APSIPA Transactions on Signal and Information Processing

223

An integrated framework for ASR post-processing with error correction, rescoring, and text-speech matching

Chin-Hung Kuo and Kuan-Yu Chen

Department of Computer Science and Information Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan

Received 31 July 2025
Revised 16 October 2025
18 November 2025
31 December 2025
Accepted 2 January 2026

Abstract

In recent years, automatic speech recognition (ASR) systems have become essential components of various computer applications, enabling seamless communication between humans and machines. *N*-best reranking and error correction are widely used post-processing techniques, each contributing to improving ASR output with distinct strengths. Building on these advantages, they propose a comprehensive framework designed to achieve superior performance. The framework begins with a text correction module to rectify and expand the original *N*-best list. This augmented list is then evaluated and reranked jointly by a text rescoring module and a text–speech matching module. The text rescoring module assesses hypotheses from a textual perspective, while the text–speech matching module evaluates each hypothesis from the alignment between speech and text. The authors explore three distinct strategies for implementing the text–speech matching module. Using the publicly available HypR benchmark, they fairly evaluated the proposed framework against other methods. Experimental results demonstrate the feasibility of the proposed framework and highlight the contributions of this study.

Keywords Automatic speech recognition, *N*-best reranking, Error correction, Text–speech matching

Paper type Research paper

1. Introduction

Automatic speech recognition (ASR) technology transforms human speech into text, enabling hands-free, intuitive operation of various devices (Hinton *et al.*, 2012; Jelinek, 1976; Rabiner and Juang, 1993). Its integration into systems allows users to control computers and other devices via voice commands, significantly enhancing usability and accessibility. ASR is pivotal in applications such as virtual assistants, smart home devices and interactive entertainment. Recent advancements in deep learning have enabled modern speech recognition models, trained on large and diverse data sets, to achieve near-human-

Funding: This work was supported by the National Science and Technology Council (NSTC), Taiwan, under Grants NSTC 112–2628-E-011–008-MY3 and NSTC 114–2640-B-002–005. Additional support was provided by the "Empower Vocational Education Research Center" at the National Taiwan University of Science and Technology (NTUST) through the Featured Areas Research Center Program, as part of the Higher Education Sprout Project funded by the Ministry of Education (MOE), Taiwan. The authors also acknowledge the National Center for High-Performance Computing, National Applied Research Laboratories (NARLabs), Taiwan, for providing essential computational and storage resources.

APSIPA Transactions on Signal and Information Processing
Vol. 15 No. 1, 2026
pp. 223-248
Emerald Publishing Limited
2048-7703
DOI 10.1108/ATSIP-07-2025-0069

level accuracy (Ao *et al.*, 2022; Hannun *et al.*, 2014; Radford *et al.*, 2023). However, deploying these models on resource-constrained devices poses significant challenges due to their large parameter sizes. Additionally, cloud-based speech recognition services are not universally viable, as they face issues related to domain mismatch, privacy concerns and operational costs. Addressing these challenges by improving recognition accuracy while maintaining model efficiency remains a key research focus.

Techniques such as *N*-best reranking (Cai *et al.*, 2023; Chiu and Chen, 2021b; Fohr and Illina, 2021; Salazar *et al.*, 2020; Shin *et al.*, 2019; Shivakumar *et al.*, 2023; Udagawa *et al.*, 2022; Xu *et al.*, 2022) and error correction (Du *et al.*, 2022; Dutta *et al.*, 2022; Errattahi *et al.*, 2018; Leng *et al.*, 2023; Leng *et al.*, 2021b; Leng *et al.*, 2021a; Ma *et al.*, 2023a; Mani *et al.*, 2020; Zhao *et al.*, 2021; Zhu *et al.*, 2021) serve as effective post-processing modules to improve speech recognition performance without modifying the ASR model architecture, training methodology or inference pipeline (Li *et al.*, 2023; Li *et al.*, 2019; Wang *et al.*, 2022b). This decoupling from the ASR system is a critical advantage, enabling these methods to be flexibly applied across different ASR systems.

N-best reranking approaches reevaluate the *N*-best hypotheses generated during ASR decoding. Instead of selecting the top-one hypothesis as the final result, these methods try to identify the hypothesis with the lowest error rate from *N*-best candidates as the output. In contrast, error correction models directly revise the ASR output by applying substitutions, insertions or deletions. Unlike *N*-best reranking methods, error correction models are not constrained by the set of *N*-best candidates and can leverage a broader vocabulary, making them particularly effective for addressing errors involving rare or out-of-vocabulary words. While error correction models offer greater flexibility and the potential for higher accuracy, *N*-best reranking methods are inherently more stable due to their limited search space.

Given the distinct strengths of *N*-best reranking methods and error correction models, their complementary nature suggests that combining these approaches could yield superior results. Building on this idea, several studies have demonstrated the effectiveness of such combinations on various benchmarks (Kang *et al.*, 2024; Kuo and Chen, 2022; Leng *et al.*, 2021b). However, a notable challenge arises: error correction models often overcorrect, producing hypotheses with strong linguistic coherence that can mislead the scoring during reranking. In other words, hypotheses modified by error correction models may be grammatically sound and semantically coherent yet deviate from the original speech content.

To address this challenge and advance the field, we propose a novel ASR post-processing framework in this study. The framework integrates a text correction module, a text rescoring module and a text–speech matching module in a pipeline. First, the error correction module generates new hypotheses by refining the *N*-best candidates produced by ASR. Next, the text rescoring module evaluates the quality of the wording and semantic coherence using text–level cues. Finally, to prevent hypotheses from drifting away from the original speech content, the text–speech matching module compares the hypotheses against the speech signal, ensuring alignment and enhancing overall accuracy. Furthermore, to implement the text–speech matching module, we design and present three distinct mechanisms, discussing their strengths and limitations in our experiments. The proposed framework integrates *c*orrection, *re*scoring and text–speech *m*atching, which we denote as CREAM for short. In our experiments, CREAM is evaluated on the HypR benchmark (Wang *et al.*, 2024) using the AISHELL-1 (Bu *et al.*, 2017), TEDLIUM-2 (Rousseau *et al.*, 2014) and LibriSpeech (Panayotov *et al.*, 2015) data sets to validate its feasibility across domains and languages.

2. Related work

To enhance recognition performance, numerous methods have been proposed to revise ASR hypotheses, which can be broadly categorized into two major approaches: *N*-best reranking and error correction.

2.1 N-best reranking methods

The primary objective of *N*-best reranking is to assign a reliable score to each hypothesis, ensuring that the score strongly negatively correlates with the error rate, where hypotheses with higher scores should correspond to lower error rates. By ranking hypotheses based on these scores, the optimal candidate from the *N*-best list is selected as the final output. Depending on the scoring granularity or computation approach, these methods can be classified into token-level, sentence-level and comparison-based approaches (Wang *et al.*, 2024).

Token-level methods use advanced language models to compute scores at the token level, evaluating the probability of each token within a hypothesis. Two key paradigms are causal language modeling (CLM) (Jelinek, 1998; Jurafsky and Martin, 2000) and masked language modeling (MLM) (Kenton and Toutanova, 2019; Xu *et al.*, 2022). For a hypothesis $X = \{x_1, x_2, \ldots, x_L\}$, CLM calculates the score by decomposing X into a sequence of conditional probabilities using the chain rule:

$$\text{Score}_{\text{CLM}}(X) = \prod_{i=1}^{L} P(x_i | x_{<i}), \tag{1}$$

where $x_{<i}$ represents the preceding tokens. While mathematically sound, CLM considers only unidirectional context, ignoring future tokens. In contrast, MLM leverages bidirectional context, computing scores based on the entire hypothesis:

$$\text{Score}_{\text{MLM}}(X) = \prod_{i=1}^{L} P(x_i | x_{\setminus i}), \tag{2}$$

where $x_{\setminus i}$ denotes the hypothesis with x_i masked.

Sentence-level methods represent entire hypotheses as vectors and convert these representations into scores for ranking (Chiu and Chen, 2021a; Xu *et al.*, 2022). During training, ground truths such as error rates, rankings or token-level scores are used to guide the model parameter updates through supervised learning. Compared to token-level methods, sentence-level approaches treat each hypothesis as a single unit, reducing computational overhead during inference. Moreover, advanced sentence embedding techniques can distill lexical, semantic and latent information into compact vector representations (Conneau *et al.*, 2017; Kenton and Toutanova, 2019; Logeswaran and Lee, 2018), making sentence-level methods both practical and efficient for generating reliable scores.

Comparison-based methods reformulate *N*-best reranking as a pairwise comparison task (Fohr and Illina, 2021; Illina and Fohr, 2021; Ogawa *et al.*, 2018). These methods evaluate hypotheses by performing one-on-one comparisons, identifying the candidate with the most "wins" as the best option. This approach simplifies the reranking process by focusing on relative correctness, avoiding the complexities of absolute scoring. However, comparison-based methods often come with higher computational costs due to the large number of pairwise evaluations.

