Записки научных
семинаров ПОМИ
Том 546, 2025 г.

A. Iudin, M. Skripkin, O. Y. Rogov, D. Korzh

CLARISPEECH: LLM-ENHANCED SPEECH RECOGNITION POST-CORRECTION

Abstract. Recent advances in Automatic Speech Recognition (ASR) have made these systems widely applicable, including in virtual assistants and web-based interfaces. However, even cutting-edge ASR models often produce errors, particularly when adapting to new speech domains. Conventional solutions involve fine-tuning ASR models on target-domain data or integrating language models (LMs) to rescore predictions. However, joint fine-tuning of ASR and LM models can be unstable, demand substantial training data, and suffer from alignment issues. Using more sophisticated language models for shallow fusion, especially large language models (LLMs), is impractical, leading to significant computational overhead. In this paper, we address these challenges by focusing on post-transcription corrections, using parameter-efficient fine-tuning of external language models while leaving the ASR system frozen. Our experiments show that this approach significantly improves accuracy and computational efficiency. Compared to the baseline ASR system, employing an ASR+LLM configuration reduces the word error rate from 12% to 10%, while increasing computational cost by less than 50%, despite an eightfold rise in the number of parameters.

§1. Introduction

Automatic speech recognition (ASR) systems have recently achieved superhuman performance. They provide high-quality transcribed text for downstream tasks such as call center automation, speech translation, auto subtitle generation, etc.

There are several main techniques to achieve high-quality ASR performance. The first is to experiment with the model's architecture and straightforwardly increase the size and diversity of the train data. Additionally, multitask training mode can significantly improve the robustness [34]. Despite the solid improvements in benchmark datasets, this approach is computationally expensive to fine-tune.

Key words and phrases: Deep Learning and Automatic Speech Recognition and Large Language Models and Natural Language Processing and Artificial Intelligence and Speech recognition methods.

The second approach is to experiment with training techniques. Supervised fine-tuning (SFT) is the most popular method requiring labeled paired audio-text data, which is expensive. SFT allows us to achieve sufficient results with relatively small data. In contrast, self-supervised learning (SSL) pre-training [9, 29] can be applied. During this stage, the model learns how to compress input data to a meaningful hidden state, significantly improving general audio understanding. This is done by solving auxiliary tasks derived from the data itself, such as predicting missing data points or parts of the text/image/audio, reconstructing input features, or contrasting representations of different augmentations of the same data instance. The audio SSL procedure is similar to the masked language modeling (MLM) task, like in BERT [15]. Although SSL drastically improves ASR quality, this stage requires a large amount of unlabeled data and GPU hours. Finally, the SFT is still needed to reach the best ASR performance, especially in domain-specific audios and tasks. Weakly supervised learning bridges the gap between supervised and unsupervised approaches using imprecise, noisy, or incomplete annotations. This paradigm encompasses techniques such as semi-supervised learning, where a small set of labeled data is complemented by a large pool of unlabeled data, and noisy-label learning, which accounts for label inaccuracies.

Hypotheses rescoring is another crucial technique that is widely used in ASR systems. There are several main rescoring approaches. First-pass rescoring (shallow-fusion) [63] is a method that uses a Language Model (LM) during hypothesis decoding. Given the hidden state from the encoder of the ASR model, the possible token distribution is computed, and then the distribution is blended with the LM distribution. Then, the token is picked from the distribution obtained in the previous step. Deep-fusion, when we train a model over a pre-trained LM and ASR. The second-pass rescoring is a post-ASR method based on rearranging the transcribed hypothesis [46]. The idea is to take the output after beam-search decoding and then rearrange hypotheses according to acoustic and language model probability with length penalty: $P_{\text{total}} = P_{\text{ASR}}(y|x) + \alpha P_{\text{LM}}(y) + \beta \operatorname{len}(y)$, where x is input audio, y is an output transcription. It allows for the incorporation of information both from the ASR system and LM. LM is usually trained on many widely available text data in all those cases and has a better language understanding, enhancing the ASR system quality.

Also, Audio Large Language Models (multimodal LLM) [5,6,25] recently have been applied to the ASR task. Audio LLM consists of the audio encoder, a language model, and a connector. Audio LLMs are usually trained for tasks such as ASR and emotion recognition. Such a setup could improve speech recognition quality and other downstream textual-speech tasks.

However, the approaches described above have potential disadvantages; thus, the place for research and transcription quality enhancement still exists. Firstly, suppose one fine-tuned the pre-trained audio model (e.g., Whisper) to improve the ASR performance of this model. In that case, it can degrade the quality of other tasks, such as language identification. Secondly, Audio LLM occasionally could not capture specific details from the audio encoder, as it happens in Visual LLMs [61]. Spoken language and conversational speech are not always grammatically accurate, so LMs, trained primarily on more formal text, could rely on language rules and provide inaccurate transcription. The deep fusion approach is often unstable to train; second-pass rescoring only selects the most probable hypothesis but cannot fix them, and shallow-fusion cannot incorporate strong LLM due to the significant computational overhead. Moreover, tuning the LM for the new words and phrases is easier than the ASR model.

Our idea is to consider strong LMs and fine-tune them on the error correction of several ASR models on different audio datasets. For this purpose, LMs will be tuned to understand the errors the ASR models are prone to. To give the LMs better context about audio, we will provide them with the top 5 hypotheses instead of only 1, the most probable transcription. This could lead to a lower Word Error Rate (WER). More about this metric in 5.1. Hypotheses 2-5 are often helpful because they may contain correct tokens not presented in the top 1 hypothesis.

Our contributions might be summarized as follows:

- We significantly improve the transcription quality compared to the stand-alone ASR. Namely, we reduce WER from 12% to 10% on average.
- We demonstrate that our post-correction LMs have good generalization ability. They improve transcriptions of different ASR models even for the out-of-the-train domain audio sources.
- We demonstrate that the Noisy Embeddings Instruction Fine Tuning (NEFTune) [12] regularization technique helps train more robust post-corrections LLMs. Furthermore, we show that even relatively small LMs are also strong correctors.

