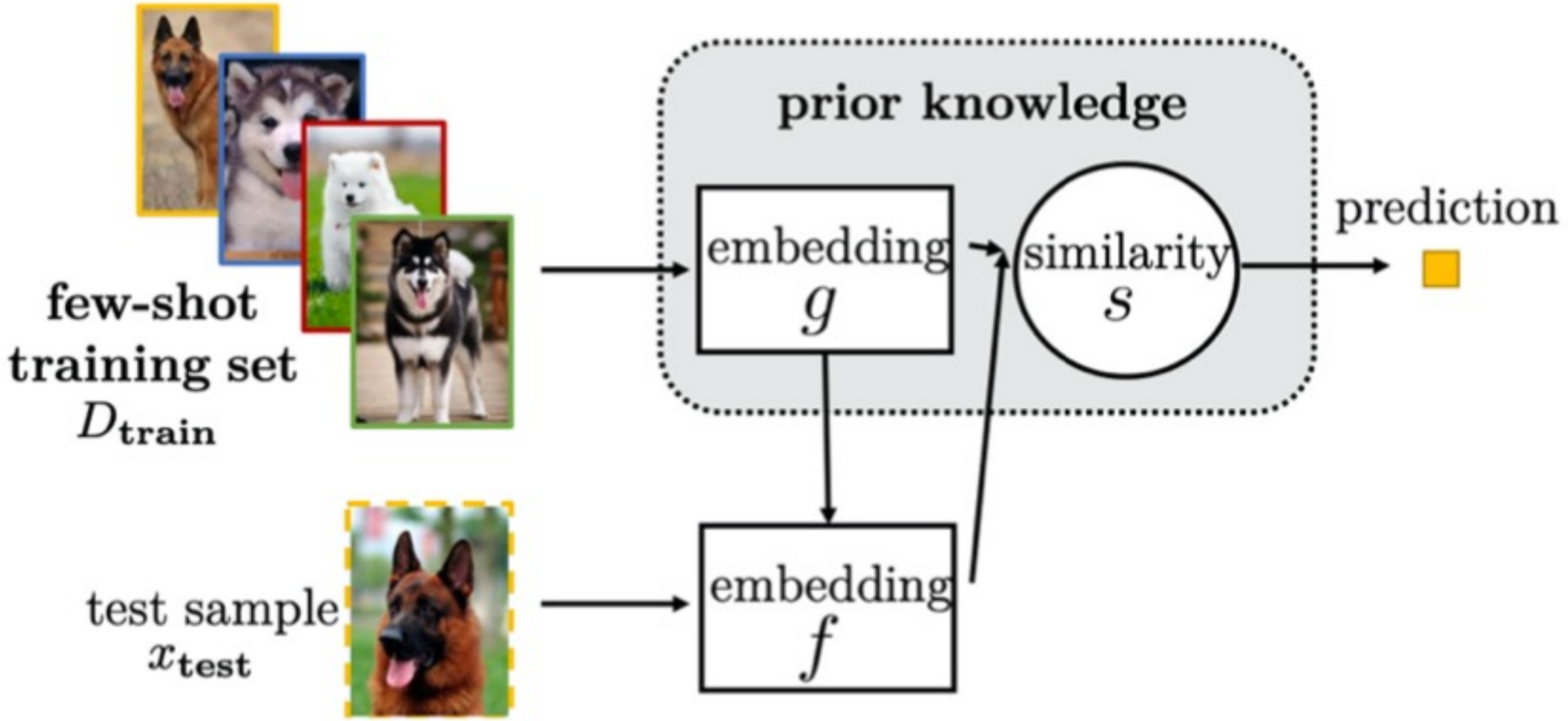
Solving the FSL problem by task-invariant embedding model



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

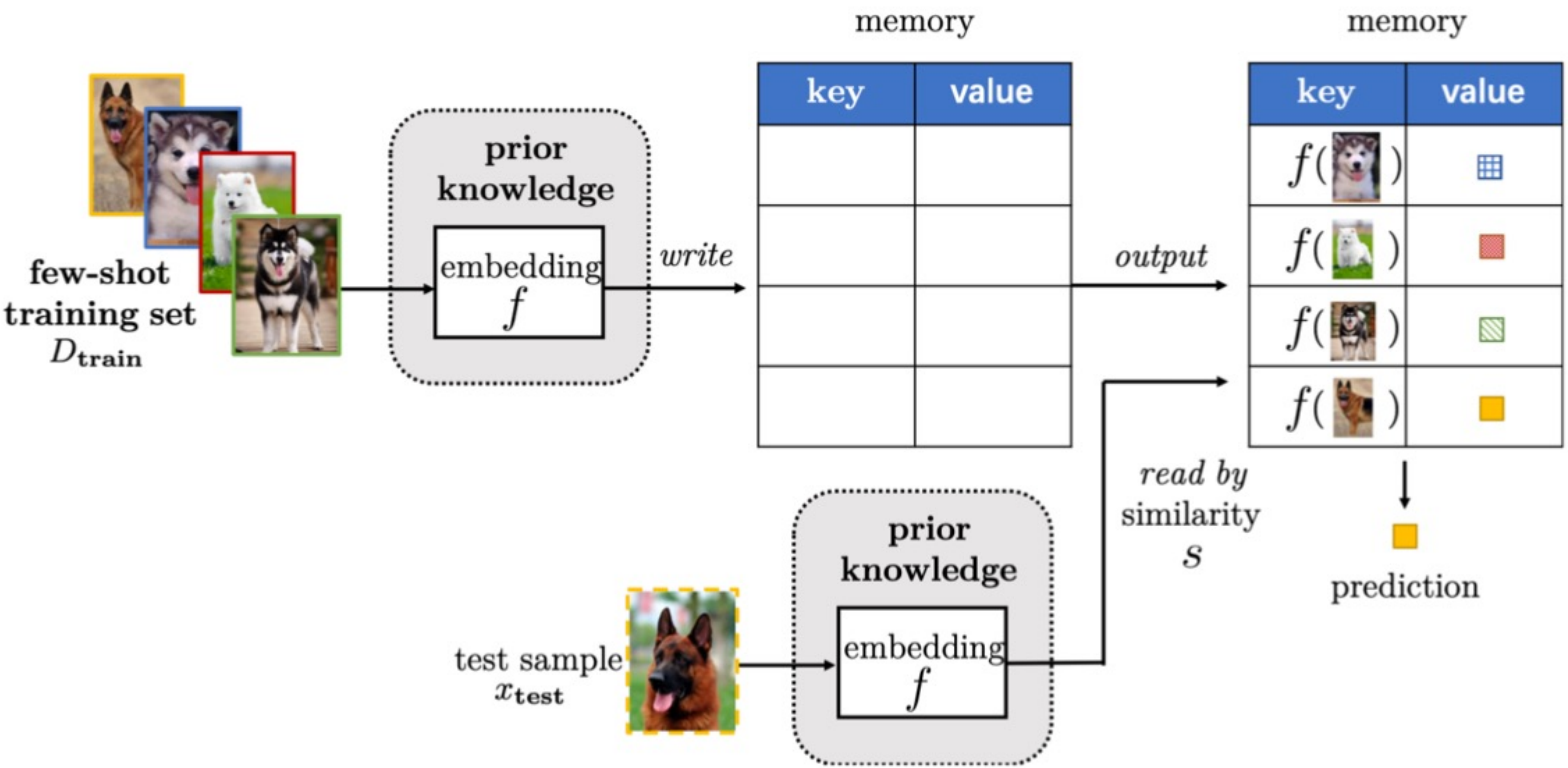
Solving the FSL problem by hybrid embedding model



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Solving the FSL problem by learning with external memory



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Characteristics of FSL Methods
Based on Learning with External Memory

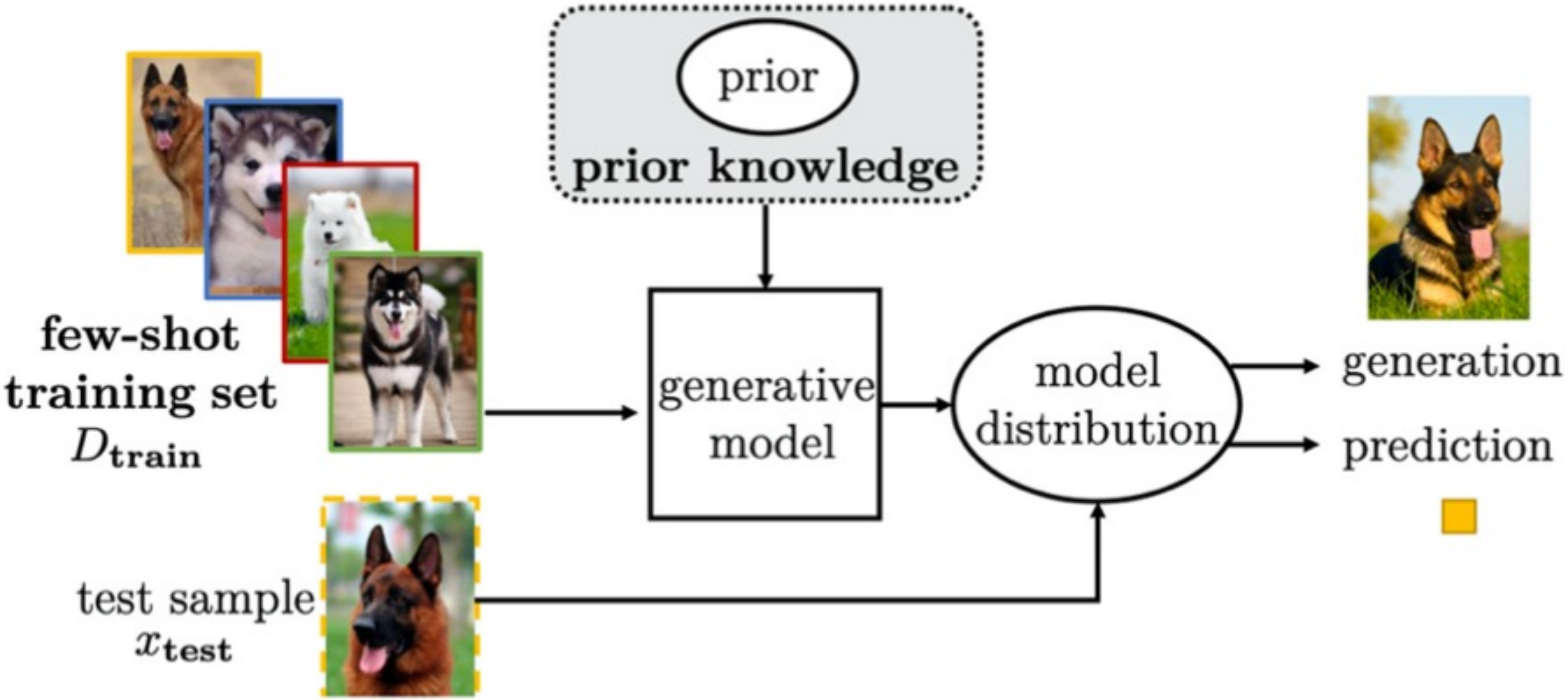
category	method	memory M key M_{key}	memory M value M_{value}	similarity s
refining representations	MANN [114]	$f(x_i, y_{i-1})$	$f(x_i, y_{i-1})$	cosine similarity
	APL [104]	$f(x_i)$	y_i	squared ℓ_2 distance
	abstraction memory [149]	$f(x_i)$	word embedding of y_i	dot product
	CMN [164]	$f(x_i)$	y_i, age	dot product
	life-long memory [65]	$f(x_i)$	y_i, age	cosine similarity
	Mem2Vec [125]	$f(x_i)$	word embedding of y_i, age	dot product
refining parameters	MetaNet [96]	$f(x_i)$	fast weight	cosine similarity
	CSNs [97]	$f(x_i)$	fast weight	cosine similarity
	MN-Net [22]	$f(x_i)$	y_i	dot product

Here, f is an embedding function usually pre-trained by CNN or LSTM.

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

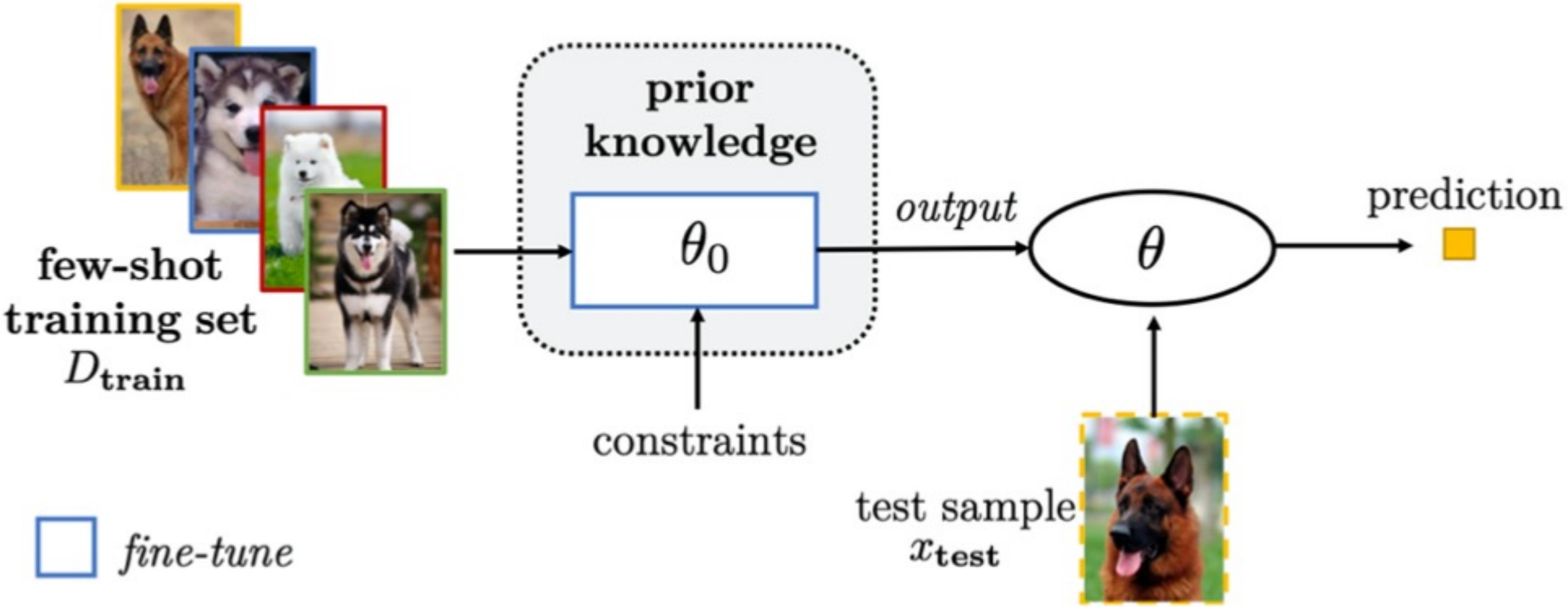
Solving the FSL problem by generative modeling



