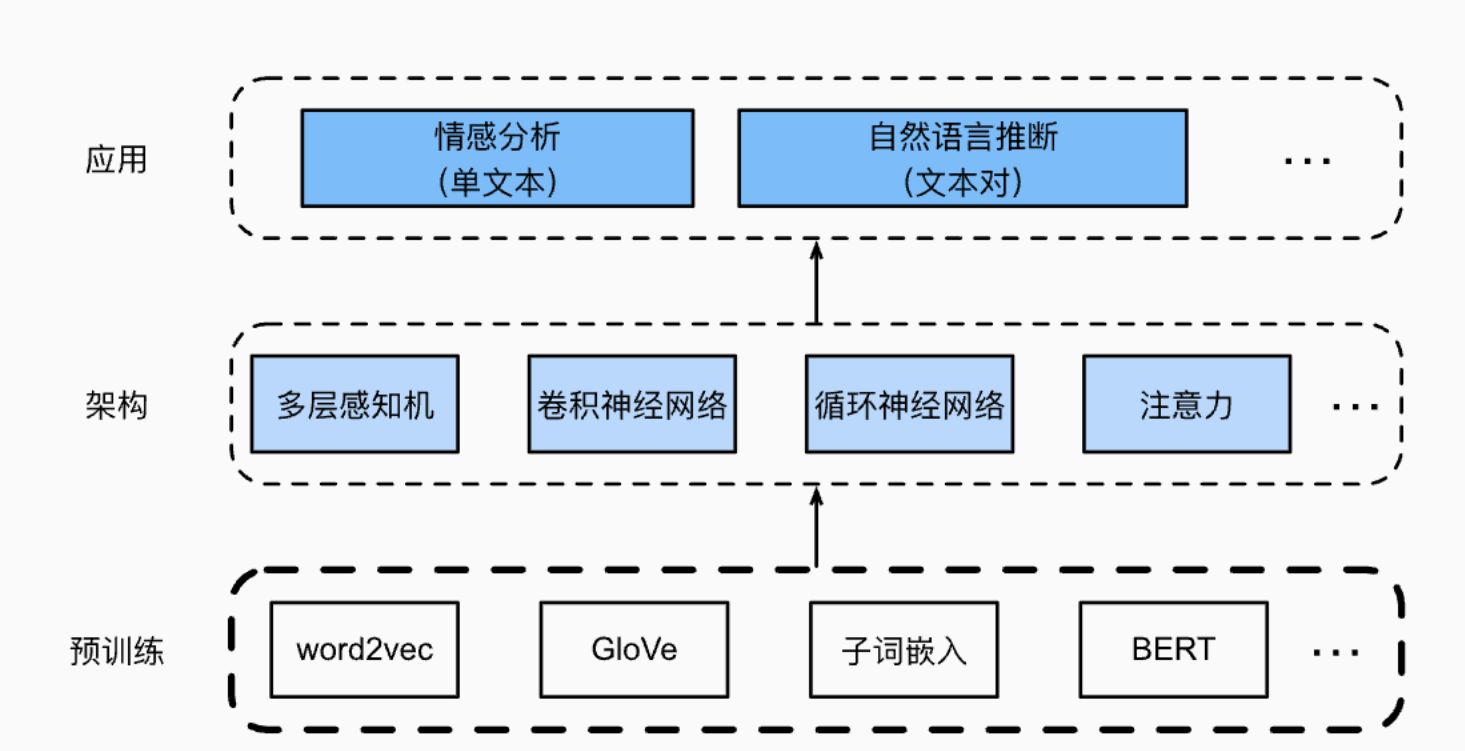
自然语言处理：预训练 2

广东工业大学
李东
dong.li@gdut.edu.cn

预训练模型在自然语言处理的作用

- 预训练好的文本表示可以放入各种深度学习架构，应用于不同自然语言处理任务（本章主要研究上游文本的预训练）¶



上节复习和补充

- 14.1. 词嵌入（word2vec）
- 14.2. 近似训练
- 14.3. 用于预训练词嵌入的数据集
- 14.4. 预训练word2vec
- 14.5. 全局向量的词嵌入（GloVe）
- 14.6. 子词嵌入

