BLSP: Bootstrapping Language-Speech Pre-training via Behavior Alignment

2024年8月12日 19:32

论文分析：语音文本模态对齐是gpt-4o关键的一步，本文在输入上做到了对齐，但是副语言信息的缺失意味着无法合成逼真人声，只完成了一部分

需要考虑的还有 流式实现的问题以及如何获取包含副语言信息特征表征的问题，但论文中迈出了SLLM获取智能这一步，模态对齐和关注续写的训练方法可以借鉴

BLSP: Bootstrapping Language-Speech Pre-training via Behavior Alignment

Chen Wang[1,3], Minpeng Liao[2], Zhongqiang Huang[2], Jinliang Lu[1,3], Junhong Wu[1,3]
Yuchen Liu[1], Chengqing Zong[1,3], Jiajun Zhang[1,3,4†]
[1] Institute of Automation, Chinese Academy of Sciences
[2] Machine Intelligence Technology Lab, Alibaba Inc.
[3] School of Artificial Intelligence, University of Chinese Academy of Sciences
[4] Wuhan AI Research
{wangchen2020}@ia.ac.cn {jjzhang,cqzong}@nlpr.ia.ac.cn
{minpeng.lmp,z.huang}@alibaba-inc.com

背景：

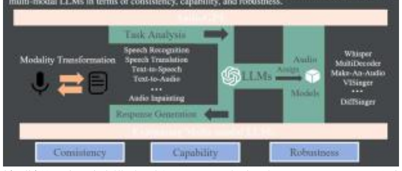
LLM已经发展到一个很高的水平 -> 语音和文本的天然对齐 -> 如何利用少量的语音数据激发LLM的语言能力 -> 将LLM能力延伸到语音模态

已有的方法：

1. 级联的系统：比如audioGPT: speechin/speechout + GPT + audio model

组装起来的用于音频分析生成的系统：能够在多轮对话中处理包含语音、音乐、声音、说话人等理解和生成的AI任务，使人类能够以前所未有的轻松方式创建丰富多样的音频内容。，

Figure 1: A high-level overview of AudioGPT. AudioGPT can be divided into four stages, including modality transformation, task analysis, model assignment, and response generation. It equips ChatGPT with audio foundation models to handle complex audio tasks and is connected with a modality transformation interface to enable spoken dialogue. We design principles to evaluate multi-modal LLMs in terms of consistency, capability, and robustness.



2. 端到端的语音语言大模型：比如 LLASM， 核心思想是stage1 利用ASR任务搭配简单的asr相关指令（帮我把这句话转成文字）构建模态适应pretrain model; stage2 构建跨模态的语音语言微调指令集进行微调，重点是指令集构建与发布， 模型结构与训练过程与本文基本一致。作者认为该文章缺点是：依赖指令集且基于asr任务导致语音特征表征出现偏差鲁棒性较差

mask: https://huggingface.co/datasets/LinkSoul/LLaSM-Audio-Instructions -> 公开指令集收集; 需要了解的是指令集构建方法，以及训练数据的构造

X_{sample} = B_INST + B_SYS + SYSTEM + E_SYS + [$content_{user}$] + E_INST + [$content_{gpt}$]
$content_{user}$ = [$audio_{token}$] + "\n" + I_{simple} (or $content_{user}$ = I_{simple} + "\n" + [$audio_{token}$])
$content_{gpt}$ = TEXT_LABEL
$audio_{token}$ = AUDIO_START_TOKEN + AUDIO_PATCH_TOKEN * [audio_token_len] + AUDIO_END_TOKEN

Figure 4: The sample sequence format for the pre-training. We follow the manner of Llama-2, and B_INST = '[INST]', E_INST = '[/INST]', B_SYS = '<<SYS>> \n', E_SYS = '\n <</SYS>> \n\n'. The SYSTEM = 'You are a helpful language and speech assistant. You are able to understand the speech content that the user provides, and assist the user with a variety of tasks using natural language.', and the TEXT_LABEL is the text label of the ASR data sample. The audio_token_len is set to 64 by default. Special audio tokens are used, AUDIO_START_TOKEN = '<au_start>', AUDIO_END_TOKEN = '<au_end>', AUDIO_PATCH_TOKEN = '<au_patch>'. The $content_{user}$ consists of the $audio_{token}$ and the I_{simple}, in which I_{simple} is a simple instruction and is randomly put before or after the $audio_{token}$. While training the BOS token and the EOS token will be added to each sample at the beginning and the end of the sequence, only the green tokens are used to compute the loss.