§2. Related Work

In this section, we discuss works relevant to the post-correction, such as main ASR and LLM models and previous approaches.

2.1. ASR models.

Most ASR models utilize Mel-Spectrograms or MFCC as input features instead of raw waveform [3, 44]. CTC-based ASRs apply Connectionist Temporal Classification (CTC) loss [24] that align input speech sequences with output text, simplifying training and forcing the model to learn the optimal alignment between speech frames and text transcriptions. During inference, the model maintains several top hypotheses over paths in beam-search. Despite its simplicity and performance, CTC has a disadvantage as its decoding lacks context information, assuming that characters are independent. Listen, Attend, and Spell (LAS) [45] uses an encoder-decoder architecture where the encoder processes the input speech signal, and the attention mechanism allows the decoder to focus on different parts of the input sequence dynamically. The Conformer model [18] integrates Convolutional Neural Networks (CNNs) with Transformer [59] architectures, enabling capturing local features through convolution and long-range dependencies through self-attention.

Whisper leverages a transformer-based encoder-decoder architecture optimized in a weakly supervised regime on a colossal amount of data. The model uses an autoregressive approach, where it encodes audio data with convolution layers and Multi-Head Attention (MHA) with residual connection [51, 54, 55] into embeddings and then iteratively predicts the next symbols token-by-token with MHA. In addition, they use the technique Pre-Layer Normalization [53], which leads to better convergency. In the original Transformer architecture, LayerNorm was used with residual connection after the transformer block. Whisper focuses on robustness and scalability to different languages and speech domains, making it adaptable to diverse datasets and conditions. However, it does not demonstrate the best results on standard benchmarks and hallucinates if the audio contains a lot of silence, as it misleads attention.

Wav2Vec2 employs a self-supervised learning approach to pre-train the model on unlabeled speech data. This model uses contrastive learning to understand audio representation as in masked language modelling. Both Whisper and Wav2Vec2 are later fine-tuned on clean labeled data.

Table 1. Brief description of used datasets.

Dataset	Description	#Train	#Test
ATIS	Airline travel information.	3964	809
CHiME4	Audio with some background noise.	9600	1320
CORAAL	Interviews with speakers born between 1888 and 2005.	3232	170
Common Voice	Publicly available voice dataset, powered by the voices of volunteer contributors.	47293	2000
LRS2	BBC television recordings.	42940	2259
LibriSpeech	Collection of audiobooks.	50000	5559
SwithBoard	Telephone speech corpus.	36539	2000
TED-LIUM 3	TED talks.	50000	1155
WSJ	Reading of Wall Street Journals.	37514	836

2.2. HyPoradise. `HyPoradise` [33] is a dataset created for the investigation of methods of ASR post-correction, especially for the LLMs' fine-tuning. Dataset consists of several popular audio datasets [8,36,38–42,60, 62] listed in Table 1.

The audio was transcribed using the Whisper-Large ASR model and divided into training and test splits. The top 5 transcription hypotheses represent each audio sample. The errors presented in the dataset can be partially classified as follows:

- **insertion** (ASR: ”various sizes to you”, Ground-Truth (GT): ”various sizes”);
- **deletion** (ASR: ”this was a great”, GT: ”this was great”);
- **consonant** (ASR: ”additionally”, GT: ”traditionally”);
- **hallucinations** due to background noise (ASR: ”sorry sorry sorry”, GT: ”despite the decline in stock prices...”).

2.3. Autoregressive Text Generation and Attention. Sequence-to-Sequence (Seq2Seq) learning is a framework designed for tasks where the input and output are sequences of potentially differing lengths. The encoder transforms input sequence $\mathbf{x}$ into a context vector $\mathbf{c} = \text{Encoder}(\mathbf{x})$, where $\mathbf{x} = (x_1, x_2, \ldots, x_T)$, while the decoder generates the output sequence $\mathbf{y}$ autoregressive: $y_t = \text{Decoder}(\mathbf{c}, y_1, \ldots, y_{t-1})$. Given an input sequence, and the output sequence, the model learns

$$P(\mathbf{y} \mid \mathbf{x}) = \prod_{t=2}^{T} P(y_t \mid y_1, y_2, \ldots, y_{t-1}, \mathbf{c}), \tag{1}$$

where y_1 represents the special start of the sequence token.

The attention mechanism enhances Seq2Seq by considering all encoder hidden states, weighting them based on their relevance to the current decoder state, and combining them with the current decoder state, where the attention scores can, for example, be computed as a dot product $e_i^t = \langle h_{xt}, h_{yi} \rangle$. And the attention scores (probability distribution of the encoder states importance) $\alpha^t = \mathrm{SoftMax}(\mathbf{e}^t)$. Further, this attention vector is processed with the decoder state. This approach is called cross-attention (attention between decoder and encoder states).

Another type of attention is a self-attention [59]. In self-attention, each encoder (decoder) state "attends" to every other encoder (decoder) token, enabling the model to capture both local and global dependencies (however, masked attention is used for the decoder to prevent foresight in future text). Given an input sequence of embeddings $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{N \times k}$, self-attention computes three projections: queries Q, keys K, and values V. The result is a weighted combination of the value vectors V, where the weights are derived from the similarity between queries Q and keys K. Self-attention enables efficient parallelization. Transformer architecture incorporates encoder self-attention with positional encoding, decoder masked self-attention and decoder-encoder cross attention, which made this architecture instrumental in achieving state-of-the-art results across various Natural Language Processing (NLP) tasks.

2.4. Common Loss Functions. Connectionist Temporal Classification (CTC) [24] loss is widely used in speech recognition tasks where explicit alignment between input frames and output symbols is not given. Consider an input sequence of T frames $x_{1:T}$ and a target sequence of U symbols $y_{1:U}$. CTC introduces an additional blank symbol $\varnothing$ (ϵ) to form an extended alignment of length L, where $U \leqslant L \leqslant T$.

