CopyNE: Better Contextual ASR by Copying Named Entities

Shilin Zhou[1], **Zhenghua Li**[1][*]; **Yu Hong**[1],
Min Zhang[1], **Zhefeng Wang**[2], **Baoxing Huai**[2]
[1]School of Computer Science and Technology, Soochow University, China
[2]Huawei Cloud, China
slzhou.cs@outlook.com; {zhli13,hongy,minzhang}@suda.edu.cn
{wangzhefeng,huaibaoxing}@huawei.com

Abstract

End-to-end automatic speech recognition (ASR) systems have made significant progress in general scenarios. However, it remains challenging to transcribe contextual named entities (NEs) in the contextual ASR scenario. Previous approaches have attempted to address this by utilizing the NE dictionary. These approaches treat entities as individual tokens and generate them token-by-token, which may result in incomplete transcriptions of entities. In this paper, we treat entities as indivisible wholes and introduce the idea of copying into ASR. We design a systematic mechanism called CopyNE, which can copy entities from the NE dictionary. By copying all tokens of an entity at once, we can reduce errors during entity transcription, ensuring the completeness of the entity. Experiments demonstrate that CopyNE consistently improves the accuracy of transcribing entities compared to previous approaches. Even when based on the strong Whisper, CopyNE still achieves notable improvements.

1 Introduction

End-to-end automatic speech recognition (ASR) systems have achieved impressive performance in general scenarios (Chan et al., 2016; Rao et al., 2017; Gulati et al., 2020; Boulianne, 2022). However, in the contextual ASR scenario where speech often contains numerous contextual entities, it remains a challenge for ASR systems to get accurate transcriptions (Alon et al., 2019; Jayanthi et al., 2023). For instance, when utilizing personal voice assistants like Siri or Alexa, it is common to encounter contextual entities such as personal names, place names, and organization names. ASR models trained solely on speech-text data often struggle to transcribe these personalized entities due to their infrequent occurrence in the training set (Sathyendra et al., 2022). Since contextual entities always cover a wealth of semantic information. It is important to improve the accuracy of transcribing entities for downstream natural language processing tasks such as information retrieval and spoken language understanding (Ganesan et al., 2021; Wu et al., 2022).

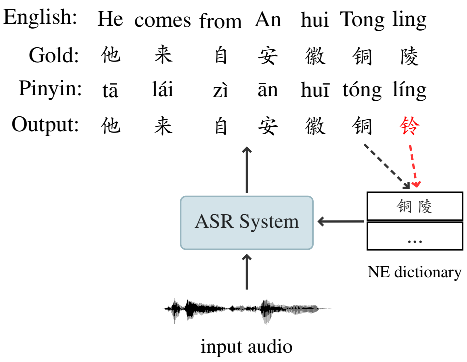
English: He comes from An hui Tong ling
Gold: 他 来 自 安 徽 铜 陵
Pinyin: tā lái zì ān huī tóng líng
Output: 他 来 自 安 徽 铜 铃
ASR System
铜 陵
...
NE dictionary
input audio

Figure 1: An example with homophonic errors. Pinyin is the Mandarin pronunciation of each token. The red text indicates the wrongly predicted token.

Recently, researchers have started leveraging the information of textual modality as additional contextual knowledge to help contextual ASR. The most typical approach, premised on the assumption that entities are already known before, use a contextual named entity (NE) dictionary as contextual knowledge (Chen et al., 2019; Jain et al., 2020; Han et al., 2021; Huber et al., 2021; Fu et al., 2023). Two representative approaches are "contextual listen, attend and spell" (CLAS) (Pundak et al., 2018) and contextual bias attention (CBA) (Zhang and Zhou, 2022). CLAS employs the knowledge of the dictionary to aid the prediction of each token. They use dictionary representation as extra inputs for token prediction in the decoder. The decoder attends to each entity, and the dictionary representation is an aggregated representation of all entities, weighted by the attention scores. CBA extends CLAS and uses an extra training loss. The loss explicitly makes use of the attention scores and force the model attend to a proper entity in the

[*] Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2675–2686
August 11-16, 2024 ©2024 Association for Computational Linguistics

dictionary if the token to be predicted is related with the entity.

Previous methods have achieved considerable improvements, especially in transcribing entities. However, they all treat entities as individual tokens. These models utilize contextual knowledge to aid in predicting independent tokens without considering the role of these tokens in constituting a complete entity. In other words, a multi-token entity is broken into isolated tokens during decoding. We argue that this is problematic. For instance, model may erroneously generate the subsequent tokens of an entity, despite correctly producing the preceding tokens. As shown in Figure 1, when transcribing the speech "他来自安徽铜陵" (He comes from Anhui Tongling), an incorrect output of "他来自安徽铜铃" (He comes from Anhui copper bell) is obtained. Despite the model's awareness of the location entity "铜陵" in the NE dictionary and its accurate prediction of the first token "铜", it mistakenly transcribes "陵" (ling) as "铃" (bell) during the token-level prediction process. This occurs because the model predicts tokens independently, neglecting the integrity of the token span as a complete entity. Furthermore, "陵" and "铃" share the same pronunciation "líng", with "铃" being a more frequently occurring token in the training data. Consequently, the model tends to generate the wrong token "铃".

