利用上下文 LLM 进行修订来改进命名实体转录

Viet Anh Trinh[1], *Xinlu He*[1], *Jacob Whitehill*[1]

[1]Computer Science Department, Worcester Polytechnic Institute, USA

vtrinh@wpi.edu, xhe4@wpi.edu, jrwhitehill@wpi.edu

Abstract

随着建模的最新进展和有监督训练数据量的增加，自动语音识别（ASR）系统在普通语音上取得了显著的性能。然而，最先进的 ASR 在命名实体上的字错误率（WER）仍然很高。由于命名实体通常是最关键的关键词，识别错误会影响所有下游应用，尤其是当 ASR 系统作为复杂系统的前端功能时。在本文中，我们引入了一种大型语言模型（LLM）修正机制，通过利用 LLM 的推理能力以及包含一组正确命名实体的局部上下文（例如，讲义），来修正 ASR 预测中的错误命名实体。最后，我们介绍了 NER-MIT-OpenCourseWare 数据集，该数据集包含来自 MIT 课程的 45 小时数据用于开发和测试。在该数据集上，我们提出的技术在命名实体上实现了最高 30 % 的相对 WER 降低。

Index Terms: contextual ASR, AI Agent

最近，通过扩大数据和模型的规模，语音识别性能显著提高。Whisper [1] 是一个在 765K 小时语音的大规模数据集上训练的模型，其表现非常出色，在标准的 Librispeech 干净语料库上实现了 2.7 % 的 WER，并在流行的语音数据集上平均达到 12.8 % 。然而，尽管在大量数据上进行了训练，Whisper（large-v3）在识别命名实体方面仍然表现不足。例如，在上下文 ASR 基准数据集 ConEC [2] 中，它对人名的 WER 为 28.9 % ，对组织名称的 WER 为 19.6 % 。

另一方面，人类具有根据知识和局部语境进行推理和联系不同命名实体的能力。例如，学生可以使用讲义幻灯片将讲座视频中的命名实体映射到幻灯片中的内容。由于命名实体涉及事件、人物姓名、产品等，这些通常依赖于局部语境并随时间变化，因此使用新的配对语音-文本数据重新训练模型的解决方案成本高昂，并且难以扩展。

在本文中，我们提出了一种创新的方法（见图 1）——受到 LLM 代理 [3] 的启发——该方法利用一个 LLM 以及语音和语义匹配组件来修正错误的命名实体。LLM 接收来自 ASR 的预测和局部上下文中过滤的信息，局部上下文包含发音类似的命名实体以及包括这些单词的句子。然后，LLM 根据这些信息修正 ASR 的预测。我们证明了我们提出的方法在不同 LLM 上工作有效，从闭源模型到开源模型，并且涵盖各种大小。此外，我们的工作在未来可以通过整合视觉信息或解决罕见词汇的错误率来扩展。此外，我们的方法是可解释的，因为 LLM 解释了它为什么进行某些修正，这不同于其他仅输出修正而不提供任何解释的注意力基础方法。最后，我们介绍了 MIT-OpenCourseWare NER 上下文数据集，其中包括语音-文本对以及局部上下文。

1. 相关工作

改进命名实体识别的一种常见方法是通过偏向于预定义词列表来调整 ASR。在 ASR 中调整的传统方法之一是浅融合 [4, 5, 6] 。在浅融合中，在推理过程中，ASR 的预测受到具有某种权重的上下文语言模型的影响。上下文语言模型通常表示为加权有限状态转换器 [7] 。浅融合是灵活的，因为它允许在推理时切换到不同的语言模型。然而，ASR 和语言模型（LM）以松散结合的方式组合，这可能导致次优的输出。此外，ASR 的性能对 ASR 和 LM 的权重组合非常敏感，并且不能很好地推广到不同的领域。