2.2 Error correction models

Error correction models improve ASR performance by directly detecting and correcting recognition errors. These models typically operate in a pipeline combining error detection and correction modules (Leng *et al.*, 2023; Leng *et al.*, 2021b; Leng *et al.*, 2021a; Wang *et al.*, 2022a). With advances in deep learning, research has increasingly focused on end-to-end sequence-to-sequence frameworks that streamline the process and minimize error propagation. Recently, modern approaches often leverage powerful pre-trained language models, such as BART and bidirectional encoder representations from transformers (BERT), as foundational components to enhance performance (Dutta *et al.*, 2022; Hrinchuk *et al.*, 2020; Shunmuga Priya *et al.*, 2022; Zhao *et al.*, 2021). To further improve robustness, some methods incorporate additional information from the ASR *N*-best list, expanding the capabilities of error correction models (Ma *et al.*, 2023a; Zhu *et al.*, 2021). Non-autoregressive decoding techniques have also been explored to accelerate inference while maintaining accuracy (Leng *et al.*, 2023; Leng *et al.*, 2021b; Leng *et al.*, 2021a).

Key differences between *N*-best reranking and error correction models can be summarized as follows. First, error correction models generate new outputs that may not belong to the original *N*-best set, potentially surpassing the performance of *N*-best reranking methods confined to the predefined hypotheses. Second, pipeline or end-to-end error correction models are inherently more complex to train, whereas *N*-best reranking methods usually rely on simpler, well-established training techniques. Third, error correction models depend solely on textual clues without direct reference to the input speech, increasing the risk of introducing unexpected errors. In contrast, *N*-best reranking methods are more conservative and stable.

2.3 Other approaches and recent advances

Beyond *N*-best reranking and error correction, recent research has explored hybrid and alternative strategies to further enhance ASR performance. Leveraging word lattices as input expands the search space for candidate hypotheses, improving recognition outcomes (Pandey *et al.*, 2022; Tsai *et al.*, 2022). However, encoding lattice structures is computationally intensive and typically requires large supervised data sets (Dai *et al.*, 2022; Ma *et al.*, 2020).

The rapid development of large language models (LLMs) has introduced zero-shot and one-shot strategies for revising ASR hypotheses, achieving remarkable performance (Brown *et al.*, 2020; Chen *et al.*, 2024; Ma *et al.*, 2023b; Ouyang *et al.*, 2022; Touvron *et al.*, 2023a; Touvron *et al.*, 2023b; Workshop *et al.*, 2022; Yang *et al.*, 2023). For *N*-best reranking, LLMs have been used to select the best hypothesis directly from the list using in-context learning, often with demonstrations in the prompt to improve task understanding (Chen *et al.*, 2024; Ma *et al.*, 2023b; Wang *et al.*, 2024).

LLMs have also been applied to error correction tasks, using advanced techniques such as multi-task learning and prompt engineering to achieve state-of-the-art results (Chen *et al.*, 2023; Ma *et al.*, 2023b; Min and Wang, 2023; Nie *et al.*, 2022; Pu *et al.*, 2023). Additionally, cross-modal fusion technologies that integrate LLMs with pre-trained speech models offer promising directions for ASR post-processing (Radhakrishnan *et al.*, 2023; Wang *et al.*, 2024). However, the large parameter sizes and high computational demands of these methods remain significant considerations.

2.4 Pipeline-style approaches

Recent ASR post-processing research has shown a growing interest in *pipeline-based* architectures, where multiple specialized modules are cascaded to incrementally refine recognition outputs. As summarized in Table 1, these frameworks generally consist of two or more stages, such as uncertainty estimation, text correction, rescoring or acoustic

Table 1. Comparison of recent pipeline-based ASR post-processing frameworks in terms of processing stages and information utilization

Method	Pipelines	Uses *N*-best list	Uses extra LM	Uses speech info
Cross-modal ASR post-processing (Du *et al.*, 2022)	*Stage 1:* Confidence estimation *Stage 2:* Error correction via cross-modal fusion (acoustic + text)	✗	✓	✓
Multi-stage LLM correction (Pu *et al.*, 2023)	*Stage 1:* Uncertainty estimation on *N*-best list *Stage 2:* Rule-guided LLM correction	✓	✓	✗
ClozeGER (Hu *et al.*, 2024)	*Stage 1: N*-best transformation into cloze format *Stage 2:* Multimodal LLM generative error correction	✓	✓	✓
MMGER (Mu *et al.*, 2024)	*Stage 1:* Multimodal speech-text encoding *Stage 2:* Multi-granularity correction (frame + utterance)	✗	✓	✓
SpeechLLM pseudo-labeling (Prakash *et al.*, 2025)	*Stage 1:* Multi-ASR hypothesis fusion *Stage 2:* LLM/SpeechLLM correction and pseudo-label generation	✓	✓	✓
LLM error correction w/ Task-Activating prompting (Yang *et al.*, 2023)	*Stage 1:* Hypothesis rescoring via LLM *Stage 2:* Error correction using task-activated prompting	✓	✓	✗
Tag and correct (Ziętkiewicz, 2022)	*Stage 1:* Neural tagger detects erroneous spans *Stage 2:* Corrector applies context-aware edits	✗	✓	✗
Pinyin regularization for LLM correction (Tang *et al.*, 2024)	*Stage 1:* Pinyin-based text regularization *Stage 2:* LLM refinement for Chinese ASR correction	✗	✓	✗

consistency checking. Their modular design allows each component to address specific error types and facilitates interpretability during inference.

Early studies like Cross-Modal ASR Post-Processing (Du *et al.*, 2022) introduced a dual-stage pipeline combining confidence estimation with cross-modal error correction, explicitly integrating both textual and acoustic cues. Subsequent works extended this paradigm toward multi-stage language-model-driven correction. For instance, Multi-stage LLM Correction (Pu *et al.*, 2023) performs uncertainty estimation on *N*-best hypotheses followed by rule-guided LLM refinement, whereas ClozeGER (Hu *et al.*, 2024) and multi-modal and multi-granularity generative error correction (MMGER) (Mu *et al.*, 2024) exploit multimodal and multigranular modeling to bridge textual and acoustic modalities. SpeechLLM-based approach (Prakash *et al.*, 2025) further combines multiple ASR outputs with generative reasoning to form pseudo-labeled data for self-improvement.

Other frameworks focus primarily on text–level enhancement. Tag and Correct (Ziętkiewicz, 2022) uses a tagging–correction pipeline to achieve high-precision localized

edits and the Pinyin Regularization method (Tang *et al.*, 2024) introduces phonetic normalization before LLM-based refinement to handle Chinese homophone ambiguities. Meanwhile, prompting-based designs such as LLM Error Correction with Task-Activating Prompting (Yang *et al.*, 2023) explore the capability of LLMs to act as both rerankers and correctors under zero-shot or few-shot conditions.

From a comparative standpoint, most existing pipelines rely on *N*-best hypotheses and external language models for reranking and text–level correction, while only a subset – such as Cross-Modal ASR Post-Processing, ClozeGER and MMGER – explicitly leverage speech information. This trend highlights the field's transition from single-model correction toward more modular, multimodal and LLM-augmented post-processing paradigms.

3. Proposed methodology

Building upon previous research, we propose a novel post-processing framework for ASR that seamlessly integrates error correction, rescoring and acoustic information to create a more stable and accurate system. More formally, this framework consists of three interconnected modules: the text correction module, the text rescoring module and the text–speech matching module, collectively referred to as CREAM. An overview of the framework is illustrated in Figure 1. These modules operate sequentially: the text correction

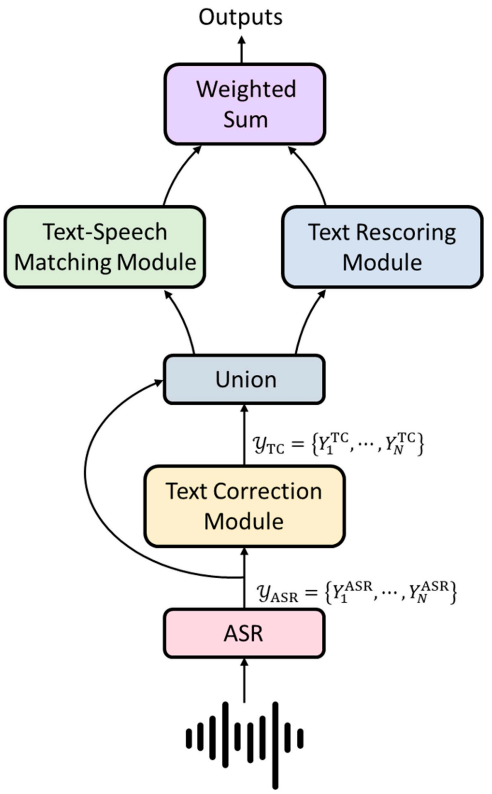

Figure 1. Overview of the proposed CREAM framework

module first refines the original ASR *N*-best hypothesis list by generating enhanced hypotheses. Subsequently, the text rescoring module evaluates the linguistic coherence and semantic quality of these hypotheses. Finally, the text–speech matching module ensures that the hypotheses align with the original speech signal. Together, these modules collaboratively identify the most accurate hypothesis, leveraging both language- and acoustic-based criteria to determine the final output.

3.1 Text correction module

In the first stage, the text correction module rewrites each hypothesis to generate a new set of candidate hypotheses. Its objective is to replace recognition errors with correct predictions, ideally aligning the new hypotheses with the ground truth. Formally, given the original ASR-generated hypotheses $\mathcal{Y}_{\text{ASR}} = \{Y_1^{\text{ASR}}, \ldots, Y_n^{\text{ASR}}, \ldots, Y_N^{\text{ASR}}\}$, the text correction module generates a new set of refined hypotheses $\mathcal{Y}_{\text{TC}} = \{Y_1^{\text{TC}}, \ldots, Y_n^{\text{TC}}, \ldots, Y_N^{\text{TC}}\}$:

$$Y_n^{\text{TC}} = \text{TC}\left(Y_n^{\text{ASR}}\right), \tag{3}$$

where $\text{TC}(\cdot)$ denotes the text correction model. To maximize the capacity of the candidates and avoid missing plausible hypotheses, the original and refined sets are merged for input into subsequent modules:

$$\mathcal{Y} = \mathcal{Y}_{\text{ASR}} \cup \mathcal{Y}_{\text{TC}} \tag{4}$$

We adopt BART and FastCorrect models for implementing the text correction module (Leng *et al.*, 2021a; Zhao *et al.*, 2021). BART refines hypotheses autoregressively, while FastCorrect applies non-autoregressive corrections. Implementation details are discussed in Section 4.2.