In this paper, we propose a new approach for contextual ASR called CopyNE. Unlike previous approaches, we view entities as indivisible wholes. To the best of our knowledge, we are the first to introduce the idea of copying into ASR. We design a systematic and effective mechanism to copy entities from a dictionary. Specifically, CopyNE uses a copy loss that guides the model to copy the correct entity from the dictionary. During inference, our CopyNE has the flexibility to either predict a token from the token vocabulary or copy an entity from the NE dictionary at each decoding step. By copying multiple tokens simultaneously, we can alleviate errors within the entity, thus ensuring the token span as a complete entity.

Experiments on Chinese Aishell (Bu et al., 2017), ST-cmds[1], and English Eng (Yadav et al., 2020) datasets show that our CopyNE achieves significant improvements across all scenarios, particularly in the contextual ASR scenario. Compared to previous methods using dictionary, CopyNE achieves relative reductions in CER of 13.5% and 20.8% on Aishell and ST-cmds in the contextual scenario. Notably, CopyNE shows more remarkable improvements when it comes to transcribing entities, with relative reductions of 55.4% and 53.9% in the NE-CER metric on Aishell and ST-cmds. Moreover, when based on Whisper (Radford et al., 2022) and evaluated in its domain of expertise, Eng dataset, CopyNE still achieves an impressive 6.4% and 16.8% relative reductions in WER and NE-WER. We will release our codes, configurations, and models at https://github.com/zsLin177/CopyNE.

2 The CTC-Transformer Model

In this work, we build our proposed approach on the end-to-end CTC-Transformer model, since it is the most widely used and achieves competitive performance in the ASR field (Hori et al., 2017; Kim et al., 2017; Miao et al., 2020; Omachi et al., 2021; Gong et al., 2022). However, it is worth noting that our idea can be applied to other ASR approaches as well.

The CTC-Transformer is built upon the seq-to-seq Transformer (Vaswani et al., 2017), with a connectionist temporal classification (CTC) layer added after the audio encoder. As shown in Figure 2, it takes a sequence of acoustic frames $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_T)$ as input and generates the corresponding transcription text $\boldsymbol{y} = (y_1, ..., y_U)$ as output. The model consists of two main components: an encoder and a decoder. First, the encoder encodes the acoustic frames $\boldsymbol{X}$ into hidden states $\boldsymbol{H} = (\boldsymbol{h}_1, ..., \boldsymbol{h}_T)$. Then, the decoder predicts the target sequence $\boldsymbol{y}$ in an auto-regressive manner. At each decoding step u, the decoder predicts the next target token y_{u+1} based on the encoder's output $\boldsymbol{H}$ and the previously predicted tokens $y_{\leq u} = (y_1, ..., y_u)$. This process is expressed as follows:

$$\boldsymbol{H} = \text{AudioEncoder}(\boldsymbol{X}) \tag{1}$$

$$\boldsymbol{d}_u = \text{Decoder}(y_{\leq u}, \boldsymbol{H}) \tag{2}$$

$$P(y_{u+1}|y_{\leq u}) = \text{softmax}(\boldsymbol{W}\boldsymbol{d}_u + \boldsymbol{b}) \tag{3}$$

Here, $\boldsymbol{d}_u \in \mathbb{R}^d$ denotes the hidden state at step u, and $P(y_{u+1}|y_{\leq u})$ is the posterior distribution of predicting token y_{u+1}. $\boldsymbol{W} \in \mathbb{R}^{|\mathcal{V}|\times d}$ and $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{V}|}$ are learned parameters, where $\mathcal{V}$ is the token vocabulary, and $|\mathcal{V}|$ is the size of the vocabulary.

[1] https://www.openslr.org/38/

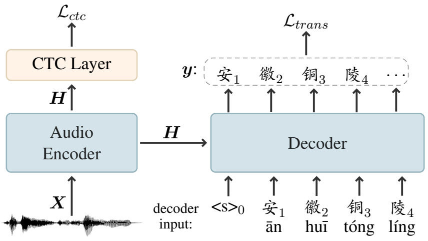

Figure 2: The CTC-Transformer model.

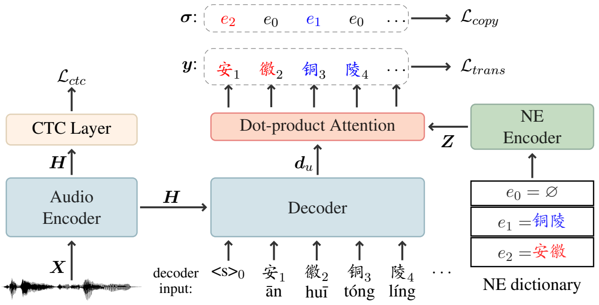

Figure 3: Our CopyNE model.

