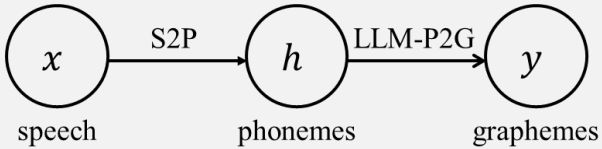
LLM-based phoneme-to-grapheme for phoneme-based speech recognition



Te Ma[1], Min Bi[2], Saierdaer Yusuyin[1], Hao Huang[1], Zhijian Ou[*3]

1 School of Computer Science and Technology, Xinjiang University, China

2 Guangxi Radio and Television Monitoring Center, Guangxi, China

3 Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, China

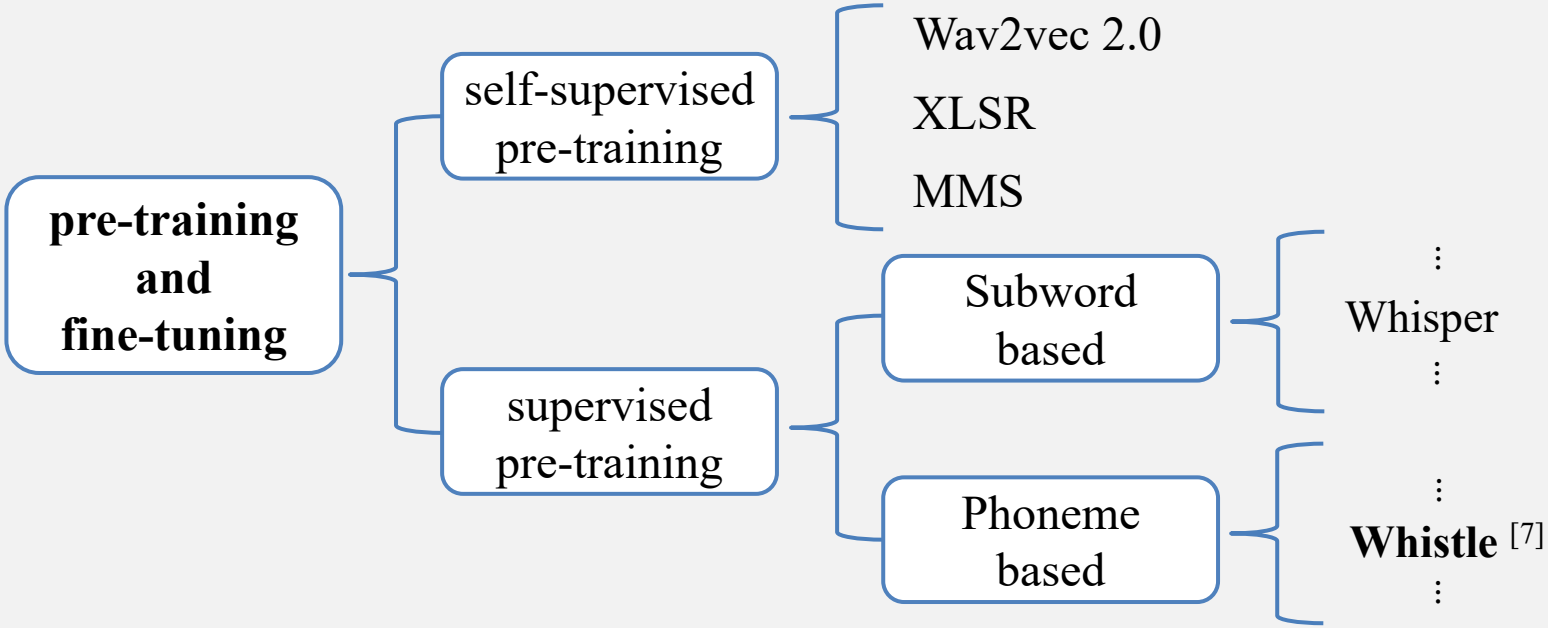
Content

1. Motivation & Background
2. Related Work
3. Method
4. Experiment
5. Conclusion

1. Motivation & Background

Motivation & Background

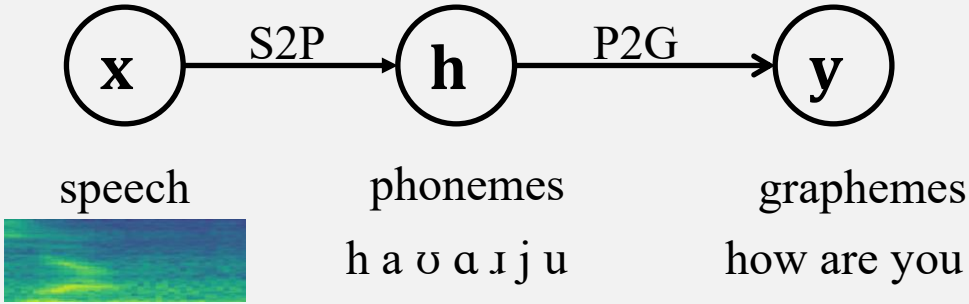
- **Data-hungry**: requiring large amounts of labeled speech for model training
- **Data-efficient**: well performance given small amounts of labeled speech



[7] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, "Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision," IEEE Transactions on Audio, Speech and Language Processing, pp. 1–14, 2025.

Motivation & Background

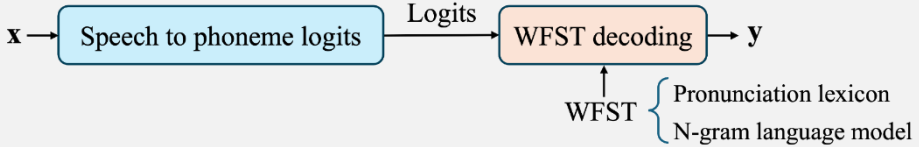
- It is found in **Whistle** [7] that when cross-lingual fine-tuning data is more limited, phoneme-based supervised pre-training achieves the most competitive results and provides high data-efficiency.
  - Presumably, this is because phoneme-based supervision enables more efficient **data sharing** than subword-based supervision.
- We speculate that modeling phonemes as an interface between speech and text in the ASR pipeline, serving as a **structural constraint**, significantly reduces the problem complexity.