The probability of obtaining the correct target sequence y given x is computed by summing over all possible alignment paths π that map to y through the mapping function $\mathcal{B}$, which removes blank symbols:

$$P(y \mid x) = \sum_{\pi \in \mathcal{B}^{-1}(y)} P(\pi \mid x). \tag{2}$$

The CTC loss function is defined as the negative log-likelihood of this probability: $L_{CTC} = -\ln P(y \mid x)$. LLMs typically learn to generate text by predicting the next token in a sequence given the previous tokens. Consider a sequence of tokens $x_{1:T}$ drawn from a vocabulary $\mathcal{V}$. At each step t, the model estimates a probability distribution $P(x_t \mid x_{1:t-1})$ over $\mathcal{V}$. The training aims to maximize the likelihood of the correct token at each timestamp, which is commonly achieved through a cross-entropy loss (or, equivalently, the negative log-likelihood). More formally, given a training set of sequences, the loss for a single sequence $x_{1:T}$ is defined as:

$$L_{CE} = -\sum_{t=1}^{T} \ln P(x_t \mid x_{1:t-1}). \tag{3}$$

This cross-entropy loss encourages the model's predicted probability distribution at each step to place a higher probability mass on the correct token.

In a post-correction input sequence is formed from the task description prompt, five hypotheses and a ground truth label y: $x = [\text{Prompt:}, \chi_1, \dots \chi_5 : y]$, where χ_i are the top i ASR hypothesis. It is worth mentioning that token generation and loss calculation (3) starts only on the y subpart of the input x and can be considered as supervised learning.

2.5. ASR Post-Correction. LMs can be classified as encoder-only, encoder-decoder, and decoder-only. Encoder-only models, such as BERT, focus on generating contextual representations for all tokens within an input sequence simultaneously, making them highly effective for understanding tasks like classification, named entity recognition and question answering. BERT employs a bidirectional transformer to pre-train on masked language modelling and next-sentence prediction, capturing context from both directions. Encoder-decoder models, exemplified by T5 and the original Transformer, use an encoder to process the input into a contextual representation and a decoder to autoregressively generate the output sequence, making them ideal for tasks like translation and summarization. Decoder-only models, such as GPTs, based on unidirectional transformer decoder, are specialized for autoregressive generation, where they predict each token based on previously generated tokens, excelling in open-ended tasks like text generation and dialogue systems. GPT-4 enhances alignment with user intent using fine-tuning and reinforcement learning from human feedback (RLHF [58]).

One way to correct hypotheses from ASR systems is to use a proprietary LLM with a great linguistic domain, e.g., in [22] authors tested a proprietary model with several datasets, like `LibriSpeech`, `TED-LIUM 3` and `Artie Bias Corpus` [43].

They investigated such ideas as hypothesis generation and selection approaches with performance dependency on the number of provided hypotheses in zero-shot and one-shot scenarios. Although they achieved good results, their approach is limited to using the proprietary LLM through API.

Cross-modal fusion with Whisper and Llama [1,56] [1] proposed an end-to-end approach, with fusion inside the self-attention layer, where Keys and Values in attention are taken from the Whisper model.

Another straightforward approach is to fine-tune the sequence-to-sequence or causal LLM. In `HyPoradise`, the authors tuned T5-0.75B [21] and Llama-13B LLMs for the post-correction and measured WER on the `HyPoradise` dataset, demonstrating promising results. However, they fine-tuned each model to each dataset separately, which, as was said, could lead to reduced generalization and decreased accuracy on out-of-the-train audios and corresponding transcriptions. In our experiments, the relative improvement of WER (ASR only vs. ASR with post-correction LLM) was better for 5 out of 8 datasets than in the `HyPoradise`.

2.6. Adversarial Attacks. Deep learning models are vulnerable to special adversarial perturbation [32]. Adversarial perturbation can be insignificant to humans but significantly affect the model's performance. Consider classification task $f_\theta : \mathbb{R}^n \to [0,1]^C$, where deep learning model f with parameters θ maps input images or audios x to class probabilities, we want to create an adversarial transformation T with parameter δ that x and $T(x,\delta)$ are classified differently: $\arg\max f_\theta(x) \neq \arg\max f_\theta(T(x,\delta))$. Among all possible transformations, the most common are additive perturbations $T(x,\delta) = x+\delta$; for additive perturbation, the adversarial objective can be described as follows in (4):

$$\min \|\delta\|_p : \arg\max f_\theta(x) \neq \arg\max f_\theta(x+\delta). \tag{4}$$

Adversarial attacks can be classified as white-box, black-box, and grey-box. A white-box adversary has full access to the fooling model's architecture, weights, gradients, and outputs. In contrast, black-box [30] adversary has only access to the final prediction or the prediction logits. Grey-box adversary has intermediate knowledge of the target fooling model. White-box attacks are simpler and simultaneously strong adversaries as they fully know the target model. Moreover, white-box adversarial perturbations are transferable to the new model. The simplest white-box attack is the fast gradient sign method (FGSM) [31] $x' = x + \epsilon\,\mathrm{sign}(\nabla_x L(x, y, \theta))$, where L is a loss function. At the same time, targeted FGSM can be described as $x' = x - \epsilon\,\mathrm{sign}(\nabla_x L(x, y_t, \theta))$, where y is a ground truth class, y_t is a target class we want the model f incorrectly predict. This attack aims to adjust input x to maximize the loss for untargeted classification or minimize the loss of the target class. These solutions can be obtained from the idea of loss maximization/minimization for $p = \infty$ using a first-order approximation.

§3. Model Description

3.1. Language model tuning. We utilized the transformer-based T5 and Qwen [10] models for our experiments. The first one has an encoder-decoder transformer architecture. The idea behind this model is that the encoder gathers all information from the instruction and hypotheses, encodes it into hidden states, transmits it through cross-attention to the decoder part, and the decoder generates its own transcription. The T5 model was fine-tuned on the `Flan` [23] dataset, leading to better generalization for tasks such as summarization, reading comparison, and comprehension. The model receives five hypotheses from Whisper on `HyPoradise` and processes them. Qwen, on the other hand, is a decoder-only model that predicts the next token with causal attention. We considered Qwen2-0.5B and 1.5B with the top 1 hypothesis generated on augmented audios by different ASR systems for experiments with `CV`. In this branch of experiments, we test LM's ability to handle poor quality data, such as augmented audio recordings, without "context" information from hypotheses 2-5. See Figure 1 for more details about our method.