神经网络常用于编码上下文信息 [8, 9, 10, 11] 。这些上下文神经网络通常与 ASR 系统一同训练。早期的工作由 [8] 引入了一种偏置编码器到编码器-解码器 ASR 模型中 [12] 。在 [9] 中，神经网络用于编码偏置词，并通过注意力机制将上下文网络嵌入与基于转导器的 ASR 的音频编码器或标签编码器融合。也有研究结合了这些技术或探索替代方法。例如，在 [13] 中，作者结合了基于字典树的偏置、浅融合及上下文神经网络。在 [14] 中，作者提出使用一个连续积分-发放 (CIF) 模型进行上下文 ASR 的协同解码。最近，也有一些与上下文语音识别相关的使用大语言模型 (LLM) 的工作 [15] 。在 [15] 中，作者提出使用偏置提示和偏置融合网络。该研究还指出，增加偏置列表的长度会显著降低性能。我们提出的解决方案通过仅选择在语境中听起来与 ASR 预测相似的命名实体来解决这个问题。这种方法在缩短偏置列表长度的同时，仍然保留最相关的偏置词。在这篇论文中，我们提出了一种新颖的方法，利用大型语言模型（LLM）的推理能力来修正自动语音识别（ASR）的预测，该方法利用了语音和可用上下文中命名实体的语义和语音信息。该方法包含几个步骤。首先，(1) 使用 ASR 模型（例如，Whisper）转录语音。接下来，(2) 使用现成的命名实体识别工具从 ASR 预测结果和上下文文档中提取命名实体。上下文可以是与语音相关的任何文本文档。例如，在转录讲座时，上下文可能包括幻灯片或阅读清单。由于上下文中词汇量较大，(3) 如果上下文中的命名实体与 ASR 预测中的命名实体类型相同（例如，人、组织）且发音相似，则从上下文中选择命名实体。最后 (4)，大型语言模型分析 ASR 预测及其命名实体，以及上下文中的命名实体及包含它们的句子。它利用这些信息和详细的指令提示（见图 3）来生成修正后的预测。由于上下文自动语音识别（ASR）数据集的数量非常有限，且通常针对工业环境或特定需求（如财报电话会议）设计，我们在自己新引入的数据集上评估了我们提出的方法。相比之下，我们更关注教育和科学领域。测试集是 2013 年秋季提供的关于动物行为的一个本科课程，从 MIT 开放课程资料中提取。我们使用 BeautifulSoup 下载了该课程的音频、视频和讲义幻灯片，最终获得了总计 17 小时的 25 个讲座音频录音。测试集标签包含约 121K 字，而上下文包含 46K 字。真实转录（由 OpenCourseWare 提供）包含 1.2K 个命名实体词，其中约 66% 出现在幻灯片中。图 2 中提供了讲义幻灯片的示例。

我们的开发集是一个来自麻省理工学院开放课程的关于大脑结构的本科学课程，开设于 2014 年春季。这个课程有 35 节课，但为了加快执行速度，我们只使用了前 3K 个话语。开发集的标签包含大约 32K 个单词，而上下文包