The loss of Transformer, $\mathcal{L}_{trans}(\boldsymbol{y})$, comes from minimizing the negative log probability of $\boldsymbol{y}$.

$$\mathcal{L}_{trans}(\boldsymbol{y}) = -\sum_{u=0}^{U-1} \log P(y_{u+1}|y_{\leq u}) \tag{4}$$

As commonly used in previous works, the CTC loss is also applied here. CTC aligns each acoustic frame with a token from left to right. For a given target sequence $\boldsymbol{y}$, there may be multiple valid alignments. The CTC loss is derived from maximizing the sum of these valid alignments, and has been proved to be able to enhance the representational capacity of the audio encoder (Kim et al., 2017). Finally, the overall loss is the a weighted sum of the $\mathcal{L}_{trans}(\boldsymbol{y})$ and $\mathcal{L}_{ctc}(\boldsymbol{y})$, as follows:

$$\mathcal{L}(\boldsymbol{y}) = \lambda\mathcal{L}_{trans}(\boldsymbol{y}) + (1-\lambda)\mathcal{L}_{ctc}(\boldsymbol{y}) \tag{5}$$

where λ is a hyper-parameter that determines the relative weight of each loss term.

In inference, the model selects the most probable transcription using beam search as follows:

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}}(\sum_{u} \log P(y_{u+1}|y_{\leq u})) \tag{6}$$

Here, there are many ways to use scores in decoding, such as combining CTC scores and Transformer scores as in training, or using CTC-prefix beam search followed by re-scoring with Transformer to select the optimal result. To compare with most previous works, we use the simplest decoding strategy, as shown in Equation 6.

3 Our CopyNE Model

This section describes our proposed CopyNE model. The basic idea is that the model incorporates a contextual NE dictionary as external knowledge and can choose to directly copy NEs from the dictionary. We design a systematic framework to implement the idea. During training, a copy loss is designed to encourage the model to copy corresponding entities from the dictionary. During inference, at each generation step, the model can either predict a single token from token vocabulary or directly copy a entity from the given dictionary.

3.1 The Model Framework

Figure 3 illustrates the framework of our CopyNE model, which shares the same encoder as the CTC-Transformer model, but with a distinct decoder. In the decoder, we introduce an extra NE encoder that takes the NE dictionary as input and encodes it into NE representations. Then, we use a dot-product attention module to compute copy probabilities based on the obtained NE representations, which are then aggregated to form the overall dictionary (Dict) representation. The decoder can not only utilize copy probabilities to select entities for copying but also leverage the Dict representation to aid in predicting the next token.

NE Representation. We denote an NE dictionary as $E = (e_0, e_1, ..., e_N)$. We use $e_0 = \varnothing$ as a pseudo entity to handle the case where the text to be transcribed has no relation to any entity and the model should not copy any entity at current step.

For each entity e_i, we apply a multi-layer LSTM as the NE encoder to encode the token sequence and use the last hidden state of the NE encoder as the entity representation. It is a popular practice in previous contextual ASR works (Pundak et al., 2018; Zhang and Zhou, 2022).

$$\boldsymbol{z}_i = \text{LSTM}(e_i) \tag{7}$$

After that, we get entity representations $\boldsymbol{Z} = (\boldsymbol{z}_0, \boldsymbol{z}_1, ..., \boldsymbol{z}_N)$, where $\boldsymbol{Z} \in \mathbb{R}^{N \times d}$.

Copy Probability. Once the NE representations are obtained, the copy probability is computed by a dot-product attention mechanism. It is used to determine which entity to copy. First, we compute the attention score $a_u^{e_i}$ for entity e_i at step u as follows:

$$a_u^{e_i} = \frac{(\boldsymbol{W}_q \boldsymbol{d}_u)^\top (\boldsymbol{W}_k \boldsymbol{z}_i)}{\sqrt{d}} \tag{8}$$

where $\boldsymbol{W_q}, \boldsymbol{W_k} \in \mathbb{R}^{d_a \times d}$ are two learned parameters. d_a denotes the dimension of the attention. After that we obtain the attention probability $P_c(e_i|y_{\leq u})$ for entity e_i by softmax.

$$P_c(e_i|y_{\leq u}) = \frac{\exp(a_u^{e_i})}{\sum_{e_j \in E} \exp(a_u^{e_j})} \tag{9}$$

Here, $P_c(e_i|y_{\leq u})$ not only represents the attention probability of e_i but also naturally serves as the copy probability for the entity. During inference, we use the copy probabilities to select the entities for copying.