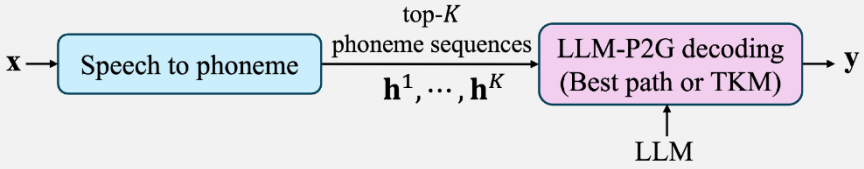
[7] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, "Whistle:Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision," IEEE Transactions on Audio, Speech and Language Processing, pp. 1–14, 2025.

Motivation & Background

- However, challenges remain for decoding in **phoneme-based ASR**.
- The widely used Weighted Finite State Transducer (WFST) based decoding has **two major drawbacks**:
  - It involves a complex pipeline, which needs construction of pronunciation lexicons and compiling of WFSTs;
  - It is not easy to effectively leverage the rich linguistic knowledge in large language models (LLMs).



(a) Phoneme-based ASR with WFST decoding



(b) Phoneme-based ASR with LLM-P2G decoding

- P2G models can be naturally trained over LLMs
- Simplifying decoding pipeline

2. Related Work

Related Work

- **The two-step idea of phoneme-based ASR** (recognizing speech to phonemes and then to graphemes) has been studied for ASR [13,14]
  - with a similar motivation as ours
  - However, prior studies do not explore using LLMs for P2G for phoneme-based ASR.
- **How to integrate LLMs into ASR?**

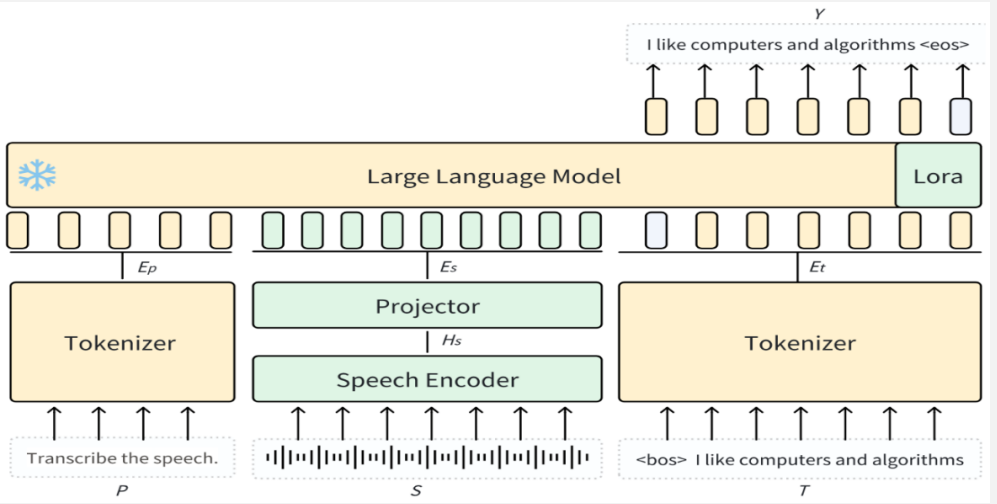
  Different interfaces between speech and languages have been studied:
  - ASR-generated text [16]: heavier than phonemes
  - Continuous embeddings of speech [17, 18, 19]