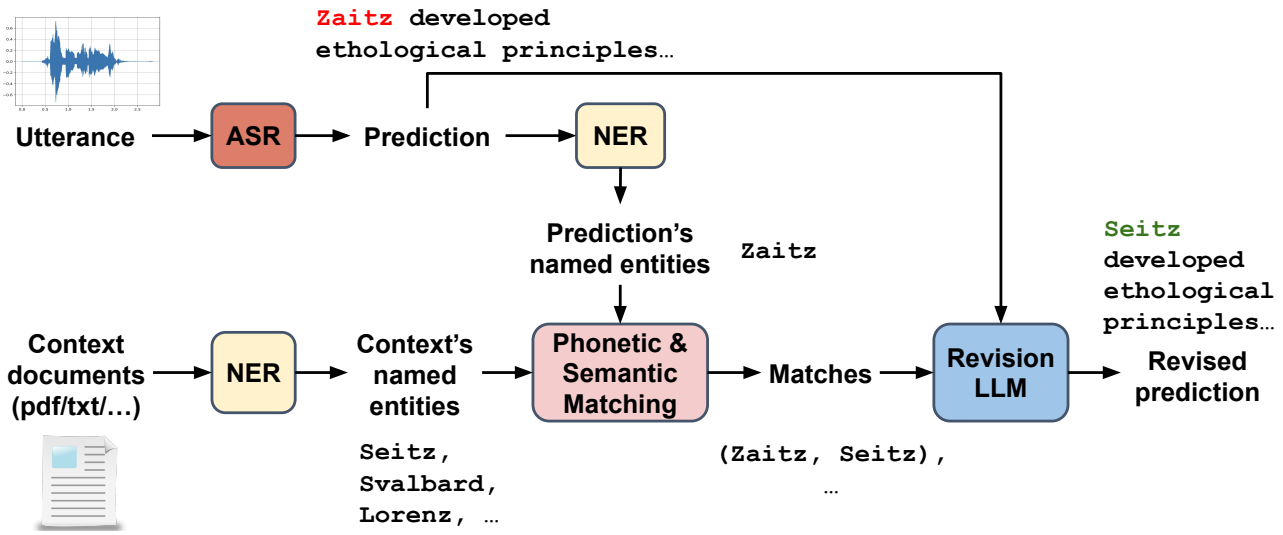

Figure 1: 提出将语音和语义信息结合起来修正 *ASR* 预测中的命名实体的方法。

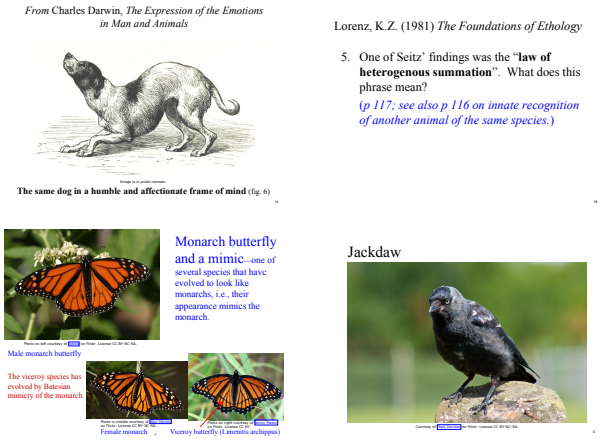

Figure 2: 课程幻灯片示例。

括来自六张讲座幻灯片的 14K 个单词。开发集中有 0.2K 个命名实体词，大约 60 % 的这些词出现在幻灯片中。

我们使用 pdfplumber [16] 将真实转录和课件幻灯片转换为文本文件，并使用 spacy `en_core_web_lg` 模型通过去除不需要的字符来清理文本，并将真实转录分割成多个句子。幻灯片包含图像、数学公式和其他元素，使它们变得嘈杂。我们使用 pydub [17] 从视频 (mp4) 中提取音频 (flac) 并将音频转换为采样率为 16,000 Hz 的单声道音频。由于每个音频文件的时长为 30 到 60 分钟，我们利用 fairseq [18] 将长音频分割成对应于 spacy 模型识别的句子的多个语音段。

2. 实验

使用该数据集 (第 ?? 节)，我们进行了实验以评估所提方法在改善命名实体的自动语音识别 (ASR) 转录准确性方面的益处，同时希望保持非命名实体的转录准确性不变。由于我们的方法侧重于在将信息提供给 LLM 之前，利用语义和语音线索筛选上下文中的相关信息，我们与两种强控制条件进行了比较：(1) 包含给定语音段对应的整个文档的 LLM 提示（不仅仅是筛选出的相关实体）；(2) 先使用 LLM 总结上下文然后将总结提供给修订 LLM。

对于第一阶段的 ASR，我们使用 Whisper large-v3。我们选择 Flair [19]，具体来说是 flair/ner-english-ontonotes-large 模型，用于命名实体识别 (NER)，因为它在标准 NER 基准上的速度和高准确性。最后，为了检测两个单词是否听起来相似，我们使用 Double Metaphone [20]，这是一种流行且强大的基于规则的语音算法，可以将单词转换为语音代码，使其在检测相似音的单词和执行姓名搜索时非常有用 [21, 22]。它不仅可以用于英语，也可以用于低资源语言 [23, 24]。

根据另一个基准测试 ASR 上下文数据集 [2]，我们关注八种特定的命名实体类型：PERSON、ORG、GPE、LOC、PRODUCT、EVENT、NORP 和 FAC。我们手动验证了测试集中 500 个句子中的命名实体，并测量基于集合的评估召回率，忽略同一命名实体在一个句子中的多次出现，因为 LLM 会替换句子中所有相同命名实体的实例。Flair 的精准率是 96 %，召回率是 86 %。由于在人群中手动标记命名实体代价高昂，而 Flair 是一个具有合理召回率的稳健 NER 模型，我们将 Flair 的命名实体结果视为基准，用于评估 Flair 检测到的命名实体上的 WER。