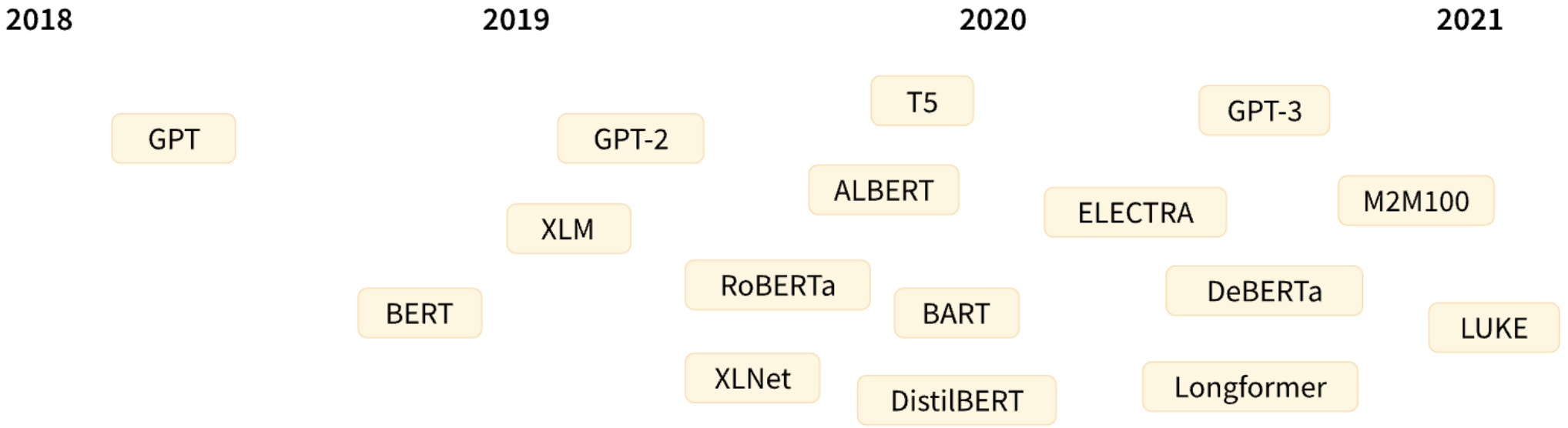
从上下文无关到上下文敏感

- 在“a crane is flying”（一只鹤在飞）
- “a crane driver came”（一名吊车司机来了）
- 上述文中，“crane”一词有完全不同的含义；
- 因此，同一个词可以根据上下文被赋予不同的表示。
- 流行的上下文敏感表示包括TagLM（language-model-augmented sequence tagger，语言模型增强的序列标记器）（Peters *et al.*, 2017）、CoVe（Context Vectors，上下文向量）（McCann *et al.*, 2017）和ELMo（Embeddings from Language Models，来自语言模型的嵌入）（Peters *et al.*, 2018）。
- 现有的监督模型是专门为给定的任务定制的。利用当时不同任务的不同最佳模型，添加ELMo改进了六种自然语言处理任务的技术水平：情感分析、自然语言推断、语义角色标注、共指消解、命名实体识别和问答。

从特定于任务到不可知任务

- 尽管ELMo显著改进了各种自然语言处理任务的解决方案，但每个解决方案仍然依赖于一个特定于任务的架构。然而，为每一个自然语言处理任务设计一个特定的架构实际上并不是一件容易的事。
- GPT（Generative Pre Training，生成式预训练）模型为上下文的敏感表示设计了通用的任务无关模型 (Radford *et al.*, 2018)。
- GPT建立在Transformer解码器的基础上，预训练了一个用于表示文本序列的语言模型。当将GPT应用于下游任务时，语言模型的输出将被送到一个附加的线性输出层，以预测任务的标签。
- 与ELMo冻结预训练模型的参数不同，GPT在下游任务的监督学习过程中对预训练Transformer解码器中的所有参数进行微调。
- GPT在自然语言推断、问答、句子相似性和分类等12项任务上进行了评估，并在对模型架构进行最小更改的情况下改善了其中9项任务的最新水平

基于 Transformer的不可知任务模型



- **2018 年 6 月**： GPT，第一个预训练的 Transformer 模型，用于各种 NLP 任务并获得极好的结果
- **2018 年 10 月**： BERT，另一个大型预训练模型，该模型旨在生成更好的句子摘要（下一章将详细介绍！）
- **2019 年 2 月**： GPT-2，GPT 的改进（并且更大）版本，由于道德问题没有立即公开发布
- **2019 年 10 月**： DistilBERT，BERT 的提炼版本，速度提高 60%，内存减轻 40%，但仍保留 BERT 97% 的性能
- **2019 年 10 月**： BART 和 T5，两个使用与原始 Transformer 模型原始架构的大型预训练模型（第一个这样做）
- **2020 年 5 月**，GPT-3，GPT-2 的更大版本，无需微调即可在各种任务上表现良好（称为零样本学习）