3.2 Text rescoring module

After the text correction module, a new set of hypotheses $\mathcal{Y}$ is obtained and is expected to have hypotheses that are close to the ground truth. In the following step, the text rescoring module is introduced to evaluate each hypothesis based on text–level regularity and semantic coherence:

$$S^{\text{TR}}(Y_i) = \text{TR}(Y_i), \tag{5}$$

where $\text{TR}(\cdot)$ represents the text rescoring model and $Y_i \in \mathcal{Y}$.

To balance performance and efficiency, we use BERT as the underlying model to compute perplexity scores for each hypothesis (Kenton and Toutanova, 2019; Udagawa *et al.*, 2022; Xu *et al.*, 2022). By leveraging its bidirectional contextual understanding, BERT enables efficient hypothesis rescoring. Implementation details are provided in Section 4.3.

3.3 Text–speech matching module

Although the hypotheses generated by the text correction module are typically fluent and semantically coherent, they may deviate from the original speech content due to the lack of acoustic information in the correction process. Such deviations are challenging to detect using the text rescoring module, as it also relies solely on text–level information. To address this, we propose the text–speech matching module to calculate the degree of alignment between each hypothesis and the corresponding speech utterance. We introduce three implementations of the module.

3.3.1 Automatic speech recognition model. Using an ASR model is a straightforward approach. For a hypothesis Y_i and its corresponding speech utterance $\mathcal{X}$, the text–speech matching score $S_{\text{ASR}}^{\text{TS}}(Y_i)$ is defined as the ASR decoding score. For example, in a connectionist temporal classification (CTC)-attention ASR model, the text–speech matching score is computed as (Watanabe *et al.*, 2017):

$$S_{\text{ASR}}^{\text{TS}}(Y_i) = \alpha s_{ctc}(Y_i|\mathcal{X}) + (1-\alpha)s_{att}(Y_i|\mathcal{X}), \tag{6}$$

where α is a weighting factor and $s_{ctc}(Y_i|\mathcal{X})$ and $s_{att}(Y_i|\mathcal{X})$ represent the CTC and attention scores, respectively. The CTC score is defined by taking the log of the summation of probabilities for all alignments Φ between Y_i and $\mathcal{X}$:

$$s_{ctc}(Y_i|\mathcal{X}) = \log \sum_{\pi \in \Phi(Y_i)} P(\pi|\mathcal{X}). \tag{7}$$

The attention score is computed by summing the log probability of every token w_i conditioned on its previous tokens:

$$s_{att}(Y_i|\mathcal{X}) = \sum_{l=1}^{L} \log P(w_l| < sos >, w_1, \ldots, w_{l-1}, \mathcal{X}), \tag{8}$$

where L denotes the number of tokens in hypothesis Y_i.

3.3.2 Multi-modal matching model. Inspired by recent advancements in multi-modal research (Chen *et al.*, 2022c; Tsimpoukelli *et al.*, 2021), we use the masked audio-text encoder (MATE) model as a judge to calculate the text–speech matching score (Cai *et al.*, 2023). MATE integrates three key components: a pre-trained masked language model (BERT), a pre-trained speech encoder [HuBERT (Hsu *et al.*, 2021) or WavLM (Chen *et al.*, 2022b)] and a convolutional neural network (CNN)-based multi-modal alignment module. For a given speech utterance $\mathcal{X}$ and hypothesis Y_i, MATE uses WavLM to convert $\mathcal{X}$ into a sequence of feature vectors and the embedding layer of BERT to map Y_i into a series of token embeddings. To reconcile the sequence lengths of the two modalities, the multi-modal alignment module downsamples the speech feature vectors and projects them into the same embedding space as the BERT token embeddings. These aligned features from both modalities are concatenated and processed by BERT, enabling the fusion of information from both speech and text. MATE is trained using the masked language model objective and contrastive loss. The former encourages the model to predict masked tokens in the text sequence, leveraging contextual and cross-modal information. The latter promotes greater similarity between paired acoustic and lexical representations while discouraging similarity between unpaired representations. This helps MATE learn robust matching relationships between text and speech. Detailed descriptions of MATE architecture and training processes can be found in Cai *et al.* (2023).

Using MATE, the text–speech matching score for a speech-hypothesis pair $(\mathcal{X}, Y_i = \{w_1, \ldots, w_L\})$ is computed in a MLM manner:

$$S_{\text{MATE}}^{\text{TS}}(Y_i) = \sum_{l=1}^{L} \log P\left(w_l|Y_{i,\setminus l}, \mathcal{X}\right), \tag{9}$$

where $Y_{i,\setminus l}$ denotes the hypothesis Y_i with the token w_l replaced by the special mask token. The underlying principle of this scoring mechanism is straightforward: hypotheses that are both textually fluent and acoustically consistent with the speech signal will receive higher

scores, while those that are incoherent or misaligned with the acoustic input will score lower. Notably, MATE not only models the relationships between text and speech but also incorporates a strong language modeling perspective, enhancing its ability to evaluate semantic and lexical coherence alongside acoustic alignment.

3.3.3 Learning by knowledge distillation. While using an ASR model to calculate the text–speech matching degree is intuitive and effective, this approach is computationally expensive and requires a large number of parameters. To address these challenges, we propose a lightweight model trained to distill knowledge from an ASR system and efficiently evaluate the matching degree. We refer to this model as MIMICA because it is designed to mimic an ASR.

To reduce the computational overhead while considering acoustic-level information, MIMICA leverages phoneme sequences derived from the *N*-best list as acoustic references. During training, for a given speech utterance $\mathcal{X}$ and its *N*-best hypotheses $\mathcal{Y}^{\text{ASR}} = \{Y_1^{\text{ASR}}, \ldots, Y_N^{\text{ASR}}\}$, each hypothesis Y_n^{ASR} is converted into a phoneme sequence H_n^{ASR} using a grapheme-to-phoneme (G2P) model. These phoneme sequences are concatenated to form a single acoustic reference:

$$H_{1:N}^{\text{ASR}} = \text{concat}\left(H_1^{\text{ASR}}, \ldots, H_N^{\text{ASR}}\right). \tag{10}$$

For a target hypothesis to be scored, its phoneme sequence is concatenated with the acoustic reference as the input to MIMICA. Following the BERT-style architecture, a [CLS] token is prepended for the score prediction and a [SEP] token is used to separate the hypotheses. Segment embeddings are used to differentiate the target hypothesis (assigned 0) and the acoustic reference (assigned 1). MIMICA is implemented as a multi-layer Transformer model followed by a simple linear layer for score prediction. After multi-round interactions among the phoneme units of the target hypothesis and the acoustic reference, the final hidden state of the [CLS] token is passed through the linear layer to come up with the text–speech matching score:

$$S_{\text{MIMICA}}^{\text{TS}}\left(Y_n^{\text{ASR}}\right) = h_{n,[\text{CLS}]}^{\text{ASR}} W, \tag{11}$$

where $h_{n,[\text{CLS}]}^{\text{ASR}}$ is the final hidden state of the [CLS] token for the target hypothesis $Y_n^{\text{ASR}} \in \mathcal{Y}^{\text{ASR}}$ and W represents the weight matrix of the linear layer. The training objective of MIMICA is to minimize the mean squared error between the decoding score $S_{\text{ASR}}\left(Y_n^{\text{ASR}}\right)$ calculated by ASR as defined in equation (6) and the predicted score $S_{\text{MIMICA}}^{\text{TS}}\left(Y_n^{\text{ASR}}\right)$:

$$\mathcal{L}_{\text{MIMICA}} = \sum_{n=1}^{N} \left(S_{\text{ASR}}\left(Y_n^{\text{ASR}}\right) - S_{\text{MIMICA}}^{\text{TS}}\left(Y_n^{\text{ASR}}\right)\right)^2. \tag{12}$$

The model architecture is depicted in Figure 2.

During inference, MIMICA computes the text–speech matching score for each hypothesis $Y_i \in \mathcal{Y}$ by concatenating its phoneme sequence H_i with the corresponding acoustic reference $H_{1:N}$. The inference process uses only phoneme sequences, eliminating the need for acoustic features paired with a full ASR system. In summary, MIMICA provides a lightweight and efficient solution by leveraging phoneme sequences from *N*-best hypotheses as acoustic references to calculate text–speech matching scores. Compared to using an ASR model directly, MIMICA significantly reduces computational complexity while maintaining robust performance.