对于修订 LLM，我们选择 Llama3-70b-8192 [25]（通过 Groq API）、GPT-4o-mini API、GPT-4o API [26] 和 Meta-Llama-3.1-70B（使用 bitsandbytes 量化），在单个 A100 GPU（80GB）上本地运行，并使用 vLLM [27]。选用 Langchain [28] 作为实验框架，因为它提供了在不同 LLM 提供商之间切换的灵活性。我们从 [29] 和 OpenAI 提示工程指南 [30] 中汲取灵感来手动设计提示。我们的提示（见图 3）旨在确保 LLM 修改实体名称而不改变其他词语，并应用推理。由于 LLM 输出除了修订预测之外的额外文本，我们指导 LLM 使用一种特殊格式进行修订预测，以 << @ 开始并以 @ >> 结束。这种方法使我们能够轻松将修订后的预测与额外文本分开，并在所有 LLM 中证明其是稳健的。

对于摘要任务（在控制条件 2 下），我们使用 GPT-4o-mini 与以下提示："简明总结以下内容，同时确保所有命名实体得到保留。为每个命名实体包含一个句子，仅使用字母和数字。内容：{ context }"。

如表 1 所示，我们提出的方法与原始的 Whisper 预测和两个控制条件相比，无论使用闭源还是开源的 LLM，也

Table 1: 对比各种方法下的命名实体（*NE*）词错误率（*WER*）（%）和非命名实体（*NNE*）词错误率（*WER*）（%）。

Method	LLM	NE	NNE	LLM	NE	NNE	LLM	NE	NNE
Whisper	None	32.3	7.0	None	32.3	7.0	None	32.3	7.0
Full context	GPT-4o-mini	43.4	7.1	GPT-4o	30.6	7.0	Llama-3.1-70B	32.6	7.2
Context summary	GPT-4o-mini	38.6	7.1	GPT-4o	54.2	7.1	Llama-3.1-70B	47.4	8.2
Proposed method	GPT-4o-mini	22.7	7.0	GPT-4o	22.9	7.0	Llama-3.1-70B	24.8	7.2

Speech recognition may incorrectly capture named entities due to similar-sounding words or uncommon entities in training data. The named entities and their probabilities are provided below as "Named entities probability" in the format: entity : probability. Review the prediction's named entities. For each entity whose probability is below asr_confidence_threshold, or missing, please complete the following steps:

- 检查上下文来确定相似发音的实体指代什么，并分析句子以理解 *ASR* 预测的实体指代什么。检查是否（*1*）两者都指代同一事物，（*2*）差异可能是由于 *ASR* 转录错误，（*3*）替换提高了准确性，保留了原句的意思，并自然地适合句子。如果这三个标准都满足，那么用相似发音的实体替换 *ASR* 预测的实体。
- 确保在评估或替换时将每个命名实体视为一个整体单元。不要在多词实体中单独修改个别词语。
- 如果不存在合适的相似发音实体，则保持实体不变。
- 不要删除上下文中缺失的实体中的任何单词。
- 保持所有其他词汇、标点符号和格式不变。

The context includes similar-sounding entities and sentences containing these named entities. The revised prediction should start with << @ and end with @ >> Context: { context }
Speech recognition prediction: { preds }
Named entities probability: { ent_prob_dict }

Figure 3: 给大型语言模型的 *ASR* 修订提示。在消融分析中，我们仅保留提示中的斜体部分。

无论 LLM 的规模如何，都表现出了显著的性能提升。使用 GPT-4o-mini 作为 LLM 时，当使用完整上下文（控制条件 1）时，WER 从 32.3 % 增加到 43.4 % 。这一增加可能归因于长的提示输入导致 LLM 出现幻觉。控制条件 2 使用完整上下文的摘要，但事实证明同样无效，因为 WER 仍然从 32.3 % 增加到 38.6 % 。然而，我们提出的方法则将 WER 降低到 22.7 % （相当于 30 % 的相对 WER 降低），同时将非实体 WER 保持在 7 % 。结果表明，结合 LLM 的推理能力与额外语音工具提供的知识是有利的。这突显出即使在大规模文本数据上训练的 LLM 可能也没有足够的语音或声学信息知识。图 4 展示了我们的提出方法如何有效修正各种 ASR 系统错误的定性例子。

我们提出的方法在不同的 LLM 模型中也展示了其有效性，比如大型的、最先进的 GPT-4o 和强大的开源 Llama-3.1-70B，如表 1 所示。对于像 GPT-4o 这样的强大模型，利用完整的上下文可使 WER 略微降低，从 32.3%

