CS-FLEURS：一个大规模多语言和代码切换的语音数据集

Brian Yan[1], Injy Hamed[2], Shuichiro Shimizu[3], Vasista Lodagala[4], William Chen[1], Olga Iakovenko[5], Bashar Talafha[6], Amir Hussein[7], Alexander Polok[8], Kalvin Chang[1], Dominik Klement[8], Sara Althubaiti[4], Puyuan Peng[9], Matthew Wiesner[7], Thamar Solorio[2], Ahmed Ali[4], Sanjeev Khudanpur[7], Shinji Watanabe[1], Chih-Chen Chen, Zhen Wu[1], Karim Benharrak[9], Anuj Diwan[9], Samuele Cornell[1], Eunjung Yeo[9], Kwanghee Choi[9], Carlos Carvalho, Karen Rosero[1]

[1]Carnegie Mellon University, [2]Mohamed bin Zayed University of Artificial Intelligence, [3]Kyoto University, [4]Humain, [5]University of Sheffield, [6]Independent, [7]Johns Hopkins University, [8]Brno University of Technology, [9]University of Texas at Austin

Abstract

我们提出了 CS-FLEURS，一个新的数据集，用于开发和评估高资源语言之外的混杂语音识别和翻译系统。CS-FLEURS 包含 4 个测试集，涵盖总计 52 种语言中的 113 种独特的混杂语言对：1）一个由真实声音阅读合成生成的混杂句子组成的 14 种 X-英语语言对集合，2）一个使用生成文本到语音技术的 16 种 X-英语语言对集合，3）一个使用生成文本到语音技术的 60 种{阿拉伯语、普通话、印地语、西班牙语}-X 语言对集合，以及 4）一个使用拼接文本到语音技术的 45 种低资源 X-英语语言对测试集。除了这四个测试集之外，CS-FLEURS 还提供了一个包含 16 种 X-英语语言对的生成文本到语音数据的训练集，时长共计 128 小时。我们的希望是，CS-FLEURS 有助于拓宽未来混杂语音研究的范围。数据集链接：`https://huggingface.co/datasets/byan/cs-fleurs`。

Index Terms: 代码切换，代码切换语音

1. 介绍

在当前大规模多语言语音处理时代，从业者正在数百种语言上开发和评估系统 [1, 2, 3, 4, 5]。这种日益增长的全球影响很大程度上归功于数据集构建工作，这些工作能够进行原始未转录语音的预训练 [6, 7] 和转录或伪标记语音的监督训练 [8, 9, 10, 11]，以及在多种语言上对标准化评估集进行广泛基准测试 [12, 13]。然而，这些数据资源主要是单语种的 utterances 集合——我们目前缺乏广泛的代码切换语音数据，其中 utterances 包含多种语言。

尽管在多语言社区中普遍存在会话中的语言混用，语言技术大规模缺乏 CS 数据可以从广泛的说话者中获取不同语言混用的语音需要自然地将多种语言混合使用的双语或多语种说话者 [14]。此外，语言混用往往零星且不可预测，使其难以在大型、精心策划的数据集中捕捉到。即使有可用数据，标准化跨语言对的数据也是一项挑战。这些因素使得创建单一的广泛代表性的语言混用语音数据集变得困难；相反，实践者专注于为他们特别感兴趣的特定社区构建专门语料库 [15, 16, 17, 18, 19, 20]。

在这项工作中，我们采取了一种替代方法：而不是依赖于收集自然对话中的语言混用语音，我们使用了合成生成的语言混用文本上的朗读语音和合成语音的组合。虽然我们的方法没有捕捉到对话中出现的语言混用语音，但我们能够使用（1）**标准代码切换模式**，（2）一个**标准文本域**，以及（3）跨语言对的一个**标准音频域**。通过在构建数据集时控制这些自然发生的变异，我们就可以问：*模型在不同语言对的代码切换语音上的表现如何?*

我们介绍了 CS-FLEURS：一个大规模多语言和混用代码的 ASR 及 ST 数据集，包含 52 种语言和 113 个独特的混用代码组合，分为三个子集：

- CS-花卉读取：14 个 X-英语配对，阅读语音
- CS-花卉-XTTS：跨越 17 种语言的 76 对，生成式 TTS
- CS-花卉-MMS：45 对英语词汇，串接式语音合成技术