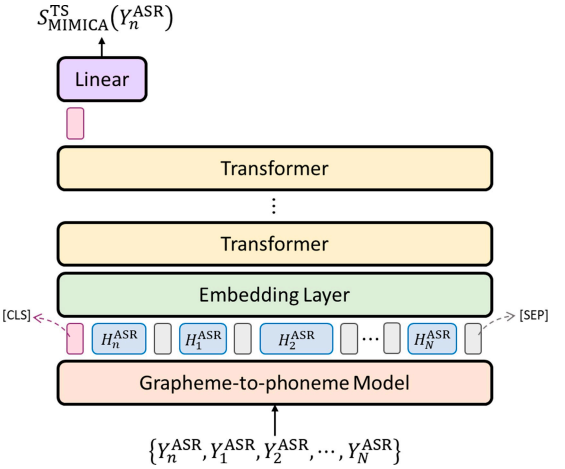

Figure 2. The model architecture of the proposed MIMICA. In this example, the target hypothesis to be scored is Y_n^{ASR} and the acoustic reference is formed by Y_1^{ASR}, …, Y_N^{ASR}

3.4 The correction, rescoring and matching framework

The CREAM framework integrates three modules – text correction, text rescoring and text–speech matching – to achieve robust ASR post-processing. The process begins with the text correction module, which expands the pool of candidate hypotheses by refining the original *N*-best list. Next, each hypothesis is evaluated based on text–level token regularity and semantic coherence by the text rescoring module, as well as the alignment between text and speech by the text–speech matching module. The final score for each hypothesis is computed by combining the rescoring and matching scores using a weighted sum. The hypothesis with the highest combined score is selected as the final output:

$$Y^* = arg\,max_{Y_i \in \mathcal{Y}}\, S^{TR}(Y_i) + \beta S^{TS}(Y_i), \tag{13}$$

where $S^{TR}(Y_i)$ represents the text rescoring score, $S^{TS}(Y_i)$ represents the text–speech matching score and β is a tunable hyperparameter that adjusts the relative influence of the two scores. This framework ensures that the selected hypothesis balances both linguistic quality and acoustic alignment, resulting in a more accurate and reliable ASR output.

It is worth noting that CREAM also follows a pipeline-style design, similar in spirit to many recent ASR post-processing frameworks summarized in Table 1. However, unlike prior systems that typically focus on one or two isolated components, CREAM integrates three complementary modules into a unified and ASR-agnostic architecture. Specifically, it performs text correction, text rescoring and text–speech matching sequentially, while maintaining modular independence that allows each component to be optimized or replaced without affecting the overall framework. This unified yet flexible pipeline enables CREAM to fully exploit textual, contextual and acoustic information, thereby achieving a more balanced tradeoff between correction accuracy, acoustic consistency and computational efficiency.

4. Experimental setup

4.1 Data sets

In this study, we adopted the publicly available speech recognition hypothesis revising benchmark, HypR [1], to evaluate the performance of our proposed CREAM framework. HypR provides speech recognition results for several publicly available speech corpora, with the speech recognition model for each data set using a Transformer-based CTC/attention architecture built with ESPnet (Watanabe *et al.*, 2018). To evaluate our framework under varying ASR quality, we leverage recognition results from HypR with and without language models to simulate low-error and high-error scenarios, respectively. Our experiments were conducted on three public speech corpora included in the benchmark: the Chinese corpus AISHELL-1 (Bu *et al.*, 2017) and the English corpora TEDLIUM-2 (Rousseau *et al.*, 2014) and LibriSpeech (Panayotov *et al.*, 2015). For each speech utterance, we extracted the top 10 hypotheses to train, tune and test all models compared in the experiments. For each corpus, we used the corresponding training sets to train the models, selected the best model checkpoints and hyperparameters based on the development set performance and finally evaluated the results on the test sets. Table 2 summarizes the statistics of the data sets used in our study.

4.2 Text correction module

We adopted BART and FastCorrect models to evaluate the versatility of our proposed framework, as these two models use different modeling philosophies. BART is a classical end-to-end sequence-to-sequence autoregressive model, while FastCorrect is a non-autoregressive model when processing generation. For BART, we fine-tuned the text correction models using pre-trained Chinese [2] and English [3] checkpoint models individually. However, since only the Chinese [4] checkpoint model is available for FastCorrect, experiments with this model were conducted exclusively on the AISHELL-1 data set. The vocabulary sizes of BART are 51,271 for Chinese and 50,265 for English, whereas FastCorrect has a vocabulary size of 32,652 for Chinese. The fine-tuning configurations are summarized in Table 3.

4.3 Text rescoring module

BERT has demonstrated its effectiveness in various tasks, including serving as a language model for speech recognition (Shin *et al.*, 2019; Udagawa *et al.*, 2022; Xu *et al.*, 2022). In

Table 2. Statistics of the speech corpora were computed using BERT's tokenizer to determine the number of tokens in each ground truth sentence

				#Tokens/sentence		
Corpus	Splits	#Utterances	#Hours	Min.	Avg.	Max.
AISHELL-1	Train	120,098	150.0	1	14.4	44
	Dev	14,326	18.0	3	14.3	35
	Test	7,176	10.0	3	14.6	37
TEDLIUM-2	Train	92,973	212.0	1	26.0	96
	Dev	507	1.6	1	38.5	143
	Test	1,155	2.6	1	26.3	132
LibriSpeech	Train	281,231	960.9	1	36.6	97
	Dev-clean	2,703	5.4	1	22.1	113
	Test-clean	2,620	5.4	1	22.0	109
	Dev-other	2,864	5.3	1	19.6	92
	Test-other	2,939	5.1	1	19.7	123

Table 3. Training configurations for the text correction module on different data sets

Model	BART			FastCorrect
Data set	AISHELL-1	TEDLIUM-2	LibriSpeech	AISHELL-1
Epochs	10	10	10	25
Batch size	128	64	64	–
Max tokens	–	–	–	9,000
Gradient accumulation	2	4	4	4
Learning rate	1e-6	1e-6	1e-6	5e-4
Warm up Ratio	0.3	0.3	0.3	–

this study, we use BERT to implement the text rescoring module and compute the perplexity score for each hypothesis (Xu *et al.*, 2022). Publicly available pre-trained models for Chinese [5] and English [6] are used to initialize the model parameters. These models are then fine-tuned using the MLM loss with a 15% masking rate applied to each training text sequence. The vocabulary sizes of BERT are 21,128 for Chinese and 30,522 for English. The fine-tuning configurations are summarized in Table 4.

4.4 Text–speech matching module

In this study, we explore three approaches for implementing the text–speech matching module: the ASR model, MATE and MIMICA. For the ASR model, we used the pre-trained ASR model provided by the HypR benchmark, keeping its parameters frozen during the experiments. The interpolation weight α between the CTC and attention scores [cf. Equation (6)] was adjusted for each data set based on the settings in the HypR benchmark.

For MATE, we followed the methodology outlined in the previous study to combine BERT with a pre-trained speech encoder to construct the multi-modal matching model (Cai *et al.*, 2023). HuBERT [7] and WavLM [8] were used as the speech encoders for Chinese and English, respectively. On top of the speech encoder, the multi-modality alignment component was built using three layers of CNN with 768 channels, strides (2, 1, 2), kernel widths (3, 1, 1) and a bottleneck adapter layer with a compression factor of 0.5. The fine-tuning hyperparameters are summarized in Table 5.

For MIMICA, hypotheses were converted to their phoneme sequences using the pypinyin [9] toolkit for Chinese, whereas a G2P [10] model was used to generate phoneme sequences for English (Lee *et al.*, 2018). The main structure of the model is built using a stack of twelve Transformer layers. During training, MIMICA learned to predict the decoding scores calculated by the ASR model. During inference, it predicted the text–speech matching score for each hypothesis in $\mathcal{Y}$. Table 6 summarizes the hyperparameters used for fine-tuning.

Table 4. MLM training configurations of the text rescoring module across different data sets

Data set	AISHELL-1	TEDLIUM-2	LibriSpeech
Epochs	10	10	10
Batch size	256	128	128
Gradient accumulation	1	2	2
Learning rate	1e-5	1e-5	1e-5
Warm up Ratio	0.01	0.01	0.01

Table 5. MATE training configurations on different data sets

Data set	AISHELL-1	TEDLIUM-2	LibriSpeech
Epochs	5	5	5
Batch size	4	4	4
Gradient accumulation	8	8	8
Learning rate	2e-5	2e-5	2e-5
Warm up Ratio	0.1	0.1	0.1

Table 6. Training configurations of MIMICA across different data sets

Data set	AISHELL-1	TEDLIUM-2	LibriSpeech
Epochs	10	10	10
Batch size	16	16	16
Gradient accumulation	2	2	8
Learning rate	1e-5	1e-5	1e-5
Warm up Ratio	0.1	0.1	0.1

5. Experiments

5.1 Main results

In the first set of experiments, we discuss the performance of our proposed CREAM framework and compare the results with basic baselines. Each of the three data sets is tested under both high-error and low-error scenarios, depending on whether an language model (LM) is used during the ASR decoding stage. Tables 7 and 8 summarize the experimental

Table 7. Performance comparison between baseline models and CREAM variants under the high-error condition. The text correction and text–speech matching modules can be implemented with different settings, while the text rescoring module is implemented using BERT

		High-error scenario			
Method		AISHELL-1	TEDLIUM-2	Libri-other	Libri-clean
Baselines					
Top-1		8.33	12.07	11.99	4.64
Oracle		4.74	8.55	8.14	2.43
BART		7.43	11.77	11.26	4.50
FastCorrect		6.72	–	–	–
BERT		6.05	10.37	9.91	3.58
CREAM					
Text correction module	Text–speech matching module				
BART	ASR	*4.91*	*9.48*	*9.12*	*3.28*
BART	MATE	5.05	10.41	9.43	3.93
BART	MIMICA	5.25	11.33	10.74	4.83
FastCorrect	ASR	5.01	–	–	–
FastCorrect	MATE	*4.90*	–	–	–
FastCorrect	MIMICA	5.02	–	–	–
BART	ASR+MATE+MIMICA	4.63	9.29	8.80	3.16
FastCorrect	ASR+MATE+MIMICA	4.73	–	–	–