$X_{instruction}$ = <BOS> + X_{sample} + <EOS> + X_{sample} + <EOS> + ⋯ + <EOS>
X_{sample} = B_INST + B_SYS + SYSTEM + E_SYS + [$content_{user}$] + E_INST + [$content_{gpt}$]
$content_{user}$ = [$audio_{token}$], $content_{gpt}$ = TEXT_RESPONSE
$audio_{token}$ = AUDIO_START_TOKEN + AUDIO_PATCH_TOKEN * [audio_token_len] + AUDIO_END_TOKEN

Figure 5: The sample sequence format for the cross-modal instruction fine-tuning. We follow the manner of Llama-2, and B_INST = '[INST]', E_INST = '[/INST]', B_SYS = '<<SYS>> \n', E_SYS = '\n <</SYS>> \n\n'. The SYSTEM = 'You are a helpful language and speech assistant. You are able to understand the speech content that the user provides, and assist the user with a variety of tasks using natural language.', and the TEXT_RESPONSE is the text response from the chatbot. The audio_token_len is set to 64 by default. Special audio tokens are used, AUDIO_START_TOKEN = '<au_start>', AUDIO_END_TOKEN = '<au_end>', AUDIO_PATCH_TOKEN = '<au_patch>'. The $content_{user}$ is the $audio_{token}$ which will be replaced by the audio embeddings during training. Each round of question and answer will be formatted as X_{sample}, which will be concatenated together with the EOS token. While training the BOS token will be added at the beginning of the sequence, and the EOS token will be added at the end of the sequence, only the green tokens are used to compute the loss.

Table 1: LLaSM-Audio-Instructions Data.

LLaSM-Audio-Instructions				
Source	Conversations	Samples	English Samples	Chinese Samples
WizardLM	80k	160k	159k	<1k
ShareGPT	23k	155k	140k	15k
GPT-4-LLM	96k	192k	128k	64k
Total	199k	508k	428k	80k

本文：

在本文中，提出了 BLSP 方法，利用现有的 ASR 训练数据，通过行为对齐引导语言语音预训练。加入冻结的语音 encoder和LLM之间的轻量级模态适配器来实现这一点，确保LLM表现出相同忽略模态（语音or文本）的生成行为。

文中主要关注续写能力，与原有LLM的能力相似。文中证明了对齐引导语音预训练的方法可以将 LLM 的能力扩展到语音，并实现与级联系统相比具有竞争力的性能，在零样本跨语言场景也可以做到同样的能力，从而实现语音识别、语音翻译、口语理解和语音对话。

主要贡献：

1. 提出BLSP方法即通过行为对齐实现语言-语音预训模型的训练，引入一种新的LLM的多模态对齐方案； 且改进了基于ASR的pretrain改为关注续写任务的pretrain， 更贴合LLM本身的能力
2. 轻量级模块适配器，训练好的encoder 和LLM中间加入适配器做到少量参数训练，多模态适配； 这样通过asr训练数据就可以做到模态适配，减少了对语音指令数据的依赖
3. 丰富的实验验证和结果分析

通过行为对齐的方法进行语言语音模型的预训练：

发现分析问题: 对几个不同的任务 续写任务， 情绪分析， 语音识别， 语音翻译任务 的 不同模态（语音 or文本）输入的特征表征进行分析，直觉上，语音对齐LLMtoken上映射的好的话，那么相同任务的不同模态的特征表征应该表现出靠近，而不同任务的相同或者不同模态都应该分离

基于ASR pretrain model的模态特征表现出了，语音表征跟语音表征在一起玩，且完全不跟指令去跟相应的text表征合作的迹象，这不是好的模态融合结果，也就是行为跟表征没有对齐，各干各的/

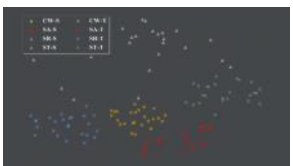


Figure 1: T-SNE visualization of feature representations learned from ASR task. Colors denote task instructions: orange for continuation writing (CW), red for sentiment analysis (SA), blue for speech recognition (SR), and gray for speech translation (ST). Shapes distinguish input modality: triangles for speech, crosses for text. Note that speech inputs result in overlapping features.

	CW-S	SA-S	SR-S	ST-S
CW-S	1.000	0.997	0.997	0.991
SA-S	0.997	1.000	0.997	0.992
SR-S	0.997	0.997	1.000	0.993
ST-S	0.991	0.992	0.993	1.000

Table 7: Average similarity between representations of the same speech inputs under different task instructions learned from ASR task.

CW	SA	SR	ST
0.270	0.106	0.328	0.176

Table 8: Average similarity between representations of paired speech/text inputs under the same task instructions learned from ASR task.