[13] "TranUSR: Phoneme-to-word transcoder based unified speech representation learning for cross-lingual speech recognition," INTERSPEECH, 2023.
[14] "Optimizing two-pass crosslingual transfer learning: Phoneme recognition and phoneme to grapheme translation," ASRU, 2023.

[16] "Can generative large language models perform ASR error correction?" arXiv, 2023.
[17] "On decoder-only architecture for speechto-text and large language model integration," ASRU 2023.
[18] "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," arXiv 2023.
[19] "An embarrassingly simple approach for LLM with strong ASR capacity," arXiv 2024.

Related Work

- **Speech Encoder + Projector + LLM:**
  - Black box
  - Needs additional projection, which is not easy, not natural



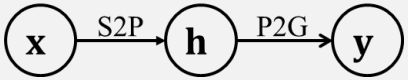
- **Our approach (LLM-P2G):**
  - Explainable
  - Natural integration
    - Both phonemes and subwords are discrete tokens
    - The IPA symbols all fall in the token set of mT5 - the LLM used in our experiments.
    - It is found that the P2G capability (called IPA transliterate) emerges in LLMs[15]

LLMs are well suited to P2G !

[15] “Emergent abilities of large language models,” Transactions on Machine Learning Research, 2022.

3. Method

Method overview



A two-step ASR architecture, which we refer to as **SPG** (speech-to-phoneme-to-grapheme):

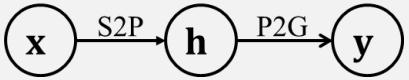
$$p(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h}} p(\mathbf{h}|\mathbf{x})p(\mathbf{y}|\mathbf{h}) \qquad (1)$$

where **x, y** and **h** denote speech, grapheme and phoneme sequence.

- **S2P (speech-to-phoneme)**
  - Modeled by $p(\mathbf{h}|\mathbf{x})$, CTC-based in our experiments
  - Obtained by fine-tuning a phoneme-based multilingual S2P backbone (Whistle) [7] over speech data with phoneme labels, which we refer to as *Whistle-S2P*.
- **P2G (phoneme-to-grapheme)**
  - Modeled by $p(\mathbf{y}|\mathbf{h})$
  - Obtained by fine-tuning an LLM

[7] S. Yusuyin, T. Ma, H. Huang, W. Zhao, and Z. Ou, "Whistle: Data-efficient multilingual and crosslingual speech recognition via weakly phonetic supervision," IEEE Transactions on Audio, Speech and Language Processing, pp. 1–14, 2025.

Method overview



A challenge in building the two-step ASR

- There seems to have *information loss* in cascading S2P and P2G.
- In decoding, the hypothesized phoneme sequence output from S2P may contain errors, which may propagate to P2G.

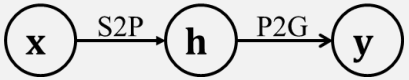
We propose two training strategies

- Data Augmentation with Noisy Phonemes (DANP)
- Top-K Marginalized (TKM) training and decoding

Data Augmentation with Noisy Phonemes (DANP)

- **Motivation**
  - If the P2G model is fine-tuned/trained using only the single annotated phoneme sequence, then there is a severe **mismatch** in training and testing for ASR.
  - The input phonemes fed to P2G in ASR testing is much **nosier**.
  - Therefore, a straightforward strategy to compensate for such a mismatch is to add noise to the input phonemes fed to P2G in training.
- **Training**
  - For CTC-based S2P, we perform *beam search* or *sampling* to generate K hypothesized phoneme sequences
  - To train P2G, the LLM is fine-tuned on K pairs of (phoneme, grapheme) sequences.
- **Decoding**
  - Using the best phoneme sequence from S2P, which is fed to P2G for decoding, called *Best Path Decode*