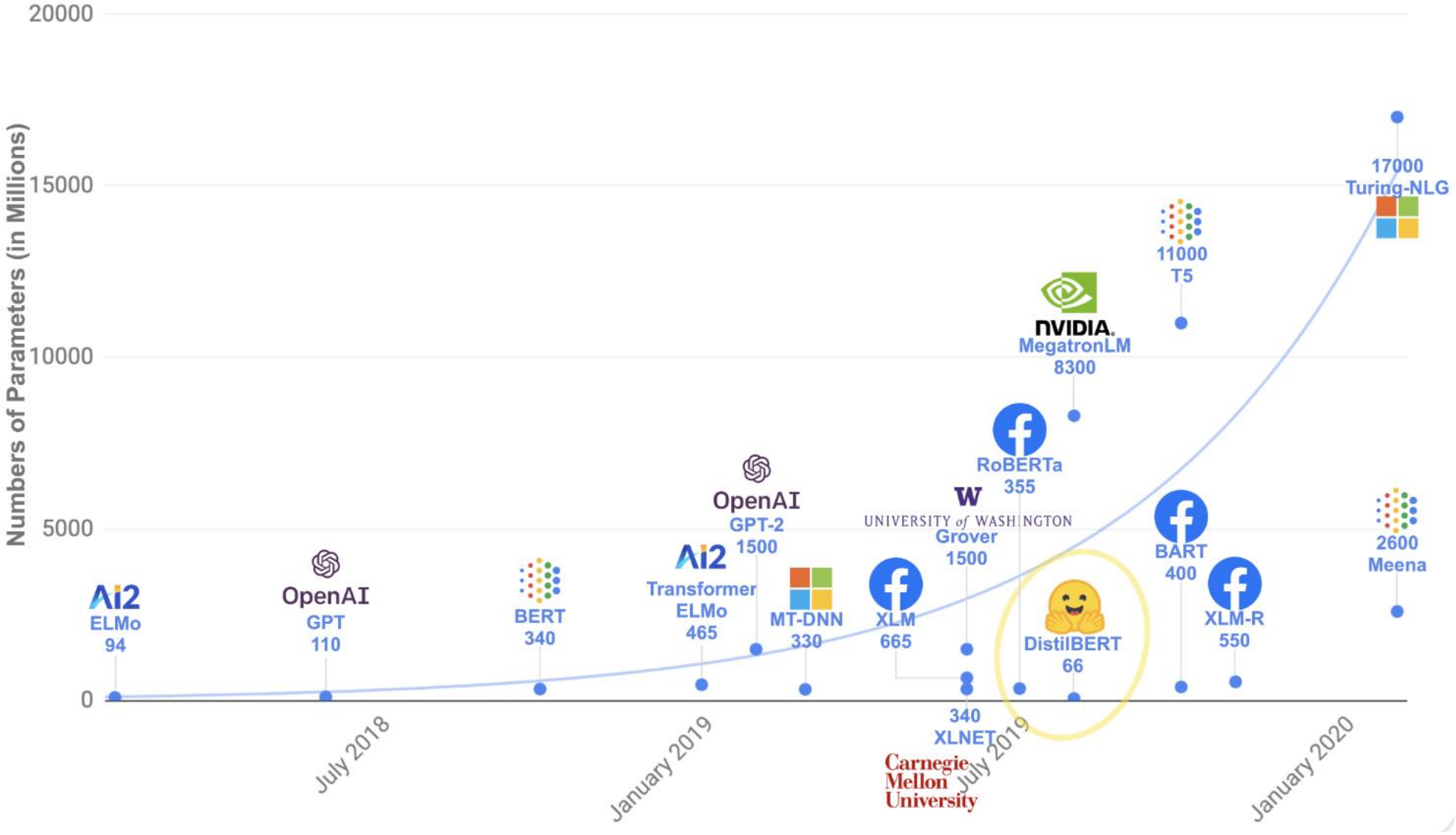
Transformer 自然语言模型分类

类型	代表模型	架构特点	训练目标	典型应用场景
GPT-like（自回归）	GPT-3/4、LLaMA	单向**解码器**架构（仅用 Transformer 解码器）	预测下一个 token（自回归生成），基于**单向语境**（仅前文）	文本生成（写作、对话、代码生成）
BERT-like（自动编码）	BERT、RoBERTa	双向**编码器**架构（仅用 Transformer 编码器）	随机掩码部分 token，预测原词（自动编码），基于**双向语境**（上下文）	文本理解（分类、NER、语义匹配）
BART/T5-like（序列到序列）	T5、BART、ChatGLM	**编码器 - 解码器**架构（同时用 Transformer 编解码器）	编码器处理输入文本，解码器生成目标文本（序列到序列），结合**双向 + 单向语境**	翻译、摘要、问答、生成（支持更复杂任务）