据我们所知，在涵盖的语言种类以及独特混用代码组合数量方面，这是最广泛的单一混用语音数

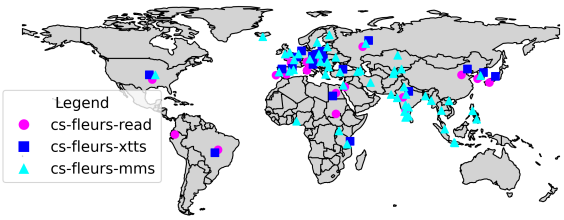

图 1: 展示了 *CS-FLEURS* 涵盖的 *52* 种语言在全球的分布情况。

表 1: *CS-FLEURS* 与其他代码切换语音语料库的对比。

Dataset	Langs	CS-Pairs	Domain	Speech Type	Hours
ArzEn [20]	2	1	Conversation	Spontaneous	12
Bangor Miami [15]	2	1	Conversation	Spontaneous	35
SEAME [16]	2	1	Conversation	Spontaneous	192
ESCWA [18]	3	2	Conversation	Spontaneous	3
MUCS [19]	3	2	Lecture	Structured	160
Soapies [17]	5	4	TV	Scripted	14
CS-花卉	**52**	**113**	**维基百科**	**读/合成**	**294**

据集合。

该数据集支持在 113 种不同的语言对上进行模型基准测试，并且可以在 16 个 X-英语语言对上训练模型。图 1 展示了语言覆盖范围，表 1 总结了该数据集并与其它代码切换语料库进行了比较。除了数据集外，这项工作还描述了几项实证发现。首先，我们展示 Whisper[2] 在转录代码切换语音方面存在困难，特别是在使用不同脚本的语言对上。然而，我们也展示了 Whisper 相对稳健地将代码切换语音翻译成英语，这表明代码切换 ST 比 ASR 更容易。最后，我们展示了合成代码切换语音数据是一种有效的训练数据增强形式，可以提高模型在已知和未知语言对上的性能。

2. 数据集

CS-FLEURS 建立在先前的一系列工作的基础上。FLoRes-101 数据集 [21] 包含了从英语维基百科翻译成 101 种语言的 3001 句句子，由人工翻译者完成。接着，FLEURS 数据集 [13] 选取了来自 FLoReS-101 的 2009 句句子，并用每种语言的三位发音人录制了读音，在 102 种语言中进行了记录。现在，CS-FLEURS 使用 FLEURS 中的 2009 条话语生成跨越 113 种语言对的代码切换文本和语音，保持与 FLEURS 相同的训练/开发/测试划分。

表 2: 涵盖的 *113* 种语言混用组合。

CS-Pairs	Matrix	Embedded	Read	XTTS	MMS
7	都给了 分 了 为了 俄 西班牙	英语	✓	✓	✓
5	这些是国家代码的缩写，无需翻译: ces cmn ita jpn kor	英语	✓	✓	
2	slk 电话	英	✓		
3	荷兰语 波兰语 土耳其语	英文		✓	✓
1	混	英文		✓	
60	ara cmn hin spa	阿拉伯 法语 德语 意大利语 葡萄牙语 波兰语 土耳其语 俄语 荷兰语 捷克语 西班牙语 普通话 日语 缅语 韩语 印地语		✓	
35	ben bul cat ceb cym ell fin guj heb hun ind isl jav kan kaz kir lav lug mal mar mya pan ron swe swh tam tel tgk tgl tha ukr urd uzb vor zho	eng			✓

表 3: *CS-FLEURS* 的汇总统计。

Statistic	阅读 Test	XTTS Train	XTTS Dev	XTTS Test1	XTTS Test2	MMS Test
Duration (hours)	17	128	15	36	42	56
Tokens (words)	128k	889k	105k	257k	300k	315k
Matrix Langs	14	16	16	16	4	45
Embedded Langs	1	1	1	1	15	1
Total CS Pairs	14	16	16	16	60	45
Same-Script Pairs	7	10	10	10	9	22
Distinct-Script Pairs	7	6	6	6	51	23