For fine-tuning Flan-T5 we utilized Low-rank adaptation (LoRA) and NEFTune.

3.2. LoRA. Low-rank adaptation [11] is a technique that is used to fine-tune deep learning models efficiently. LoRA reduces trainable parameters in large models by decomposing learnable parameters into low-rank factors, cutting computational costs and memory usage while maintaining performance. During LoRA fine-tuning, the models' weights are frozen and not updated. In contrast, matrices A and B of a small rank are created

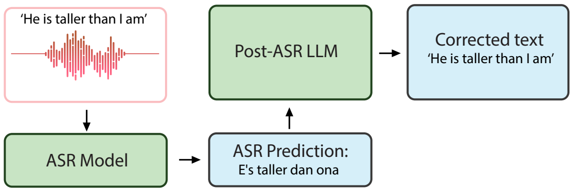

Figure 1. Overview of the considered method. ASR model receives audio as an input. This model generates 5 possible transcriptions with beam-search. These hypotheses transferred to LLM, which does post-correction.

to be learned during fine-tuning. They have shapes (d, r) and (r, d), respectively, where d is the dimensionality of the hidden layer and r is the rank of LoRA, which is relatively small (1-16). A is initialized with Normal noise (however, it can be initialized differently, e.g., using singular value decomposition of the W), B is initialized with zeros, and then they are learned through the optimization with numerical gradient optimization. The formula for the new weights W_{LoRA} is (5)

$$W' = W_{\text{frozen}} + A \times B \tag{5}$$

3.3. NEFTune. NEFTune is a regularization method that samples a ϵ noise vector with a uniform distribution $\epsilon \sim \text{Uniform}(-1, 1)$. This noise is then scaled and added to the embeddings. The scaling factor is equal to the regularization parameter α divided by the product of the sequence length L and the size of the embeddings d. More details can be found in (6).

$$X'_{emb} = X_{emb} + (\frac{\alpha}{\sqrt{Ld}})\epsilon \tag{6}$$

It led to better instruction execution without additional computational overhead, as the corresponding article shows.

3.4. PGD attack on audio data. Projected gradient descent (PGD) is a white-box attack [2], commonly used to evaluate the robustness of deep learning models. It is a stronger adversary than FGSM because more steps are used for the generation. The PGD is described in Algorithm 1.

As ASR models are also sensitive to adversarial attacks [4, 50, 52], we decided to investigate the robustness of our post-correction LMs against

Algorithm 1 PGD algorithm

```
Require: $\epsilon$ – maximum perturbation,
    $\alpha$ – step size,
    $N$ – number of steps,
    $y_{\text{target}}$ – adversarial text,
    $x$ – original input audio,
    $\varkappa$ – amplitude range of $x$,
    $x_1$ – copy of $x$.
Ensure: adversarial perturbed audio.
for $t \in [1, N]$ do
    $x_{t+1} \leftarrow \text{clip}_{\varkappa,\epsilon}(x_t - \alpha \,\text{sign}\, \nabla_x \, \text{L}_{\text{CTC}}(f(x_t), y_{\text{target}}))$
end for
return $x_N$
```

them. For this purpose, we attack the Whisper ASR model using PGD and CTC-loss in an autoregressive regime. We took the audio, converted it into a tensor, then iteratively attacked it with PGD and saved the result. Then, we decoded the attacked audio as usual and transferred it into LLM.

§4. Dataset

We used `HyPoradise`[1] train and test splits made by authors, except for `LibriSpeech-clean` and `WSJ`, which test splits were used as validation sets in our experiments. All Whisper transcriptions for Table 2 were also provided by authors. Most of the info can be taken from section 2.2. In `HyPoradise` dataset, from 1.3% to 13% of tokens are not presented in hypotheses. As we can see in Figure 2a, `CORAAL` and `Common Voice` sub-datasets have about 9% and 13% of tokens that are not presented in 1-5 transcriptions, while `TED3` and `LRS2` number of this tokens is low. In most cases, there are not a lot of tokens, which are not presented in the top 1 hypothesis, like in Figure 2b. Here, all datasets can be divided into 2 groups:

(1) `LRS2`, `LibriSpeech-other`, `TED3`, where probability almost the same.
(2) `ATIS`, `CHiME4`, `CORAAL`, `SwitchBoard`, `Common Voice`, where probabilities that token was not generated by ASR are higher.

[1] HyPoradise `https://huggingface.co/datasets/PeacefulData/HyPoradise-v0`HuggingFace Hub.

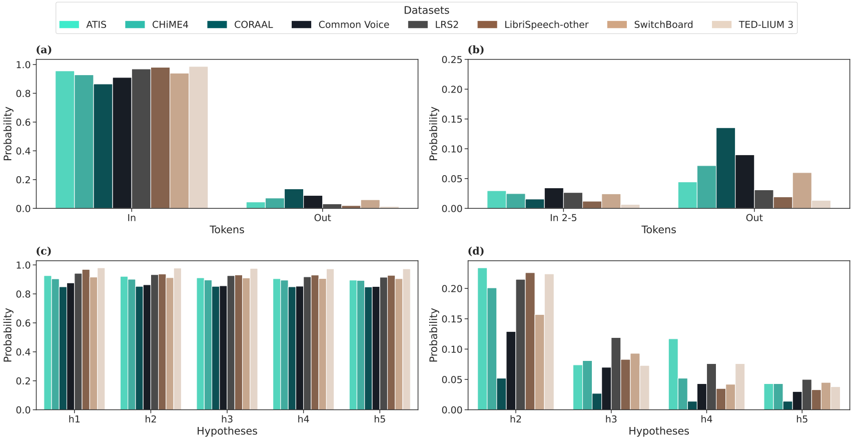

Figure 2. Probabilities of finding tokens from Ground Truth in or out of hypotheses. Each color represents a dataset from HyPoradise. Horizontal axes represent probabilities of: **(a)** token presented or not presented/out of hypotheses; **(b)** tokens in or out of hypotheses for the set of tokens exclusive for 2-5 hypotheses; **(c)** finding the GT token in each hypothesis; **(d)** in 2-5 provided that we no longer consider a token in hypothesis i if it has already occurred in hypothesis $j > i$.

