Joint Token-level and Phrase-level Contextual Biasing for Automatic Speech Recognition with Large Language Models

Anonymous ACL submission

Abstract

End-to-end Automatic Speech Recognition (ASR) models often face challenges in accurately transcribing contextually relevant keywords, such as proper nouns or user-specific entities. Existing approaches leverage large language models (LLMs) to improve keyword recognition through token-level or phrase-level biasing. However, token-level approaches struggle to ensure holistic generation of keyword phrases, while phrase-level approaches may compromise the accuracy of non-keyword transcriptions. To overcome these limitations, we propose a novel joint approach that integrates token-level and phrase-level biasing, leveraging their complementary strengths. Our approach incorporates LLMs using a late-fusion mechanism, combining ASR and LLM outputs at both token and phrase levels. Experiments on Chinese and English datasets demonstrate that our approach achieves state-of-the-art performance on keyword-related metrics while preserving the high accuracy on non-keyword text. Ablation studies also confirm that the token-level and phrase-level components both significantly contribute to the improvement, complementing each other in our joint approach. The code and models will be publicly available at `https://github.com/`.

1 Introduction

Current end-to-end Automatic Speech Recognition (ASR) models, such as Whisper (Radford et al., 2022; Gandhi et al., 2023), have demonstrated impressive performance in transcribing common words. However, these models often struggle with accurately transcribing contextually relevant keywords (Alon et al., 2019; Yang et al., 2024b; Zhou et al., 2024b). Such keywords may include proper nouns or user-specific entities, such as contact names from a phone's address book. These keywords often convey important semantics, which are essential for understanding a sentence's meaning and performing related tasks accurately. In many scenarios, contextual keywords frequently occur within predefined contexts, such as contact names stored in a phone or previously searched terms. This contextual information is typically readily accessible. To improve the recognition of these keywords during transcription, prior researches have explored the use of contextual keyword dictionaries (Pundak et al., 2018; Alon et al., 2019; Lakomkin et al., 2024).

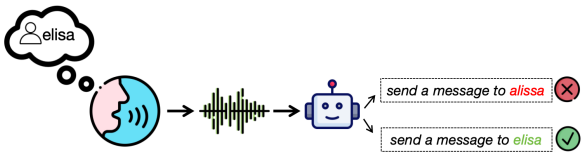

Figure 1: An example that illustrates the wrongly transcribed keywords by the ASR model.

Large language models (LLMs) have recently demonstrated exceptional capabilities in contextual modeling and reasoning across diverse tasks (OpenAI, 2023; Touvron et al., 2023; Yang et al., 2023; Abdin et al., 2024). These strengths align closely with the demands of contextual ASR, where the accurate recognition of user-relevant keywords heavily depends on contextual information. Consequently, there has been growing interest in leveraging LLMs to help ASR model better recognize keywords (Sun et al., 2024; Lakomkin et al., 2024; Yang et al., 2024b). Typically, contextual keyword dictionaries are provided to LLMs in the form of a prompt C. Based on this, Sun et al. (2024) proposed a two-pass approach. In the first pass, the ASR model generates multiple transcriptions. These candidates are then fed into the LLM, which computes second-pass scores conditioned on C. The LLM calculates the likelihood of each token sequentially during decoding and aggregates these token-level scores to produce an overall score for each candidate. The candidate with the highest

score is selected as the final result. In addition, Lakomkin et al. (2024) and Yang et al. (2024b) proposed early-fusion based methods. Beyond providing the contextual prompt C, these methods directly feed audio input into the LLM via adapters and train the model to predict transcription text. Similar to the two-pass approach, the decoding process in early-fusion methods operates at the token level, where the LLM generates transcription text in a token-by-token, autoregressive manner. Both approaches share the characteristic of relying on token-level operations during decoding. For this reason, we categorize them as token-level biasing methods in this paper.

However, keywords are often multi-token phrases, and token-level biasing approaches lack a holistic understanding of the entire keyword phrase. This limitation can result in incomplete generation of keywords. To address this issue, Zhou et al. (2024b) proposed a phrase-level biasing approach, introducing a phrase-level copy loss to guide the ASR model in selecting the correct keyword phrase. This approach allows the model to directly copy all tokens of a keyword phrase from a predefined dictionary, significantly improving the overall recall of keywords. However, the effectiveness of the phrase-level approach depends on the accurate selection of keywords from the dictionary. Zhou et al. (2024b) relied exclusively on speech information from the ASR model for keyword selection, which may result in incorrect choices. This, in turn, may adversely affect the recognition accuracy of non-keyword text.

In this paper, we propose a joint approach that integrates token-level and phrase-level approaches to improve ASR keyword recognition. Figure 2 illustrates the framework of our joint approach. Unlike previous works that utilize LLMs through two-pass or early-fusion strategies, we introduces a novel late-fusion based approach to incorporate LLMs. The overall process is as follows: the keyword list is first provided to the LLM in the form of a prompt. During transcription, the knowledge of ASR and LLM is fused in a late-fusion manner to improve keyword recognition. Specifically, in our token-level biasing, we fuse the token-level logit scores from the ASR and LLM to guide the model in generating tokens. In phrase-level biasing, we further fuse the hidden representations of the ASR and LLM to enable the model to select the correct keyword phrase from the dictionary. Finally, the results from token-level and phrase-level biasing are jointly considered through a re-normalization process to produce the final result.