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Solving the FSL problem by fine-tuning existing parameter θ_0 by regularization



Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Characteristics for FSL Methods

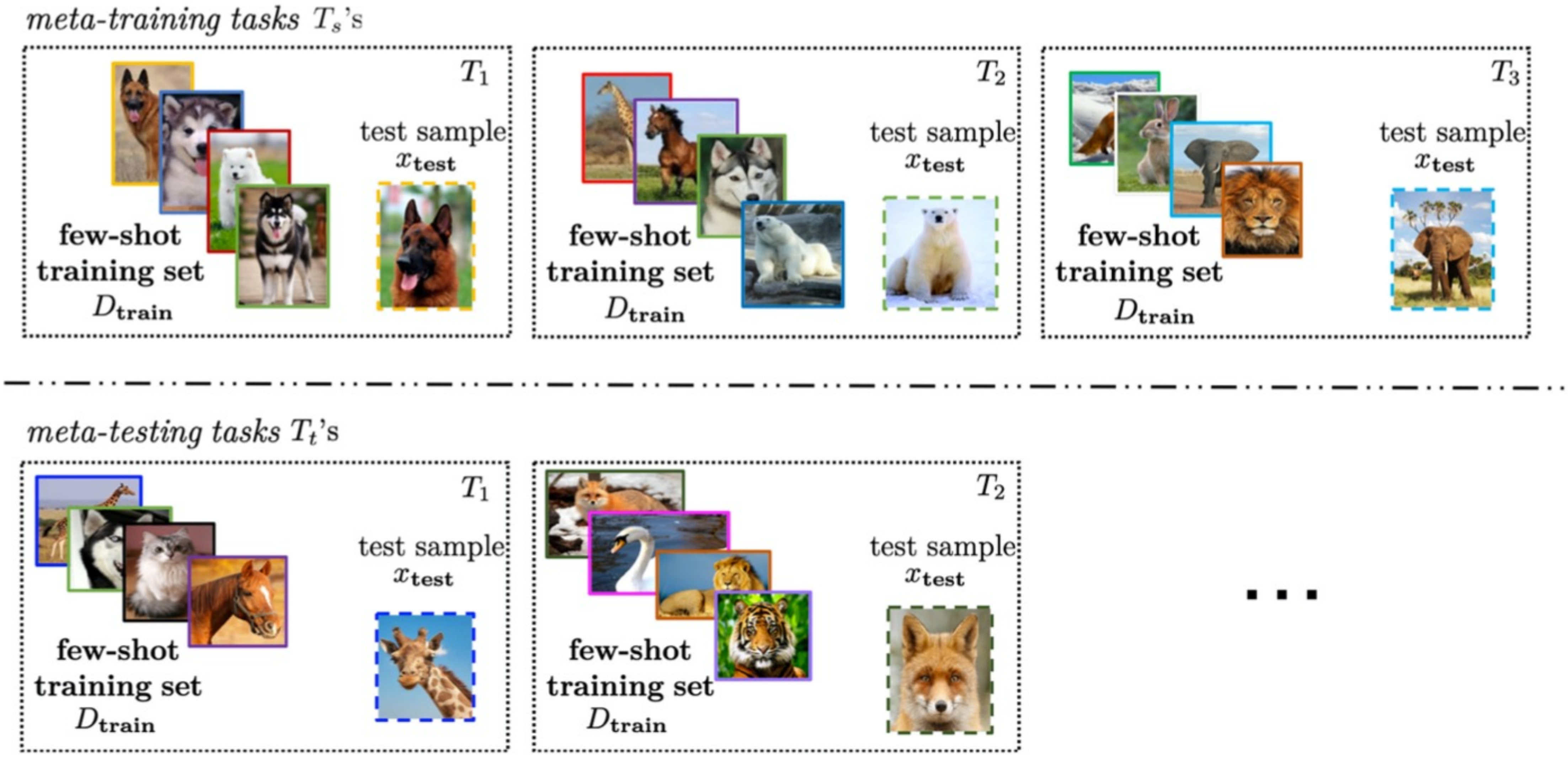
Focusing on the Algorithm Perspective

strategy	prior knowledge	how to search θ of the h^* in $\mathcal{H}$
refining existing parameters	learned θ_0	refine θ_0 by D_{train}
refining meta-learned parameters	meta-learner	refine θ_0 by D_{train}
learning the optimizer	meta-learner	use search steps provided by the meta-learner

Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Solving the FSL problem by meta-learning

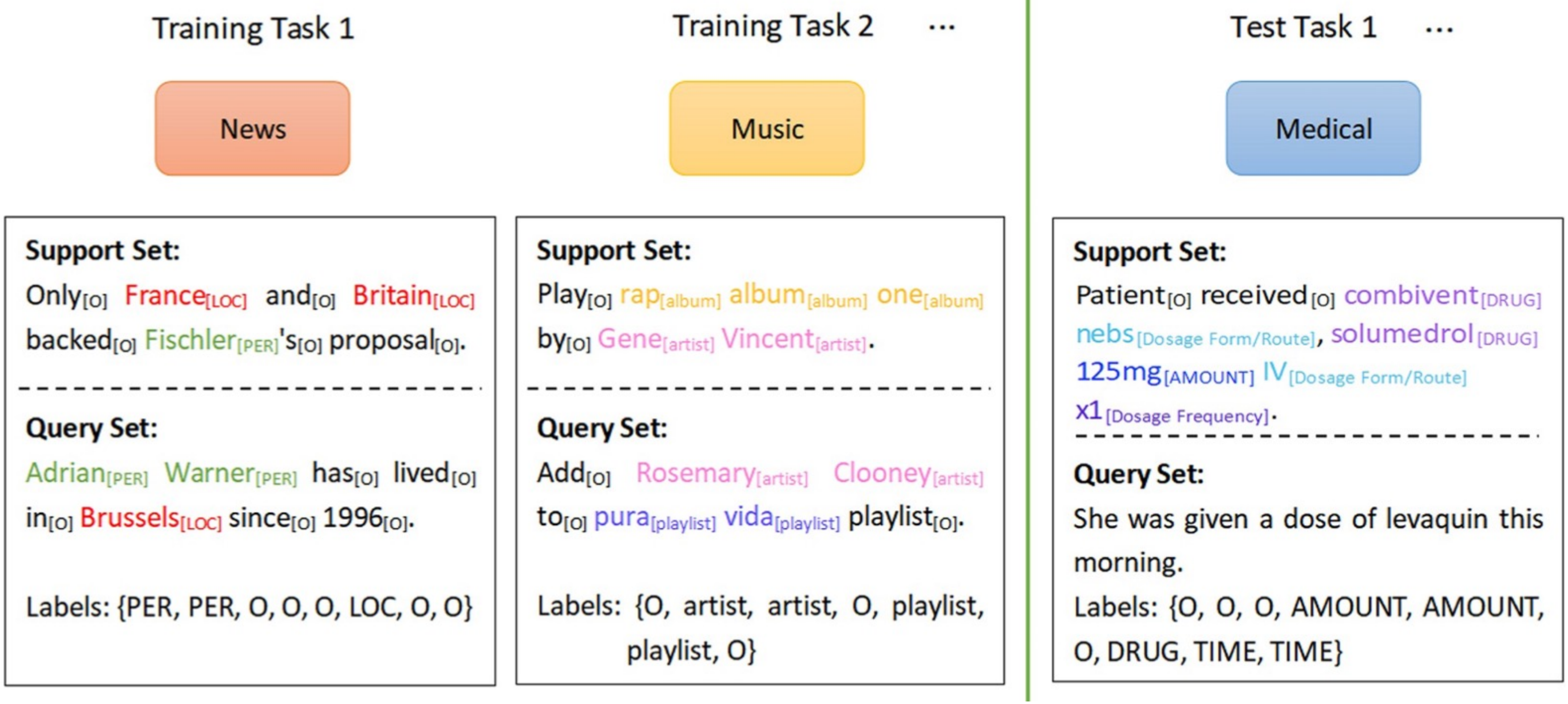


Source: Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.

Few-Shot Learning (FSL)

Meta-learning

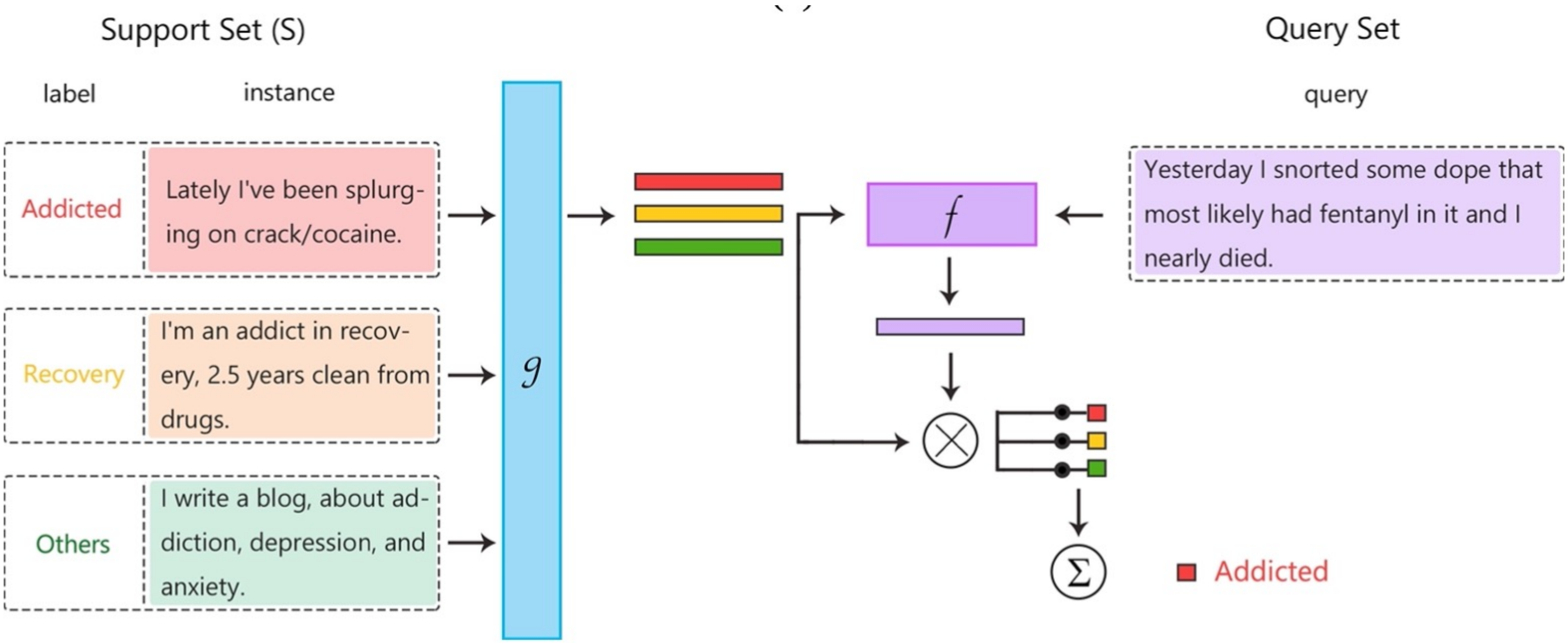
Each task mimics the few-shot scenario, and can be completely non-overlapping.
Support sets are used to train; query sets are used to evaluate the model



Source: Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).

Few-Shot Learning (FSL)

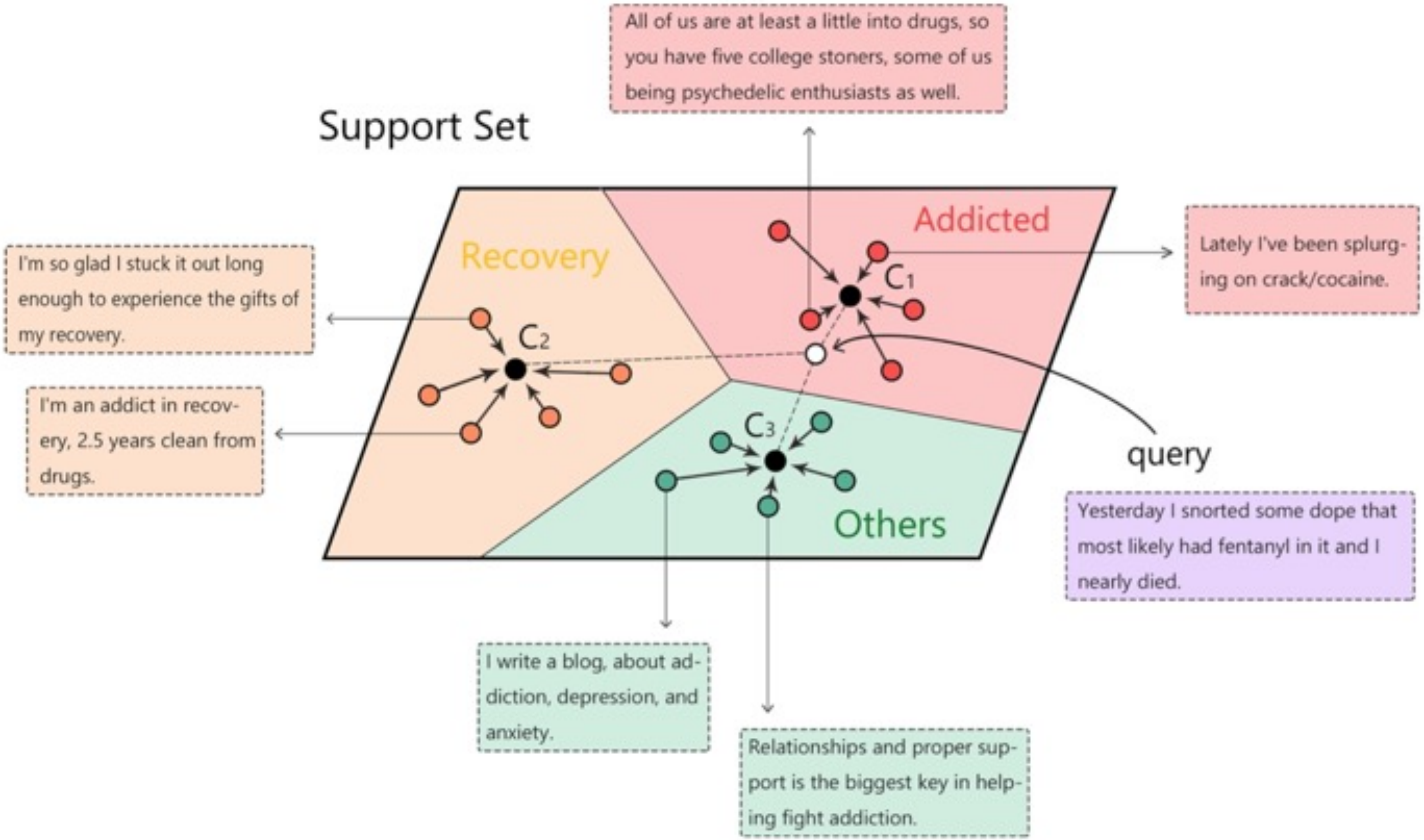
Matching networks



Source: Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).