解决： 提出**行为对齐概念**，让模型不再分开关注语音文本表征，而是关注指令和输出的对齐，忽视输入模态的不一样。具体分为两步：

step1:

sists of two steps. In the first step, we use speech transcripts as input and instruct the LLM to generate responses using the following prompt:

```
###[Human]:<instruction><transcript>\n\n\n
###[Assistant]:
```

step2:

transcript改成对应的speech signal， step1生成的结果作为ground truth进行有监督训练

sponses produced in the first step as the ground truth for supervised learning using language modeling loss with the following prompt:

```
###[Human]:<instruction><speech>\n\n\n
###[Assistant]:<response>
```

此外: 在几个任务中，训练时重点关注的续写任务，避免了asr预训练中容易出现的过拟合问题

关于adapter, 是比较常见的模态融合方法，修改参数量少，好收敛。冻结encoder和LLM让adapter学习一个模态表征映射就可以

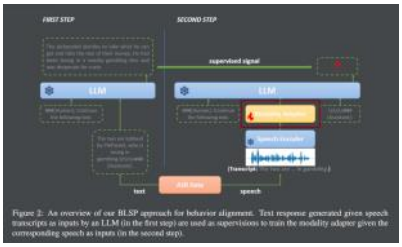


Figure 2: An overview of our BLSP approach for behavior alignment. Text response generated given speech transcripts as inputs by an LLM (in the first step) are used as supervisions to train the modality adapter given the corresponding speech as inputs (in the second step).

实验：

论文提供了两个版本模型： BLSP 和BLSP-RP 一个只关注续写一个关注续写和识别数据比例9：1

主要关注的对比模型是CTC+LLM，也是encoder和LLM的组合

1. **在各个不同任务上的结果：**

Method	ASR (WER↓)		ST (BLEU↑)		SLU (ACC↑)		
	LibriSpeech test-clean	TED-LIUM 3	MUST-C	CoVoST 2.0	SNIPS fgit-close	FSC	SLUE VoxCeleb
Text+LLM	0.0	5.6	0.0	14.5	19.7	21.9	86.5	72.4	75.0
Whisper+LLM	3.4	5.9	4.5	20.4	16.6	16.9	83.2	58.3	74.1
CTC+LLM	6.2	10.8	8.4	20.7	15.3	15.2	79.0	60.4	74.7
ASR pretraining	—	3.7	—	4.5	0.0	0.0	0.0	0.0	0.0
BLSP	—	10.4	—	23.1	12.3	12.7	75.8	60.9	76.0
+RP	—	6.4	—	8.1	14.9	13.8	76.8	77.3	75.5

Table 2: Overview of BLSP results on zero-shot speech-to-text tasks. For each ASR test set, we report two WER scores: on the left for the standalone ASR component of a cascaded system, and on the right for instructing the LLM to repeat the words recognized by the ASR component. The BLEU scores for the ST test sets are averaged across multiple translation directions.

2. 然后验证了基于blsp方法预训练的模型比基于asr预训练模型在下游任务上更鲁棒：

Method	en-de	en-es	en-fr	en-it	en-nl	en-pt	en-ro	en-ru
w/o pretraining	21.1 / 74.4	25.4 / 78.1	20.9 / 75.6	20.6 / 76.1	23.6 / 76.8	25.3 / 76.7	16.4 / 74.7	13.7 / 73.5
ASR pretraining	23.7 / 78.6	27.9 / 78.7	32.1 / 77.7	23.3 / 78.2	25.4 / 78.7	27.3 / 79.6	18.6 / 77.4	14.9 / 76.2
BLSP	23.3 / 77.7	27.4 / 79.5	31.9 / 76.5	23.2 / 79.0	26.4 / 80.0	28.5 / 80.4	19.2 / 78.6	18.6 / 77.3

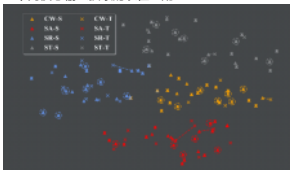
Table 4: ST results (BLEU / COMET) of fine-tuned models on MUST-C.

3. QA 能力验证：

Method	Accept Rate (%)
Text+LLM	94.5
Whisper+LLM	91.3
CTC+LLM	88.6
ASR pretraining	0.0
BLSP	88.5
+RP	88.3

Table 3: ChatGPT evaluation using acceptable rate.

4. 不同模态输入获取的表征匹配：



在附录 D 中，我们提供了定量证据，证明我们的 BLSP 模型可以在不同的指令下为相同的语音输入生成不同的表示，并且当给出相同的指令时，成对语音和文本输入的表示非常匹配。这些结果表明，BLSP 方法有效地对齐同一空间中的语音和文本输入，从而将 LLM 的指令跟踪能力扩展到语音输入。

5. 多轮对话能力，模态可以对齐的话，多轮对话对于LLM来说也不成问题：

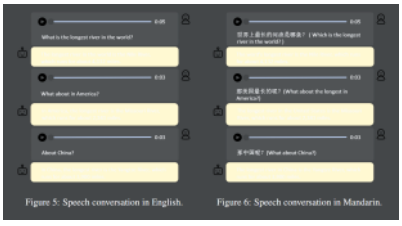


Figure 5: Speech conversation in English.

Figure 6: Speech conversation in Mandarin.