Additionally, we checked the distribution of tokens in the audio transcription for the test sets of the HyPoradise dataset. Figure 2c shows that the concentration of GT-text tokens out of the total number of words in all five hypotheses was high. Also, we saw that some tokens might be included in hypotheses $n + 1$ if they were not included in n, as shown in Figure 2d. If a word is included in hypothesis n, we break the loop. Otherwise, we check if it is included in $n+1$. For example, if a token is included in both hypothesis n and in hypothesis $n+1, n+2, \ldots$, we count it only in hypothesis n and not in $n + 1, n + 2, \ldots$. In all sub-datasets except ATIS, the probability of finding a token for GT that has not been previously submitted slightly decreases. For CORAAL this probability is relatively low, but the problem with this dataset is poor quality of transcription made by the ASR model.

Also, we added the data from the [35][2] to increase the number of speakers and improve generalization. We randomly chose the ASR model among Whisper-Small and Whisper-Base instead of Whisper-Large to reduce transcription quality as an augmentation technique. The ASR system receives audio, pre-processes it, encodes it into the sequence of feature representation, and processes it in the decoder. For example, Whisper loads the audio, pads it, trims it, and converts it to a Mel Spectrogram. For our experiments, we collect the hypotheses of the top 5 ASR transcriptions (regarding the sequence's log prob) for the given audio using Beam-Search. The text has been processed using a normalizer: all letters have been converted to lowercase, all numbers have been replaced by letters and punctuation marks have been removed. The weights of ASR models were frozen as we did not fine-tune them during our experiments.

ASR transcriptions were partially augmented to simulate two common error types: random deletion of a word and random insertion. Insertion was made using context BERT embeddings with the NLPAug [7] library. These augmentations forced the model to check the relevant information in distributed hypotheses, not only the most probable one.

We considered the Mozilla's `Common Voice 18 (CV18)`[3] English part for the second branch of our experiments. We took 900000 audios for training and validation. Each audio was randomly augmented with several transformations: voice activity detection, Gaussian noise, room impulse response (RIR), gain, polarity inversion, pitch shift, low-pass filtering, color noise, and audio pre-emphasis. Pre-emphasis is defined as follows: $x_l = x_l - \gamma x_{l-1}$, where $l = 1, L$ is an amplitude index of a given audio of length L, and γ is the pre-emphasis factor (we used $\gamma = 0.97$). Each enhancement was applied independently. Augmented train audios were transcribed with Whisper Tiny, Base, and Medium. To better evaluate the generalization power of our model, the non-augmented test audios (`CV18` test set) were transcribed with Whisper Small, Large, and Wav2Vec2 base models. We provided the LLM-only top 1 ASR hypothesis in this branch of experiments.

§5. Experiments

5.1. Metrics. The primary metric is already mentioned WER, which computes word-level Levinstein distance (7)

[2] Crowdsourced high-quality UK and Ireland English Dialect speech data set - `https://openslr.org/83/`OpenSLR.

[3] Common Voice `https://commonvoice.mozilla.org/ru/datasets`

$$WER = \frac{S + D + I}{N} = \frac{S + D + I}{S + D + C}, \tag{7}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions, C is the number of correct words, and N is the number of words in the reference text.

5.2. Experiment Setup. We used LoRA to fine-tune Flan-T5 and Qwen, allowing fast fine-tuning with limited resources. We considered several layers to optimize with LoRA in our Flan-T5 model. Moreover, we used NEFTune with alpha `0.1`. α value adjustment was performed on the same validation set, and a subset of the training set was used for model fine-tuning. We tested several α values and achieved the lowest validation WER with `0.1`, so this value was used to fine-tune T5 model in other experiments. The results of fine-tuned Flan-T5 with and without NEFTune models are presented in Table 2.

A PGD attack, explained in Section 3.4, was performed for each audio. To create adversarial audios, we allowed the gradient to flow from raw amplitudes to final logits during model inference. We took the logits, calculated the CTC loss, entered an α value equal to 0.007, and clamped the audio from 0 to 1 and from 0 to a maximum perturbation equal to 0.07. We repeat with 15 steps. This attack result demonstrates the ability of our model to post-correct poor hypotheses and the robustness of our model.

We used AdamW optimizer [27] with learning rate `1e-4` and `0.1` warm-up ratio, NVidia's fused implementation, which combines all Adam operations in one. Forward pass was mixed-precision [13] with `bfloat16` type [14]. The batch size for both training and validation was set to 16. The inference beam-search has 6 beams, and sampling parameters `temperature 0.8`, `top_k 40`, `top_p 0.75`.

5.3. Baselines. The default results are obtained with ASR models. For the baseline, we considered HyPoradise Llama [4] and ROVER [37] method. However, ROVER managed to improve WER only for several sub-datasets, thus we have decided not to include it in the main text. For Llama we took an original prompt template and eval script from their GitHub repository [5].

[4] Llama HyPoradise `https://huggingface.co/GenSEC-LLM/SLT-Task1-Llama2-7b-HyPo-baseline`

[5] HyPoradise GitHub `https://github.com/Hypotheses-Paradise/Hypo2Trans?tab=readme-ov-file`

Table 2. WER (%) results of different target modules for LoRA and NEFTune fine-tuning with T5 model. Lower is better.

Test Set	**Whisper**	**HyPo**	**QV**	**QKVO**	**Full**	**Full+ NEFTune**
ATIS	8.40	**1.12**	2.72	1.67	1.75	**1.58**
CHiME4	12.05	**4.76**	6.21	5.32	6.00	**4.84**
CORAAL	24.38	23.42	23.65	27.56	23.60	**22.47**
CV	15.72	13.66	11.4	11.39	11.01	**10.85**
LRS2	12.89	16.07	10.18	10.16	9.93	**8.72**
LS-other	**5.15**	11.57	5.29	5.30	5.36	8.37
SWBD	17.03	**14.75**	15.45	**14.75**	15.24	14.94
TD-3	4.77	9.12	**4.03**	4.09	4.20	4.54
Avg.	12.44	11.8	9.86	10.03	9.63	9.54

§6. Results

This section compares our models with Whisper's and Wav2Vec2's top 1 hypothesis, and demonstrates the result on the `HyPoradise` and `CV` datasets.