We conduct experiments on both Chinese and English datasets (Chen et al., 2022; Zhou et al., 2024a; Wang et al., 2024). Experiments demonstrate that our proposed approaches achieve performance comparable to state-of-the-art methods, particularly on keyword-related metrics, while maintaining the accuracy on non-keyword text. Ablation studies show that both the token-level and phrase-level components play crucial roles in the joint framework. The joint strategy effectively integrates the strengths of both approaches, achieving further improvements. The contributions can be summarized as follows:

- We propose a joint approach that integrates token-level and phrase-level approaches, leveraging the strengths of both to improve ASR keyword recognition.
- Unlike previous works that utilize LLMs through two-pass or early-fusion strategies, our joint approach introduces a late-fusion based approach to incorporate LLMs with ASR.
- We evaluate our joint model on both Chinese and English datasets. The results show that our approach achieves performance on par with previous approaches and even surpasses them on keyword-related metrics. Additional analysis further confirms the effectiveness of our joint strategy. The codes and models will be released at `https://github.com/`.

2 Proposed Approach

The core idea of our approach is biasing the model to transcribe contextual keywords more accurately through LLM intervention. We achieve this by first injecting the knowledge of contextual keywords into the LLM. Then, we incorporate LLM-derived contextual information to bias the ASR model to transcribe the keywords in the speech input. Specifically, we propose a joint biasing framework that leverages LLMs to assist ASR models from both token-level and phrase-level perspectives. Our approach comprises four main components: knowledge injection, token-level biasing, phrase-level biasing, and joint modeling. Following sections will introduce each of these components in detail.

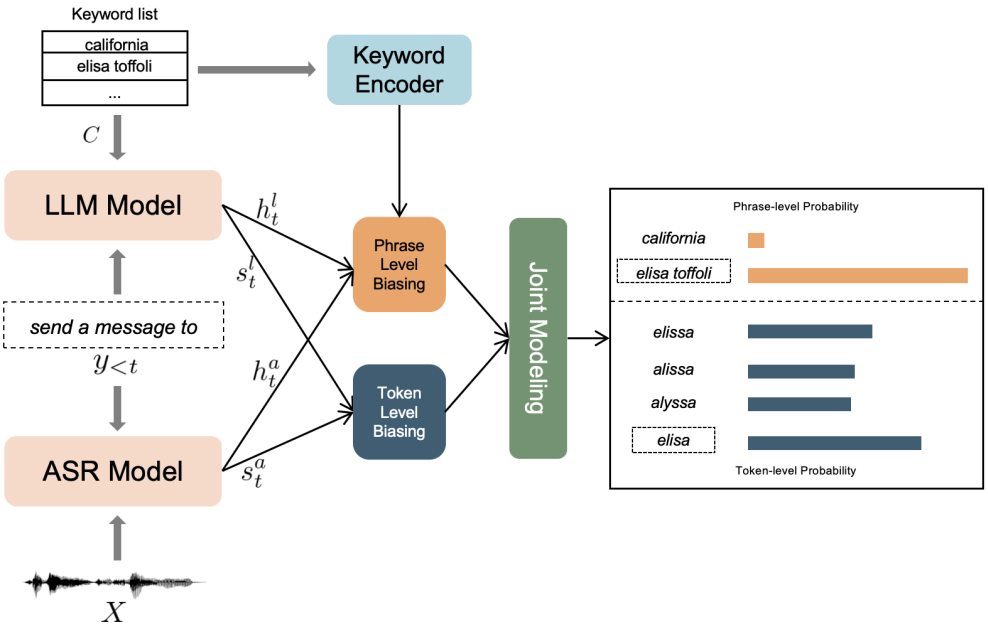

Figure 2: An example of transcribing the speech "send a message to elisa toffoli". Here, X is the speech input. C is the contextual prompt. $y_{<t}$ is the previously generated text. s_t^l, h_t^l and s_t^a, h_t^a are the logit score and hidden state of the LLM and ASR model, respectively.

2.1 Konwledge Injection

For a given keyword list $K = (k_1, k_2, ..., k_N)$ with N keywords, we concatenate the keywords into a single contextual prompt C and instruct the LLM to pay more attention to the keywords during the subsequent transcription process. Specifically, the prompt C is formulated as follows: *Transcribe the speech into text. The following keywords are likely to appear in the transcribed text output. Use relevant keywords to improve transcription accuracy and ignore irrelevant ones. The keywords are* k_1, k_2, ..., k_N. *The text corresponding to the speech is:*

In both two-pass (Li et al., 2023; Sun et al., 2024) and early-fusion (Lakomkin et al., 2024; Yang et al., 2024b) approaches, the keyword list is also provided to the LLM in the form of a prompt. However, in addition to the keyword prompt, two-pass approach requires candidate transcriptions as input to the LLM, while early-fusion approach requires audio representations to be processed by the LLM. In contrast, our approach only provides the keyword list as prior knowledge to the LLM. This design not only enables a highly efficient one-pass process but also eliminates the need for fine-tuning the model to handle audio inputs.