Few-Shot Learning (FSL)

Prototypical network



Source: Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).

Few-Shot Learning (FSL) for medical text

Study	Year	Data source	Research aim	Size of training set	Number of entities / classes	Entity type of training domain	Entity type of test domain
Alicia Lara-Clares and Ana Garcia-Serrano[44]	2019	MEDDOCAN shared task dataset[45]	NER	500 clinical cases, with no reconstruction	29	Clinical	Clinical
Ferré et al.[46]	2019	BB-norm dataset from the Bacteria Biotope 2019 Task[47]	Entity Normalization	Original dataset with no reconstruction and zero-shot	Not mentioned *	Biological	Biological
Hou et al.[48]	2020	Snips dataset[49]	Slot Tagging (NER)	1-shot and 5-shot	7	Six of Weather, Music, PlayList, Book (including biomedical), Search Screen (including biomedical), Restaurant and Creative Work.	The remaining one
Sharaf et al.[50]	2020	ten different datasets collected from the Open Parallel Corpus (OPUS)[51]	Neural Machine Translation (NMT)	Sizes ranging from 4k to 64k training words (200 to 3200 sentences), but reconstructed	N/A †	Bible, European Central Bank, KDE, Quran, WMT news test sets, Books, European Medicines Agency (EMEA), Global Voices, Medical (ufal-Med), TED talks	Bible, European Central Bank, KDE, Quran, WMT news test sets, Books, European Medicines Agency (EMEA), Global Voices, Medical (ufal-Med), TED talks
Lu et al.[52]	2020	MIMIC II[22] and MIMIC III[23], and EU legislation dataset[53]	Multi-label Text Classification	5-shot for MIMIC II and III, 50-shot for EU legislation	MIMIC II: 9 MIMIC III: 15 EU legis-lation: 5	Medical	Medical

Source: Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).

Few-Shot Learning (FSL) for medical text

Study	Year	Data source	Research aim	Size of training set	Number of entities / classes	Entity type of training domain	Entity type of test domain
Lu et al.[80]	2021	Constructed and shared a novel dataset [††] based on Weibo for the research of few-shot rumor detection, and use PHEME dataset[81]	Rumor Detection (NER)	For the Weibo dataset: 2-way 3-event 5-shot 9-query; for PHEME dataset: 2-way 2-event 5-shot 9-query	Weibo: 14 PHEME: 5	Source posts and comments from Sina Weibo related to COVID-19	Source posts and comments from Sina Weibo related to COVID-19
Ma et al. [82]	2021	CCLE, CERES-correctedCRISPR gene disruption scores, GDSC1000 dataset, PDTC dataset and PDX dataset[‡‡]	Drug-response Predictions	1-shot, 2-shot, 5-shot and 10-shot	N/A[†]	Biomedical	Biomedical
Kormilitzin et al.[83]	2021	MIMIC-III[23] and UK-CRIS datasets[30, 31]	NER	25%, 50%, 75% and 100% of the training set, with no reconstruction	7	Electronic health record	Electronic health record
Guo et al.[84]	2021	Abstracts of biomedicalliteratures (from relation extraction task of BioNLP Shared Task 2011 and 2019[47]) and structured biological datasets	NER	100%, 75%, 50%, 25%, 0% of training set, with no reconstruction	Not mentioned [*]	Biomedical entities	Biomedical entities
Lee et al.[85]	2021	COVID19-Scientific[86], COVID19-Social[87] (fact-checked by journalists from a website called Politi-fact.com), FEVER[88] (Fact Extraction and Verification, generated by altering sentences extracted from Wiki-pedia to promote research onfact-checking systems)	Fact-Checking (close to Text Classification)	2-shot, 10-shot and 50-shot	Not mentioned [*]	Facts about COVID-19	Facts about COVID-19

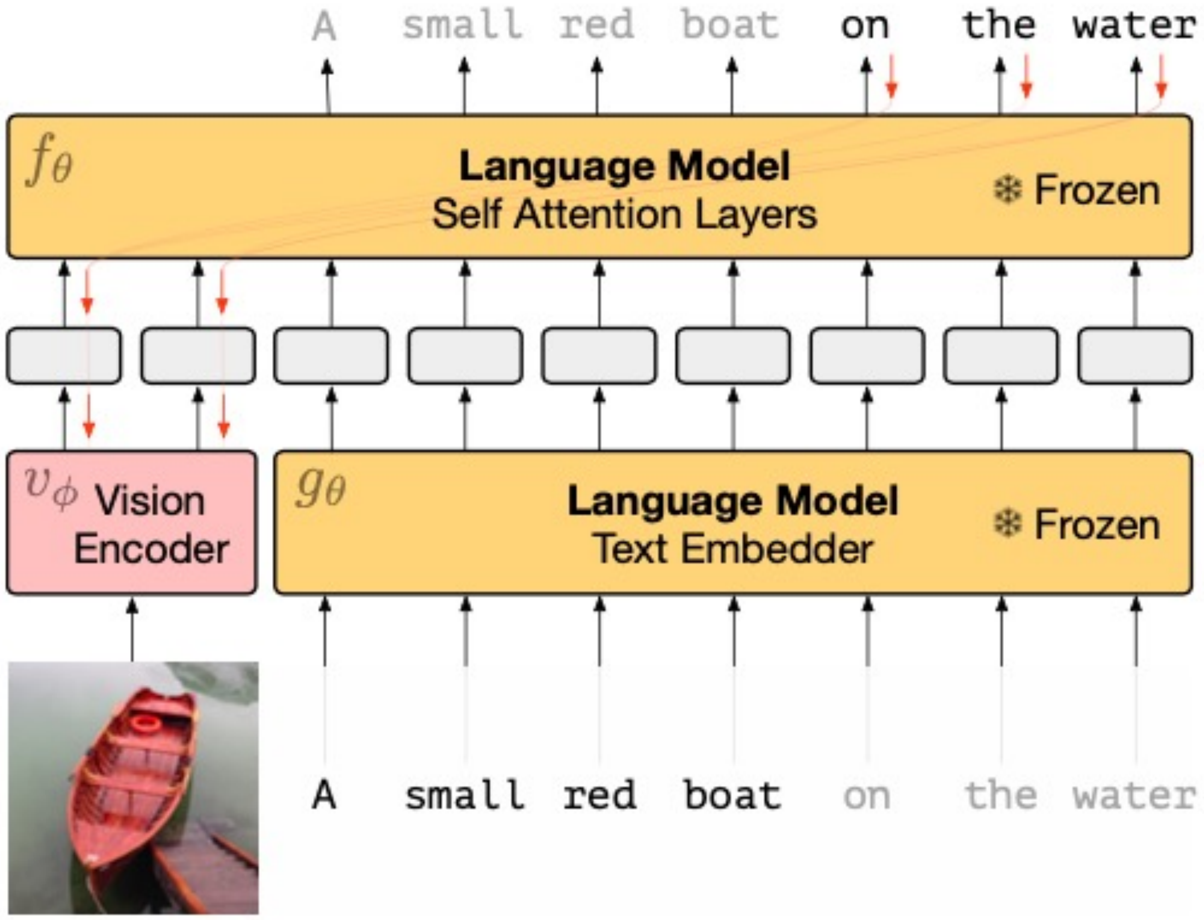
Source: Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).

Multimodal Few-Shot Learning with Frozen Language Models

Curated samples with about five seeds required to get past well-known language model failure modes of either repeating text for the prompt or emitting text that does not pertain to the image.
These samples demonstrate the ability to generate open-ended outputs that adapt to both images and text, and to make use of facts that it has learned during language-only pre-training.

Source: Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models." Advances in Neural Information Processing Systems 34 (2021): 200-212.

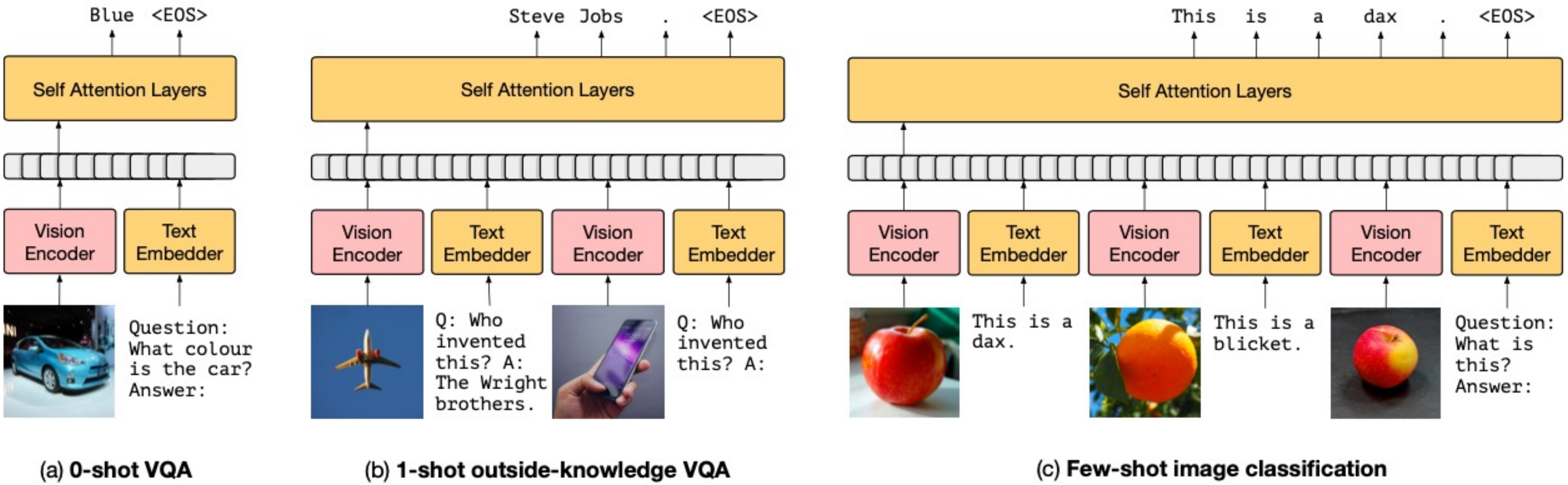
Multimodal Few-Shot Learning with Frozen Language Models



Gradients through a frozen language model's self attention layers are used to train the vision encoder.

Source: Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models." Advances in Neural Information Processing Systems 34 (2021): 200-212.

Multimodal Few-Shot Learning with Frozen Language Models



Inference-Time interface for *Frozen*. The figure demonstrates how we can support (a) visual question answering, (b) outside-knowledge question answering and (c) few-shot image classification via in-context learning.

Source: Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models." Advances in Neural Information Processing Systems 34 (2021): 200-212.

Multimodal Few-Shot Learning with Frozen Language Models

Examples of (a) the Open-Ended miniImageNet evaluation (b) the Fast VQA evaluation.

Source: Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models." Advances in Neural Information Processing Systems 34 (2021): 200-212.