Dict Representation. With copy (attention) probabilities, we can obtain the Dict representation $\boldsymbol{r}_u$ at decoding step u. It is used to help the prediction of subsequent tokens. Specifically, $\boldsymbol{r}_u \in \mathbb{R}^d$ is computed by weighted summing the entity representations with the copy (attention) probabilities.

$$\boldsymbol{r}_u = \sum_{e_i \in E} P_c(e_i|y_{\leq u}) \boldsymbol{z}_i \tag{10}$$

Dict-enhanced Prediction. Finally, we get the overall Dict representation and copy probabilities. Following Pundak et al. (2018), the Dict representation is applied to help the generation of the next token. So Equation 3 is extended as follows:

$$P(y_{u+1}|y_{\leq u}, E) = \text{softmax}(\boldsymbol{W}\,[\boldsymbol{d}_u, \boldsymbol{r}_u] + \boldsymbol{b}) \tag{11}$$

3.2 Training

During training, to guide the model in selecting correct entities from the NE dictionary for copying, we introduce an additional copy loss $\mathcal{L}_{copy}$. First, based on the ground truth transcription $\boldsymbol{y}$ and the NE dictionary, we construct a copy target σ_{u+1} for each decoding step u, telling the model whether to copy an entity from the dictionary or not, and which one to copy. Then we compute the copy loss $\mathcal{L}_{copy}$ according to the copy target σ_{u+1} and the copy probability $P_c(\sigma_{u+1}|y_{\leq u})$.

The Computation of Copy Loss. Provided that we have an NE dictionary E^b, we construct a copy target, denoted as σ_{u+1}, for decoding step u. In order to build the copy target, we perform maximum matching on the transcription text $\boldsymbol{y}$ from left to right based on the dictionary E^b. If the token span $\boldsymbol{y}_{i,j} = (y_i, ..., y_j)$ matches the k-th entity e_k in E^b, then we set the copy target $\sigma_i = e_k$, and $\sigma_{i+1 \sim j} = \varnothing$. This indicates that the model can copy the k-th entity from the dictionary at decoding step $i-1$, but cannot copy any entity from decoding step i to $j-1$. When it comes to a span of length 1, i.e., $i = j$, during the left-to-right maximum matching process, we also set σ_i to $\varnothing$[2].

For example, in the instance shown in Figure 3, the span “安徽” (An hui) matches the second entity in the dictionary, and the span “铜陵” (Tong ling) matches the first entity in the dictionary. This means that at steps 0 and 2, the model can choose to copy the second and first entities from the dictionary, respectively. Therefore, $\sigma_1 = e_2$ and $\sigma_3 = e_1$, while $\sigma_2 = \varnothing$ and $\sigma_4 = \varnothing$.

After constructing all the copy targets $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_U)$, we can compute the copy loss as follows:

$$\mathcal{L}_{copy}(\boldsymbol{\sigma}) = -\sum_{u=0}^{U-1} \log P_c(\sigma_{u+1}|y_{\leq u}) \tag{12}$$

where $P_c(\sigma_{u+1}|y_{\leq u})$ is the copy probability computed in Equation 9, meaning the probability of copying entity σ_{u+1} at decoding step u. It is worth noting that the copy loss and the bias loss in CBA have fundamental differences. The bias loss in CBA provides information to each token, including tokens within entities, about which entity to attend to. In contrast, our copy loss solely instructs the model to copy the entity from the dictionary at the first token of the entity.

Finally, the loss in our CopyNE model is formed as follows:

$$\mathcal{L} = \lambda \mathcal{L}_{trans}(\boldsymbol{y}) + (1-\lambda)\mathcal{L}_{ctc}(\boldsymbol{y}) + \mathcal{L}_{copy}(\boldsymbol{\sigma}) \tag{13}$$

Dictionary Construction. To construct the copy target and compute the copy loss, we should first build a contextual NE dictionary for training. Provided that the entities have been labeled in the

[2] Please note that in this paper, we primarily focus on entities with a length greater than 1, and therefore only retain such entities in our dictionary.

dataset, we build a NE dictionary E^b for each data batch following previous works.

Firstly, to construct E^b, we extract all entities in the instances of this batch and add them to the dictionary. For instances that do not contain any entities, in order to ensure an adequate number of positive examples, we randomly select one or two substrings of length 2 or 3 from the transcription and include them in the dictionary as pseudo entities. In order to improve the ability of copying the correct entity from a wide range of entities, we also extract additional negative entities from the training set. We analyze the influence of the quantity of negative entities on the model. Due to page constraints, we have included this section in §C.

3.3 Inference

During inference, unlike previous token-level approaches, our model has the flexibility to predict either a token from the vocabulary or an entity from the NE dictionary. By copying the tokens of an entity at once, our CopyNE model can avoid errors that occur when predicting multiple tokens separately. As shown in Figure 3, our CopyNE model can directly copy the two entities “安徽” and “铜陵” from the dictionary.