Table 2: 利用我们提出的方法中的 *GPT-4o-mini*，对比不同语音识别系统中命名实体（*NE*）*WER*（%）和非命名实体（*NNE*）*WER*（%）。

ASR	Params	Before revision		After revision	
		NE	NNE	NE	NNE
Whisper small	244 M	39.3	9.4	29.5	9.4
Whisper medium	769 M	36.0	8.0	26.8	8.0
Whisper large v3	1.54 B	32.3	7.0	22.7	7.0
Canary-1B	1B	36.6	8.3	26.8	8.3

Ground truth:
now this is a difficult concept but let me try to go through it anyway because seitz was very important in working out some of the principles of ethological investigation and the principles of ethology that lorenz was talking about
Canary 1B:
now this is a difficult concept ...zeitz ...lawrence ...
Whisper:
now this is a difficult concept...zaitz ...lawrence ...
Canary and Whisper after revision:
now this is a difficult concept ...seitz ...lorenz ...

Figure 4: 一个例子展示了使用我们提出的方法对各种 *ASR* 系统进行修正的结果，包括真实情况、原始预测和修正后的预测（使用 *GPT-4o-mini*）。省略号表示为简洁起见而省略的正确转录的单词。

Table 3: 命名实体 *WER*（%）和非命名实体 *WER* 评估使用完整提示或简短提示

Method	Revision LLM	NE (%)	NNE
Full prompt	GPT-4o-mini	22.7	7.0
Short prompt	GPT-4o-mini	22.7	7.0
Full prompt	Llama-3.1-70B	24.8	7.2
Short prompt	Llama-3.1-70B	32.3	7.0

降至 30.6%，如表 1 所示。然而，使用我们提出的方法，WER 显著降低至 22.9%。

为了评估我们提出的方法在不同模型规模和语音识别系统中的有效性，我们使用 Whisper Small、Whisper Medium 和 Nemo Canary-1B [31] 进行实验，这是另一个最先进的 ASR 模型。结果如表 2 所示。正如表 2 所示，我们的方法在不同的模型规模和架构上表现出良好的泛化能力。对于 ASR 模型 Canary-1B，命名实体 WER 从 36.6 % 降低到 26.8 %。同样，命名实体 WER 在 Whisper small 中从 39.3 % 降低到 29.5 %，在 Whisper medium 中从 36.0 % 降低到 26.8 %。

在运行时间成本方面，提取每个话语中的命名实体的时间为 0.013 秒（在 9427 个测试话语中取平均）。使用 GPT-4o-mini 修正 ASR 预测的时间为 1.2 秒，平均于 750 个包含命名实体的测试话语。

2.1. 消融研究

我们尝试仅保留提示中与输入、输出和一般指令相关的最关键部分，如图 3 中斜体部分所示。该简化提示的结果汇报在表 3 中。如图所示，完整提示和简短提示在 GPT-4o-mini 上表现良好，而开源的 Llama-3.1-70B 在完整提示下实现了更好的性能，24.8 %，相比简短提示下的 32.3 %。这表明在此情况下，付费 API GPT-4o-mini 可能比开源 Llama-3.1-70B 更加强大，推理能力更好。开源模型似乎需要更多详细指令才能取得良好结果，而 GPT-4o-mini 可以在较少指令情况下表现良好。

最后，我们还进行了一项不需要 LLM 的实验，采用一种简单的策略，在 ASR 预测中随机用上下文中发音相似的命名实体替换掉命名实体。在命名实体上，使用随机替换的 WER 是 25.3 %，而使用 GPT-4o-mini 的提出方法是 22.7 %。这可能是因为较简单的基线在不理解上下文的情况下替换命名实体，这在上下文数据较为嘈杂时尤其成为问题。例如，对于一个其真实转录为 “one of those students became very well known margaret mead ” 的话语，将命名实体替换为一个随机选择的发音相似的实体会导致 “...margaret mit ”。然而，使用我们提出的方法（使用 GPT-4o-mini），ASR 的转录被正确修改为 “...mead ”。在本文中，我们介绍了一种新方法来利用 LLM 修正语音识别系统的预测。具体来说，我们采用一个语音和语义匹配组件，从上下文中选择与 ASR 预测中最相关的命名实体。我们将我们提出的方法与使用完整上下文或上下文摘要的强对照进行比较，并证明我们的方法在新引入的测试集上可实现最多 30 % 的相对 WER 降低。

在我们的工作中，我们主要关注命名实体，因为它们的定义是标准化的，并且 NER 的工具已广泛可用。然而，随着稀有词检测模型的发展，该方法可以扩展到修订稀有词。此外，该方法还可以扩展以整合视觉信息，例如图像描述或光学字符识别（OCR），以便更好地利用可用的上下文文档。

3. References

[1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *International conference on machine learning*. PMLR, 2023, pp. 28 492–28 518.