GPT-3: Language Models are Few-Shot Learners

Language Models are Few-Shot Learners

Tom B. Brown* Benjamin Mann* Nick Ryder* Melanie Subbiah*
Jared Kaplan† Prafulla Dhariwal Arvind Neelakantan Pranav Shyam
Girish Sastry Amanda Askell Sandhini Agarwal Ariel Herbert-Voss
Gretchen Krueger Tom Henighan Rewon Child Aditya Ramesh
Daniel M. Ziegler Jeffrey Wu Clemens Winter
Christopher Hesse Mark Chen Eric Sigler Mateusz Litwin Scott Gray
Benjamin Chess Jack Clark Christopher Berner
Sam McCandlish Alec Radford Ilya Sutskever Dario Amodei

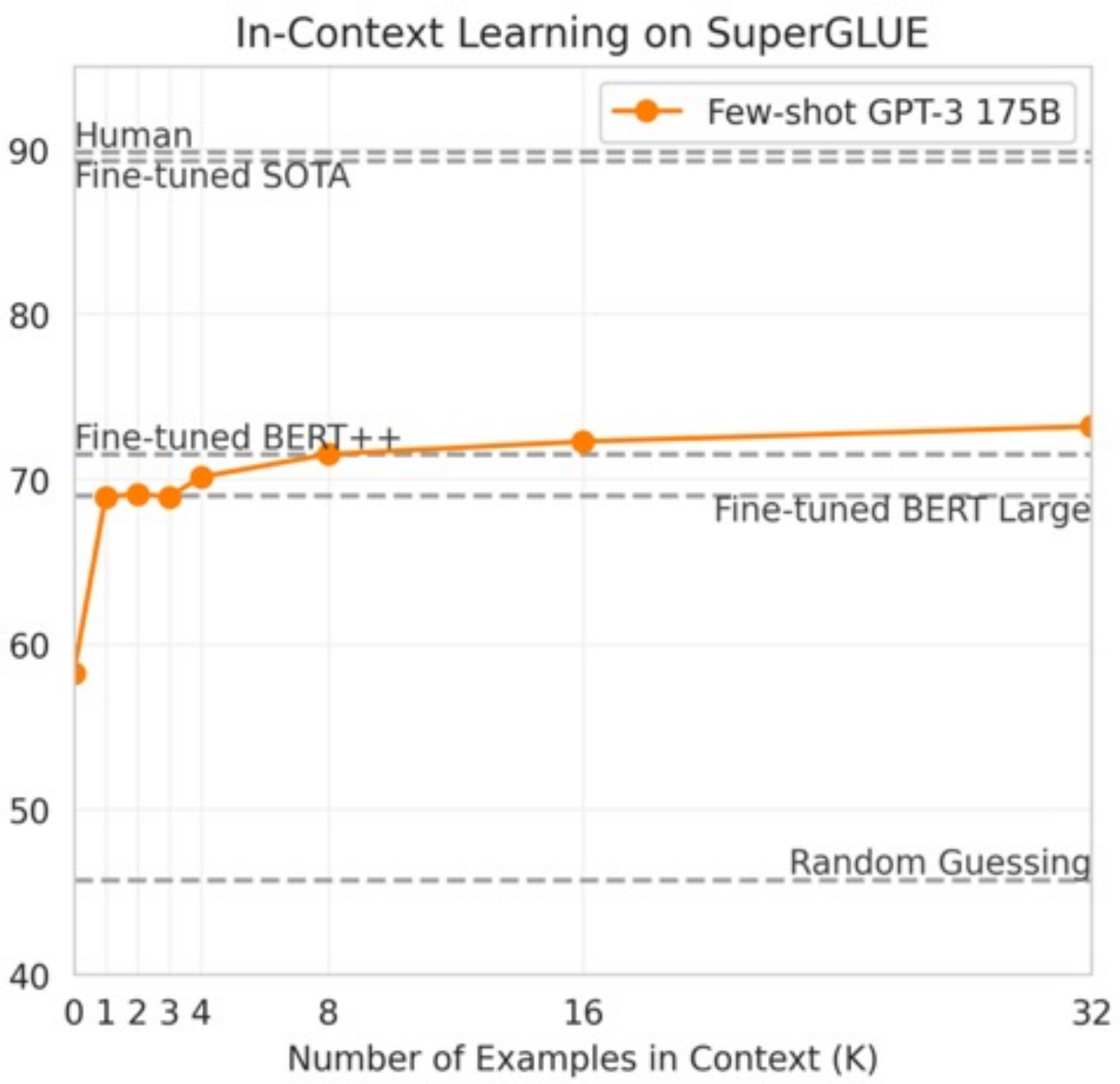
This work was funded by **OpenAI**.
All models were trained on **V100** GPU's on part of a high-bandwidth cluster provided by Microsoft.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

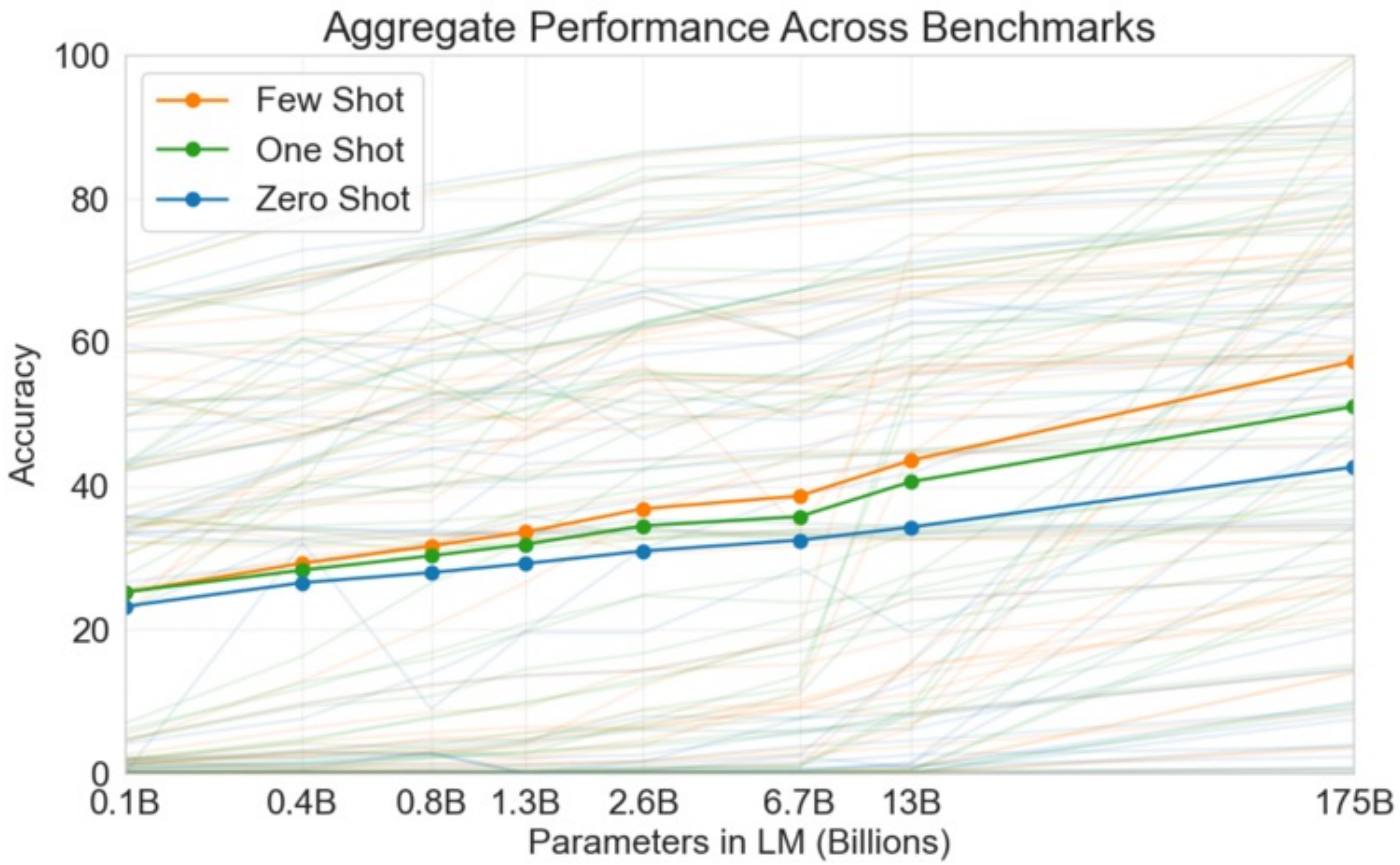
GPT-3: Language Models are Few-Shot Learners



Performance on SuperGLUE increases with model size.
A value of K = 32 means that our model was shown 32 examples per task, for 256 examples total divided across the 8 tasks in SuperGLUE.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners



Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Performance on cloze and completion tasks.

Setting	LAMBADA (acc)	LAMBADA (ppl)	StoryCloze (acc)	HellaSwag (acc)
SOTA	68.0[a]	8.63[b]	**91.8**[c]	**85.6**[d]
GPT-3 Zero-Shot	**76.2**	**3.00**	83.2	78.9
GPT-3 One-Shot	**72.5**	**3.35**	84.7	78.1
GPT-3 Few-Shot	**86.4**	**1.92**	87.7	79.3

GPT-3 significantly improves SOTA on LAMBADA while achieving respectable performance on two difficult completion prediction datasets.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Results on three Open-Domain QA tasks

Setting	NaturalQS	WebQS	TriviaQA
RAG (Fine-tuned, Open-Domain) [LPP+20]	**44.5**	**45.5**	**68.0**
T5-11B+SSM (Fine-tuned, Closed-Book) [RRS20]	36.6	44.7	60.5
T5-11B (Fine-tuned, Closed-Book)	34.5	37.4	50.1
GPT-3 Zero-Shot	14.6	14.4	64.3
GPT-3 One-Shot	23.0	25.3	**68.0**
GPT-3 Few-Shot	29.9	41.5	**71.2**

GPT-3 is shown in the few-, one-, and zero-shot settings, as compared to prior SOTA results for closed book and open domain settings.
TriviaQA few-shot result is evaluated on the wiki split test server.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

GPT-3 results on a selection of QA / RC tasks.

Setting	ARC (Easy)	ARC (Challenge)	CoQA	DROP
Fine-tuned SOTA	**92.0**[a]	**78.5**[b]	**90.7**[c]	**89.1**[d]
GPT-3 Zero-Shot	68.8	51.4	81.5	23.6
GPT-3 One-Shot	71.2	53.2	84.0	34.3
GPT-3 Few-Shot	70.1	51.5	85.0	36.5

CoQA and DROP are F1 while ARC reports accuracy.
See the appendix for additional experiments. a[KKS+20] b[KKS+20] c[JZC+19] d [JN20]

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Setting	En→Fr	Fr→En	En→De	De→En	En→Ro	Ro→En
SOTA (Supervised)	**45.6**[a]	35.0 [b]	**41.2**[c]	40.2[d]	**38.5**[e]	**39.9**[e]
XLM [LC19]	33.4	33.3	26.4	34.3	33.3	31.8
MASS [STQ+19]	37.5	34.9	28.3	35.2	35.2	33.1
mBART [LGG+20]	-	-	29.8	34.0	35.0	30.5
GPT-3 Zero-Shot	25.2	21.2	24.6	27.2	14.1	19.9
GPT-3 One-Shot	28.3	33.7	26.2	30.4	20.6	38.6
GPT-3 Few-Shot	32.6	39.2	29.7	40.6	21.0	39.5

Few-shot GPT-3 outperforms previous unsupervised NMT work by 5 BLEU when translating into English reflecting its strength as an English LM.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Performance of GPT-3 on SuperGLUE compared to fine-tuned baselines and SOTA

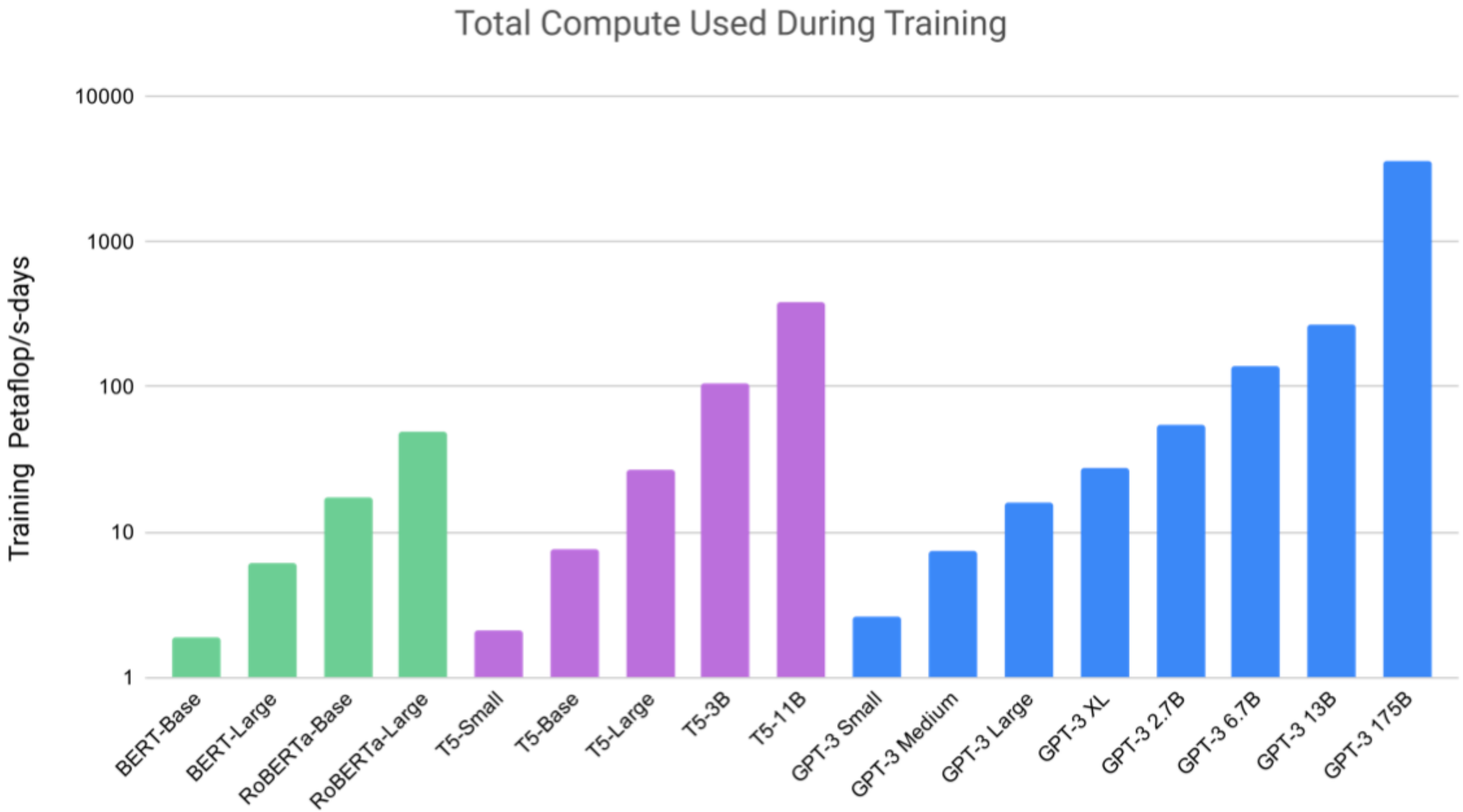
	SuperGLUE Average	BoolQ Accuracy	CB Accuracy	CB F1	COPA Accuracy	RTE Accuracy
Fine-tuned SOTA	**89.0**	**91.0**	**96.9**	**93.9**	**94.8**	**92.5**
Fine-tuned BERT-Large	69.0	77.4	83.6	75.7	70.6	71.7
GPT-3 Few-Shot	71.8	76.4	75.6	52.0	92.0	69.0

	WiC Accuracy	WSC Accuracy	MultiRC Accuracy	MultiRC F1a	ReCoRD Accuracy	ReCoRD F1
Fine-tuned SOTA	**76.1**	**93.8**	**62.3**	**88.2**	**92.5**	**93.3**
Fine-tuned BERT-Large	69.6	64.6	24.1	70.0	71.3	72.0
GPT-3 Few-Shot	49.4	80.1	30.5	75.4	90.2	91.1

GPT-3 few-shot is given a total of 32 examples within the context of each task and performs no gradient updates.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners



GPT-3 3B is almost 10x larger than RoBERTa-Large (355M params),
both models took roughly 50 petaflop/s-days of compute during pre-training

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

Human accuracy in identifying whether short (~200 word) news articles are model generated

	Mean accuracy	95% Confidence Interval (low, hi)	t compared to control (p-value)	"I don't know" assignments
Control (deliberately bad model)	86%	83%–90%	-	3.6 %
GPT-3 Small	76%	72%–80%	3.9 (2e-4)	4.9%
GPT-3 Medium	61%	58%–65%	10.3 (7e-21)	6.0%
GPT-3 Large	68%	64%–72%	7.3 (3e-11)	8.7%
GPT-3 XL	62%	59%–65%	10.7 (1e-19)	7.5%
GPT-3 2.7B	62%	58%–65%	10.4 (5e-19)	7.1%
GPT-3 6.7B	60%	56%–63%	11.2 (3e-21)	6.2%
GPT-3 13B	55%	52%–58%	15.3 (1e-32)	7.1%
GPT-3 175B	52%	49%–54%	16.9 (1e-34)	7.8%

This table compares mean accuracy between five different models, and shows the results of a two-sample T-Test for the difference in mean accuracy between each model and the control model (an unconditional GPT-3 Small model with increased output randomness).