我们遵循矩阵语言框架模型 [22]；对于每一对代码切换，一种语言，称为矩阵，提供语法结构，而另一种语言，称为嵌入，提供插入句子中的单词或形态单位。我们将语言对称为矩阵-嵌入；例如，汉语-英语指的是带有英语单词嵌入的汉语矩阵句子。表 2 展示了 CS-FLEURS 涵盖的代码切换语言对在 CS-花卉读取、CS-花卉-XTTS 和 CS-花卉-MMS 子集中的细分。表 3 提供了额外的汇总统计数据。

2.1. 收集跨越 14 种语言配对的阅读语音

CS-花卉读取包含来自 14 种-X 语言对的双语发言者朗读的内容（每个语言对有 1-4 名发言者，整体男女比例为 2:1）。CS-花卉读取使用 FLEURS 的测试集生成，并且旨在用于模型基准测试。因此，该集合已经过人工验证。在语言覆盖方面，构建此集合的限制因素是双语朗读者的可用性。最后，在这个集合中我们将 14 种非英语语言视为矩阵语言，将英语视为嵌入语言。

2.1.1. 生成准确且流畅的代码切换文本

我们使用大型语言模型（LLM）作为生成代码切换文本的主干，用于 CS-花卉读取。具体来说，通过利用来自 FLEURS 测试集的 X-英语平行单语句子，我们提示 GPT-4o[23] 以三种不同的方式生成等效的代码切换句：

- *GPT-基础*：仅喂入选配的单语句子
- *GPT-EC*：输入成对的单语句子和一组有效的英文单词进行嵌入，这些单词由等价约束理论 [24] 确定，以不违反两种语言的句法规则，如在 [25] 中实现的那样
- *GPT-预测*：输入成对的单语句子和一组有效的英文单词以进行嵌入，这组单词由一个预测模型 [26] 确定，该模型给定一个英语句子后，能够预测出可以在相应代码切换句中嵌入的可能性较高的英文单词。

表 4 显示了一个给定句子对的完整提示示例。*GPT-EC* 和 *GPT-预测*提示包括用于嵌入到矩阵语言中的英语关键词。此外，为了鼓励形态代码切换，即一种以上的语言出现在同一单词内，我们用表中所示的情境示例来提示 GPT-4o。14 种语言中有 10 种被如此提示。[1]

生成的代沟不保证是 (1) 混合语言的，(2) 准确无误的，并保留原始单语句子的意义，或 (3) 流畅自然，看似由人类产生的。我们请双语读者拒绝那些不是混合语言或不准确的句子。句子不会因为流畅性而被拒绝，但我们收集 0 到 2 的评分，

表 4: 提示由三个主要部分组成：*(1)* 任务描述及要合成的配对单语句（用黑色标出）；*(2)* 在 *GPT-EC* 和 *GPT-预测*情况下的有效英文单词集（用蓝色标出）；以及 *(3)* 在这种类型切换常见语言中的形态学代码切换示例（用紫色标出）。

你是一名双语西班牙语-英语使用者，你将帮助把西班牙语和英语句子翻译成一种混码句。这种混码句应包含两种语言的单词，每个单词都用正确的语言书写。我们将提供给你西班牙语和英语句子除了需要出现在代码混合句子中的一些英文关键词之外 。
你需要在适当的情况下生成形态学代码切换以实现更好的流畅度。以下是一个示例：
我的朋友在被透露电影结局时生气了。
我的朋友在有人剧透电影结局时惊慌失措。英语关键词：惊慌失措，宠坏
一段流畅的代码混合句子是：Mi amigo se frickeo cuando le spoilearon el final de la película.
请生成以下句子的代码混排形式：
西班牙语句子：{Spanish_sentence}
英语句子：{English_sentence}
{英文关键词：{English_keywords}}
仅提供代码混排后的句子，不包含解释或额外文本。

表 5: 人类对跨越 *14* 种 *X*-英语组合的 *LLM* 生成的代码混用文本进行验证。*CMI*=代码混用指数。

		流畅性				
Prompt	Reject(%)	0(%)	1(%)	2(%)	Avg	CMI
GPT-基础	**8.54**	31.66	25.44	34.36	1.03	25.50
GPT-EC	12.15	44.71	21.22	21.92	0.74	30.98
GPT-预测	11.76	19.26	25.25	43.73	**1.28**	21.05