Specifically, at step u, our prediction is based on both the model's probability for a token v, i.e., $P(v|\hat{y}\leq u, E)$, and the copy probability for an entity e, i.e., $P_c(e|\hat{y}_{\leq u})$. The former represents the probability of predicting a token v from the token vocabulary, while the latter is normalized on all entities, originally indicating the attention probability over entity e, which can be naturally interpreted as the probability of copying entity e from the dictionary. To consider both probabilities on the same scale, we devise an elegant decoding strategy by taking use of the copy probability of $\varnothing$, i.e., $P_c(\varnothing|\hat{y}_{\leq u})$, and re-normalize the probabilities to create an unified searching space Q.

$$Q(i|\hat{y}_{\leq u}) = \begin{cases} P_c(\varnothing|\hat{y}_{\leq u})P(i|\hat{y}_{\leq u}, E), & i \in \mathcal{V} \\ P_c(i|\hat{y}_{\leq u}), & i \in E, i \neq \varnothing \end{cases} \tag{14}$$

Here, to ensure the sum of the probabilities of all elements is 1, we use $P_c(\varnothing|\hat{y}_{\leq u})$ as a prior probability, representing the probability of the text to be transcribed has no relation with the entities in the dictionary and the text should be generated from the token vocabulary. If the element is from the token vocabulary $\mathcal{V}$, we obtain the probability by multiplying the prior probability and the model's probability for the token. Otherwise, we use the copy probability directly.

However, in our experiments, we observe that the model occasionally selects irrelevant entities for copying. To enhance the quality of copying, we introduce a confidence threshold γ during decoding to filter out low-confidence copies. Specifically, we set $P_c(i|\hat{y}_{\leq u}) = 0, i \in E, i \neq \varnothing$, and $P_c(\varnothing|\hat{y}_{\leq u}) = 1$ when $\max\{P_c(i|\hat{y}_{\leq u})|i \in E, i \neq \varnothing\} < \gamma$. This means that if the model's maximum copy probability over real entities is less than γ, it is prevented from copying entities from the dictionary and instead generates tokens from the token vocabulary. In section 4.2, we discuss the influence of the γ in detail.

Finally, we use beam search to select the best element at each step to form the final prediction[3].

$$\hat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}}(\sum_{u} \log Q(i|y_{\leq u})) \tag{15}$$

4 Experiments

4.1 Experimental Setup

Datasets. Experiments on Chinese Mandarin are conducted on two widely used datasets, Aishell (Bu et al., 2017) and ST-cmds[4]. We use the Eng dataset released by Yadav et al. (2020) to perform experiments on English. Furthermore, to compare the performance of different methods in contextual ASR scenarios where speeches contain entities, we extract instances containing entities from the dev and test sets, forming the corresponding “$*$-NE” datasets. Detailed introduction about the datasets can be found in §A.

NE Dictionary. Aishell and ST-cmds were released without entity annotations. In contrast, the Eng dataset was simultaneously released with audio, transcribed text, and corresponding entity annotations. Chen et al. (2022) further annotated entities for Aishell. So, in our experiments with Aishell and Eng, we use the releated entities to build the NE dictionary. For ST-cmds, we use HanLP[5] to get three types of entities: person, location, and organization.

Evaluation Metrics. Character error rate (CER) and word error rate (WER) are used to assess the overall performance of models in Mandarin and

[3] It has to be noted that the partially predicted $\hat{\boldsymbol{y}}$ is still encoded at token-level.

[4] https://www.openslr.org/38/

[5] https://github.com/hankcs/HanLP

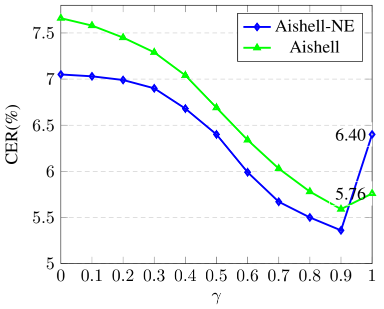

Figure 4: Effect of the Confidence Threshold γ.

English ASR tasks. In this paper, to evaluate the model's entity transcription accuracy, we also employ NE-CER and NE-WER metrics (Han et al., 2021). We align the predicted hypothesis and reference using the minimum edit distance algorithm, and subsequently calculate NE-C(W)ER by measuring the C(W)ER between the entity text in the reference and its counterpart in the hypothesis.

Parameter Setting. The parameter setting in our work is the same as that in most previous works, and the detailed descriptions can be found in §B. To ensure a fair comparison with prior works, we carefully reproduced the CLAS (Pundak et al., 2018) and CBA (Zhang and Zhou, 2022). Moreover, to verify the effectiveness of our approach on pretrained large models, we also conducted experiments on OpenAI Whisper (Radford et al., 2022). Specifically, we use the Whisper model as our transformer encoder and decoder. We choose seeds randomly to run models for 3 times and report the average results.

4.2 Results

Analysis about γ. We first investigate the influence of the copy threshold γ during inference. Figure 4 illustrates how the CER changed on the Aishell dev and Aishell-NE dev with different γ values. Our findings reveal that when the threshold is low, the CER is high, indicating that copying results in more errors when the model copies entities with low confidence. As we increase the γ, the CER decreases, indicating improved reliability of our CopyNE when the model had higher confidence. However, when the threshold becomes too high (above 0.9), the model has fewer opportunities to choose to copy entities, resulting in a higher CER. This happens because it becomes more difficult for the model to trigger the copy mechanism. So, we set γ to 0.9 for all experiments and discussions to enhance the robustness of our model.