Table 8. Performance comparison between baseline models and CREAM variants under the low-error condition. The text correction and text–speech matching modules can be implemented with different settings, while the text rescoring module is implemented using BERT

Method		Low-error scenario			
		AISHELL-1	TEDLIUM-2	Libri-other	Libri-clean
Baselines					
Top-1		7.22	9.72	7.21	*2.58*
Oracle		4.14	6.28	4.45	1.34
BART		7.53	10.66	7.42	3.05
FastCorrect		6.81	–	–	–
BERT		5.55	8.44	6.86	2.73
CREAM					
Text correction modul	Text–speech matching module				
BART	ASR	*5.30*	*8.43*	*7.21*	*2.89*
BART	MATE	5.70	9.63	7.39	3.48
BART	MIMICA	5.79	9.80	7.28	2.95
FastCorrect	ASR	*5.09*	–	–	–
FastCorrect	MATE	5.21	–	–	–
FastCorrect	MIMICA	5.29	–	–	–
BART	ASR+MATE+MIMICA	4.98	8.13	6.72	2.70
FastCorrect	ASR+MATE+MIMICA	4.73	–	–	–

results. Several noteworthy observations emerge from these experiments. First, Top-1 represents the ASR output without any post-processing and Oracle refers to selecting the hypothesis with the lowest error rate from the 10 best candidates for each speech utterance. Comparing the two baselines, the performance gap highlights room for improving speech recognition results and the potential for ASR post-processing research. Second, we present the performance achieved by BART, FastCorrect and BERT as references. BART and FastCorrect are error correction models, whereas BERT is an *N*-best reranking method. In the high-error condition, all three models effectively enhance recognition performance. However, in the low-error scenario, BART fails to outperform Top-1, while FastCorrect and BERT still demonstrate improvements in most cases. These results indicate that although ASR post-processing typically improves performance, error correction models may sometimes degrade accuracy compared to the original ASR output. This occurs because such models rely solely on textual cues when making corrections, thereby increasing the risk of introducing spurious errors due to the lack of direct reference to the input speech. Table 9 presents several correction examples produced by BART, along with the corresponding ASR output (Top-1) and the ground truth (GT) for comparison. As shown, without guidance from acoustic information, error correction models like BART may mislead the original ASR output toward commonly occurring phrases such as “办公,” “projecting,” or “god bless you.” However, these phrases drift entirely away from the actual spoken content.

Next, we turn the focus to the proposed CREAM framework, which has different settings because the text correction module and text–speech matching module can be implemented by using various strategies. Among all configurations, the text rescoring module is implemented using BERT. At first glance, regardless of the specific implementation, CREAM consistently outperforms the baselines in most cases. Upon closer examination, FastCorrect proves to be more effective than BART for the text correction module, aligning with their standalone performance. Regarding the text–speech matching module, ASR

Table 9. Examples from the AISHELL-1, TEDLIUM-2 and LibriSpeech data sets. GT denotes the ground truth, top-1 indicates the ASR output and BART represents the corrected sentence produced by BART

Data set	Type	Sentence
AISHELL-1 (BAC009S0906W0191)	GT	包括北方干旱半干旱草原地区和青藏高原草原地区
	Top-1	包括北方甘汉办干汉草园地区和青脏高园草园地区
	BART	包括北方甘汉办公汉草园地区和倾盆高园草园
TEDLIUM-2 (DanBarber_ 2010-0016300–0016543)	GT	We're basically a world unto ourselves
	Top-1	Or basically a world onto ourselves
	BART	Or basically projecting the world onto ourselves
LibriSpeech (6432–63723 - 0036)	GT	So king got bail who put it up
	Top-1	So king god bail who put it up
	BART	So king god bless you who put it up

outperforms both MATE and MIMICA in most settings. This is expected, as MIMICA is designed as a lightweight alternative to ASR, predicting the matching scores without relying on acoustic features and an ASR model during inference. As a result, although MIMICA does not obtain superior results, it has very scalable usability, flexibility and generalization capabilities. MATE, on the other hand, is a multi-modality mixture model that functions as an acoustic-enhanced language model. Its advantages over ASR include simplicity during inference and the absence of the need for an ASR model for alignment. However, compared to MIMICA, MATE relies on acoustic feature vectors and its computational cost is higher due to token-wise probability predictions.

Each strategy for implementing the text–speech matching module has distinct characteristics. ASR evaluates the matching degree by calculating the alignment between acoustic features and text, providing a classical and reliable approach. MIMICA translates hypotheses into phoneme sequences and evaluates matching scores from a phoneme-level perspective, offering a lightweight and efficient solution. MATE enriches text with acoustic features to build an acoustic-enhanced language model, leveraging masked token prediction for scoring. We are thus curious about whether they could complement each other and enhance the performance further. Therefore, we combined all three methods to calculate the text–speech matching score using a simple linear weighted combination. When paired with either BART or FastCorrect, this combined strategy achieves the best results across all cases, as shown at the bottom of Tables 7 and 8. In conclusion, integrating ASR, MIMICA and MATE efficiently enhances the text–speech matching score. The combination leverages the unique strengths of each method, resulting in improved performance across all tested scenarios. These results demonstrate the potential for a comprehensive model that benefits from the diverse attributes of the individual components.

5.2 Ablation study

In this ablation study, we evaluate the contribution of each component within the CREAM framework and analyze the impact of the N-best list size. AISHELL-1 is selected as a representative data set and the best configuration, in which BART is used for the text correction module, BERT for the text rescoring module and ASR for the text–speech matching module, is adopted for experimental validation.

5.2.1 Contribution of components. We first use AISHELL-1 as a case study to investigate the contribution of each component in the CREAM framework, with the results

summarized in Table 10. Six configurations are listed and numbered from 1 to 6. Configurations 1 and 2 demonstrate the individual contributions of the text–speech matching and text rescoring modules, respectively. Configurations 3, 4 and 5 combine any two of the three components, while Configuration 6 represents the case where all three modules are jointly applied. At first glance, the results may seem inconclusive; however, several noteworthy patterns emerge upon closer examination.

When the text correction module is fixed (either enabled or disabled), we further examine the experimental results by enabling only the text rescoring module (i.e. no. 2 and 5) or only the text–speech matching module (i.e. no. 1 and 4) to compare their independent effects. The findings reveal that these two modules exhibit complementary behavior under different word error rate conditions. In the high-error scenario, the text rescoring module achieves better performance; specifically, experiment no. 2 outperforms no. 1 and no. 5 outperforms no. 4. In contrast, in the low-error scenario, the text–speech matching module proves more effective, as experiment no. 1 performs better than no. 2 and no. 4 surpasses no. 5. This is because, in high-error cases, candidate sentences often contain numerous mistakes and selecting the more fluent option can effectively reduce the error rate, making rescoring the dominant factor. In contrast, when the error rate is already low and the candidate sentences are mostly fluent, rescoring offers limited additional benefit, whereas the text–speech matching module becomes crucial by identifying and selecting the candidate most consistent with the acoustic signal.

Subsequently, when both the text rescoring and text–speech matching modules are enabled simultaneously (i.e. no. 3 and 6), the results – consistent with expectations – show that combining the two yields the best overall performance, regardless of whether the text correction module is included, since they exploit distinct yet highly complementary sources of information.

Next, we examine the role of the text correction module. In most configurations, incorporating this module improves performance. However, when it is combined with the text rescoring module but without the text–speech matching module (i.e. no. 5), its inclusion actually degrades performance, particularly under the low-error scenario. This observation suggests that the text correction module may sometimes transform recognition candidates into more fluent yet acoustically inconsistent sentences. When only textual scores are considered (as in the rescoring module), such misleading candidates may be preferentially selected, leading to a higher overall error rate.

Overall, these experiments validate the design of the CREAM framework. By appropriately organizing and integrating the modules, the framework effectively reduces ASR recognition errors through the complementary use of correction, rescoring and matching strategies.

Table 10. Experimental results of the ablation study on the contribution of each component within the CREAM framework

No.	Text correction module	Text rescoring module	Text–speech matching module	High-error scenario	Low-error scenario
1	✗	✗	✓	8.33	7.79
2	✗	✓	✗	6.44	8.43
3	✗	✓	✓	6.05	5.55
4	✓	✗	✓	8.33	7.78
5	✓	✓	✗	7.79	10.62
6	✓	✓	✓	4.91	5.30

5.2.2 Impact of the N-best list size. We further analyze the impact of the *N*-best list size on recognition performance. For this experiment, we use the best configuration, where BART is used for the text correction module, BERT for the text rescoring module and ASR for the text–speech matching module. The experimental results are summarized in Table 11, with *N*-best list sizes of 5, 10, 20 and 50. From the results, we observe that CREAM achieves improved recognition accuracy as the *N*-best list size increases – the larger the candidate set, the lower the recognition error rate. This improvement occurs because a larger *N*-best list provides greater hypothesis diversity, increasing the likelihood of including a more accurate candidate. However, it should also be noted that the computational cost increases proportionally with the candidate size. Therefore, although larger *N*-best lists can further improve performance, the choice of candidate number should be balanced against the available computational resources and the desired level of accuracy.