其中 0 表示不自然，1 表示某种程度上的自然，2 表示完美自然。

如表 5 所示，不同提示的拒绝率相当一致。在流畅性方面，*GPT-预测*得分最高，而 *GPT-EC* 则最低；这一结果与通过代码混用指数（CMI）[27] 测量到的切换频率相关联。我们还注意到，*GPT-EC* 以高频率嵌入功能词，这使得双语读者感到相当不自然。另一方面，*GPT-预测*主要嵌入内容词，导致超过 80%的生成被评定为某种程度上或完全自然。

2.1.2. 阅读和记录代码切换语音

为了便于收集朗读的语音，我们开发了一个开源工具包。[2] 我们向 21 位双语读者分发了生成的混用文本。对于每个句子，我们随机选择了上述

[1] 10 个带有形态学示例的：ara、ces、deu、ita、kor、por、rus、slk、spa、tel、tam；4 个不带的：cmn、fra、hin、jpn

[2] https://github.com/brianyan918/sentence-recorder/tree/codeswitching

三个提示中的一个，并在所有语言对中标准化了选择过程。双语读者被提供了单语参考以及混用语言的句子。他们被指示进行以下操作：1) 验证该句子确实是混用了语言，2) 该句子与单语参考具有相同的总体含义，3) 阅读并录制该句子，4) 对该句子的流畅度进行评分。录音随后由人工听者进行了验证。

2.2. 跨 111 种语言对的语音合成

接下来，我们描述两个合成语音集，旨在通过涵盖额外的语言对并提供代码切换训练数据来补充 CS-花卉读取。

CS-花卉-XTTS 包含来自 XTTS-v2 模型 [28] 的合成语音，支持 17 种语言，涵盖了 16 种英语对以及 60 种非英语对（15 种阿拉伯语-X、15 种印地语-X、15 种普通话-X、15 种西班牙语-X）。对于这 16 种英语对，我们生成训练集、开发集和测试集，并将这 16 种非英语语言视为矩阵，而将英语作为嵌入——这些语言对中有 12 个与 CS-花卉读取重叠。60 种非英语语言对仅用于测试，我们将阿拉伯语、印地语、普通话或西班牙语视为矩阵，另一种语言作为嵌入——这些语言对均不与其他两个子集重叠。

CS-花卉-MMS 包含来自 MMS TTS 模型 [5] 的 45 种 X-英语语言对的合成语音，其中许多是资源较少的语言——这些语言对中只有 10 个与另外两个子集重叠。此集合仅用于测试，并且我们视非英语语言为矩阵。

两组数据均未经过人工验证，原因是缺乏双语人士。相反，我们使用语言通用的强制对齐来过滤低质量生成。

2.2.1. 廉价且灵活地生成代码切换文本

虽然在 §2.1.1 中描述的大型语言模型骨干可以生成自然且形态丰富的代码切换，但除了高昂的计算（或 API）成本外，还有一些缺点限制了其可扩展性。基于 LLM 的方法需要人工验证来拒绝大约 14 种 X-英语语言对中的 10%的句子；对于资源较少的语言，合理的预期是拒绝对会更高。

此外，我们发现基于大语言模型的方法产生的代码切换频率范围很广，如 CMI[27] 所测量的，在不同的语言对之间。这种差异显示在表 6 的第一行：12 种经人类验证的语言对 CS-花卉读取之间的 CMI 的标准差为 10.8。因此，我们选择了一种更简单和严格的方法来生成代码切换文本在 CS-花卉-XTTS 和 CS-花卉-MMS 中，称为对齐然后交换。

表 6: 对比由 *LLM* 提示和先对齐后交换（*Swap*）生成的文本在 *12* 种 *X*-英语语言组合中的代码切换频率，该频率通过代码混合指数 *(CMI)* 来衡量。*M*=平均值；*SD*=标准差。