Results on Chinese. Table 1 and 2 show the CER and NE-CER of different models on the Chinese dataset. In Table 1, we note that while our primary focus is improving transcription of NEs, we also achieve significant improvements in overall text transcription. Without Whisper, our CopyNE model outperforms the previous CBA approach with a 3.2% relative CER reduction on the Aishell Test and 7.7% on the ST-cmds Test. In contextual ASR scenarios, the improvements are even more pronounced, with a 13.5% relative CER reduction on the Aishell-NE Test and 20.8% on the ST-cmds-NE Test. Even with the powerful Whisper, our CopyNE consistently excels, especially on the ST-cmds dataset, with relative reductions of 8.6% and 15.7% on the two test sets, respectively. Additionally, we observed that CLAS performs well on Aishell, closely matching CopyNE, but its performance on ST-cmds is comparatively weaker, sometimes even worse than the Whisper baseline, a reverse pattern also seen with CBA. In contrast, CopyNE consistently performs well across different datasets, demonstrating its better adaptability.

We also present an improved model, i.e. CopyNE†, which features a more powerful conformer encoder. The results show that CopyNE† can achieve further improvements compared to CopyNE.

In this paper, our main goal is improving the transcription of NEs. From the results presented in Table 2, it is evident that our approach exhibits significant improvements in entity transcription compared to previous methods. When not using Whisper, our CopyNE model achieves an impressive relative NE-CER reduction of 55.4% on the Aishell-NE Test and 53.9% on the ST-cmds-NE Test. Even based on the powerful Whisper model, our CopyNE continues to achieve remarkable improvements, with a relative NE-CER reduction of 25.4% and 26.7% on the two test sets. This demonstrates that copying entities from the dictionary significantly improves the accuracy of transcribing entities.

Results on English. Whisper (Radford et al., 2022) has shown strong performance in English, so we directly use it as our baseline for experiments on English. As seen in Table 3, CopyNE still outperforms other methods, achieving a 5.2% relative WER reduction compared to CLAS in the general scenario on the Eng test dataset. In contextual scenarios, CopyNE demonstrates 6.4% relative WER reductions and 16.8% relative NE-WER reductions.

Model	Aishell		Aishell-NE		ST-cmds		ST-cmds-NE	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Joint CTC-Transformer	6.12	6.70	7.36	9.00	10.63	10.56	13.67	13.63
CLAS (Pundak et al., 2018)	6.04	6.72	7.06	8.73	10.10	10.09	12.64	12.85
CBA (Zhang and Zhou, 2022)	6.11	6.56	6.73	8.00	10.73	10.72	12.69	12.43
CopyNE	5.59	6.35	5.36	6.92	9.76	9.89	9.90	9.84
CopyNE†	**4.49**	**5.03**	**4.31**	**5.05**	**7.78**	**7.80**	**7.71**	**7.40**
Whisper	5.28	5.97	6.32	7.68	9.14	9.06	12.22	12.35
+ CLAS	**4.50**	5.23	**5.30**	6.72	9.10	9.20	12.12	12.25
+ CBA	5.41	6.06	6.13	7.60	7.96	7.87	10.03	9.96
+ CopyNE	4.71	**5.10**	5.40	**6.42**	**7.35**	**7.19**	**8.98**	**8.40**

Table 1: CER on Chinese datasets in general scenarios (Aishell, ST-cmds) and contextual scenarios (Aishell-NE, ST-cmds-NE). † means the model with an improved 12-layer Conformer (Gulati et al., 2020) encoder and averages the parameters of the best 10 epochs when decoding.

Model	Aishell-NE		ST-cmds-NE	
	Dev	Test	Dev	Test
Joint CTC-Transformer	11.64	14.03	21.63	21.41
CLAS (Pundak et al., 2018)	11.24	13.12	19.70	20.10
CBA (Zhang and Zhou, 2022)	7.78	9.44	15.72	15.92
CopyNE	**3.00**	**4.21**	**7.60**	**7.34**
w/o Dict repr $\boldsymbol{r}_u$	4.05	5.27	9.35	9.21
Whisper	10.31	12.24	21.30	21.83
+ CLAS	8.97	11.64	20.82	21.07
+ CBA	9.13	11.79	15.91	15.41
+ CopyNE	**6.74**	**8.79**	**11.93**	**11.29**

Table 2: NE-CER (%) on the Chinese datasets.

Model	Eng$_{.W}$		Eng-NE$_{.W}$		Eng-NE$_{.NW}$	
	Dev	Test	Dev	Test	Dev	Test
Whisper	8.54	8.73	8.53	8.71	28.16	26.61
+ CLAS	7.90	8.28	7.86	8.31	27.23	26.55
+ CBA	9.17	9.52	9.21	9.49	30.01	30.82
+ CopyNE	**7.47**	**7.85**	**7.42**	**7.78**	**23.29**	**22.09**

Table 3: Results on the English datasets. W and NW denote WER and NE-WER respectively.