2.2 Token-level Biasing

Token-level biasing integrates the semantic information from the LLM and the acoustic information from the ASR model during the token-by-token decoding process. We propose a training-free late-fusion strategy that fuses the information from the LLM and ASR during the decoding. Specifically, at decoding step t, the LLM produces the logit score $\boldsymbol{s}_t^l$ for all tokens in the token vocabulary $\mathcal{V}$ based on the contextual prompt C and the previously generated text $y_{<t}$. Similarly, the ASR model produces the logit score $\boldsymbol{s}_t^a$ for the next token based on $y_{<t}$ and the speech input X. The $\boldsymbol{s}_t^l$ and $\boldsymbol{s}_t^a$ model the probabilities for the next token from semantic and acoustic perspectives, respectively.

$$\boldsymbol{h}_t^l, \boldsymbol{s}_t^l = \text{LLM}(y_{<t}, C) \quad (1)$$

$$\boldsymbol{h}_t^a, \boldsymbol{s}_t^a = \text{ASR}(y_{<t}, X) \quad (2)$$

Here, $\boldsymbol{h}_t^l$ and $\boldsymbol{h}_t^a$ are the hidden states of the LLM and ASR model.

We combine $\boldsymbol{s}_t^l$ and $\boldsymbol{s}_t^a$ to compute the final probability of generating next token. However, since $\boldsymbol{s}_t^l$ and $\boldsymbol{s}_t^a$ originate from different modalities, and the computation of the two scores relies on different information, they cannot be directly compared. For example, the computation of $\boldsymbol{s}_t^l$ depends solely on $y_{<t}$ and C. At the beginning of generation, $y_{<t}$ contains limited information, causing the LLM to predict within a broad semantic space. Excessive reliance on $\boldsymbol{s}_t^l$ may lead to the model deviating from the true transcription corresponding to the speech

input. On the contrary, $\boldsymbol{s}_t^a$ is based on the speech input X and $y_{<t}$, which may align better with the speech contents. Therefore, inspired by Chen et al. (2024), we employ a score fusion strategy based on uncertainty to balance the predictions of the LLM and ASR. The final token-level fusion formulas are as follows:

$$\boldsymbol{s}_t = \boldsymbol{s}_t^a + \text{sigmoid}(\boldsymbol{u}_t^a) \cdot \boldsymbol{s}_t^l \tag{3}$$

$$\boldsymbol{u}_t^a = -\boldsymbol{p}_t^a \cdot \log(\boldsymbol{p}_t^a) \tag{4}$$

$$p_{tok}(y_t|y_{<t}, C, X) = [\text{softmax}(\boldsymbol{s}_t)]_{y_t} \tag{5}$$

Here, $\boldsymbol{u}_t^a$ represents the uncertainty of the ASR model. $\boldsymbol{p}_t^a$ is the probability of the ASR model obtained from applying softmax to $\boldsymbol{s}_t^a$. It means that when the ASR model has high uncertainty, for example, if a keyword is difficult to distinguish based on speech information alone, more weight is given to the LLM's prediction. Conversely, at the beginning of the generation, when the ASR model is more confident, greater weight is assigned to the ASR's prediction. $\boldsymbol{s}_t$ and $p_{tok}(y_t|y_{<t}, C, X)$ represent the final fused token-level score and probability, respectively.

2.3 Phrase-level Biasing

Unlike the token-level approach, which applies biasing to each individual token in the vocabulary, the phrase-level approach biases the model to focus on entire key phrases, enabling it to transcribe target keywords as cohesive units.

Specifically, we design a phrase fusion module to integrate the phrase-level representation of each keyword, the contextual semantic information from the LLM, and the speech information from the ASR to determine the phrase-level probability for each keyword. At decoding step t, the module estimates the probability of generating a complete phrase from the keyword list.

First, we obtain the phrase-level representation $\boldsymbol{r}_i$ for each keyword k_i in the keyword list K using a keyword encoder. Following CopyNE (Zhou et al., 2024b), we use a light-weight LSTM as the keyword encoder to encode each keyword. We take the hidden state of the last token as the phrase-level representation $\boldsymbol{r}_i$.

$$\boldsymbol{r}_i = \text{LSTM}(k_i) \tag{6}$$

Then, for contextual semantic information and acoustic information, we use the hidden states $\boldsymbol{h}_t^l$ and $\boldsymbol{h}_t^a$ from the LLM and ASR model, as shown in equation 1 and 2. With $\boldsymbol{r}_i$, $\boldsymbol{h}_t^l$, and $\boldsymbol{h}_t^a$ obtained, we compute the phrase-level probability $p_{phr}(k_i|y_{<t}, C, X)$ for each keyword k_i using a dot-product attention mechanism.