Type	ara	ces	cmn	deu	fra	hin	ita	jpn	kor	por	rus	spa	M	SD
LLM	12.3	30.4	24.1	27.8	36.0	18.5	37.0	18.5	7.0	37.2	17.7	39.8	25.5	10.8
Swap	29.4	20.1	14.1	16.8	15.0	13.6	16.8	11.2	25.9	17.0	20.6	15.5	17.3	4.0

对齐然后交换的过程很简单。我们首先使用 AwesomeAlign[29] 获得词级对齐，这是一个经过五种语言配对的平行文本针对词级对齐目标进行微调的 104 种语言的 mBERT 模型 [30]。值得注意的是，AwesomeAlign 可以推广到其他在 mBERT 预训练中包含但在词级对齐微调阶段未见过的语言 [29]。接下来我们随机选择 30%用 Stanza[31] 标记的名词、动词、副词和形容词，与对齐嵌入语言中的词语进行交换。

此外，我们通过按照它们在单语嵌入语言句子中原始出现的顺序插入单词来处理一对一到多词对齐（矩阵到嵌入）。我们还通过对汉语和日语进行基于字符的语言重新分词为词汇单位来处理这些语言，使用了 Stanza[31]；FLEURS 中的韩语文本已经是分词状态，因此不需要这一步。

如表 6 所示，先对齐再交换（Swap）方法得到的 CMI 在语言对之间更为一致，其标准偏差仅为 4.0，降低了 60%。因此，这种更简单但更一致的方法对于资源较少的语言和罕见语言对生成代码切换文本是更优选的。

2.2.2. 生成式代码切换合成与 XTTS

XTTS-v2[28] 是一个多语言训练的 TTS 模型，基于 GPT-2 编码器预测 VQ-VAE 单元，这些单元随后隐式地条件化 HiFi-GAN 解码器。该模型在 17 种语言上进行了训练，通过在文本输入开头附加一个语言 ID 标记（例如 [en]）来区分这些语言。XTTS-v2 还通过使 GPT-2 编码器和 VQ-

表 7: 过滤与接受的合成语音的比较。过滤是通过强制对齐得分（*FAS*）完成的。

Subset	XTTS-测试 1 FAS↑	CER↓	UTMOS↑	SCD↓	MMS-测试 FAS↑	CER↓	UTMOS↑	SCD↓
Filtered	-0.68	32.95	2.46	1.12	-0.96	43.10	3.02	1.90
Accepted	-0.20	18.46	2.84	1.07	-0.47	40.07	3.04	1.57

VAE 解码器依赖于说话人信息来纳入语音转换组件；实际上，只需目标说话人的单一发音即可启用语音转换。

要使用 XTTS-v2 合成代码切换语音，我们将生成的代码切换文本与矩阵语言 ID 标记一起输入到模型中。普通话、日语和韩语文本被罗马化处理，与原始 XTTS-v2 训练的方式相同。我们使用来自 FLEURS 数据集的矩阵语言发言人来进行声音转换部分。XTTS-v2 产生 24khz 的语音，然后我们将之降采样到 16khz。

2.2.3. 级联式代码切换合成与 MMS-TTS

为了覆盖额外的语言对，我们采用了一种连接式方法，该方法使用 MMS-TTS[5] 生成分段的单语 TTS。我们将这些单语片段连接起来，在中间插入 100 毫秒的静音。由于 MMS-TTS 是一个单一说话人风格的 VITS 模型，因此最终得到的混合语言语音包含不自然的伪影和说话人的变化；但是内容是正确的。与 XTTS-v2 的输出不同，MMS-TTS 的输出中没有任何带口音的语音。

2.2.4. 通用强制对齐滤波

为了进行质量控制，我们使用了一个通用的强制对齐模型 MMS-ZS[32]，该模型被训练用来将数千种语言的语音与罗马化文本进行对齐——此模型提供了一种语言通用的理解度衡量标准。在每组语言中，过滤掉长度标准化后的强制对齐得分（FAS）最低的 5%的发音。

在表 7 中，我们报告了过滤与接受部分的 XTTS-TEST1 和 MMS 测试的平均 FAS，以及 Whisper-Large-v3[2] 字符错误率（CER），UTMOS (1-5) [33] 自动自然度指标，以及 Pyannote-3.1[34] 说话人分离模型识别的说话人变化数量（SCD）。对于 XTTS-TEST1，过滤对 Whisper CER 有较大影响，并对 UTMOS 有中等影响，表明生成合成