5.3 Compared with other models

5.3.1 Performance comparison. In the third set of experiments, we compare the proposed CREAM framework with several baseline models, categorized into *N*-best reranking and error correction models. These models are implemented by the HypR project and detailed settings and configurations can be found in the HypR documentation (Wang *et al.*, 2024). The results are shown in Table 12. Based on the results, *N*-best reranking methods generally outperform error correction models, with a noticeable performance gap. This discrepancy may stem from the fact that error correction models often produce outputs that are fluent, smooth and fluid but may drift from the original speech content. Within the *N*-best reranking methods, MLM outperforms CLM, as MLM leverages bi-directional contextual information, whereas CLM is limited to uni-directional modeling. While RescoreBERT is more time-efficient than MLM, this comes at the cost of reduced performance. Among the reranking methods, PBERT achieves the best performance due to its contrastive training objective, which effectively distinguishes the hypothesis with the lowest word error rate from the others (Chiu and Chen, 2021b). Interestingly, the LLM-Reranking method, which uses a large language model (LLM) to select the best hypothesis, still leaves room for improvement.

Regarding error correction models, BART and FastCorrect serve as representative examples. On the AISHELL-1 data set, FastCorrect demonstrates superior performance compared to BART. However, compared with Top-1 listed in Tables 7 and 8, BART also achieves consistent improvements across both Chinese and English corpora.

Despite the widespread success of LLMs in various applications, using an LLM to address speech recognition errors (i.e. the LLM-Correction model) does not yet yield competitive results on these data sets. The results suggest that the use of LLMs exhibits certain instability. To replicate successful outcomes across different data sets, further exploration is needed in areas such as better prompt design and fine-tuning strategies, which may help achieve more satisfactory and stable results.

Table 11. Performance of CREAM with different *N*-best list sizes (5, 10, 20 and 50)

N-best list size	High-error scenario	Low-error scenario
5	5.25	5.61
10	4.91	5.30
20	4.68	5.10
50	4.43	4.88

Table 12. Experimental results of various post-processing models and the proposed CREAM framework

Method	High-error scenario				Low-error scenario			
	AISHELL-1	TEDLIUM-2	Libri-other	Libri-clean	AISHELL-1	TEDLIUM-2	Libri-other	Libri-clean
N-best reranking method								
CLM	6.80	10.61	9.99	3.64	6.55	8.92	7.05	2.84
MLM (BERT)	6.05	10.37	9.91	3.58	5.55	8.44	6.86	2.73
RescoreBERT	6.12	10.61	9.98	3.69	5.56	8.77	6.96	2.79
PBERT	5.89	10.35	9.97	3.63	5.35	8.51	6.81	2.74
LLM-Reranking (ChatGPT 3.5)	7.36	11.67	11.33	4.90	8.60	11.15	9.72	5.72
Error correction model								
BART	7.43	11.77	11.26	4.50	7.53	10.66	7.42	3.05
FastCorrect	6.72	–	–	–	6.81	–	–	–
LLM-Correction (ChatGPT 3.5)	7.71	13.95	12.77	6.07	7.40	14.02	10.70	6.01
CREAM								
CREAM (BART)	4.63	9.29	8.80	3.16	4.98	8.13	6.72	2.70
CREAM (FastCorrect)	4.73	–	–	–	4.73	–	–	–

Finally, when comparing CREAM to these baselines, the proposed framework consistently achieves the best results across all data sets and settings (i.e. with and without LM). The superior performance of CREAM can be attributed to its meticulous design, which incorporates an error correction module to expand the pool of candidates and reweights each hypothesis based on token regularity and text–speech matching perspectives. By carefully integrating these components, CREAM effectively minimizes recognition errors and demonstrates its robustness across diverse scenarios.

5.3.2 Model size and computational efficiency. Although CREAM outperforms most baselines in accuracy, its main limitations lie in model size and computational latency. Table 13 reports the parameter counts and the average computation time per utterance (in seconds) for several baselines and for CREAM. Overall, CREAM is approximately 2–3× larger than the baselines, primarily because it consists of three modules. In the text correction module, BART and FastCorrect have about 140 M and 65 M parameters, respectively. The text rescoring module, implemented with BERT, contains about 102 M parameters. For the text–speech matching module, we present three variants – ASR, MATE and MIMICA – with sizes of roughly 30 M, 200 M and 102 M parameters, respectively.

Beyond parameter counts, CREAM's computational latency also varies depending on the chosen configuration. Generally, using FastCorrect for text correction is faster and smaller than BART. Among the text–speech matching variants, MATE is both the largest and the slowest, while MIMICA runs faster than ASR but has a slightly larger model size. Because CREAM operates in a pipeline-style workflow, inference incurs cumulative latency across modules, resulting in relatively longer per-utterance processing time. Although the text rescoring and text–speech matching modules can, in principle, be executed in parallel to increase throughput, we measure latency sequentially to reflect the worst-case scenario. Nevertheless, reducing CREAM's parameter footprint and improving inference speed remain important directions for future work.

Table 13. Latency and number of model parameters of various post-processing models and the proposed CREAM framework

Method			Model parameters	Latency
N-best reranking method				
MLM (BERT)			102m	0.113
Error correction model				
BART			140m	0.112
FastCorrect			65 m	0.014
CREAM				
Text correction module	Text rescoring module	Text–speech matching module		
BART	BERT	ASR	272M	1.741
BART	BERT	MATE	442M	3.050
BART	BERT	MIMICA	344M	1.461
FastCorrect	BERT	ASR	197M	0.512
FastCorrect	BERT	MATE	367M	1.797
FastCorrect	BERT	MIMICA	269M	0.380
BART	BERT	ASR+MATE+MIMICA	574M	3.596
FastCorrect	BERT	ASR+MATE+MIMICA	499M	2.175

5.4 Further analyses

Though ASR post-processing models generally enhance recognition performance, we are curious whether the corrections primarily involve common function words or more meaningful named entities. To investigate this, we conducted a set of experiments on the AISHELL-NER data set (Chen *et al.*, 2022a), which is based on the AISHELL-1 corpus and includes manually annotated named entities. The test set contains 898 person names, 1,330 location names and 1,165 organization names. Table 14 summarizes the named entity recall rates of various models.

First, all models outperform the Top-1 baseline. We also observe a consistent trend where lower ASR error rates are linked to higher NER accuracy. However, this relationship is not entirely linear. For example, in the high-error scenario, FastCorrect achieves a lower speech recognition error rate than BART, yet BART attains a higher named entity recall rate. Moreover, within the CREAM framework, the choice of the text–speech matching module affects named entity recognition. When BART is used for the text correction module, pairing it with ASR as the text–speech matching module results in a lower speech recognition error rate compared to MATE or MIMICA. However, the named entity recall rate is higher when BART is paired with MATE. Similarly, when FastCorrect is used for text correction, MATE remains the superior choice for the text–speech matching module, yielding better named entity recognition results than ASR or MIMICA in both low-error and high-error scenarios. These findings implicitly suggest that a greater contribution from the language model leads to improved named entity recognition results. Therefore, MATE, which functions as an acoustic-enhanced language model, tends to outperform both ASR and MIMICA. Finally, combining ASR, MATE and MIMICA achieves the best named entity recall rates in both low-error and high-error cases, as expected. This set of experiments demonstrates that the CREAM framework effectively enhances recognition results for both critical components, such as named entities and commonly used words in speech utterances.

Table 15 presents a real example extracted from the AISHELL-1 data set. For this speech utterance, the ground truth is “盲目捐款没有益 (i 4) 处”. Three corresponding speech recognition hypotheses are labeled as (1), (2) and (3). Hypothesis (1) is the perfect result. Hypothesis (2) contains the homophone “易 (i 4)” and is phrased awkwardly. Hypothesis (3)

Table 14. Named entity recall rates of various models on the AISHELL-NER corpus

Method		High-error scenario	Low-error scenario
Baselines			
Top-1		70.85	73.71
BART		74.36	73.80
FastCorrect		73.65	74.83
BERT		77.28	77.95
CREAM			
Error correction module	Text–speech matching module		
BART	ASR	79.66	78.46
BART	MATE	*80.31*	*78.72*
BART	MIMICA	78.49	77.48
FastCorrect	ASR	78.69	79.16
FastCorrect	MATE	*79.25*	*79.66*
FastCorrect	MIMICA	78.28	78.40
BART	ASR+MATE+MIMICA	*80.70*	79.55
FastCorrect	ASR+MATE+MIMICA	79.69	*80.40*

Table 15. An example extracted from the AISHELL-1 data set

Ground truth	盲目捐款没有益(i 4)处(ch u 4) (Donating blindly is of no benefit)		
ID	Hypotheses	Text rescoring Module	Text–speech Matching module
(1)	盲目捐款没有益(i 4)处(ch u 4) (Donating blindly is of no benefit)	−8.9	−0.5
(2)	盲目捐款没有易(i 4)处(ch u 4) (Blind donations have no alternative disposition)	−26.9	−4.2
(3)	盲目捐款没有意(i 4)义(i 4) (Donating blindly is meaningless)	−5.9	−15.2

includes the homophone "意 (i 4)" and a substitution error "义 (i 4)," yet the overall sentence remains coherent. The scores for these three hypotheses, evaluated by the text rescoring module and the text–speech matching module, are listed in the table. The text rescoring module assigns scores based solely on textual information, whereas the text–speech matching module evaluates the degree of alignment between speech and text. Due to their differing focuses, some discrepancies in scoring arise. For example, the grammar and semantics of hypothesis (3) are correct and reasonable, while those of hypothesis (2) are problematic. Consequently, the text rescoring module assigns a higher score to hypothesis (3) than to hypothesis (2). In contrast, the text–speech matching module assigns a higher score to hypothesis (2) than to hypothesis (3), as it evaluates the alignment between text pronunciation and speech. Additionally, while both "没有意义" and "没有益处" are valid expressions, "没有意义" is more commonly used in Chinese. As a result, the text rescoring module assigns a slightly higher score to hypothesis (3) than to hypothesis (1). This example demonstrates that our proposed CREAM framework, by integrating the two scoring modules to evaluate correctness from both perspectives, is not only a reasonable and feasible approach but also explains why it can achieve superior results.