First, $\boldsymbol{h}_t^l$ and $\boldsymbol{h}_t^a$ are concatenated and passed through a linear layer to obtain the attention query $\boldsymbol{q}_t$. Then, we calculate the attention score between the query $\boldsymbol{q}_t$ and each phrase-level representation $\boldsymbol{r}_i$. Finally, a softmax function is applied to obtain the phrase-level probability $p_{phr}(k_i|y_{<t}, C, X)$.

$$\boldsymbol{q}_t = \text{Linear}([\boldsymbol{h}_t^l; \boldsymbol{h}_t^a]) \tag{7}$$

$$p_{phr}(k_i|y_{<t}, C, X) = \frac{\exp(\boldsymbol{q}_t \cdot \boldsymbol{r}_i)}{\sum_{j=1}^{N} \exp(\boldsymbol{q}_t \cdot \boldsymbol{r}_j)} \tag{8}$$

2.4 Joint Modeling

The token-level probability $p_{tok}(y_t|y_{<t}, C, X)$ guides the model on which token to generate next, while the phrase-level probability $p_{phr}(k_i|y_{<t}, C, X)$ informs the model to pick an entire keyword phrase. Different choices between generating individual tokens and entire keywords can result in varying final outcomes. So it is important to jointly consider the two probabilities to make the best transcription. However, it is non-trivial to directly combine the two probabilities, as they are computed based on different information and have different scales. To make the two probabilities comparable, we use a simple yet effective method to normalize them to the same scale following Zhou et al. (2024b).

Specifically, we add a fake keyword k_0 to the keyword list K. When the model should generate tokens rather than select a keyword, we train it to predict k_0. The equation (8) becomes:

$$p_{phr}(k_i|y_{<t}, C, X) = \frac{\exp(\boldsymbol{q}_t \cdot \boldsymbol{r}_i)}{\sum_{j=0}^{N} \exp(\boldsymbol{q}_t \cdot \boldsymbol{r}_j)} \tag{9}$$

The phrase-level probability of the fake keyword k_0, i.e., $p_{phr}(k_0|y_{<t}, C, X)$ means the model should generate tokens in the next step. Accordingly, it can be treated as the prior probability of generating tokens. Finally, the two sorts of probability, i.e., $p_{tok}(y_t|y_{<t}, C, X)$ and $p_{phr}(k_i|y_{<t}, C, X)$, can be jointly normalized to the same scale.

$$p_{joi}(z_i|y_{<t}) = \begin{cases} p_{phr}(k_0|y_{<t}) p_{tok}(z_i|y_{<t}), & z_i \in \mathcal{V} \\ p_{phr}(z_i|y_{<t}), & z_i \in K, z_i \neq k_0 \end{cases} \tag{10}$$

Here, $\mathcal{V}$ is the token vocabulary. z_i is a token from $\mathcal{V}$ or a keyword from K. For abbreviation, we omit the condition C and X in the equation. After normalization, tokens in the vocabulary and keywords in the keyword list form a unified space $\mathcal{Z} = \mathcal{V} \cup K$. The probabilities from token-level and phrase-level biasing are jointly normalized in $p_{joi}(z_i|y_{<t})$. We finally generate the final transcription by doing beam search over $p_{joi}(z_i|y_{<t})$.

3 Experiments

3.1 Experimental Setup

Datasets. We conduct experiments on Chinese and English datasets. For Chinese, we use the Aishell dataset (Bu et al., 2017; Chen et al., 2022) and the RWCS-NER dataset (Zhou et al., 2024a). The Aishell dataset contains 150 hours of clean speeches recorded in quiet environments. RWCS-NER is a test set proposed to evaluate the performance of Chinese Spoken NER in real-world scenarios. It covers two domains: open-domain daily conversations (DC) and task-oriented intelligent cockpit instructions (ICI). Both the Aishell and RWCS-NER datasets are annotated with transcriptions and the named entities. We utilize the entities as the corresponding keyword list.

For English, we use the Slidespeech dataset (Wang et al., 2024), a large-scale corpus enriched with slide information. Each speech sample is paired with its corresponding transcription as well as a keyword list extracted from the associated slide. We use the 473-hour L95 subset of Slidespeech as the training set.

Backbone Models. Our joint model has two main components: the ASR model and the LLM model. For the ASR model, we select the Whisper-small model[1] (Radford et al., 2022). As for the LLM model, we choose the Qwen2-1.5B[2] (Yang et al., 2024a) and the Phi-3.5-mini[3] (Abdin et al., 2024) for the experiments on Chinese and English respectively.