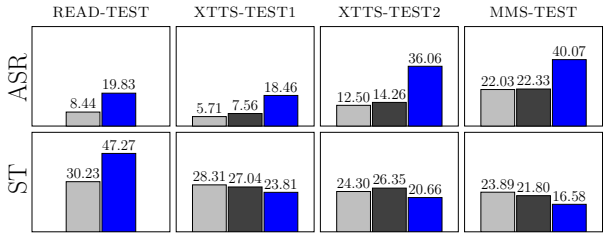

图 2: *Whisper-Large-v3 ASR（CER↓）和翻译成英语（BLEU↑）在 CS-FLEURS 测试集上的表现*（蓝色）。两个单语对照组的表现也显示出来进行比较（灰色）。

方法容易产生不可理解且不自然的语音。另一方面，对于 MMS-TEST，滤波对 CER 和 UTMOS 的影响较小，表明级联合成的出错概率较低；然而，SCD 显示这种方法会导致频繁的说话人变化。

3. 实验

在本节中，我们描述了基于 CS-FLEURS 的几个实证发现。对于 ASR，我们测量大小写不敏感且无标点字符错误率（CER）。对于 ST（到英语），我们测量大小写不敏感且无标点的 BLEU [35]。

3.1. 基准测试

*Whisper 在处理混杂语言的语音与单语语音时表现如何？*为了回答这个问题，我们将 ASR 和 ST（转为英语）的性能在每个 CS-FLEURS 测试集与两个单语对照组进行比较，这两个对照组包含各自 CS-FLEURS 集合中所有矩阵语言的单语语音：1) 原始 FLEURS 和 2) FLEURS 的 TTS 版本，这是一个单语版本。后者称为 TTS FLEURS，使我们能够将基于 TTS 的混杂语音的结果与匹配的基于 TTS 的单语语音进行比较，消除了 Whisper 可能对 XTTS 或 MMS-TTS 语音不够鲁棒的可能性。

如图 2 所示，CS-FLEURS 上的 ASR CER 比原始单语 FLEURS 高出 2 倍多——这种退化在非英语语言对集上最为明显（XTTS-TEST2）。然而，在 XTTS 和 MMS 集合中，ST 性能在单语和代码切换语音之间的差异较低。此外，Whisper 实际上在阅读测试上的表现优于原始 FLEURS——这可以归因于高频率的 X-英语代码切换（见表 6）。这些结果表明了**直接翻译代码切换语音比依赖转录**

表 8: 同一脚本与不同脚本语言对上 *Whisper-Large-v3* 语音识别（*CER*↓）性能的比较。

	阅读	XTTS		MMS
CS Pair Type	Test	Test1	Test2	Test
Same Script	7.32	8.49	9.92	28.26
Distinct Script	32.33	37.17	40.67	51.62

表 9: 训练数据扩增的效果结果。

Data	花朵		CS-花卉读取	
Augment	12 Seen	2 Unseen	12 Seen	2 Unseen
	14.38	8.93	31.77	29.62
✓	**12.67**	**8.51**	**26.24**	**27.77**

错误更少，尽管我们将非英语目标的基准测试工作留给了未来的研究。

Whisper 在不同语言对的混合语语音上表现如何？如表 8 所示，不同脚本语言对的性能明显低于相同脚本语言对——平均字符错误率高 3 倍。这些结果表明**代码切换 ASR 性能受限于交替生成两种不同脚本的能力**在单个话语级别的解码中。

3.2. 训练

最后，我们调查了问题：合成数据对训练模型有多大用处？为了解答这一问题，我们使用 ESPnet 工具包训练了两个模型 [36]：一个使用原始的 FLEURS 训练数据，另一个则使用的是原始的 FLEURS 加上 CS-FLEURS(XTTS-TRAIN)。这两个模型都按照 [37] 中描述的自我条件下的 XLSR 基础配方进行了相同次数的迭代训练。然后我们在读取测试上进行评估，该数据集包含 14 种语言组合；其中 12 种在 XTTS-训练中可见，而另外 2 种则不可见。我们还报告了这 12 种已知和 2 种未知的语言对的矩阵语言的单语 FLEURS 结果。如表 9 所示，**在合成代码切换数据上的训练提高了已见和未见过的语言对的性能**。