6. Conclusions

In this study, we explored ASR post-processing technology and proposed a novel framework called CREAM, which comprises three key components: the text correction module, the text rescoring module and the text–speech matching module. For the text–speech matching module, we introduced three different implementation methods, each with distinct advantages, limitations and suitable use cases. We evaluated the performance of CREAM on three widely used data sets, demonstrating its effectiveness and robustness. In future work, we plan to explore potential modifications and optimizations for each module, including leveraging LLMs to enhance their implementation. We also aim to investigate the feasibility of transforming CREAM into an end-to-end framework and test its performance on more challenging data sets. Ultimately, we aspire to adapt this framework to a broader range of applications, ensuring consistent and robust improvements in speech recognition performance.

Notes

[1.] https://github.com/Alfred0622/HypR

[2.] https://huggingface.co/fnlp/bart-base-chinese

[3.] https://huggingface.co/facebook/bart-base

[4.] https://github.com/microsoft/NeuralSpeech/tree/master/FastCorrect

[5.] https://huggingface.co/bert-base-chinese

[6.] https://huggingface.co/bert-base-uncased

[7.] https://huggingface.co/TencentGameMate/chinese-hubert-base

[8.] https://huggingface.co/microsoft/wavlm-base

[9.] https://github.com/mozillazg/python-pinyin

[10.] https://github.com/Kyubyong/g2p

References

Ao, J., Wang, R., Zhou, L., Wang, C., Ren, S., Wu, Y., Liu, S., Ko, T., Li, Q., Zhang, Y., Wei, Z., Qian, Y., Li, J. and Wei, F. (2022), "SpeechT5:unified-modal encoder-decoder pre-training for spoken language processing", in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, *Association for Computational Linguistics*, *Dublin*, *Ireland*, pp. 5723-5738.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. and Agarwal, S. (2020), "Language models are few-shot learners", *Advances in Neural Information Processing Systems*, Vol. 33, pp. 1877-1901.

Bu, H., Du, J., Na, X., Wu, B. and Zheng, H. (2017), "Aishell-1: an open source Mandarin speech corpus and a speech recognition baseline", in *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and SpeechI/O Systems and Assessment (O-COCOSDA)*, *IEEE*, pp. 1-5.

Cai, J., Sunkara, M., Li, X., Bhatia, A., Pan, X. and Bodapati, S. (2023), "Masked audio text encoders are effective multi-modal rescorers", in *The 61st Annual Meeting Of The Association For Computational Linguistics*.

Chen, B., Xu, G., Wang, X., Xie, P., Zhang, M. and Huang, F. (2022a), "AISHELL-NER: named entity recognition from Chinese speech", in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics*, *Speech and Signal Processing (ICASSP)*, pp. 8352-8356.

Chen, C., Hu, Y., Yang, C.-H.H., Liu, H., Siniscalchi, S.M. and Chng, E.S. (2023), "Generative error correction for code-switching speech recognition using large language models", *arXiv preprint arXiv:2310.13013*.

Chen, C., Hu, Y., Yang, C.-H.H., Siniscalchi, S.M., Chen, P.-Y. and Chng, E.-S. (2024), "Hyporadise: an open baseline for generative speech recognition with large language models", *Advances in Neural Information Processing Systems*, Vol. 36.

Chen, S., Wang, C., Chen, Z., Wu, Y., Liu, S., Chen, Z., Li, J., Kanda, N., Yoshioka, T., Xiao, X. and Wu, J. (2022b), "Wavlm: large-scale self-supervised pre-training for full stack speech processing", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 16 No. 6, pp. 1505-1518.

Chen, Z., Zhang, Y., Rosenberg, A., Ramabhadran, B., Moreno, P.J., Bapna, A. and Zen, H. (2022c), "MAESTRO: matched speech text representations through modality matching", in *Interspeech 2022*, pp. 4093-4097.

Chiu, S.-H. and Chen, B. (2021a), "Innovative BERT-based reranking language models for speech recognition", in *Proceedings of SLT*.

Chiu, S.-H. and Chen, B. (2021b), "Innovative BERT-based reranking language models for speech recognition", in *2021 IEEE Spoken Language Technology Workshop (SLT)*, *IEEE*, pp. 266-271.

Conneau, A., Kiela, D., Schwenk, H., Barrault, L. and Bordes, A. (2017), "Supervised learning of universal sentence representations from natural language inference data", in *Proceedings of the 2017Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Copenhagen, Denmark*, pp. 670-680.

Dai, L., Chen, L., Zhou, Z. and Yu, K. (2022), "LatticeBART: lattice-to-lattice pre-training for speech recognition", in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6112-6116.

Du, J., Pu, S., Dong, Q., Jin, C., Qi, X., Gu, D., Wu, R. and Zhou, H. (2022),"Cross-modal ASR post-processing system for error correction and utterance rejection", *ArXiv, abs/2201.03313*, available at: https://api.semanticscholar.org/CorpusID:245837194

Dutta, S., Jain, S., Maheshwari, A., Pal, S., Ramakrishnan, G. and Jyothi, P. (2022), "Error correction in ASR using sequence-to-sequence models", *arXiv preprint arXiv:2202.01157*.

Errattahi, R., El Hannani, A. and Ouahmane, H. (2018), "Automatic speech recognition errors detection and correction: a review", *Procedia Computer Science*, Vol. 128, pp. 32-37.

Fohr, D. and Illina, I. (2021), "BERT-based semantic model for rescoring N-best speech recognition list", in *Interspeech 2021*, pp. 1867-1871.

Hannun, A.Y., Case, C., Casper, J., Catanzaro, B., Diamos, G.F., Elsen, E., Prenger, R.J., Satheesh, S., Sengupta, S., Rao, V., Coates, A. and Ng, A. (2014), "Deep speech: scaling up end-to-end speech recognition", *ArXiv, abs/1412.5567*.

Hinton, G., Deng, L., Yu, D., Dahl, G.E., Mohamed, A-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T.N. and Kingsbury, B. (2012), "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups", *IEEE Signal Processing Magazine*, Vol. 29 No. 6, pp. 82-97.

Hrinchuk, O., Popova, M. and Ginsburg, B. (2020), "Correction of automatic speech recognition with transformer sequence-to-sequence model", in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7074-7078.

Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhotia, K., Salakhutdinov, R. and Mohamed, A. (2021), "Hubert: self-supervised speech representation learning by masked prediction of hidden units", *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, Vol. 29, pp. 3451-3460.

Hu, Y., Chen, C., Qin, C., Zhu, Q., Chng, E. and Li, R. (2024), "Listen again and choose the right answer: a new paradigm for automatic speech recognition with large language models", in Ku, L.-W., Martins, A. and Srikumar, V. (Eds), *Findings of the Association for Computational Linguistics: ACL 2024*, *Association for Computational Linguistics*, Bangkok, Thailand, pp. 666-679, doi: 10.18653/v1/2024.findings-acl.37.

Illina, I. and Fohr, D. (2021), "DNN-based semantic rescoring models for speech recognition", in *Text, Speech, and Dialogue*, Springer International Publishing, Cham, pp. 357-370.

Jelinek, F. (1976), "Continuous speech recognition by statistical methods", *Proceedings of the IEEE*, Vol. 64 No. 4, pp. 532-556.

Jelinek, F. (1998), *Statistical Methods for Speech Recognition*, MIT Press.

Jurafsky, D. and Martin, J.H. (2000), *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, Upper Saddle River, NJ.

Kang, I.E., Van Gysel, C. and Siu, M.-H. (2024), "Transformer-based model for ASR N-best rescoring and rewriting", in *Interspeech 2024*, pp. 3505-3509, doi: 10.21437/Interspeech.2024-1449.

Kenton, J.D.M.-W.C. and Toutanova, L.K. (2019), "Bert: pre-training of deep bidirectional transformers for language understanding", in *Proceedings of NAACL-HLT*, Minneapolis, MN, Vol. 1, p. 2.

Kuo, C.-H. and Chen, K.-Y. (2022), "Correcting, rescoring and matching: an n-best list selection framework for speech recognition", in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), IEEE*, pp. 729-734.

Lee, Y., Shon, S. and Kim, T. (2018), "Learning pronunciation from a foreign language in speech synthesis networks", *arXiv preprintarXiv:1811.09364*.

Leng, Y., Tan, X., Zhu, L., Xu, J., Luo, R., Liu, L., Qin, T., Li, X., Lin, E. and Liu, T.-Y. (2021a), "Fast correct: fast error correction with edit alignment for automatic speech recognition", *Computation and Language*, Vol. 34, pp. 21708-21719, doi: 10.48550/arXiv.2105.03842.