3.2 Training

Loss Function. The loss function $\mathcal{L}$ of our model is composed of two parts: the token-level loss $\mathcal{L}_{tok}$ and the phrase-level loss $\mathcal{L}_{phr}$. First, given the ground-truth transcription $Y = y_1, y_2, \ldots, y_T$ of the speech, $\mathcal{L}_{tok}$ is the negative log-likelihood loss to generate the transcription Y.

$$\mathcal{L}_{tok} = -\sum_{t=1}^{T} p_{tok}(y_t|y_{<t}, C, X) \quad (11)$$

Second, the phrase-level loss $\mathcal{L}_{phr}$ is used to teach the model to choose the correct keywords from the keyword list K. Given K, we apply the maximum matching algorithm to find all the phrases in the transcription Y that also listed in the keyword list K. Through this process, we can obtain a sequence $P = p_1, p_2, \ldots, p_T$. $p_i = k_j$ if the j-th keyword is matched at time step i. Otherwise, $p_i = k_0$, which is a fake keyword indicating no keyword is matched. So the phrase-level loss $\mathcal{L}_{phr}$ is the loss to generate the phrase-level sequence P.

$$\mathcal{L}_{phr} = -\sum_{t=1}^{T} p_{phr}(p_t|p_{<t}, C, X) \quad (12)$$

Finally, the total loss $\mathcal{L}$ is the sum of $\mathcal{L}_{tok}$ and $\mathcal{L}_{phr}$.

$$\mathcal{L} = \mathcal{L}_{tok} + \mathcal{L}_{phr} \quad (13)$$

Training Details. We train our model on the Aishell training set and the L95 subset of Slidespeech respectively. In the whole training process, the LLM model is frozen. Since the token-level biasing requires the ASR model and the LLM model to have the same token vocabulary, we replace the tokenizer of the ASR model with the one in LLM (Chen et al., 2024). Therefore, during the training process, we only update the ASR decoder, Keyword Encoder, and the linear layer in the phrase-level biasing module. The number of parameters to be updated is about 300M.

We use the AdamW optimizer (Loshchilov et al., 2017) to train our model. The learning rate is set to 1e-4 and the batch size is 60. The model is trained for 40 epochs with the first 4 epochs used for warm-up. The training process takes about 40 hours on three A100 40G GPUs. We report the results of the best model on the validation set.

3.3 Evaluation

Baselines. On the Chinese dataset, we first fine-tune the Whisper model (Radford et al., 2022) on the Aishell training set as the initial baseline. Additionally, the Whisper baseline also corresponds to our joint model with both token-level and phrase-level biasing removed. CopyNE (Zhou et al.,

[1]https://huggingface.co/openai/whisper-small
[2]https://huggingface.co/Qwen/Qwen2-1.5B
[3]https://huggingface.co/microsoft/Phi-3.5-mini-instruct

Model	Aishell Entity Char Ratio=9.58%				DC Entity Char Ratio=8.34%				ICI Entity Char Ratio=18.86%			
	CER↓	B-CER↓	U-CER↓	R↑	CER↓	B-CER↓	U-CER↓	R↑	CER↓	B-CER↓	U-CER↓	R↑
Whisper	5.2	10.4	4.7	80.6	12.8	22.9	11.7	71.1	11.5	30.7	6.9	40.8
CopyNE	4.6	3.4	4.7	94.4	12.2	14.9	11.8	82.0	8.9	16.8	7.0	70.0
Ours	**3.7**	**2.2**	**3.8**	**96.4**	**11.0**	**11.4**	**10.8**	**86.6**	**7.7**	**10.9**	**6.8**	**79.8**
Ours w/o P	4.3	7.0	4.0	86.9	11.6	17.9	10.8	77.3	10.2	20.1	7.6	61.8
Ours w/o T	4.5	2.7	4.7	95.5	12.1	13.2	11.8	84.3	8.6	14.7	7.0	73.1

Table 1: Results on Aishell, DC, and ICI test sets. Entity Char Ratio is the ratio of the entity characters in the dataset. w/o P and w/o T indicate the exclusion of the phrase-level and token-level modules, respectively.

Model	Slidespeech Keyword Ratio=6.72%			
	WER↓	B-WER↓	U-WER↓	R↑
Whisper	10.9	8.0	11.2	92.3
CopyNE	10.9	6.7	11.2	93.5
MaLa-ASR	**9.1**	5.5	**9.4**	94.9
Ours	10.8	**5.3**	11.2	**95.2**

Table 2: Results on the Slidespeech test set.

2024b) is the second baseline. It is a phrase-level biasing model which also utilizes the keyword dictionary to improve the accuracy of keywords.

On English, besides the Whisper and CopyNE, we also compare with MaLa-ASR (Yang et al., 2024b), a state-of-the-art model on the Slidespeech dataset. It's a token-level biasing model that follows the early-fusion paradigm and utilizes both the keyword dictionary and the LLM model.

To ensure a fair comparison with the baselines, all models are evaluated using the same keyword list. Following MaLa-ASR, we set the keyword list size to 50 for the primary experiments. We will also evaluate the effect of the keyword list size in the ablation studies.

Metrics. Following Zhou et al. (2024b) and Yang et al. (2024b), we report the character error rate (CER) and the word error rate (WER) for the Chinese and English respectively. In addition to the CER/WER, to evaluate the performance on the keywords, we also report the keyword related metrics including biased character error rate (B-CER), biased word error rate (B-WER), and keyword recall (Recall). B-CER and B-WER measure the error rate of the transcriptions corresponding to the keywords. Recall is the percentage of the keywords that are correctly and fully predicted.