Top-K Marginalized (TKM) **training**



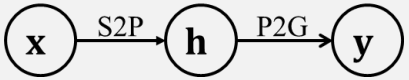
- **Motivation**
  - The DANP strategy mainly address *the mismatch in training and testing* when P2G decoding is based on the 1-best phoneme sequence generated from S2P.
  - Ideally, in P2G decoding, it would be better to marginalize over multiple hypothesized **h** to decode.
  - The RAG-Sequence technique in Retrieval-Augmented Generation (RAG)
    - Treat the retrieved document as a latent variable that is marginalized to get *the marginal likelihood via a top-K approximation*

We maximize *the marginal likelihood p(**y**|**x**) via a top-K approximation*

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{h} \in \text{top } K(\mathrm{p}(\mathbf{h}|\mathbf{x}))} \mathrm{p}(\mathbf{h}|\mathbf{x})\mathrm{p}(\mathbf{y}|\mathbf{h}) = \sum_{k=1}^{K} \mathrm{p}(\mathbf{h}^{(k)}|\mathbf{x}) \prod_{i=1}^{L} p(y_i|\mathbf{h}^{(k)}, y_{1:i-1})$$

[21] "Retrieval-augmented generation for knowledge-intensive NLP tasks," NeurIPS, 2020.

Top-K Marginalized (TKM) **decoding**

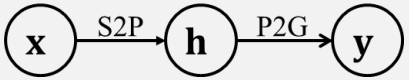


- We first run S2P beam search to obtain a set of top-K phoneme sequences $\mathbf{h}^{(1)}, \cdots, \mathbf{h}^{(K)}$
- For each phoneme sequence candidate $\mathbf{h}^{(k)}$, the LLM-P2G generates S text predictions, producing a total of $K \times S$ grapheme sequence results.
- After de-duplication, we obtain a set of grapheme sequences $\mathbf{Y}$. Each grapheme sequence $\mathbf{y} = y_1, \cdots, y_L$ from $\mathbf{Y}$ is scored as follows:

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) \approx \sum_{k=1}^{K} \mathrm{p}(\mathbf{h}^{(k)}|\mathbf{x}) \prod_{i=1}^{L} \mathrm{p}(y_i|\mathbf{h}^{(k)}, y_{1:i-1}) \quad (2)$$

- The top-S grapheme sequence in $\mathbf{Y}$ are obtained, which can be further re-scored, by combining Eq. (2) with in-domain language model scores.

Randomized Top-K Marginalized (TKM) **training**



- **A Variation of TKM training where more randomness is introduced**
  - Every time the training instance (**x, y**) is in a training minibatch, instead of always taking the top-K hypothesized phoneme sequences (as ranked by the S2P model), we randomly draw *n* hypothesized phoneme sequences from top-K for marginalization (n < K)
  - The training objective can be reformulated as:

$$\mathrm{p}(\mathbf{y}|\mathbf{x}) \approx \sum_{j=1}^{n} \mathrm{p}\left(\mathbf{h}^{(k_j)}|\mathbf{x}\right) \mathrm{p}\left(\mathbf{y}|\mathbf{h}^{(k_j)}\right)$$

    where $k_1, \cdots, k_n$ are uniformly drawn from $1, \cdots, K$
  - The advantages include
    - Better generalization, as it reduces over-reliance on a specific S2P ranking;
    - More robust to noisy S2P, as it helps when real-world S2P returns imperfect results.

4. Experiment

Experiment Setup

- **Baseline models**
  - *Using Whistle as the CTC-based acoustic backbone* (trained over CV 10 languages)
  - Fine-tuned with either phoneme or subword labels
  - WFST decoding
- **LLM-P2G model (ours)**:
  - **S2P model:** *Using Whistle as the CTC-based acoustic backbone*
  - **P2G model:** mT5-base (580 M)
- **Task:** Crosslingual speech recognition on Polish and German
- **Dataset:**
  - Common Voice 11.0 **Polish, German** (130h each): unseen languages for Whistle, in-domain languages for mT5.
- **Metric:** WER (Word Error Rate)