Transformer 自然语言模型联系与共性

1. **底层架构同源**：均基于 Transformer，依赖自注意力机制捕捉长距离依赖。
2. **预训练范式相同**：先在海量文本无监督预训练，再微调至下游任务（迁移学习）。
3. **相互借鉴优化**：
    1. 如 GPT-4 引入部分双向语境优化理解（类似 BERT 的 MLM 思想）；
    2. BERT 系列模型（如 ALBERT）借鉴 GPT 的层归一化等训练技巧；
    3. ChatGLM 等模型融合 BERT 编码器与 GPT 解码器，实现对话能力（类似 BART 架构）。
4. **趋势**：近年模型趋向融合双向与单向优势（如 Google PaLM 结合双向编码与自回归生成），通用型大模型（如 GPT-4、Claude 3）逐渐模糊三类边界，通过更复杂预训练目标实现 “理解 + 生成” 一体化。

Transformer 是大模型



DeepSeek R1 与 ChatGPT o3mini 的对比

对比项	ChatGPT 03 mini	DeepSeek R1
架构类型	密集 Transformer（全参数激活）	混合专家模型（MoE），每次激活2/16专家
总参数量	约2000亿	6710亿
单token激活参数	全部参数	370亿
训练计算量	约 120万 A100 小时	266.4万 H800 GPU 小时
上下文窗口	200K tokens（输出限制100K）	128K tokens
安全响应率	仅1.19%不安全响应	11.98% 不安全响应

BERT 参数量计算

- 只有编码器
- 嵌入层+n x Transformer 编码器
- 嵌入层=字典大小 x 隐藏单元个数

  = 30k x H
- Transformer 编码器= n x （ 4 x H^2+ 2x H x 4H）

  = n x 12 H^2

$Bert_{Base}$=30k x 768+ 12 x 12 x 768^2=108M(110M)

$Bert_{Large}$=30k x 1024+ 24 x 12 x 1024^2=332M(340M)

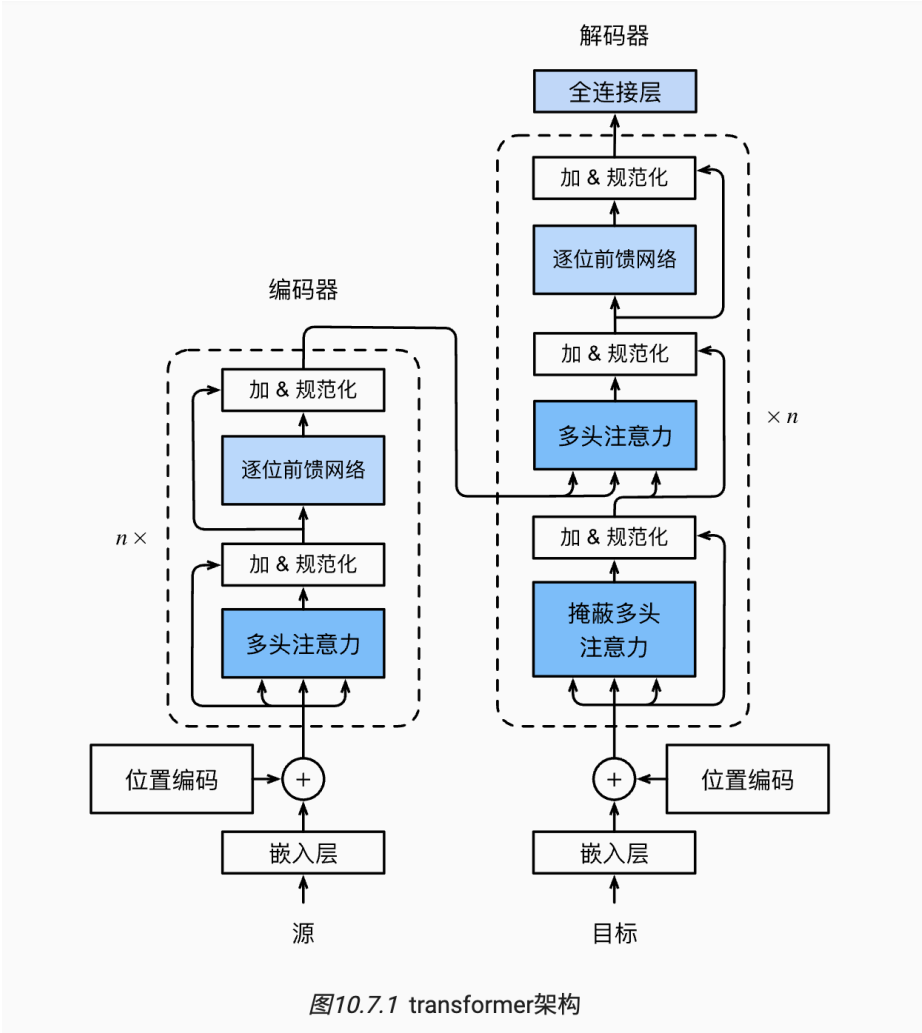


图10.7.1 transformer架构