On the contrary, U-CER and U-WER are the CER and WER of the transcriptions that do not contain keywords, which can also reflect the impact of different models on non-keyword texts.

3.4 Results

Main Results. For Chinese, we evaluate the models on the test sets of Aishell, DC, and ICI. Since the models are fine-tuned on the Aishell training set, their performance on DC and ICI reflects their ability to generalize to out-of-domain scenarios. Table 1 presents the results across all three datasets.

Overall, our model achieves the best performance across all metrics, particularly those related to keywords. Specifically, compared to CopyNE, our joint model achieves absolute reductions in B-CER of 1.28%, 3.51%, and 5.9% on Aishell, DC, and ICI, respectively. Similarly, for Recall, our model outperforms CopyNE with absolute improvements of 2%, 4.59%, and 9.96% on the three datasets. Notably, the advantages of our method become even more pronounced when the keyword density is higher. For instance, in ICI, where the ratio of entity characters reaches 18%, the improvement achieved by our method is the largest among the three datasets.

Table 2 presents the results on the test set of the English Slidespeech dataset. Overall, our model outperforms Whisper and CopyNE and achieves performance comparable to MaLa-ASR. A closer comparison with MaLa-ASR reveals that our model excels in keyword-related metrics, such as B-WER and Recall, but falls behind in WER and U-WER. We hypothesize two main reasons.

First, the LLM used in MaLa-ASR is the 7B Vicuna (Chiang et al., 2023), which is significantly larger than the 3.8B Phi model employed in our joint approach. Second, MaLa-ASR adopts an early-fusion design, where the audio input is directly fed into the LLM. This allows the model to fully utilize the parameters of the LLM to fit the data. Given that non-keyword text dominates the dataset (approximately 95%), the early-fusion model effectively learns to transcribe non-keyword text. However, its ability to handle keywords is comparatively weaker. In contrast, our model em-

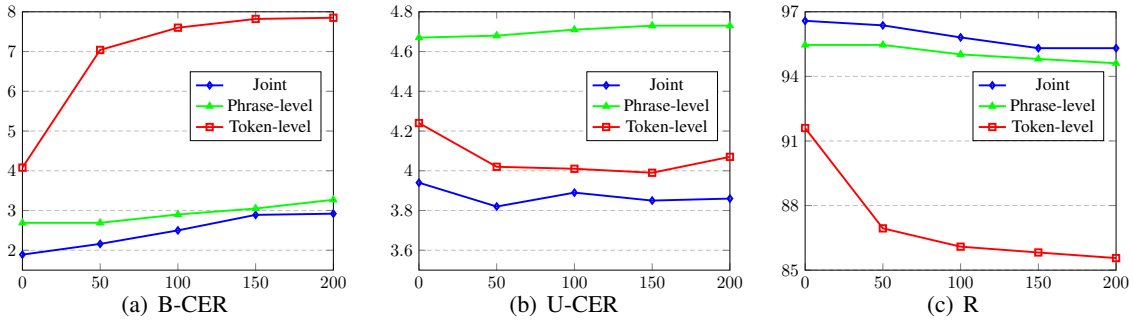

Figure 3: Effect of the keyword list size.

ploys a late-fusion approach, where the LLM is primarily used to enhance the generation of keyword-related text. As a result, we achieve superior performance on metrics associated with keywords.

Token-level vs. Phrase-level. To evaluate the effectiveness of token-level and phrase-level biasing, we conduct ablation studies on the three Chinese datasets. The lower half of Table 1 reports the results after removing the phrase-level biasing module (w/o P) and the token-level biasing module (w/o T), respectively. The results indicate that removing phrase-level biasing while retaining token-level biasing leads to a significant decrease in performance on keyword-related metrics (B-CER, Recall), with minimal impact on non-keyword metrics (CER, U-CER). Conversely, removing token-level biasing results in a substantial degradation in non-keyword metric, while its impact on keyword-related metrics is less pronounced. Despite these degradations, both configurations outperform the Whisper baseline, which lacks any biasing mechanism.

These findings suggest that token-level and phrase-level biasing contribute to recognition accuracy from complementary perspectives. Our phrase-level biasing effectively improves keyword recognition accuracy without compromising non-keyword accuracy, while token-level biasing enhances the recognition of non-keyword text. This is likely because the training data for text-based LLMs contains significantly more non-keyword text than keyword text, leading the LLM's better capability on the non-keyword text.

Effect of the Keyword List Size. The size of the keyword list significantly impacts model performance. When the list becomes larger, it introduces many noisy keywords that are irrelevant to the context, which can degrade the model's ability. Therefore, it is essential to analyze the robustness of different models. As shown in Figure 3, we examine the effect of keyword list size on the Aishell dataset. Since the overall CER is minimally affected by the list size, we focus on the variations in the B-CER, U-CER, and Recall.

First, it can be observed that the joint method consistently achieves the best performance across all settings of keyword list size in terms of B-CER, U-CER, and Recall. Second, the analysis reveals that as the keyword list size increases, the performance of all methods declines to some extent. Among them, the token-level biasing shows the largest decline, indicating poor stability, whereas the phrase-level biasing demonstrates the smallest decline, reflecting superior robustness. The joint approach effectively integrates the strengths of both token-level and phrase-level biasing, inheriting the robustness of the phrase-level approach, and thus exhibits good stability too.