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

GPT-3: Language Models are Few-Shot Learners

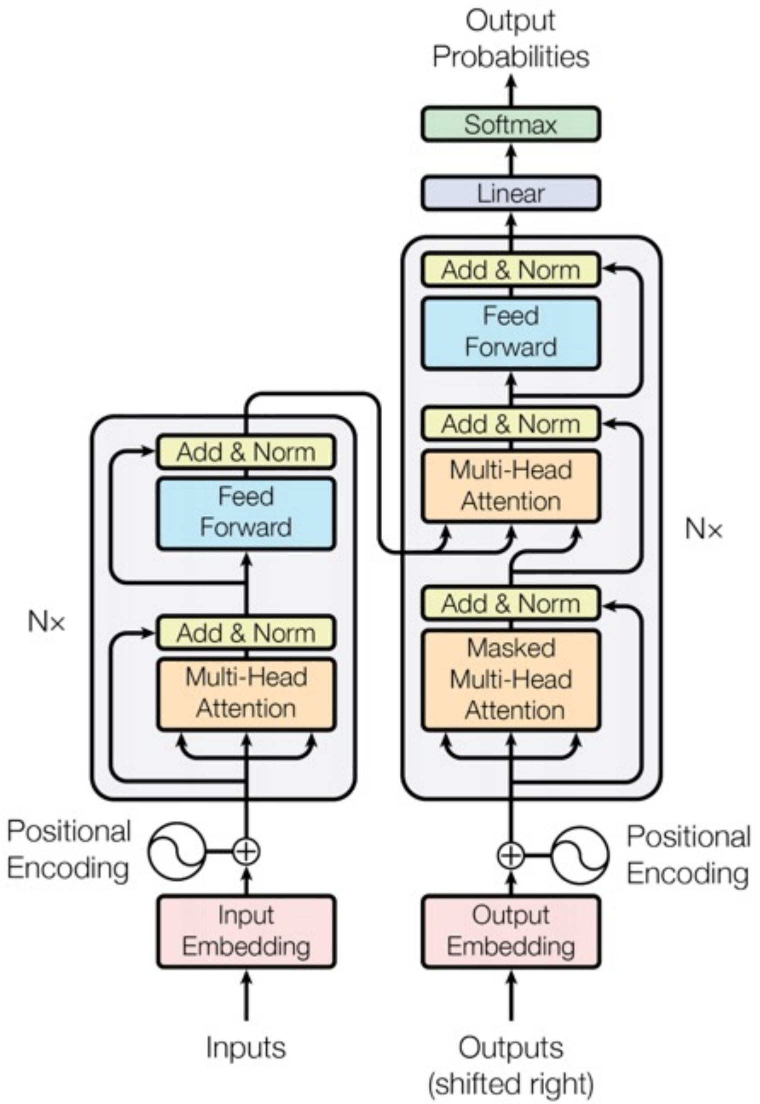
The GPT-3 generated news article that humans had the greatest difficulty distinguishing from a human written article (accuracy: 12%)

Title: United Methodists Agree to Historic Split
Subtitle: Those who oppose gay marriage will form their own denomination
Article: After two days of intense debate, the United Methodist Church has agreed to a historic split - one that is expected to end in the creation of a new denomination, one that will be "theologically and socially conservative," according to The Washington Post. The majority of delegates attending the church's annual General Conference in May voted to strengthen a ban on the ordination of LGBTQ clergy and to write new rules that will "discipline" clergy who officiate at same-sex weddings. But those who opposed these measures have a new plan: They say they will form a separate denomination by 2020, calling their church the Christian Methodist denomination.
The Post notes that the denomination, which claims 12.5 million members, was in the early 20th century the "largest Protestant denomination in the U.S.," but that it has been shrinking in recent decades. The new split will be the second in the church's history. The first occurred in 1968, when roughly 10 percent of the denomination left to form the Evangelical United Brethren Church. The Post notes that the proposed split "comes at a critical time for the church, which has been losing members for years," which has been "pushed toward the brink of a schism over the role of LGBTQ people in the church." Gay marriage is not the only issue that has divided the church. In 2016, the denomination was split over ordination of transgender clergy, with the North Pacific regional conference voting to ban them from serving as clergy, and the South Pacific regional conference voting to allow them.

Source: Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901

Transformer (Attention is All You Need)

(Vaswani et al., 2017)

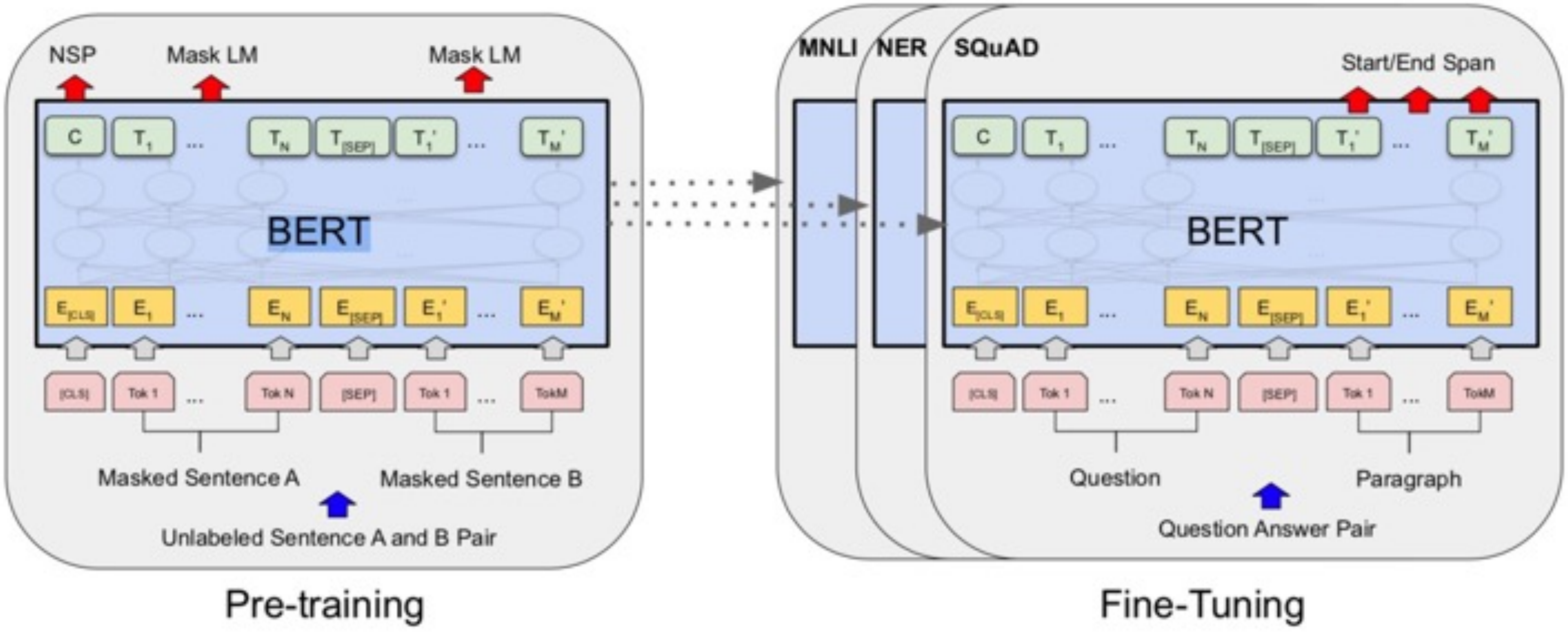


Source: Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In *Advances in neural information processing systems*, pp. 5998-6008. 2017.

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

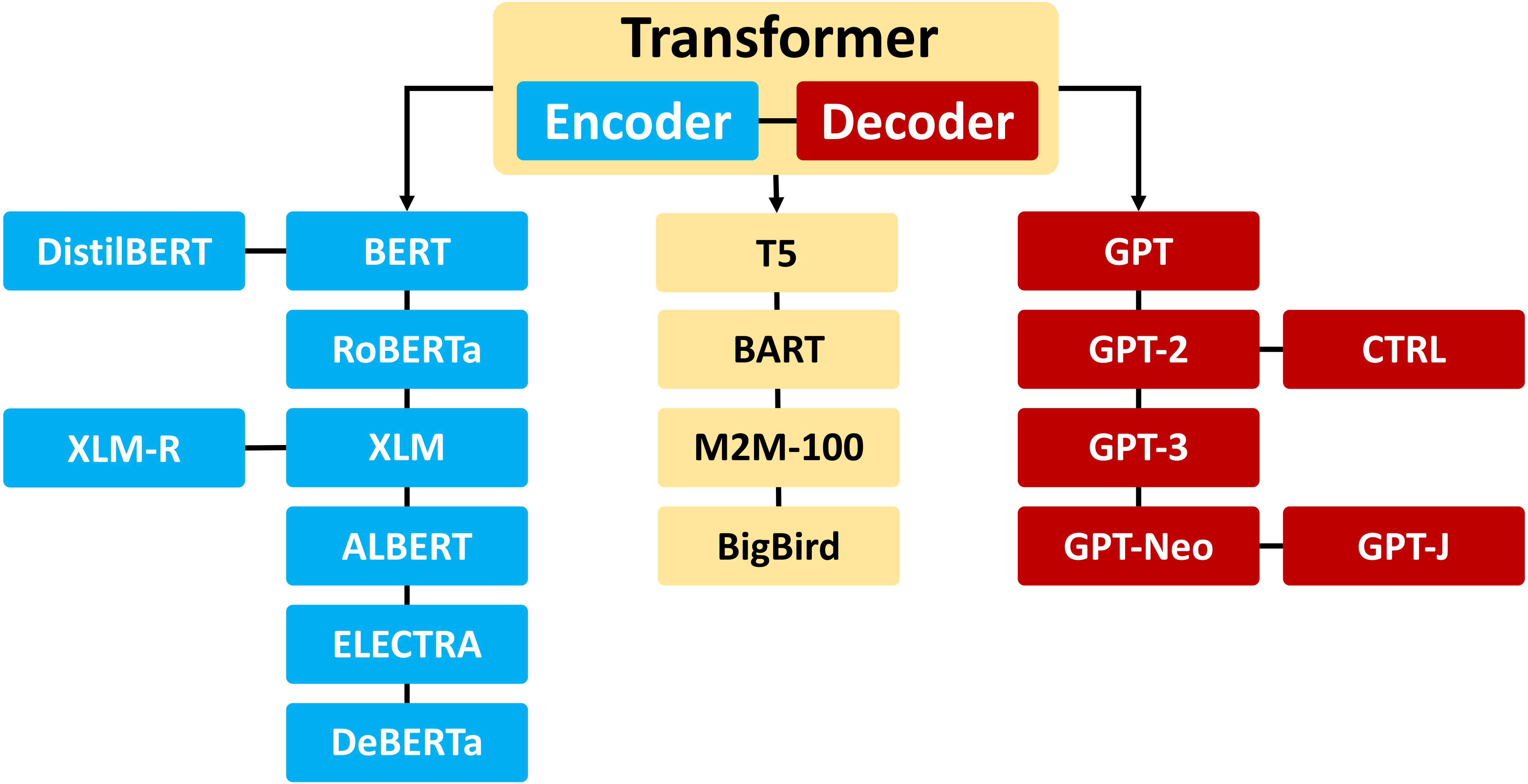
BERT (Bidirectional Encoder Representations from Transformers)

Overall pre-training and fine-tuning procedures for BERT



Source: Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805.