The results on `HyPoradise` dataset test split are presented in Table 2. Our model outperforms Whisper on all test sub-datasets except for `LibriSpeech-other`. On `ATIS` sub-datasets, the WER of our model compared to Whisper is 80% smaller; this could be the cause of the dataset's nature, such as Whisper translating mainly audio without splitting some place names into parts. For example, Whisper writes ”washington dc” when the GT variant is ”washington d c,” and there are a lot of geographical places in this dataset. The second dataset by WER improvement is `CHiME4`. This one consists of audio with different background noises, sometimes leading to inaccuracy or even hallucinations. However, 1 to 5 hypotheses still provide enough information for LLM to correct errors. Compared to HyPo, our model demonstrates close results to the competitor for the sub-datasets where HyPo has lower WER. However, for `CV`, `LS-other`, `LRS2` and `TD-3` sub-datasets, our model significantly outperforms HyPo.

We also present model fine-tuning results with LoRA and NEFTune. First, we show which LLM's target module yields the largest WER improvement, and second, how it affects the quality of the post-correction

Table 3. Examples of ASR post-correction on CV18 test set. LLM is QWEN2-1.5B.

ASR Model	**Ground truth**	**ASR prediction**	**LLM correction**	**WER ASR**	**WER LLM**
Whisper-Base	the chersky range is part of the south siberian system	the cheer sky range is part of the south cerebrian system	the chersky range is part of the south siberian system	30.0	20.0
Whisper-Base	interment in the woodlands cemetery	intermined in the woodland symmetry	interment in the woodland cemetery	60.0	20.0
Whisper-Base	basil of annonay france	basil of annoyed france	basil of annonay france	25.0	0.0
Whisper-Base	bundesliga where he played for two seasons	bunder siglor where he played for two seasons	bundesliga where he played for two seasons	28.6	0.0
Whisper-Large	he was born in pittsburgh	hΓ¤n oli nainen pittsburghissa	he had been in pittsburgh pennsylvania	100.0	60.0
Wav2Vec2	its stigmas are bilobed	tstigmas are billo bed	its stigmas are bilobed	100.0	0.0
Wav2Vec2	flett played rugby union for edinburgh university	lit play rug beunon far edinburg universipi	flett played rugby union for edinburgh university	100.0	0.0
Wav2Vec2	i am sure of it	am shored offit	i am sure of it	80.0	0.0
Wav2Vec2	the area is roughly triangular	he areas rochly criangular	the area is roughly craggy	100.0	20.0

Table 4. WER on augmented Common Voice EN test set for different ASR models with and without post-correction. Lower is better.

LLM\ASR	**Whisper-Small**	**Whisper-Large**	**Wav2Vec2-Base**
No LLM	30.1	28.9	43.9
Qwen2-1.5B	19.8	18.5	30.5
Qwen2-0.5B	21.6	18.5	32.7

in Table 2. Here, `QV` and `QKVO` refer to attention modules we fine-tune, and "Full" refers to all linear modules, i.e., attention and post-attention `T5LayerFF` parts. So `Full` can be interpreted as `QKVO + wi_0 + wi_1 +`

Table 5. WER on different datasets. The LLMs were fine-tuned on the CV18 EN train set. Lower is better.

LLM\Dataset	**ATIS**	**CHiME4**	**CORAAL**	**LRS2**
No LLM	18.2	18.0	27.4	25.9
Qwen2-1.5B	13.5	17.0	54.6	15.9
Qwen2-0.5B	12.6	17.4	64.6	17.7

`wo`. One can see that tuning different modules can sometimes lead to decreased and increased WER on different datasets, like `QKVO`, compared to `QV` reduced WER on `ATIS` but increased it on `LibriSpeech-other`. Overall, "Full+NEFTune" performs best even though embeddings with LoRA are frozen during training. "Full+NEFTune" we used as "Ours" in Table 2. This shows that NEFTune with LoRA can effectively fine-tune the LLM for post-correction with less resource utilization.

In addition, we compared different ASR-only systems to ASR-LLM Post-Correction with small language models, namely built by Alibaba Qwen2-0.5B and Qwen2-1.5B, on `CV18` dataset. Also, we checked how LLM fine-tuning on one dataset can affect performance on other out-of-domain data. See Table 4. As we can see, even the 0.5B model can perform post-correction effectively, reducing WER from 43.9 to 32.7 on Wav2Vec2 on the `CV18` dataset. These results demonstrate that even a relatively small LM with a reach language domain can perform WER reduction for all three ASR systems tested, so the method can also be applied to Whisper and Wav2Vec. Examples of Qwen correction are presented in Table 3.

6.1. Robustness of models. Another experiment was to attack Whisper on a subset of `CV` with a PGD attack to evaluate the robustness of the ASR model against adversarial attacks and, more importantly, the robustness of our post-correction LLMs against adversarially obtained transcriptions. Additionally to our model, we tested the HyPo model using the same data to measure its robustness. Both models improve transcription quality even in the presence of various noises in the text data, but our one is slightly better (WER for Whisper - 24.38, HyPo - 23.59, Ours - 22.37).

This could be due to NEFTune regularization, which adds noise to model embeddings.

Also, we tested the inference of Qwen, which we trained on CV18, on the out-of-domain datasets: ATIS, CHiME4, CORAAL, and LRS2. Our trained small language models improve transcription quality on unseen data, except for CORAAL, whose transcription is still quite difficult. The results are summarized in Table 5.

These two examples demonstrate the robustness of our methods when dealing with unseen or attacked data.