Moreover, an intriguing phenomenon is observed in Figure 2(b): when the keyword list size increases, the U-CER of the token-level method unexpectedly decreases, which contradicts the general trend observed in B-CER and Recall. We hypothesize that this is because the extended keyword list results in a longer prompt C being fed into the LLM, which activates its general language modeling capabilities and improves the transcription accuracy for non-keyword text.

4 Related Works

4.1 Contextual Biasing

Using keyword dictionaries to bias ASR models for better transcription has been widely studied. Depending on the granularity of the biasing approach, these approaches can be categorized into token-level and phrase-level approaches.

Token-level. Pundak et al. (2018) proposed to use an additional encoder to encode each keyword in the keyword list and applying an attention mechanism to produce a vector representation of the entire list. This representation is then provided as an auxiliary input alongside the ASR input, biasing the model toward generating tokens from the keyword list during token-by-token transcription. Subsequent works have refined this setup in various ways, such as adding noise during training to improve the model's robustness (Alon et al., 2019), employing better keyword encoders (Fu et al., 2023), or adopting strategies to ensure that the keyword dictionary excludes irrelevant keywords (Jayanthi et al., 2023). However, token-level approaches lack a holistic understanding of keywords as complete phrases, often leading to incomplete keyword generation and lower recall rates.

Phrase-level. Phrase-level approaches focus more on generating complete forms of target keywords compared to token-level approaches, ensuring the overall correctness of the keywords. Zhou et al. (2024b) first introduced the copy mechanism into ASR, allowing the model to directly copy target keywords from a predefined keyword list. Similarly, Sudo et al. (2024) proposed merging the keyword list with the original token vocabulary into an extended vocabulary, enabling the model to predict entire phrases directly during inference. While phrase-level approaches significantly improve the accuracy and integrity of keyword generation, they often reduce the accuracy of non-keyword text to some extent.

In contrast, our work introduces a joint biasing approach that integrates token-level and phrase-level methods. It achieves a significant improvement in keyword accuracy without compromising the accuracy of non-keyword text.

4.2 Using LLMs in ASR

Language models (LMs) acquire rich linguistic knowledge through pre-training on large-scale unlabeled text. Traditionally, n-gram LMs have been widely used in ASR systems to enhance the naturalness of generated text (Yao et al., 2021; Chorowski and Jaitly, 2016; Sriram et al., 2017). Recently, the emergence of LLMs with powerful contextual modeling and reasoning capabilities has prompted researchers to focus on leveraging these models to enhance ASR performance, particularly in keyword recognition. These approaches can be broadly categorized into two types: early-fusion and two-pass approaches.

Early-fusion. In early-fusion approaches, speech features and text prompts are combined at the initial input stage. Specifically, the keyword list is formatted into a prompt, which is paired with the speech input. The speech input is first processed by a speech encoder to extract features, which are then concatenated with the embeddings of the text prompt. The fused representation is subsequently fed into the LLM. The LLM is directly trained to generate transcriptions based on this fused input (Yang et al., 2024b; Lakomkin et al., 2024; Bai et al., 2024).

Two-pass. Unlike early-fusion approaches, two-pass approaches do not involve speech features. Instead, they leverage LLMs to rescore or reorder all candidate texts produced by the ASR system and select the highest-scoring text as the final output. For example, Sun et al. (2024) explored the application of in-context learning in the rescoring stage, achieving significant improvements in keyword recognition. Ogawa et al. (2024) investigated the impact of historical conversational context under the two-pass framework. However, the two-pass processing may introduce additional latency, which is undesirable in real-time applications.

In this work, we propose a novel late-fusion-based approach that leverages LLMs to assist ASR in keyword recognition. Unlike early-fusion approaches, our approach does not require training the model to support speech inputs. In addition, our approach operates in a one-pass manner, where the LLM intervenes directly during the decoding stage, biasing the model to better recognize keywords.

5 Conclusion

In this paper, we propose a joint approach that combines token-level and phrase-level biasing approaches to enhance keyword recognition in ASR systems. Through the introduction of a late-fusion mechanism, our approach effectively leverages the advanced contextual modeling capabilities of LLMs to assist ASR models, achieving superior accuracy on keywords while preserving robust performance on non-keyword text. We hope our work can pave the way for new applications of LLMs in ASR systems and provide a foundation for further research in the area of ASR keyword recognition.

Limitations

While our approach demonstrates significant improvements in keyword recognition compared to traditional ASR models, leveraging LLMs introduces additional computational and resource overhead. In addition, our analysis shows that as the size of the keyword dictionary increases, the performance tends to degrade, particularly when using late-fusion based token-level biasing with LLMs. In future work, we aim to explore strategies to mitigate the challenges associated with large keyword dictionaries, ensuring both efficiency and accuracy.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, et al. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

Uri Alon, Golan Pundak, and Tara N Sainath. 2019. Contextual speech recognition with difficult negative training examples. In *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6440–6444.