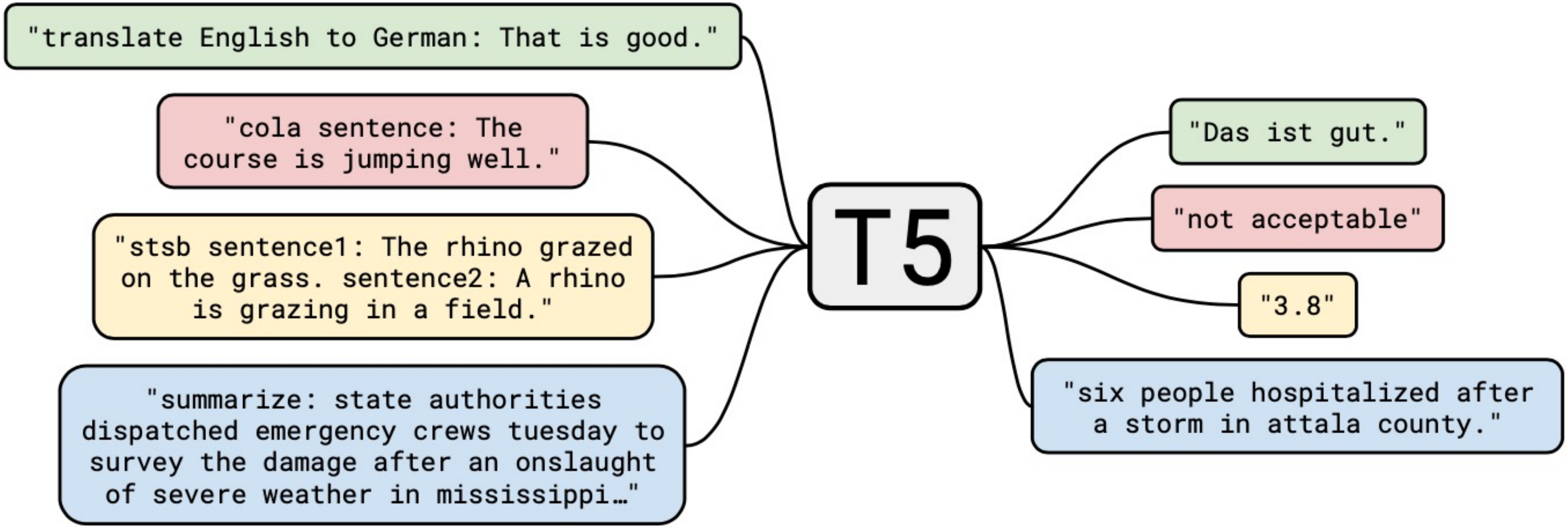
Transformer Models



Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.

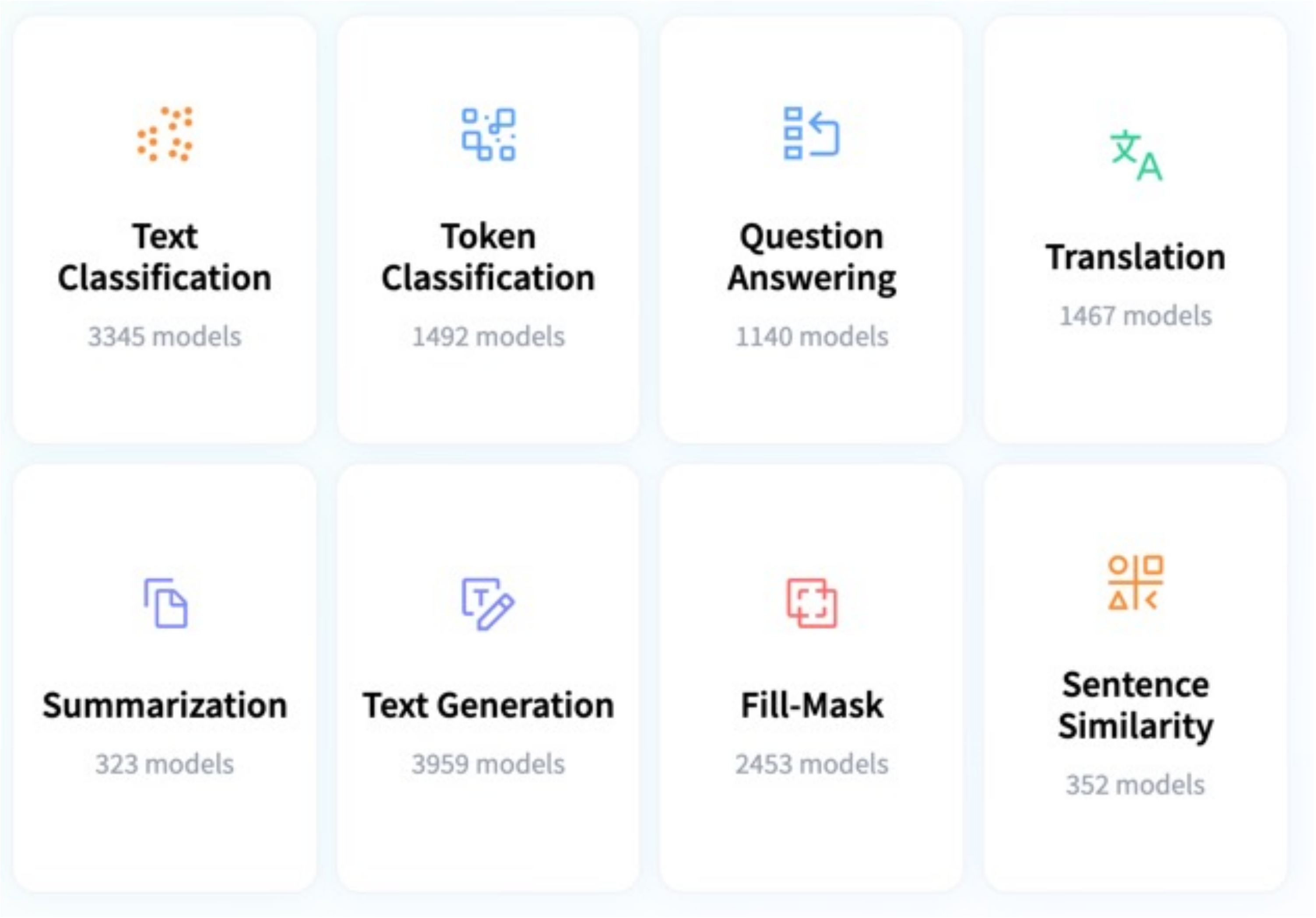
T5

Text-to-Text Transfer Transformer



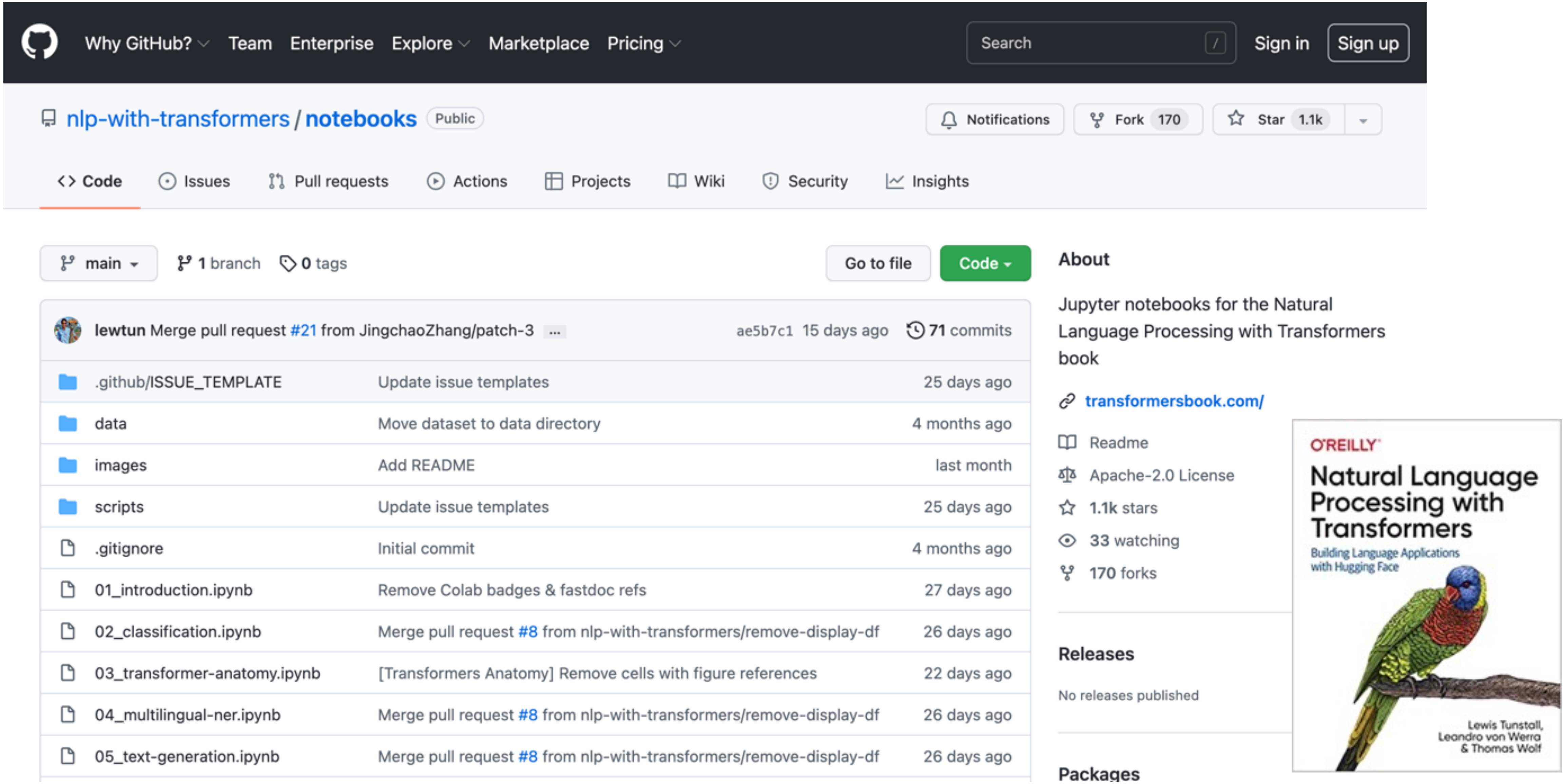
Source: JColin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu (2019). "Exploring the limits of transfer learning with a unified text-to-text transformer." arXiv preprint arXiv:1910.10683 (2019).

Hugging Face Tasks
Natural Language Processing



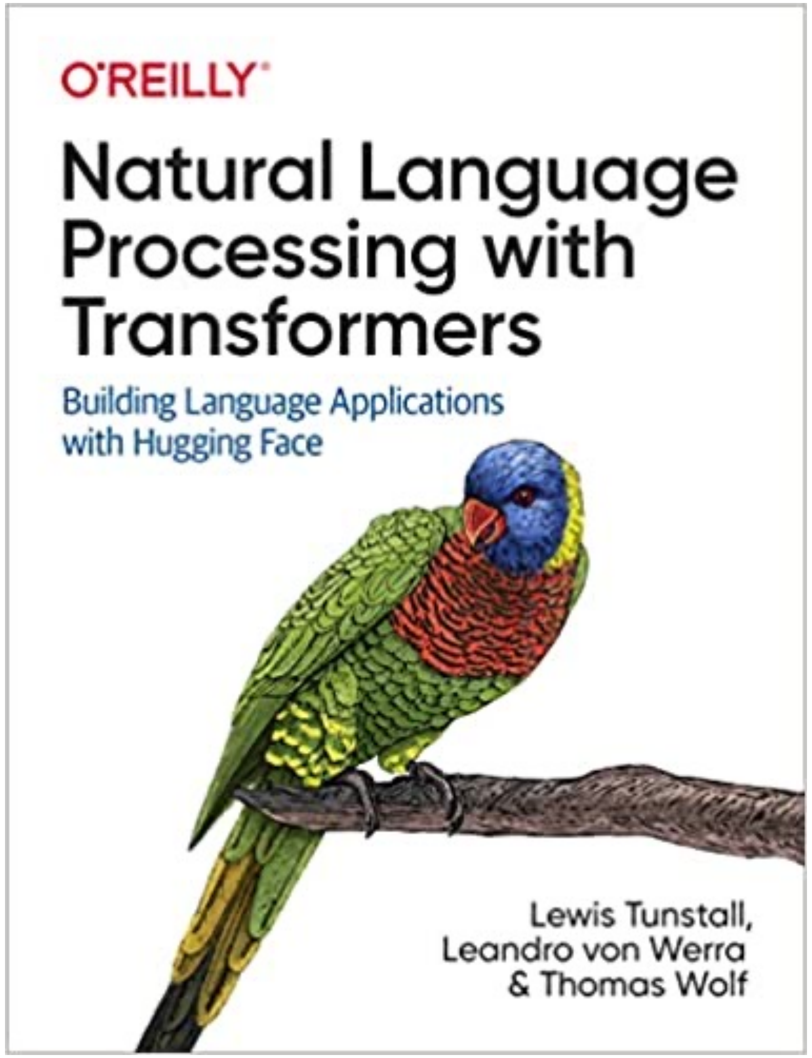
https://huggingface.co/tasks

NLP with Transformers Github



https://github.com/nlp-with-transformers/notebooks

NLP with Transformers Github Notebooks



Running on a cloud platform

To run these notebooks on a cloud platform, just click on one of the badges in the table below:

Chapter	Colab	Kaggle	Gradient	Studio Lab
Introduction	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Classification	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Transformer Anatomy	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Multilingual Named Entity Recognition	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Text Generation	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Summarization	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Question Answering	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Making Transformers Efficient in Production	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Dealing with Few to No Labels	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Training Transformers from Scratch	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab
Future Directions	Open in Colab	Open in Kaggle	Run on Gradient	Open Studio Lab