Main result: LLM-P2G based vs WFST-based

Table 1: *Word error rate (WER) comparison for Whistle fine-tuning (FT) models and LLM-P2G models. Results are shown for Polish and German, with and without language model (LM). Under any column, except "Whistle Subword FT", the other four rows share the same acoustic model (or say S2P), called the Whistle-S2P model. For Whistle models, "w/o LM" means beam search, while "w LM" means decoding with the WFST framework. For LLM-P2G, "w/o LM" means beam search, while "w LM" means using additional re-scoring with LM. NA denotes not applied.*

Model	Polish				German			
	130 h		20 h		130 h		20 h	
	w/o LM	w LM	w/o LM	w LM	w/o LM	w LM	w/o LM	w LM
Whistle Phoneme FT	NA	4.30	NA	16.27	NA	15.73	NA	30.71
Whistle Subword FT	5.84	3.82	**17.59**	**13.84**	14.09	14.01	**27.78**	**28.04**
LLM-P2G	5.71	5.04	23.75	21.56	14.76	14.39	32.26	31.45
LLM-P2G + DANP	4.44	4.18	19.99	19.05	13.86	13.63	30.49	29.97
LLM-P2G + randomized TKM	**4.01**	**3.68**	19.19	17.36	**13.44**	**13.03**	29.20	28.78

- **130h**: LLM-P2G with r-TKM: reducing WERs by 3.6% for Polish, 6.9% for German, with p-value=1e-4 (3.82 vs 3.68) and 8e-23 (14.01 vs 13.03)
- **20h**: the performance depends on the amounts of pre-training data and fine-tuning data; The percentages for German and Polish in mT5-base pre-training data are 3.05% and 2.15% respectively. (*not really a concern, LLM-P2G scales well with more data*)

Ablation study: DANP and TKM

Table 2: *Word error rates (WERs) for LLM-P2G with different settings of DANP. After de-duplication, the data size augmented by random sampling is about 32 times.*

DANP strategy	Polish		German	
	w/o LM	w LM	w/o LM	w LM
1-beam	5.71	5.04	14.76	14.67
sampling	5.09	4.93	14.82	14.65
32-beam	4.62	4.36	14.17	14.04
64-beam	4.72	4.36	14.17	13.97
32-beam + sampling	4.51	4.27	14.01	13.91
96-beam + sampling	4.66	4.26	13.86	13.64
+ multiple checkpoints	**4.44**	**4.18**	**13.86**	**13.63**

- Different data augmentation settings are compared; but cannot exploit multiple hypothesized **h** to decode.

Table 3: *Word error rates (WERs) for LLM-P2G with different settings of TKM training and decoding.*

TKM strategy	Polish		German	
	w/o LM	w LM	w/o LM	w LM
top-32	16.55	16.12	21.69	21.31
top-8	4.31	3.80	13.58	13.18
rand. 8 of top-32	**4.01**	**3.68**	**13.44**	**13.03**

- Random sampling of 8 sequences out of top-32 strategy achieves sufficient diversity while reducing noise, and obtains the best performance, which is the main result shown in Table 1.

Ablation study: TKM training and decoding

Table 4: *Comparison of word error rate (WERs) for LLM-P2G, using different training and decoding strategies. r-TKM denotes randomized TKM training.*

Lang.	Train	Best Path Decode w LM	TKM Decode w LM
Polish	DANP	4.18	4.06
	r-TKM	3.99	**3.68**
German	DANP	13.63	13.59
	r-TKM	13.42	**13.03**

- We compare two training methods: DANP and Randomized TKM, abbreviated as r-TKM.
- Each is evaluated using two decoding strategies: Best path (top-1), and TKM decoding (top-K).

5. Conclusion

Conclusion

- We propose **LLM-P2G,** a two-step ASR architecture, consisting of speech-to-phoneme (S2P) and LLM-based phoneme-to-grapheme (P2G).
- We propose **two training strategies**, which effectively overcomes potential information loss in cascading S2P and P2G models.
    - Data Augmentation with Noisy Phonemes (DANP)
    - Top-K Marginalized (TKM) training and decoding
- **Results:** LLM-P2G not only outperforms WFST-based ASR systems for crosslingual ASR but also simplifies the decoding pipeline.

The research question: what interface between speech and languages we should use?

Phonemes as a speech-language interface for ASR applications are promising

- can enable efficient decomposition and cooperation between machine ears and brain
- leveraging acoustic and language large pre-trained models
- offering an important direction for future research.

Thank You!

The code, models and data for LLM-P2G are released at

https://github.com/thu-spmi/CAT/blob/master/docs/whatsnew.md