Lecture17 Multitask Learning

The Limits of Single-task Learning

- 性能得到很大改善
- 只要数据集够大，就能获得很好的结果
- 如果想获得更通用的AI功能，需要对单一模型进行某种更持续的学习
- 大部分从随机开始，仅有一部分有预训练
    - 在NLP中，在word vectors上取得了很大的成功
    - 为什么不尝试预训练整个模型？

Why has weight & model sharing not happened as much in NLP?

- NLP需要很多不同的推理
- 需要短期和长期记忆
- NLP被分为中间任务和单独任务以取得进展
- 如果想要做更通用的，它必须是无监督的任务，且特征不监督，而NLP不会完全无监督，因为到最后语言需要监督

为不同的NLP任务考虑统一的多任务模型，需要这个多任务模型决定如何迁移知识而不是手工分配；需要模型了解自己如何进行域适应，以及如何分享权重。
统一的多任务模型可以①更容易适应新任务②简化部署到生产的时间③降低标准，让更多人解决新任务④潜在地转向持续学习

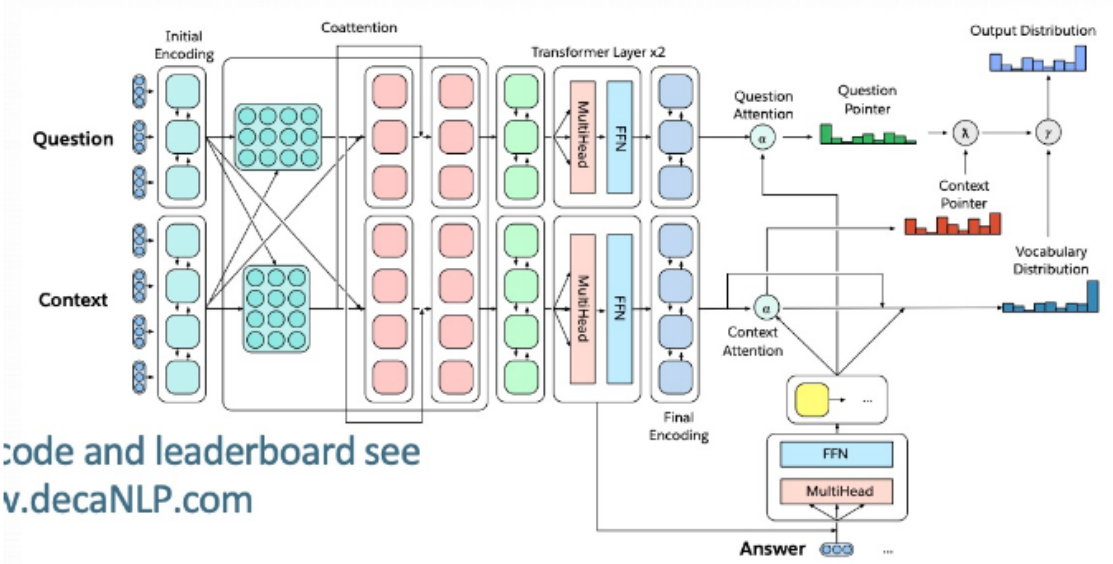
How to express many NLP tasks in the same framework?

- 序列标记。像NER或特定方面的情绪，或一个词是正面还是负面的，我们想要分类的特定背景
- 文本分类。
- Seq2seq。机器翻译，摘要，问答

Designing a model for decaNLP

- 没有任务特定的模块或参数
- 希望得到所需模型有能力根据不同任务进行内部调整，并自己做决定
- 应该为看不见的任务留下zero-shot推断的可能性

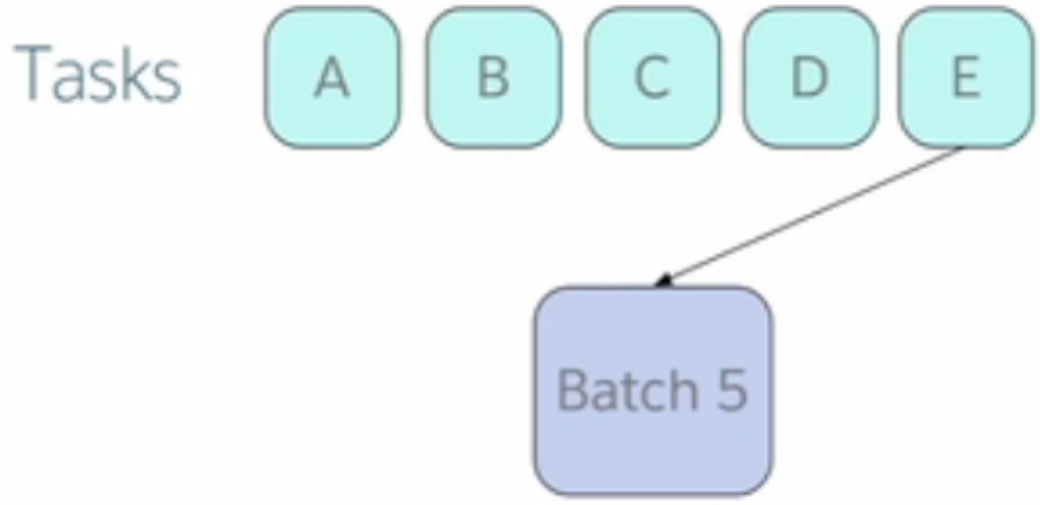
Multitask Question Answering Network (MQAN)



- 固定的 GloVe 词嵌入 + 字符级的 n-gram 嵌入 → Linear → Shared BiLSTM with skip connection
- 从一个序列到另一个序列的注意力总结，并通过跳过连接再次返回
- 分离BiLSTM以减少维数，两个变压器层，另一个BiLSTM
- 自回归解码器使用固定的 GloVe 和字符 n-gram 嵌入，两个变压器层和一个LSTM层来参加编码器最后三层的输出
- LSTM解码器状态用于计算上下文与问题中的被用作指针注意力分布问题
- 对上下文和问题的关注会影响两个**开关**：gamma决定是复制还是从外部词汇表中选择 lambda决定是从上下文还是在问题中复制

Training Strategies

- fully joint。从每一个不同的任务拿一个mini batch，只需要在那个任务上训练那个 mini
  batch



但是结果不是很好

- Anti-Curriculum Pre-training。从最困难的任务开始