Dict size	Dev		Test	
	CER	NE-CER	CER	NE-CER
×0.85	5.65	4.27	7.29	5.83
×0.90	5.56	3.90	7.15	5.26
×0.95	5.44	3.37	7.01	4.68
×1	**5.36**	**3.00**	**6.92**	**4.21**
×2	5.61	3.50	7.12	4.82
×3	5.85	3.92	7.39	5.18
×4	6.02	4.09	7.66	5.56

Table 4: The impact of the NE dictionary.

Additionally, we observed that CBA lags behind the Whisper baseline. We suspect that this might be due to its approach of encouraging the model to generate entity tokens by modifying Whisper's output logits, which can disrupt the model's overall probability distribution, especially given Whisper's strong fit on English data. On the contrary, our CopyNE is more stable.

4.3 Impact of the NE dictionary

Following previous works (Pundak et al., 2018; Han et al., 2021), we report the main results using exact NE dictionaries from the test sets. However, when collecting dictionaries in real scenarios, to ensure the coverage, many unrelated noisy NEs are inevitably added to the dictionary. To analyze the impact of noisy entities on CopyNE, we extract entities from the training set that are not included in the test set as noisy NEs. From the corresponding ×2, ×3, and ×4 rows in Table 4, we can see that the introduction of noisy NEs results in a reduction in the model's performance. Nevertheless, even with the addition of 6k noisy NEs, resulting in the dictionary size being quadrupled (×4), CopyNE continues to outperform CLAS and CBA, despite their reliance on the precise dictionary.

In the more rare cases where some NEs are out of the dictionary (OOD), to analyze CopyNE's performance in OOD scenarios, we designate some low-frequency NEs from the original dictionary as OOD NEs. These NEs are removed, and decoding is performed using the remaining NEs. From the relevant rows in Table 4, it can be observed that this primarily impacts NE-CER since CopyNE cannot copy missing NEs. However, even when the OOD proportion reaches 15% (Dict size = ×0.85), CopyNE still shows commendable performance.

4.4 Qualitative Analysis

CopyNE demonstrates significant improvements. To gain further insight into CopyNE's performance, we conduct a qualitative analysis of its generations.

	Transcriptions	Dictionary
English	A company in Yangluo...	阳逻 (Yangluo) 杨丙卿 (Yang Bingqing) 冈山 (Gangshan) 桃太郎体育馆 (Taotailang gym)
Gold	阳逻的一家公司...	
CLAS	扬罗的一家公司...	
CBA	阳罗的一家公司...	
CopyNE	[阳逻]的一家公司...	
English	Yang Bingqing served as manager...	
Gold	杨丙卿担任经理...	
CLAS	杨澄清担任经理...	
CBA	杨炳卿担任经理...	
CopyNE	[杨丙卿]担任经理...	
English	The Taotailang gym in Gangshan.	
Gold	冈山的桃太郎体育馆	
CLAS	冈山的淘汰郎体育馆	
CBA	刚山的淘汰狼体育馆	
CopyNE	[冈山]的[桃太郎体育馆]	

Table 5: Generations of different models. Red text indicates errors, while text enclosed in square brackets represents entities that were copied from the dictionary.

Table 5 shows examples of transcriptions from different ASR models. We can see that in the second example, where CBA successfully identified the correct person entity “杨丙卿” from the dictionary and produced a transcription that is close to gold, it still made a mistake by transcribing “炳” instead of “丙” due to the same pronunciation (bǐng). In contrast, CopyNE can copy all the tokens of the entity from the NE dictioanry. For example, in the third example “冈山的桃太郎体育馆” (The Taotailang gym in Gangshan), CopyNE directly copies the location entity “冈山” (Gangshan) and the organization entity “桃太郎体育馆” (Taotailang gym), achieving a completely correct transcription.

5 Related Works

Contextual ASR. Researchers have explored various approaches to help models in the contextual ASR scenario, with the primary approaches being the utilization of external dictionaries and language models (LMs). CLAS (Pundak et al., 2018) was the first to introduce the use of dictionary to aid in prediction. Alon et al. (2019) extend CLAS by adding phonetically similar alternative terms to the dictionary as negative examples, aiming to improve the model's ability to distinguish entities with similar pronunciations. Huber et al. (2021) propose to utilize the representation of a single entry in the dictionary that is most relevant to the current decoding status. Fu et al. (2023) propose to apply the character-based NE encoder to better capture acoustic features useful for transcribing rare entities. Different from our CopyNE, these methods all treat entities as individual tokens, which may result in incomplete NE transcriptions.

LMs trained on large-scale text data can learn rich linguistic and contextual knowledge, and thus can be used to assist contextual ASR. There are typically two approaches to leverage LMs for contextual ASR. The first approach involves using the dedicated LM to encourage the generation of entity tokens during decoding. (Novak et al., 2012; Aleksic et al., 2015; Zhao et al., 2019a). The second approach is multi-modal pre-training. Researchers have explored joint pre-training of speech and text models, aiming to leverage information from both modalities, and have achieved promising results (Chung et al., 2021; Ao et al., 2022; Zhang et al., 2022). However, compared to using contextual dictionaries, models that rely on LMs tend to have much more parameters, which means that training and deploying require more time and computational resources.