Ye Bai, Jingping Chen, Jitong Chen, Wei Chen, Zhuo Chen, Chuang Ding, Linhao Dong, Qianqian Dong, Yujiao Du, Kepan Gao, et al. 2024. Seed-asr: Understanding diverse speech and contexts with llm-based speech recognition. *arXiv preprint arXiv:2407.04675*.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 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 Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5.

Boli Chen, Guangwei Xu, Xiaobin Wang, Pengjun Xie, Meishan Zhang, and Fei Huang. 2022. Aishell-ner: Named entity recognition from chinese speech. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8352–8356.

Chen Chen, Ruizhe Li, Yuchen Hu, Sabato Marco Siniscalchi, Pin-Yu Chen, EngSiong Chng, and Chao-Han Huck Yang. 2024. It's never too late: Fusing acoustic information into large language models for automatic speech recognition. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Jan Chorowski and Navdeep Jaitly. 2016. Towards better decoding and language model integration in sequence to sequence models. *arXiv preprint arXiv:1612.02695*.

Xuandi Fu, Kanthashree Mysore Sathyendra, Ankur Gandhe, Jing Liu, Grant P. Strimel, Ross McGowan, and Athanasios Mouchtaris. 2023. Robust acoustic and semantic contextual biasing in neural transducers for speech recognition. In *2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

Sanchit Gandhi, Patrick von Platen, and Alexander M Rush. 2023. Distil-whisper: Robust knowledge distillation via large-scale pseudo labelling. *arXiv preprint arXiv:2311.00430*.

Sai Muralidhar Jayanthi, Devang Kulshreshtha, Saket Dingliwal, Srikanth Ronanki, and Sravan Bodapati. 2023. Retrieve and copy: Scaling asr personalization to large catalogs. *arXiv preprint arXiv:2311.08402*.

Egor Lakomkin, Chunyang Wu, Yassir Fathullah, Ozlem Kalinli, Michael L Seltzer, and Christian Fuegen. 2024. End-to-end speech recognition contextualization with large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 12406–12410. IEEE.

Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu. 2023. Prompting large language models for zero-shot domain adaptation in speech recognition. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Ilya Loshchilov, Frank Hutter, et al. 2017. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 5.

Atsunori Ogawa, Naoyuki Kamo, Kohei Matsuura, Takanori Ashihara, Takafumi Moriya, Takatomo Kano, Naohiro Tawara, and Marc Delcroix. 2024. Applying llms for rescoring n-best asr hypotheses of casual conversations: Effects of domain adaptation and context carry-over. *arXiv preprint arXiv:2406.18972*.

OpenAI. 2023. GPT-4 technical report. *ArXiv preprint*, abs/2303.08774.

Golan Pundak, Tara N Sainath, Rohit Prabhavalkar, Anjuli Kannan, and Ding Zhao. 2018. Deep context: end-to-end contextual speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 418–425.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision. *arXiv preprint arXiv:2212.04356*.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. 2017. Cold fusion: Training seq2seq models together with language models. *arXiv preprint arXiv:1708.06426*.

Yui Sudo, Yosuke Fukumoto, Muhammad Shakeel, Yifan Peng, and Shinji Watanabe. 2024. Contextualized automatic speech recognition with dynamic vocabulary. *arXiv preprint arXiv:2405.13344*.

Chuanneng Sun, Zeeshan Ahmed, Yingyi Ma, Zhe Liu, Lucas Kabela, Yutong Pang, and Ozlem Kalinli. 2024. Contextual biasing of named-entities with large language models. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 10151–10155. IEEE.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Haoxu Wang, Fan Yu, Xian Shi, Yuezhang Wang, Shiliang Zhang, and Ming Li. 2024. Slidespeech: A large scale slide-enriched audio-visual corpus. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 11076–11080. IEEE.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Guanrou Yang, Ziyang Ma, Fan Yu, Zhifu Gao, Shiliang Zhang, and Xie Chen. 2024b. Mala-asr: Multimedia-assisted llm-based asr. *arXiv preprint arXiv:2406.05839*.

Zhuoyuan Yao, Di Wu, Xiong Wang, Binbin Zhang, Fan Yu, Chao Yang, Zhendong Peng, Xiaoyu Chen, Lei Xie, and Xin Lei. 2021. Wenet: Production oriented streaming and non-streaming end-to-end speech recognition toolkit. In *Proc. Interspeech*, Brno, Czech Republic. IEEE.

Shilin Zhou, Zhenghua Li, Chen Gong, Lei Zhang, Yu Hong, and Min Zhang. 2024a. Chinese spoken named entity recognition in real-world scenarios: Dataset and approaches. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1872–1884, Bangkok, Thailand. Association for Computational Linguistics.

Shilin Zhou, Zhenghua Li, Yu Hong, Min Zhang, Zhefeng Wang, and Baoxing Huai. 2024b. CopyNE: Better contextual ASR by copying named entities. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2675–2686, Bangkok, Thailand. Association for Computational Linguistics.