6.2. Computational overhead. The computational cost of our LLMs is relatively small compared to the ASR model. Whisper requires about 4.2 seconds for the CV example with beam-search decoding, while Flan-T5 post-correction takes 1.8 seconds. Thus, ASR with post-correction takes about 6 seconds on average, less than 1.5 times longer than the average ASR. In addition, we can try to use different implementations of attention [47, 48] or quantization [49], which can optimize the inference time and memory with a possible small quality degradation.

6.3. Limitations. We should admit that our study has several limitations. The LLM model heavily depends on relatively good hypotheses produced by ASR models because these hypotheses are the only source for the language model to perform post-correction, except for its general language understanding. The post-correction LLM cannot fix the text well if the ASR model has serious hallucinations due to loud noise or silence. In addition, if train and test domains differ significantly (e.g., average train sentence length is much smaller than in the test, such as CV and CORALL), the results might be negative. But despite all of that, overall WER reduction leads to more accurate translations in most cases, regardless of the audio source.

§7. Conclusion

In this article, we tackled the problem of ASR post-correction. We proposed using the Flan-T5 and Qwen2 models, fine-tuning them on a combination of datasets with comprehensive audio and text-level augmentations. Our approach successfully corrects errors in several ASR systems, enhancing transcription from diverse audio domains distinct from the train set speech domain. Our methods demonstrated prominent and robust ASR post-correction results even against adversarial attacks.

Future work might be devoted to language classification, expanding the procedure to a multilingual setting, quality improvement, punctuation, and model distillation to make it suitable for low-resource applications.

References

1. H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, *Llama 2: Open Foundation and Fine-Tuned Chat Models*, ArXiv preprint arXiv:2307.09288 (2023).
2. A. Madry, *Towards Deep Learning Models Resistant to Adversarial Attacks*, ArXiv preprint arXiv:1706.06083 (2017).
3. A. Hannun, *et al.*, *Deep Speech: Scaling Up End-to-End Speech Recognition*, ArXiv preprint arXiv:1412.5567 (2014).
4. R. Olivier and B. Raj, *Recent Improvements of ASR Models in the Face of Adversarial Attacks*, Interspeech, 2022. Available: `https://arxiv.org/abs/2203.16536`.
5. 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).
6. Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, J. Zhou, *Qwen-Audio: Advancing Universal Audio Understanding via Unified Large-Scale Audio-Language Models*, ArXiv preprint arXiv:2311.07919 (2023).
7. E. Ma, *NLP Augmentation*, 2019. Available: `https://github.com/makcedward/nlpaug`.
8. S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, *et al.*, *CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings*, ArXiv preprint arXiv:2004.09249 (2020).
9. A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, *wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations*, in: Advances in Neural Information Processing Systems, 2020, pp. 12449–12460.
10. J. Bai, *et al.*, *Qwen Technical Report*, ArXiv preprint arXiv:2309.16609 (2023).
11. E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *LoRA: Low-Rank Adaptation of Large Language Models*, ArXiv preprint arXiv:2106.09685 (2021).
12. N. Jain, *et al.*, *Neftune: Noisy Embeddings Improve Instruction Finetuning*, ArXiv preprint arXiv:2310.05914 (2023).
13. P. Micikevicius, S. Narang, J. Alben, G. Diamos, E. Elsen, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, *et al.*, *Mixed Precision Training*, ArXiv preprint arXiv:1710.03740 (2017).
14. D. Kalamkar, *et al.*, *A Study of BFLOAT16 for Deep Learning Training*, ArXiv preprint arXiv:1905.12322 (2019).
15. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding*, ArXiv preprint arXiv:1810.04805 (2018).
16. A. Radford, *Improving Language Understanding by Generative Pre-Training*, 2018.

17. A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, *Language Models Are Unsupervised Multitask Learners*, 2019.
18. A. Gulati, *et al.*, *Conformer: Convolution-Augmented Transformer for Speech Recognition*, ArXiv preprint arXiv:2005.08100 (2020).
19. S. Schneider, A. Baevski, R. Collobert, and M. Auli, *wav2vec: Unsupervised Pre-Training for Speech Recognition*, ArXiv preprint arXiv:1904.05862 (2019).
20. D. Dai, Y. Sun, L. Dong, Y. Hao, S. Ma, Z. Sui, and F. Wei, *Why Can GPT Learn In-Context? Language Models Implicitly Perform Gradient Descent as Meta-Optimizers*, ArXiv preprint arXiv:2212.10559 (2022).
21. C. Raffel, *et al.*, *Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer.* — Journal of Machine Learning Research (2020), 1–67.
22. R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, *Can Generative Large Language Models Perform ASR Error Correction?*, ArXiv preprint arXiv:2307.04172 (2023).
23. J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, *Finetuned Language Models Are Zero-Shot Learners*, ArXiv preprint arXiv:2109.01652 (2021).
24. A. Graves, S. FernΓЎndez, F. Gomez, and JΓjrgen Schmidhuber, *Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks*, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 369–376.
25. C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, *Salmonn: Towards Generic Hearing Abilities for Large Language Models*, ArXiv preprint arXiv:2310.13289 (2023).
26. B. T. Polyak, *Some Methods of Speeding Up the Convergence of Iteration Methods.* — USSR Computational Mathematics and Mathematical Physics (1964), 1–17.
27. I. Loshchilov, *Decoupled Weight Decay Regularization*, ArXiv preprint arXiv:1711.05101 (2017).
28. D. P. Kingma, *Adam: A Method for Stochastic Optimization*, ArXiv preprint arXiv:1412.6980 (2014).
29. W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, *HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units.* — IEEE/ACM Transactions on Audio, Speech, and Language Processing (2021), 3451–3460.
30. M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, *Square Attack: A Query-Efficient Black-Box Adversarial Attack via Random Search*, in: European Conference on Computer Vision, 2020, pp. 484–501.
31. I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and Harnessing Adversarial Examples*, ArXiv preprint arXiv:1412.6572 (2014).
32. C. Szegedy, *Intriguing Properties of Neural Networks*, ArXiv preprint arXiv:1312.6199 (2013).
33. C. Chen, *et al.*, *HyPoradise: An Open Baseline for Generative Speech Recognition with Large Language Models*, Thirty-Seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track, 2023.