Leng, Y., Tan, X., Wang, R., Zhu, L., Xu, J., Liu, W., Liu, L., Li, X.-Y., Qin, T., Lin, E. and Liu, T.-Y. (2021b), "Fast correct 2: fast error correction on multiple candidates for automatic speech recognition", in *Findings of the Association for Computational Linguistics: EMNLP 2021, Punta Cana, Dominican Republic Association for Computational Linguistics*, pp. 4328-4337.

Leng, Y., Tan, X., Liu, W., Song, K., Wang, R., Li, X.-Y., Qin, T., Lin, E. and Liu, T.-Y. (2023), "Soft correct: error correction with soft detection for automatic speech recognition", *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37 No. 11, pp. 13034-130342.

Li, K., Mahadeokar, J., Guo, J., Shi, Y., Keren, G., Kalinli, O., Seltzer, M.L. and Le, D. (2023), "Improving fast-slow encoder based transducer with streaming deliberation", in *ICASSP 2023 - 2023IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10095651.

Li, Q., Zhang, C. and Woodland, P.C. (2019), "Integrating source-channel and attention-based sequence-to-sequence models for speech recognition", in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 39-46, doi: 10.1109/ASRU46091.2019.9003837.

Logeswaran, L. and Lee, H. (2018), "An efficient framework for learning sentence representations", in *International Conference on Learning Representations*.

Ma, R., Li, H., Liu, Q., Chen, L. and Yu, K. (2020), "Neural lattice search for speech recognition", in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7794-7798.

Ma, R., Gales, M.J.F., Knill, K.M. and Qian, M. (2023a), "N-best T5: robust ASR error correction using multiple input hypotheses and constrained decoding space", in *Interspeech 2023*, pp. 3267-3271.

Ma, R., Qian, M., Manakul, P., Gales, M. and Knill, K. (2023b), "Can generative large language models perform ASR error correction?", *arXiv preprint arXiv:2307.04172*.

Mani, A., Palaskar, S., Meripo, N.V., Konam, S. and Metze, F. (2020), "ASR error correction and domain adaptation using machine translation", in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE*, pp. 6344-6348.

Min, Z. and Wang, J. (2023), "Exploring the integration of large language models into automatic speech recognition systems: an empirical study", in *International Conference on Neural Information Processing, Springer*, pp. 69-84.

Mu, B., Wan, X., Zheng, N., Zhou, H. and Xie, L. (2024), "MMGER: multi-modal and multi-granularity generative error correction with LLM for joint accent and speech recognition", *IEEE Signal Processing Letters*, Vol. 31, pp. 1940-1944, doi: 10.1109/LSP.2024.3432275.

Nie, M., Yan, M., Gong, C. and Chuxing, D. (2022), "Prompt-based reranking language model for ASR", in *INTERSPEECH*, pp. 3864-3868.

Ogawa, A., Delcroix, M., Karita, S. and Nakatani, T. (2018), "Rescoring N best speech recognition list based on one-on-one hypothesis comparison using encoder-classifier model", *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6099-6103.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A. and Schulman, J. (2022), "Training language models to follow instructions with human feedback", *Advances in Neural Information Processing Systems*, Vol. 35, pp. 27730-27744.

Panayotov, V., Chen, G., Povey, D. and Khudanpur, S. (2015), "Librispeech: an ASR corpus based on public domain audio books", in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, *IEEE*, pp. 5206-5210.

Pandey, P., Torres, S.D., Bayer, A.O., Gandhe, A. and Leutnant, V. (2022), "Lattention: lattice-attention in ASR rescoring", in *ICASSP2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7877-7881, doi: 10.1109/ICASSP43922.2022.9746737.

Prakash, J., Kumar, B., Hacioglu, K., Sharma, B., Gopalan, S., Chetlur, M., Venkatesan, S. and Stolcke, A. (2025), "Better pseudo-labeling with multi-ASR fusion and error correction by SpeechLLM", in *Interspeech 2025*, pp. 579-583, doi: 10.21437/Interspeech.2025-1707.

Pu, J., Nguyen, T.-S. and Stüker, S. (2023), "Multi-stage large language model correction for speech recognition", *arXiv preprintarXiv:2310.11532*.

Rabiner, L. and Juang, B. (1993), *Fundamentals of Speech Recognition, Prentice-Hall Signal Processing Series: Advanced Monographs*, PTR Prentice Hall.

Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C. and Sutskever, I. (2023), "Robust speech recognition via large-scale weak supervision", in *International Conference on Machine Learning, PMLR*, pp. 28492-28518.

Radhakrishnan, S., Yang, C.-H., Khan, S., Kumar, R., Kiani, N., Gomez-Cabrero, D. and Tegner, J. (2023), "Whispering LLaMA: a cross-modal generative error correction framework for speech recognition", in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Singapore*, pp. 10007-10016.

Rousseau, A., Deléglise, P. and Esteve, Y. (2014), "Enhancing the TEDLIUM corpus with selected data for language modeling and more TED talks", in *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), European Language Resources Association (ELRA)*, Reykjavik, Iceland, pp. 3935-3939.

Salazar, J., Liang, D., Nguyen, T.Q. and Kirchhoff, K. (2020), "Masked language model scoring", in Jurafsky, D., Chai, J., Schluter, N. and Tetreault, J. (Eds), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, *Association for Computational Linguistics*, pp. 2699-2712.

Shin, J., Lee, Y. and Jung, K. (2019), "Effective sentence scoring method using Bert for speech recognition", in *Asian Conference on Machine Learning, PMLR*, pp. 1081-1093.

Shivakumar, P.G., Kolehmainen, J., Gu, Y., Gandhe, A., Rastrow, A. and Bulyko, I. (2023), "Discriminative speech recognition rescoring with pre-trained language models", *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1-7.

Shunmuga Priya, M., Karthika Renuka, D. and Ashok Kumar, L. (2022), "Towards improving speech recognition model with post processing spell correction using BERT", *Journal of Intelligent and Fuzzy Systems*, Vol. 43 No. 4, pp. 4873-4882.

Tang, Z., Wang, D., Huang, S. and Shang, S. (2024), "Pinyin regularizationin error correction for Chinese speech recognition with large language models", in *Interspeech 2024*, pp. 1910-1914, doi: 10.21437/Interspeech.2024-987.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S. and Bikel, D. (2023b), "Llama 2: open foundation and fine-tuned chat models", *arXivpreprint arXiv:2307.09288*.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F. and Rodriguez, A. (2023a), "LLAMA: open and efficient foundation language models", *arXivpreprint arXiv:2302.13971*.

Tsai, S-F., Kuo, S.-C., Lyu, R.-Y. and Jang, J.-S.R. (2022), "Ensemble and re-ranking based on language models to improve ASR", *2022 13th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 185-189.

Tsimpoukelli, M., Menick, J.L., Cabi, S., Eslami, S., Vinyals, O. and Hill, F. (2021), "Multimodal few-shot learning with frozen language models", *Advances in Neural Information Processing Systems*, Vol. 34, pp. 200-212.

Udagawa, T., Suzuki, M., Kurata, G., Itoh, N. and Saon, G. (2022), "Effect and analysis of large-scale language model rescoring on competitive ASR systems", in *Interspeech 2022*, pp. 3919-3923.

Wang, H.-W., Yan, B.-C., Wang, Y.-C. and Chen, B. (2022a), "Effective ASR error correction leveraging phonetic, semantic information and N-best hypotheses", in *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 117-122.

Wang, W., Hu, K. and Sainath, T.N. (2022b), "Deliberation of streaming RNN-transducer by non-autoregressive decoding", in *ICASSP2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7452-7456, doi: 10.1109/ICASSP43922.2022.9746390.

Wang, Y.-W., Lu, K.-H. and Chen, K.-Y. (2024), "HypR: a comprehensive study for ASR hypothesis revising with a reference corpus", in *Interspeech 2024*, pp. 3495-3499.

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

Watanabe, S., Hori, T., Kim, S., Hershey, J.R. and Hayashi, T. (2017), "Hybrid CTC/attention architecture for end-to-end speech recognition", *IEEE Journal of Selected Topics in Signal Processing*, Vol. 11 No. 8, pp. 1240-1253.

Workshop, B., Scao, T.L., Fan, A., Akiki, C., Pavlick, E., Ilić, S., Hesslow, D., Castagn, R., Luccioni, A. S., Yvon, F., *et al*. (2022), "Bloom: A176b-parameter open-access multilingual language model", *arXivpreprint arXiv:2211.05100*.

Xu, L., Gu, Y., Kolehmainen, J., Khan, H., Gandhe, A., Rastrow, A., Stolcke, A. and Bulyko, I. (2022), "Rescorebert: discriminative speech recognition rescoring with Bert", in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, *IEEE*, pp. 6117-6121.

Yang, C.-H.H., Gu, Y., Liu, Y.-C., Ghosh, S., Bulyko, I. and Stolcke, A. (2023), "Generative speech recognition error correction With Large language models and task-activating prompting", in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 1-8.

Zhao, Y., Yang, X., Wang, J., Gao, Y., Yan, C. and Zhou, Y. (2021), "BART based semantic correction for mandarin automatic speech recognition system", in *Proceedings of INTER SPEECH*.

Zhu, L., Liu, W., Liu, L. and Lin, E. (2021), "Improving ASR error correction using n-best hypotheses", in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, *IEEE*, pp. 83-89.

Ziętkiewicz, T. (2022), "Tag and correct: high precision post-editing approach to correction of speech recognition errors", *2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pp. 939-942, available at: https://api.semanticscholar.org/CorpusID:252650761

Corresponding author
Kuan-Yu Chen can be contacted at: kychen@mail.ntust.edu.tw