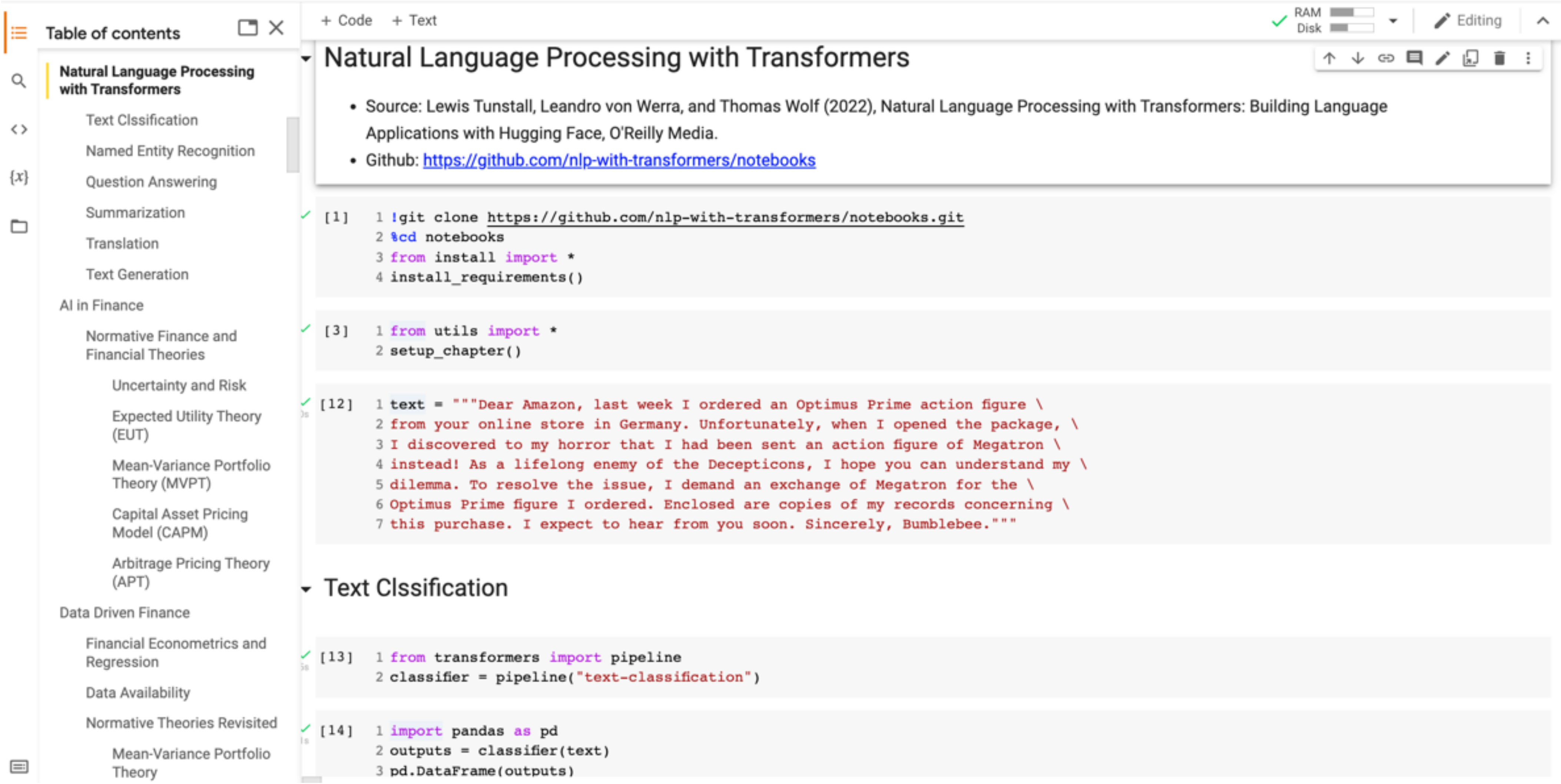
Nowadays, the GPUs on Colab tend to be K80s (which have limited memory), so we recommend using Kaggle, Gradient, or SageMaker Studio Lab. These platforms tend to provide more performant GPUs like P100s, all for free!

https://github.com/nlp-with-transformers/notebooks

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

NLP with Transformers

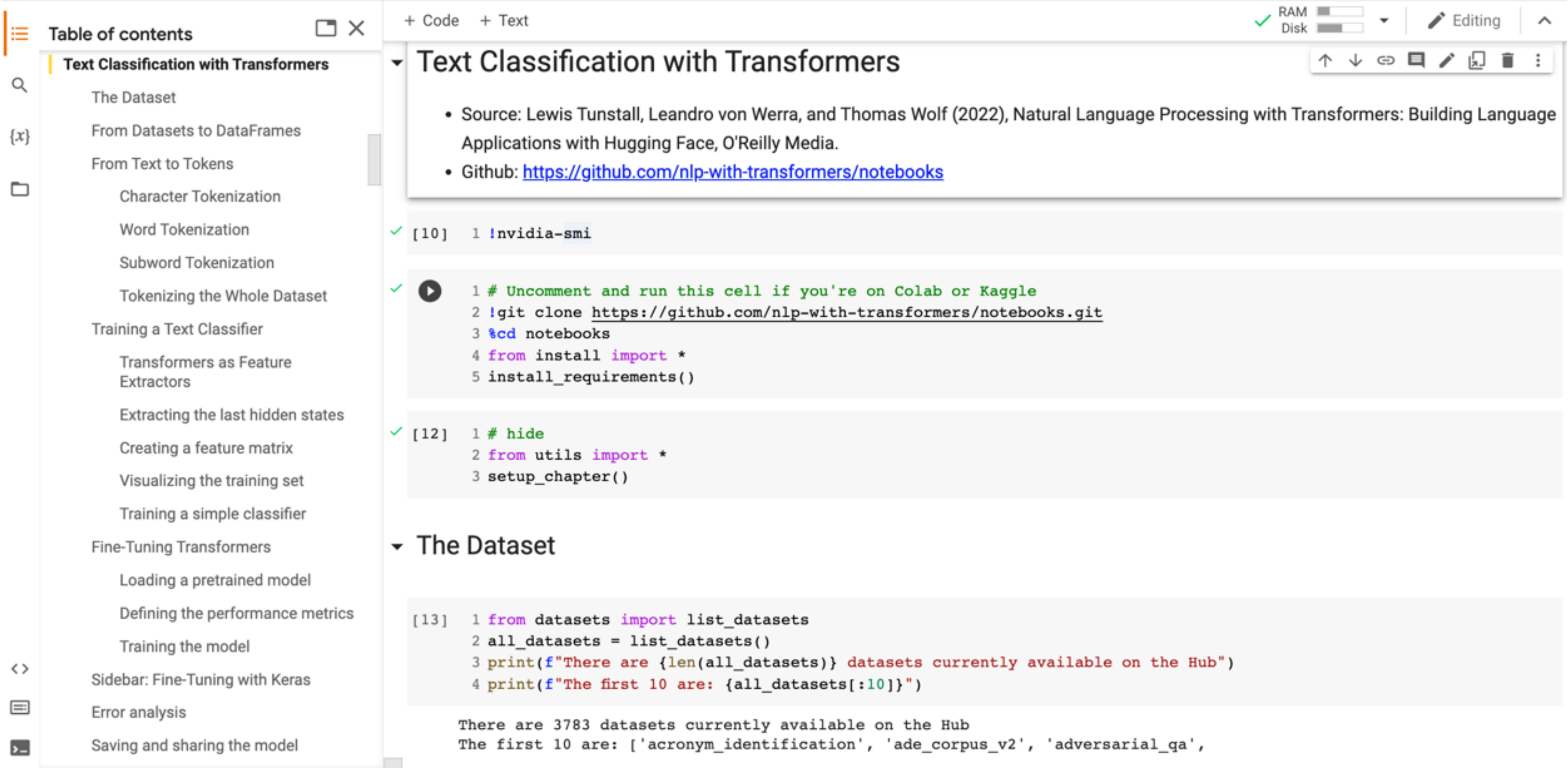


https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Text Classification



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

Named Entity Recognition (NER)

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share A

+ Code + Text

RAM Disk Editing

Multilingual Named Entity Recognition (NER)

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```
#NER: https://huggingface.co/tasks/token-classification
!pip install transformers
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

```
from transformers import pipeline
classifier = pipeline("ner")
classifier("Hello I'm Omar and I live in Zürich.")
```

```
No model was supplied, defaulted to dbmdz/bert-large-cased-finetuned-conll03-english (https://huggingface.co/dbmdz/bert-large-cased-finetuned-conll03-eng
[{'end': 14,
  'entity': 'I-PER',
  'index': 5,
  'score': 0.99770516,
  'start': 10,
  'word': 'Omar'},
 {'end': 35,
  'entity': 'I-LOC',
  'index': 10,
  'score': 0.9968976,
  'start': 29,
  'word': 'Zürich'}]
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help Saving...

Text Summarization

Comment Share

+ Code + Text

RAM Disk Editing

Text Summarization

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

```
#Source: https://huggingface.co/tasks/summarization
!pip install transformers
from transformers import pipeline
classifier = pipeline("summarization")
text = "Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents as of 2018, in an area of more than
classifier(text, max_length=30)
```

```
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 (https://huggingface.co/sshleifer/distilbart-cnn-12-6)
Your min_length=56 must be inferior than your max_length=30.
[{'summary_text': ' Paris is the capital and most populous city of France, with an estimated population of 2,175,601 residents . The City of Paris'}]
```

```
#!pip install transformers
text = """Dear Amazon, last week I ordered an Optimus Prime action figure \
from your online store in Germany. Unfortunately, when I opened the package, \
I discovered to my horror that I had been sent an action figure of Megatron \
instead! As a lifelong enemy of the Decepticons, I hope you can understand my \
dilemma. To resolve the issue, I demand an exchange of Megatron for the \
Optimus Prime figure I ordered. Enclosed are copies of my records concerning \
this purchase. I expect to hear from you soon. Sincerely, Bumblebee."""
from transformers import pipeline
summarizer = pipeline("summarization")
outputs = summarizer(text, max_length=45, clean_up_tokenization_spaces=True)
print(outputs[0]['summary_text'])
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Text Generation

Comment Share

+ Code + Text

RAM Disk Editing

Text Generation

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

[9]

```
#Source: https://huggingface.co/tasks/text-generation
#!pip install transformers
from transformers import pipeline
generator = pipeline('text-generation', model = 'gpt2')
generator("Hello, I'm a language model", max_length = 30, num_return_sequences=3)
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
[{'generated_text': "Hello, I'm a language model.\n\nBut then, one day, I'm not trying to teach the language in my head.\n\n"},
 {'generated_text': "Hello, I'm a language model. I'm an implementation for the type system. I'm working with types and programming language constructs. I a
 {'generated_text': "Hello, I'm a language modeler, not a programmer. As you know, languages are not a linear model. The thing that jumps out at"}]
```

```
from transformers import pipeline
generator = pipeline('text-generation', model = 'gpt2')
outputs = generator("Once upon a time", max_length = 30, num_return_sequences=3)
print(outputs[0]['generated_text'])
```

```
Setting `pad_token_id` to `eos_token_id`:50256 for open-end generation.
Once upon a time, every person who ever saw Jesus, knew that He was Christ. And even though he might not have known Him, He was
```

[1]

```
from transformers import pipeline
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

Question Answering and Dialogue Systems

Editing

Question Answering and Dialogue Systems

- Source: Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Github: https://github.com/nlp-with-transformers/notebooks

Question Answering

[3]

```
!pip install transformers
from transformers import pipeline
qamodel = pipeline("question-answering")
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
qamodel(question = question, context = context)
```

```
from transformers import pipeline
qamodel = pipeline("question-answering")
question = "Where do I live?"
context = "My name is Michael and I live in Taipei."
qamodel(question = question, context = context)
#{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

```
No model was supplied, defaulted to distilbert-base-cased-distilled-squad (https://huggingface.co/distilbert-base-cased-distilled-squad)
{'answer': 'Taipei', 'end': 39, 'score': 0.9730741381645203, 'start': 33}
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Question Answering

[12]

```python
from transformers import pipeline
qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "What causes precipitation to fall?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
gravity
```

[13]

```python
from transformers import pipeline
qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "What is another main form of precipitation besides drizzle, rain, snow, sleet and hail?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
graupel
```

```python
#from transformers import pipeline
#qamodel = pipeline("question-answering", model ='deepset/roberta-base-squad2')
question = "Where do water droplets collide with ice crystals to form precipitation?"
context = """In meteorology, precipitation is any product of the condensation of atmospheric water vapor that falls under gravi
output = qamodel(question = question, context = context)
print(output['answer'])
```

```
within a cloud
```

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Table of contents

Section

Question Answering and Dialogue Systems

Question Answering (QA)

Question Answering and Dialogue Systems

BERT for Question Answering

Source: Apoorv Nandan (2020), BERT (from HuggingFace Transformers) for Text Extraction, https://keras.io/examples/nlp/text_extraction_with_bert/

Description: Fine tune pretrained BERT from HuggingFace Transformers on SQuAD.

Introduction

This demonstration uses SQuAD (Stanford Question-Answering Dataset). In SQuAD, an input consists of a question, and a paragraph for context. The goal is to find the span of text in the paragraph that answers the question. We evaluate our performance on this data with the "Exact Match" metric, which measures the percentage of predictions that exactly match any one of the ground-truth answers.

We fine-tune a BERT model to perform this task as follows:

1. Feed the context and the question as inputs to BERT.
2. Take two vectors S and T with dimensions equal to that of hidden states in BERT.
3. Compute the probability of each token being the start and end of the answer span. The probability of a token being the start of the answer is given by a dot product between S and the representatio of the token in the last layer of BERT, followed by a softmax over all tokens. The probability of a token being the end of the answer is compute similarly with the vector T.
4. Fine-tune BERT and learn S and T along the way.

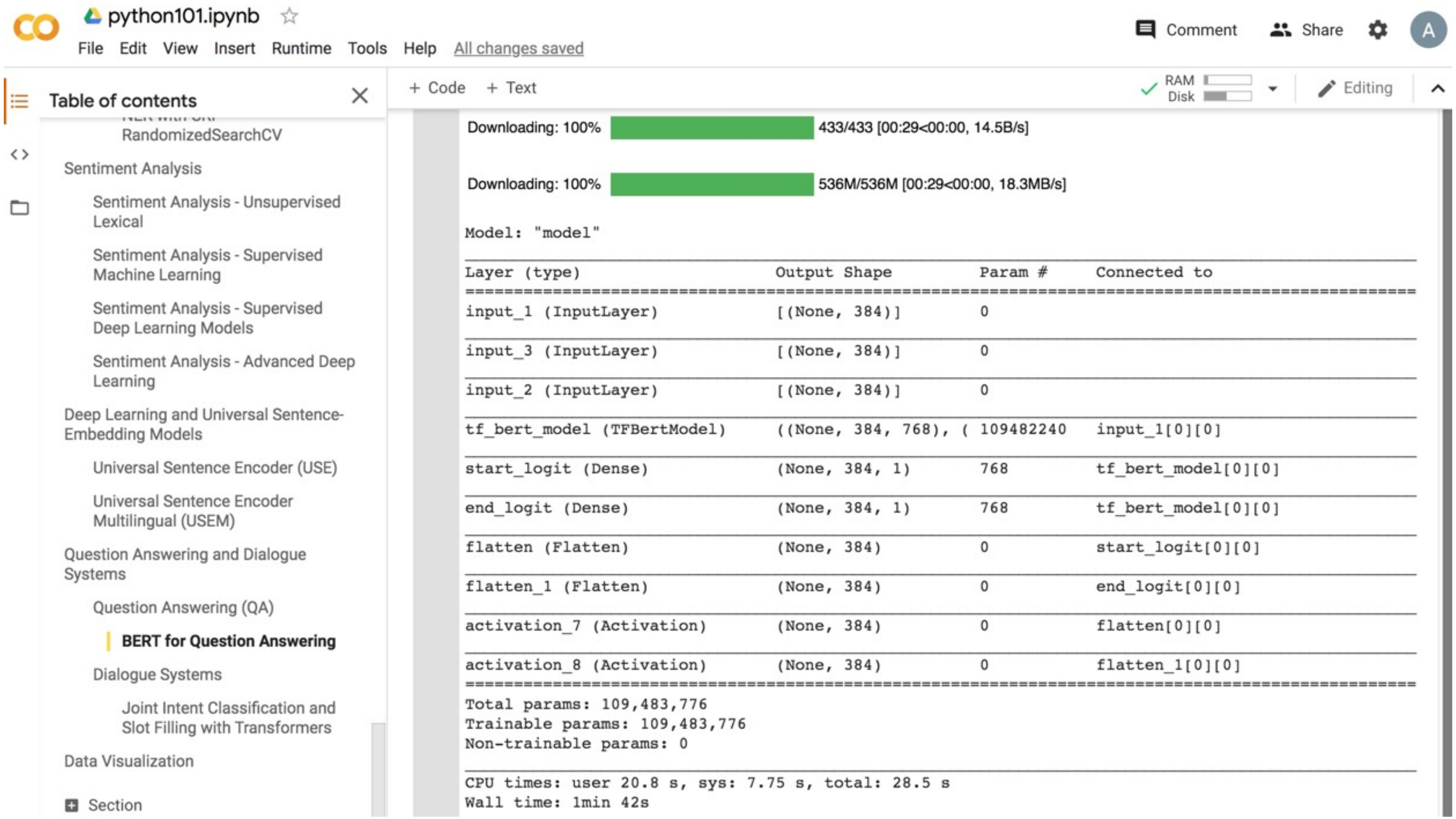
References:

- BERT
- SQuAD

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT

python101.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text RAM Disk Editing

Table of contents

Dialogue Systems

Dialogue Systems