34. A. Radford, J. Wook Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, *Robust Speech Recognition via Large-Scale Weak Supervision*, in: International Conference on Machine Learning, 2023, pp. 28492–28518.
35. I. Demirsahin, O. Kjartansson, A. Gutkin, and C. Rivera, *Open-Source Multi-Speaker Corpora of the English Accents in the British Isles*, in: Proceedings of The 12th Language Resources and Evaluation Conference (LREC), 2020, pp. 6532–6541.
36. C. T. Hemphill, J. J. Godfrey, and G. R. Doddington, *The ATIS Spoken Language Systems Pilot Corpus*, in: Speech and Natural Language: Proceedings of a Workshop Held at Hidden Valley, Pennsylvania, June 24–27, 1990, 1990.
37. J. G. Fiscus, *A Post-Processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)*, in: 1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings, 1997, pp. 347–354.
38. J. S. Chung, A. Senior, O. Vinyals, and A. Zisserman, *Lip Reading Sentences in the Wild*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6447–6456.
39. V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, *LibriSpeech: An ASR Corpus Based on Public Domain Audio Books*, in: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015, pp. 5206–5210.
40. J. J. Godfrey, E. C. Holliman, and J. McDaniel, *SWITCHBOARD: Telephone Speech Corpus for Research and Development*, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on, 1992, pp. 517–520.
41. FranΓ§ois Hernandez, *et al.*, *TED-LIUM 3: Twice as Much Data and Corpus Repartition for Experiments on Speaker Adaptation*, in: Speech and Computer: 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18–22, 2018, Proceedings 20, 2018, pp. 198–208.
42. D. B. Paul and J. Baker, *The Design for the Wall Street Journal-Based CSR Corpus*, in: Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23–26, 1992, 1992.
43. J. Meyer, L. Rauchenstein, J. D. Eisenberg, and N. Howell, *Artie Bias Corpus: An Open Dataset for Detecting Demographic Bias in Speech Applications*, in: Proceedings of the Twelfth Language Resources and Evaluation Conference, 2020, pp. 6462–6468.
44. D. Amodei, *et al.*, *Deep Speech 2: End-to-End Speech Recognition in English and Mandarin*, in: International Conference on Machine Learning, 2016, pp. 173–182.
45. W. Chan, N. Jaitly, Q. Le, and O. Vinyals, *Listen, Attend and Spell: A Neural Network for Large Vocabulary Conversational Speech Recognition*, in: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016, pp. 4960–4964.
46. C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, *et al.*, *State-of-the-Art Speech Recognition with Sequence-to-Sequence Models*, in: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4774–4778.

47. T. Dao, D. Y. Fu, S. Ermon, A. Rudra, and C. Ré, *FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness*, in: Advances in Neural Information Processing Systems (NeurIPS), 2022.
48. T. Dao, *FlashAttention-2: Faster Attention with Better Parallelism and Work Partitioning*, in: International Conference on Learning Representations (ICLR), 2024.
49. J. Lin, J. Tang, H. Tang, S. Yang, W.-M. Chen, W.-C. Wang, G. Xiao, X. Dang, C. Gan, and S. Han, *AWQ: Activation-Aware Weight Quantization for LLM Compression and Acceleration*, MLSys, 2024.
50. X. Zhang, H. Tan, X. Huang, D. Zhang, K. Tang, and Z. Gu, *Adversarial Example Attacks Against ASR Systems: An Overview*, in: 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC), 2022, pp. 470–477.
51. K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
52. N. Carlini and D. Wagner, *Audio Adversarial Examples: Targeted Attacks on Speech-to-Text*, in: 2018 IEEE Security and Privacy Workshops (SPW), 2018, pp. 1–7.
53. R. Xiong, Y. Yang, D. He, K. Zheng, S. Zheng, C. Xing, H. Zhang, Y. Lan, L. Wang, and T. Liu, *On Layer Normalization in the Transformer Architecture*, in: International Conference on Machine Learning, 2020, pp. 10524–10533.
54. W. S. McCulloch and W. Pitts, *A Logical Calculus of the Ideas Immanent in Nervous Activity.* — The Bulletin of Mathematical Biophysics (1943), 115–133.
55. J. Orbach, *Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms.* — Archives of General Psychiatry (1962), 218–219.
56. S. Radhakrishnan, C.-H. Huck Yang, S. A. Khan, R. Kumar, N. A. Kiani, D. Gomez-Cabrero, and J. N. Tegner, *Whispering LLaMA: A Cross-Modal Generative Error Correction Framework for Speech Recognition*, ArXiv preprint arXiv:2310.06434 (2023).
57. R. Ma, M. Qian, M. Gales, and K. Knill, *ASR Error Correction Using Large Language Models*, ArXiv preprint arXiv:2409.09554 (2024).
58. L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, *Training Language Models to Follow Instructions with Human Feedback*, in: Advances in Neural Information Processing Systems, 2022, pp. 27730–27744.
59. A. Vaswani, *Attention Is All You Need*, in: Advances in Neural Information Processing Systems, 2017.
60. R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, *Common Voice: A Massively-Multilingual Speech Corpus*, ArXiv preprint arXiv:1912.06670 (2019).
61. P. Rahmanzadehgervi, L. Bolton, M. R. Taesiri, and A. T. Nguyen, *Vision Language Models Are Blind*, ArXiv preprint arXiv:2407.06581 (2024).
62. C. Farrington and N. Schilling, *Contextualizing the Corpus of Regional African American Language, DC: AAL in the NationвЂ™s Capital.* — American Speech: A Quarterly of Linguistic Usage (2019), 21–35.

63. S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. N. Sainath, and K. Livescu, *A Comparison of Techniques for Language Model Integration in Encoder-Decoder Speech Recognition*, in: 2018 IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 369–375.

AIRI,
Moscow Technical University
of Communications and Informatics
E-mail: iudin@airi.net

AIRI, Skoltech
E-mail: Matvey.Skripkin@airi.net
E-mail: o.rogov@airi.net
E-mail: korzh@airi.net

Поступило 28 февраля 2025 г.