4. 结论

CS-FLEURS 是一个用于开发和评估跨越 52 种语言及 113 种独特混合使用的语言对的混合读取/合成语音数据集。通过使用这个控制了代码切换模式、文本领域和音频领域的数据集，我们发现跨不同脚本语言对转录语音仍然很困难。

5. References

[1] X. Li, F. Metze, D. R. Mortensen, A. W. Black, and S. Watanabe, "Asr2k: Speech recognition for around 2000 languages without audio," *Proc. Interspeech 2022*, pp. 4885–4889, 2022.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc ICML*. PMLR, 2023, pp. 28 492–28 518.

[3] Y. Zhang, W. Han, J. Qin, Y. Wang, A. Bapna, Z. Chen, N. Chen, B. Li, V. Axelrod, G. Wang *et al.*, "Google usm: Scaling automatic speech recognition beyond 100 languages," *arXiv preprint arXiv:2303.01037*, 2023.

[4] Y. Peng, J. Tian, W. Chen, S. Arora, B. Yan, Y. Sudo, M. Shakeel, K. Choi, J. Shi, X. Chang, J. weon Jung, and S. Watanabe, "Owsm v3.1: Better and faster open whisper-style speech models based on e-branchformer," in *Interspeech 2024*, 2024, pp. 352–356.

[5] V. Pratap, A. Tjandra, B. Shi, P. Tomasello, A. Babu, S. Kundu, A. Elkahky, Z. Ni, A. Vyas, M. Fazel-Zarandi *et al.*, "Scaling speech technology to 1,000+ languages," *Journal of Machine Learning Research*, vol. 25, no. 97, pp. 1–52, 2024.

[6] J. Kahn, M. Riviere, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *Proc ICASSP*. IEEE, 2020, pp. 7669–7673.

[7] C. Wang, M. Riviere, A. Lee, A. Wu, C. Talnikar, D. Haziza, M. Williamson, J. Pino, and E. Dupoux, "Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation," in *Proc ACL-IJCNLP*, 2021.

[8] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc ICASSP*. IEEE, 2015, pp. 5206–5210.

[9] V. Pratap, Q. Xu, A. Sriram, G. Synnaeve, and R. Collobert, "Mls: A large-scale multilingual dataset for speech research," in *Interspeech 2020*, 2020, pp. 2757–2761.

[10] R. Ardila, M. Branson, K. Davis, M. Kohler, J. Meyer, M. Henretty, R. Morais, L. Saunders, F. Tyers, and G. Weber, "Common voice: A massively-multilingual speech corpus," in *Proceedings of the Twelfth Language Resources and Evaluation Conference*, 2020, pp. 4218–4222.

[11] X. Li, S. Takamichi, T. Saeki, W. Chen, S. Shiota, and S. Watanabe, "Yodas: Youtube-oriented dataset for audio and speech," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[12] J. Shi, D. Berrebbi, W. Chen, E.-P. Hu, W.-P. Huang, H.-L. Chung, X. Chang, S.-W. Li, A. Mohamed, H. yi Lee, and S. Watanabe, "Ml-superb: Multilingual speech universal performance benchmark," in *Interspeech 2023*, 2023, pp. 884–888.

[13] A. Conneau, M. Ma, S. Khanuja, Y. Zhang, V. Axelrod, S. Dalmia, J. Riesa, C. Rivera, and A. Bapna, "Fleurs: Few-shot learning evaluation of universal representations of speech," in *2022 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2023.

[14] P. Auer, *Code-switching in conversation: Language, interaction and identity*. Routledge, 2013.

[15] M. Deuchar, P. Davies, J. Herring, M. C. P. Couto, and D. Carter, "Building bilingual corpora," *Advances in the Study of Bilingualism*, pp. 93–111, 2014.

[16] D.-C. Lyu, T. P. Tan, E. Chng, and H. Li, "Seame: a mandarin-english code-switching speech corpus in south-east asia." in *Interspeech*, vol. 10, 2010, pp. 1986–1989.