```
#Source: Olivier Grisel (2020), Transformers (BERT fine-tuning): Joint Intent Classification and S
#https://github.com/m2dsupsdlclass/lectures-labs/blob/master/labs/06_deep_nlp/Transformers_Joint_I
```

Joint Intent Classification and Slot Filling with Transformers

The goal of this notebook is to fine-tune a pretrained transformer-based neural network model to convert a user query expressed in English into a representation that is structured enough to be processed by an automated service.

Here is an example of interpretation computed by such a Natural Language Understanding system:

```
>>> nlu("Book a table for two at Le Ritz for Friday night",
        tokenizer, joint_model, intent_names, slot_names)
```

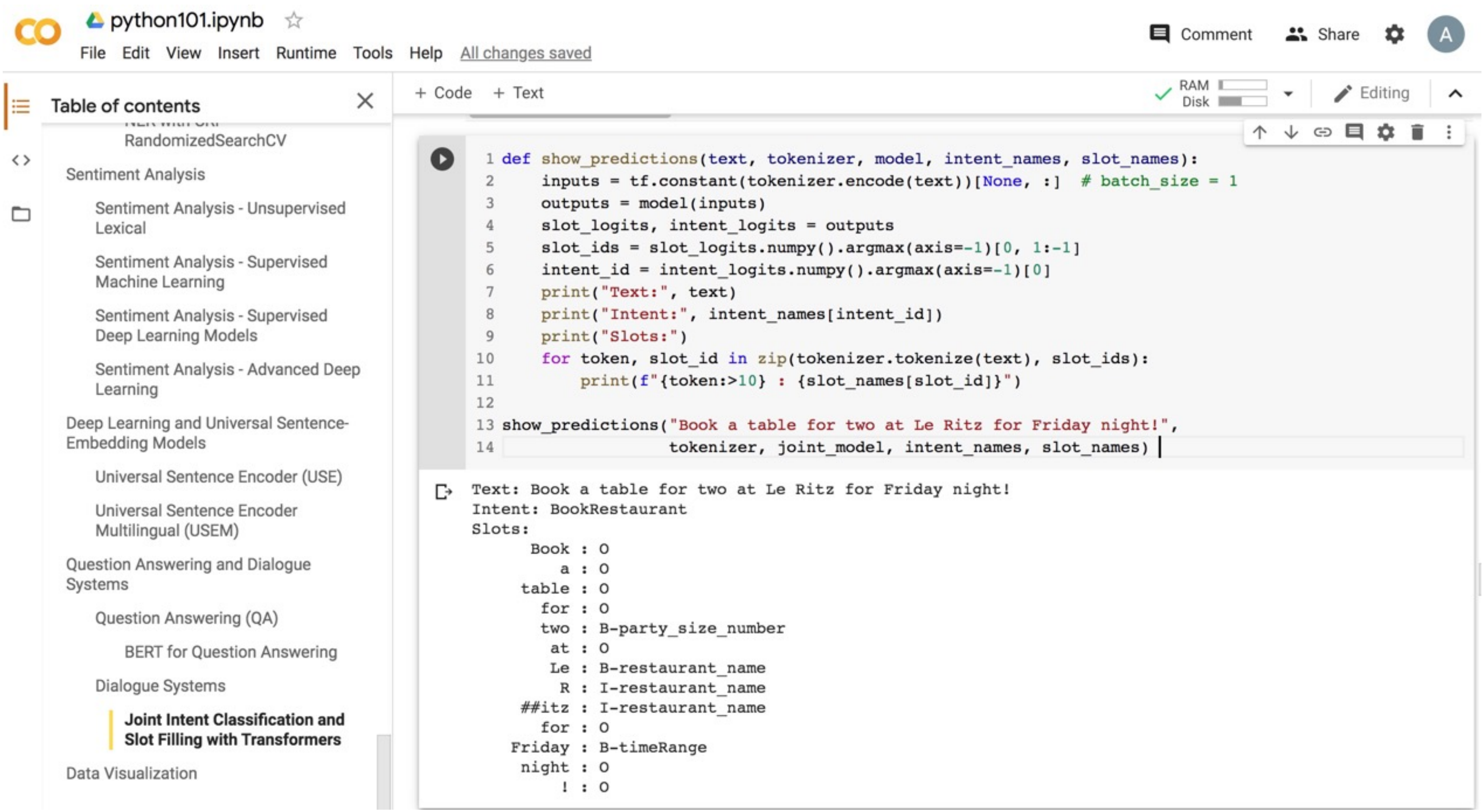
```
{
    'intent': 'BookRestaurant',
    'slots': {
        'party_size_number': 'two',
        'restaurant_name': 'Le Ritz',
        'timeRange': 'Friday night'
    }
}
```

Intent classification is a simple sequence classification problem. The trick is to treat the structured knowledge extraction part ("Slot Filling") as token-level classification problem using BIO-annotations:

https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

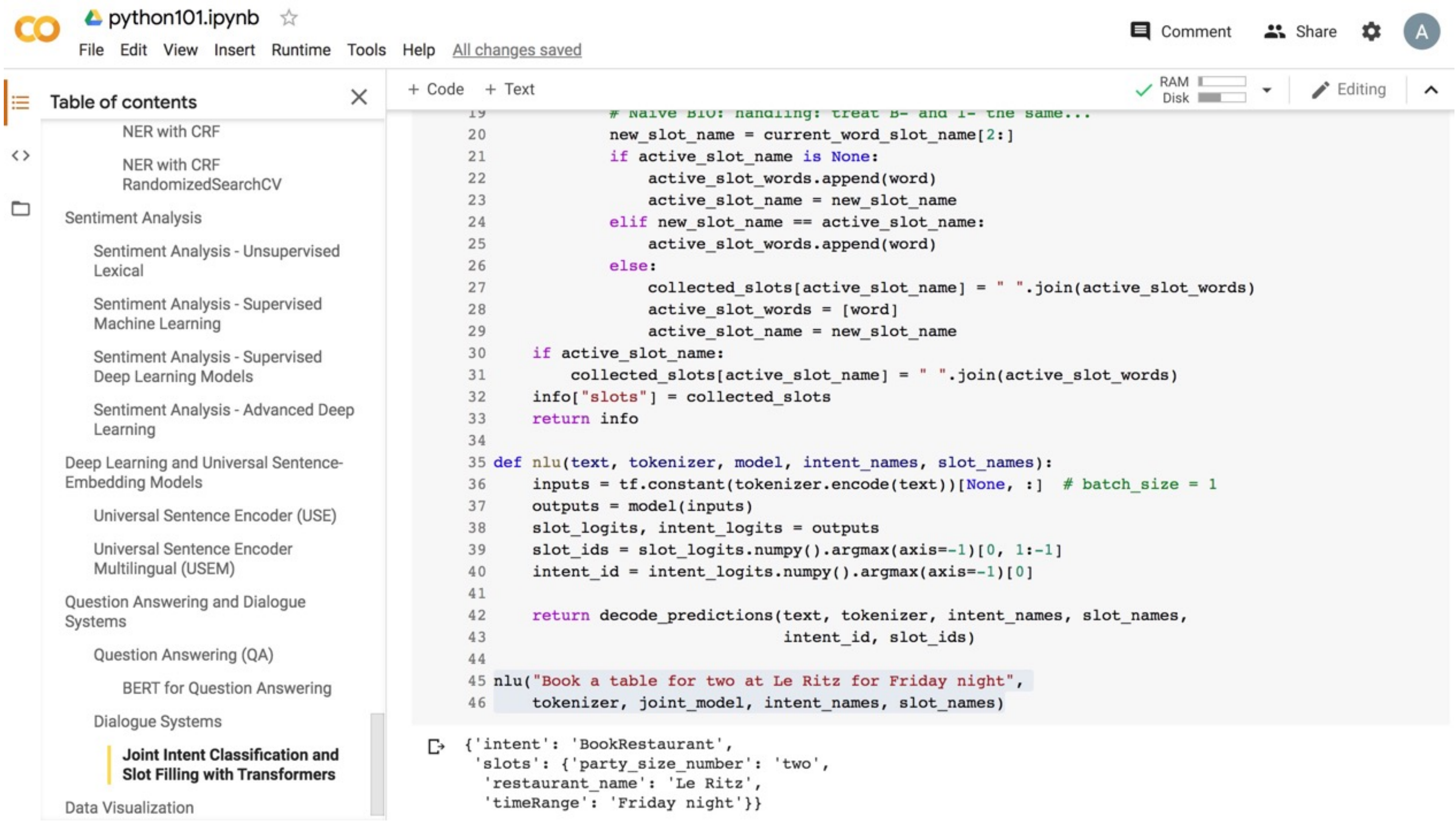
https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Python in Google Colab (Python101)

https://colab.research.google.com/drive/1FEG6DnGvwfUbeo4zJ1zTunjMqf2RkCrT



https://tinyurl.com/aintpupython101

Summary

- Deep Learning
  - Transfer Learning
    - Pre-training, Fine-Tuning (FT)
- Few-Shot Learning (FSL)
  - Meta Learning: Learn to Learn
- One-Shot Learning (1SL)
- Zero-Shot Learning (0SL)(ZSL)

References

- Lewis Tunstall, Leandro von Werra, and Thomas Wolf (2022), Natural Language Processing with Transformers: Building Language Applications with Hugging Face, O'Reilly Media.
- Denis Rothman (2021), Transformers for Natural Language Processing: Build innovative deep neural network architectures for NLP with Python, PyTorch, TensorFlow, BERT, RoBERTa, and more, Packt Publishing.
- Savaş Yıldırım and Meysam Asgari-Chenaghlu (2021), Mastering Transformers: Build state-of-the-art models from scratch with advanced natural language processing techniques, Packt Publishing.
- Sowmya Vajjala, Bodhisattwa Majumder, Anuj Gupta (2020), Practical Natural Language Processing: A Comprehensive Guide to Building Real-World NLP Systems, O'Reilly Media.
- Dipanjan Sarkar (2019), Text Analytics with Python: A Practitioner's Guide to Natural Language Processing, Second Edition. APress.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. (2020) "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901
- Yaqing Wang, Quanming Yao, James T. Kwok, and Lionel M. Ni (2020). "Generalizing from a few examples: A survey on few-shot learning." ACM computing surveys (csur) 53, no. 3 (2020): 1-34.
- Yao Ge, Yuting Guo, Yuan-Chi Yang, Mohammed Ali Al-Garadi, and Abeed Sarker (2022). "Few-shot learning for medical text: A systematic review." arXiv preprint arXiv:2204.14081 (2022).
- Maria Tsimpoukelli, Jacob L. Menick, Serkan Cabi, S. M. Eslami, Oriol Vinyals, and Felix Hill (2021). "Multimodal few-shot learning with frozen language models." Advances in Neural Information Processing Systems 34 (2021): 200-212.
- Benjamin Bengfort, Rebecca Bilbro, and Tony Ojeda (2018), Applied Text Analysis with Python: Enabling Language-Aware Data Products with Machine Learning, O'Reilly.
- Charu C. Aggarwal (2018), Machine Learning for Text, Springer.
- Gabe Ignatow and Rada F. Mihalcea (2017), An Introduction to Text Mining: Research Design, Data Collection, and Analysis, SAGE Publications.
- Rajesh Arumugam (2018), Hands-On Natural Language Processing with Python: A practical guide to applying deep learning architectures to your NLP applications, Packt.
- Jake VanderPlas (2016), Python Data Science Handbook: Essential Tools for Working with Data, O'Reilly Media.
- Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." arXiv preprint arXiv:1810.04805.
- The Super Duper NLP Repo, https://notebooks.quantumstat.com/
- Jay Alammar (2018), The Illustrated Transformer, http://jalammar.github.io/illustrated-transformer/
- Jay Alammar (2019), A Visual Guide to Using BERT for the First Time, http://jalammar.github.io/a-visual-guide-to-using-bert-for-the-first-time/
- NLP with Transformer, https://github.com/nlp-with-transformers/notebooks
- Min-Yuh Day (2022), Python 101, https://tinyurl.com/aintpupython101