

- 2022.03 - Now

机器学习工程师, 语音识别与语言技术 @ Zoom

Tech stack: Torch/Pytorch, ONNX, Kubernetes, Huggingface

• 实现并训练语音识别和文本标点模型, 从零开始构建丹麦语模型。测试数据集上的初始词错误率 (WER) 约为**8%**, 通过音素数据增强进一步降低了WER, 优于**MS Teams**的结果。初始大小写和标点的整体F1约为**70%**。

• 实现、测试、维护和评估推理系统的关键功能, 包括**解码、多头注意力 (MHA) 时间戳生成和后处理代码**。

• 在LAS/seq2seq模型上实现基于多头注意力 (MHA) 的时间对齐, 从而提供良好的单词级时间戳, 以满足多语言转录的业务需求。[为此申请了一项美国专利]

• 撰写meta-prompt, 使用开源LLM模型 (比如Mistral MoE 8x7B) 生成几百个对话场景prompt, 并结合LLM模型生成的数字序列的不同读法 (reading format)。将两者结合创建对话场景下的带数字文本, 将文本交由同事使用内部/微软tts服务生成音频, 从而得到测试集和部分训练数据。将训练数据对产线模型进行微调后, 在另外的内部的digits数据集上有少量提升 (Absolute digit WER 降低约0.4%)

• (进行中) 实验性质的对音频大模型的ASR能力进行内部评估, 比如Qwen Audio, Whisper

• 独立完成**Whisper**推理支持的实现, 优化和性能评估 (WER, RTF/latency/throughput), 使用内部in-house VAD (人声检测) 模型和开源的WhisperX, 相比OpenAI的实现取得了更高的吞吐, 并在多数测试集上实现了更低的wer。

• 与下游Web和基础设施团队合作, 简化离线转录系统架构和部署, 以支持**35+**种单语言模型在**10+**个不同区域的**5+**个不同下游服务中使用。

• 从头开始建立系统化的模型管理实践, 以应对复杂的业务和语言需求。

• 评估语音识别系统的性能, 寻找最合适的参数以减少实时因素或提高系统的吞吐量。

• 密切关注客户报告的产线错误, 并通过代码或数据质量方面的改进进行修复。

• 与硬件制造商 (如Nvidia) 合作, 评估专有推理系统的推理性能, 以降低推理的资源成本。

• 一些正在进行的前沿探索: 例如LLM重评分/生成式错误修正等。

• 2020.09 - 2021.12

数据科学家 @ Barclays

Tech stack: Spark / PySpark, Amazon Deep Java Library (DJL), Tensorflow / Keras, Pandas, Jupyter, 预训练Transformer / 似然比

• 公司地址匹配和实体匹配, 没有内部GPU和可用的标签数据。使用主动学习方法解决。从构建一些小数据集开始, 只使用外部数据并训练一个XGBoost树, 然后在边界上标记样本并以迭代方式微调BERT模型。在CPU集群上使用我自己从头开始构建的基于DJL的管道完成了600万内部配对样本的推断。该模型在嘈杂的测试数据集上实现了**94%的F1得分 (起初为89%)**。模型在我们的Spark集群上以分布式方式进行离线推断。对于600万配对样本, 运行时间不到1小时 (在拥有80个CPU的集群上)。

• 使用历史平均值和Informer模型 (Transformer的变体) 预测聚合的用户交易活动 (转账数和转账金额)。然后构建反事实评估, 以评估系统停机对银行造成的财务损失, 并找出系统可靠性的关键时期。

• 维护团队的Spark集群, 并通过结合DJL / PySpark UDF与模型建立分布式推理管道。与我的一位同事合作, 我们创建了一个团队范围的包, 只需4行代码即可启动Spark会话, 这显著减少了非分布式计算背景的同事使用Spark的开销。

• 2019.08 - 2020.09

Java 后端开发工程师 @ Barclays

Tech stack: Openshift (Kubernetes), GridGain, Maven, Gradle, Wiremock, Mockito, Spring Boot

• 端到端功能开发、测试 (单元、功能、性能)、部署 (CD)

• 为现有API添加缓存层, 以减少重复数据访问的延迟

• 将遗留代码迁移到内部Spring Boot模板, 并进行重构以提高代码可读性和性能

• 从头开始构建便捷的内部工具 (如git hooks) 和脚本 (python / bash), 以自动化软件开发过程

教育背景

2018 - 2019	硕士, 网络科学与大数据分析 @ 伦敦大学学院, 优异
2016 - 2018	学士, 互联网计算 @ 利物浦大学*, 一等荣誉
2014 - 2016	学士, 信息与计算机科学 @ 西交利物浦大学*

\*注:2+2模式 (前两年在中国苏州, 后两年在英国利物浦), 双学位。

个人项目

• 2024.06 -

关于病历数据在LLM上的微调与评估

(进行中) 使用数十万条病例数据, 在科室分类, 病历总结, 出院证明等任务上对不同的LLM foundation模型进行全量微调和(Q)Lora微调, 评估其效果。我们计划后续将数据开源。

技术分享

• "Accelerating Deep Learning on the JVM with Apache Spark and NVIDIA GPUs"

作者: Haoxuan Wang, Qin Lan [AWS], Carol McDonald [Nvidia]; 链接: [https://www.infoq.com/articles/deep-learning-apache-spark-nvidia-gpu/?itm\\_source=articles\\_about\\_ai-ml-data-eng&itm\\_medium=link&itm\\_campaign=ai-ml-data-eng](https://www.infoq.com/articles/deep-learning-apache-spark-nvidia-gpu/?itm_source=articles_about_ai-ml-data-eng&itm_medium=link&itm_campaign=ai-ml-data-eng)

早期项目

• 2019.06 - 2019.09

项目实习 (硕士学位论文) @ Astroscreen

社交媒体帖子语言来源识别 (推文和Gab帖子) 项目。 完成了Gab.com的语言 (帖子) 数据收集爬虫, 使用正则表达式进行数据预处理, 通过微调BERT和XLNet构建模型来分类这些数据的来源, 使用t-SNE可视化结果, 进行了"leave-one-hashtag-out"交叉验证, 并使用一些常见指标 (准确率、F1分数、混淆矩阵、马修斯相关系数) 评估模型。 微调后, XLNet在标签均衡的测试数据集上显示了86%的F1分数, 而在随机均衡的测试数据集上为97%。结果显示了使用模型进行来源检查的潜力, 也表明了避免数据泄漏的重要性。

• 2019.02 - 2019.03

将BERT和嵌入集成到CommonsenseQA挑战中

我们将Google BERT微调到CommonsenseQA挑战1.0 (每个问题有3个选项), 然后集成Conceptnet Numberbatch和ELMo嵌入, 试图提高模型性能。 该挑战包括一组需要人类常识知识的选择题。我们使用BERT + ELMo在验证集上达到了68.79%的准确率 (仅使用BERT: 67.47%; BERT + Numberbatch: 67.68%) 。