[2] R. Huang, M. Yarmohammadi, J. Trmal, J. Liu, D. Raj, L. P. Garcia, A. V. Ivanov, P. Ehlen, M. Yu, D. Povey *et al.*, “Conec: Earnings call dataset with real-world contexts for benchmarking contextual speech recognition,” in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 3700–3706.

[3] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, “A survey on large language model based autonomous agents,” *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[4] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 1–5828.

[5] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang *et al.*, “Streaming end-to-end speech recognition for mobile devices,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6381–6385.

[6] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, “Shallow-fusion end-to-end contextual biasing.” in *Interspeech*, 2019, pp. 1418–1422.

[7] M. Mohri, F. Pereira, and M. Riley, “Weighted finite-state transducers in speech recognition,” *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[8] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, “Deep context: end-to-end contextual speech recognition,” in *2018 IEEE spoken language technology workshop (SLT)*. IEEE, 2018, pp. 418–425.

[9] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, “Context-aware transformer transducer for speech recognition,” in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 503–510.

[10] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, “Contextual adapters for personalized speech recognition in neural transducers,” in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 8537–8541.

[11] X. Gong, Y. Wu, J. Li, S. Liu, R. Zhao, X. Chen, and Y. Qian, “Advanced long-content speech recognition with factorized neural transducer,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2024.

[12] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, “Listen, attend and spell,” *arXiv preprint arXiv:1508.01211*, 2015.

[13] D. Le, M. Jain, G. Keren, S. Kim, Y. Shi, J. Mahadeokar, J. Chan, Y. Shangguan, C. Fuegen, O. Kalinli *et al.*, “Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion,” *Interspeech*, 2021.

[14] M. Han, L. Dong, S. Zhou, and B. Xu, “Cif-based collaborative decoding for end-to-end contextual speech recognition,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6528–6532.

[15] X. Gong, A. Lv, Z. Wang, and Y. Qian, "Contextual biasing speech recognition in speech-enhanced large language model," *Proc. Interspeech. ISCA*, pp. 257–261, 2024.

[16] [Online]. Available: https://github.com/jsvine/pdfplumber

[17] [Online]. Available: https://github.com/jiaaro/pydub

[18] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[19] A. Akbik, T. Bergmann, D. Blythe, K. Rasul, S. Schweter, and R. Vollgraf, "FLAIR: An easy-to-use framework for state-of-the-art NLP," in *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 2019, pp. 54–59.

[20] L. Philips, "The double metaphone search algorithm," *C/C++ Users Journal*, 2000.

[21] P. Blair and K. Bar, "Jrc-names-retrieval: A standardized benchmark for name search," in *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 2024, pp. 9589–9603.

[22] F. Trias, H. Wang, S. Jaume, and S. Idreos, "Named entity recognition in historic legal text: A transformer and state machine ensemble method," in *Proceedings of the Natural Legal Language Processing Workshop 2021*, 2021, pp. 172–179.

[23] N. Luitel, N. Bekoju, A. K. Sah, and S. Shakya, "Contextual spelling correction with language model for low-resource setting," in *2024 International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2024, pp. 582–589.

[24] A. Kunchukuttan, S. Jain, and R. Kejriwal, "A large-scale evaluation of neural machine transliteration for indic languages," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 3469–3475.

[25] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[26] OpenAI, "Open ai. hello gpt-4o," 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/

[27] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica, "Efficient memory management for large language model serving with pagedattention," in *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[28] [Online]. Available: https://github.com/langchain-ai/langchain

[29] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[30] [Online]. Available: https://platform.openai.com/docs/guides/prompt-engineering

[31] O. Kuchaiev, J. Li, H. Nguyen, O. Hrinchuk, R. Leary, B. Ginsburg, S. Kriman, S. Beliaev, V. Lavrukhin, J. Cook *et al.*, "Nemo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.