[17] E. van der Westhuizen and T. Niesler, "A first South African corpus of multilingual code-switched soap opera speech," in *Proc LREC*, 2018.

[18] S. A. Chowdhury, A. Hussein, A. Abdelali, and A. Ali, "Towards one model to rule all: Multilingual strategy for dialectal code-switching arabic asr," in *Interspeech 2021*, 2021, pp. 2466–2470.

[19] A. Diwan, R. Vaideeswaran, S. Shah, A. Singh, S. Raghavan, S. Khare, V. Unni, S. Vyas, A. Rajpuria, C. Yarra, A. Mittal, P. K. Ghosh, P. Jyothi, K. Bali, V. Seshadri, S. Sitaram, S. Bharadwaj, J. Nanavati, R. Nanavati, and K. Sankaranarayanan, "Mucs 2021: Multilingual and code-switching asr challenges for low resource indian languages," in *Interspeech 2021*, 2021, pp. 2446–2450.

[20] I. Hamed, N. Habash, S. Abdennadher, and N. T. Vu, "ArzEn-ST: A three-way speech translation corpus for code-switched Egyptian Arabic-English," in *Proc WANLP*, 2022, pp. 119–130.

[21] N. Goyal, C. Gao, V. Chaudhary, P.-J. Chen, G. Wenzek, D. Ju, S. Krishnan, M. Ranzato, F. Guzmán, and A. Fan, "The flores-101 evaluation benchmark for low-resource and multilingual machine translation," *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 522–538, 2022.

[22] C. Myers-Scotton, *Duelling languages: Grammatical structure in codeswitching*. Oxford University Press, 1997.

[23] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[24] S. Poplack, "Sometimes i' ll start a sentence in Spanish y termino en Español: Toward a typology of code-switching," *The bilingualism reader*, vol. 18, no. 2, pp. 221–256, 1980.

[25] G. Kuwanto, C. Agarwal, G. I. Winata, and D. T. Wijaya, "Linguistics theory meets llm: Code-switched text generation via equivalence constrained large language models," *arXiv preprint arXiv:2410.22660*, 2024.

[26] I. Hamed, N. Habash, S. Abdennadher, and N. T. Vu, "Investigating lexical replacements for arabic-english code-switched data augmentation," in *Proc LoResMT*, 2023, pp. 86–100.

[27] B. Gambäck and A. Das, "Comparing the level of code-switching in corpora," in *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC)*, Reykjavik, Iceland, 2014, pp. 1850–1855.

[28] E. Casanova, K. Davis, E. Gölge, G. Göknar, I. Gulea, L. Hart, A. Aljafari, J. Meyer, R. Morais, S. Olayemi *et al.*, "Xtts: a massively multilingual zero-shot text-to-speech model," *arXiv preprint arXiv:2406.04904*, 2024.

[29] Z.-Y. Dou and G. Neubig, "Word alignment by fine-tuning embeddings on parallel corpora," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2112–2128.

[30] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings NAACL*, J. Burstein, C. Doran, and T. Solorio, Eds. Association for Computational Linguistics, 2019, pp. 4171–4186.

[31] P. Qi, Y. Zhang, Y. Zhang, J. Bolton, and C. D. Manning, "Stanza: A python natural language processing toolkit for many human languages," *arXiv preprint arXiv:2003.07082*, 2020.

[32] J. Zhao, V. Pratap, and M. Auli, "Scaling a simple approach to zero-shot speech recognition," *arXiv preprint arXiv:2407.17852*, 2024.

[33] T. Saeki, D. Xin, W. Nakata, T. Koriyama, S. Takamichi, and H. Saruwatari, "Utmos: Utokyo-sarulab system for voicemos challenge 2022," in *Interspeech 2022*, 2022, pp. 4521–4525.

[34] H. Bredin, "pyannote.audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe," in *Proc. INTERSPEECH 2023*, 2023.

[35] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, 2018.

[36] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech 2018*, 2018, pp. 2207–2211.

[37] W. Chen, B. Yan, J. Shi, Y. Peng, S. Maiti, and S. Watanabe, "Improving massively multilingual asr with auxiliary ctc objectives," in *Proc ICASSP*. IEEE, 2023, pp. 1–5.