The Copy Mechanism. The copy mechanism can be traced back to the pointer network (Vinyals et al., 2015), which can predict output sequences from the input. The copy mechanism (Gu et al., 2016) extends the pointer network by enabling the model to generate sequences that are not present in the input. According to the source of copying, it can be divided into copying from input text, from document, and from external dictionary.

Copying from Input Text. The copy mechanism is commonly used to copy text from input text. For instance, in text summarization tasks, it is common to employ the copy mechanism to copy keywords from the input text (Cheng and Lapata, 2016; Xu et al., 2020). In grammatical error correction tasks, where only a small portion requires correction, the copy mechanism is used to copy the correct text from the input text (Zhao et al., 2019b).

Copying from Document. In addition to copying from input text, the copy mechanism can be employed to copy text from other texts when the input text is not available. Lan et al. (2023) introduced the copy mechanism in decoder-only language models, where text fragments are selected from a vast amount of documents to generate the target text.

Copying from External Dictionary. In this paper, we introduce a systematic framework, which seamlessly integrates the process of copying from an external dictionary to aid in generation. We believe that our framework can also be applied to other generation tasks.

6 Conclusion

In this paper, we consider entities as indivisible elements and introduce a copy mechanism into ASR for the first time to assist in transcribing entities. We devise a systematic copy framework that can copy all the tokens of an entity from the NE dictionary at once, preserving the token span as a complete entity. Our approach demonstrates substantial improvements on both English and Chinese datasets. In summary, CopyNE represents a significant advancement in contextual ASR, providing a promising direction in this field.

Limitations

From our experiments, we have found that an excessive number of noisy entities can impact the performance. As part of our future work, we intend to explore methods for dynamically filtering out interfering entities from the dictionary during the decoding process.

Acknowledgements

We would like to thank the anonymous reviewers for their valuable comments. This work was supported by National Natural Science Foundation of China (Grant No. 62176173 and 62336006), and a Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

Petar Aleksic, Mohammadreza Ghodsi, Assaf Michaely, Cyril Allauzen, Keith Hall, Brian Roark, David Rybach, and Pedro Moreno. 2015. Bringing contextual information to google speech recognition.

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.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. 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*, pages 5723–5738.

Gilles Boulianne. 2022. Phoneme transcription of endangered languages: an evaluation of recent ASR architectures in the single speaker scenario. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2301–2308.

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.

William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

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.

Zhehuai Chen, Mahaveer Jain, Yongqiang Wang, Michael L Seltzer, and Christian Fuegen. 2019. Joint grapheme and phoneme embeddings for contextual end-to-end asr. In *2019 Conference of the International Speech Communication Association (Interspeech)*, pages 3490–3494.

Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 484–494.

Yu-An Chung, Chenguang Zhu, and Michael Zeng. 2021. SPLAT: Speech-language joint pre-training for spoken language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1897–1907.

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.

Karthik Ganesan, Pakhi Bamdev, Jaivarsan B, Amresh Venugopal, and Abhinav Tushar. 2021. N-best ASR transformer: Enhancing SLU performance using multiple ASR hypotheses. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 93–98.

Zhuo Gong, Daisuke Saito, Sheng Li, Hisashi Kawai, and Nobuaki Minematsu. 2022. Can we train a language model inside an end-to-end ASR model? - investigating effective implicit language modeling.

In *Proceedings of the Second Workshop on When Creative AI Meets Conversational AI*, pages 42–47.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1631–1640.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Minglun Han, Linhao Dong, Shiyu Zhou, and Bo Xu. 2021. Cif-based collaborative decoding for end-to-end contextual speech recognition. In *2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6528–6532.

Takaaki Hori, Shinji Watanabe, and John Hershey. 2017. Joint CTC/attention decoding for end-to-end speech recognition. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 518–529.

Christian Huber, Juan Hussain, Sebastian Stüker, and Alexander Waibel. 2021. Instant one-shot word-learning for context-specific neural sequence-to-sequence speech recognition. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–7. IEEE.

Mahaveer Jain, Gil Keren, Jay Mahadeokar, Geoffrey Zweig, Florian Metze, and Yatharth Saraf. 2020. Contextual rnn-t for open domain asr. *arXiv preprint arXiv:2006.03411*.

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*.

Suyoun Kim, Takaaki Hori, and Shinji Watanabe. 2017. Joint ctc-attention based end-to-end speech recognition using multi-task learning. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4835–4839.

Tian Lan, Deng Cai, Yan Wang, Heyan Huang, and Xian-Ling Mao. 2023. Copy is all you need. In *The Eleventh International Conference on Learning Representations*.

Haoran Miao, Gaofeng Cheng, Changfeng Gao, Pengyuan Zhang, and Yonghong Yan. 2020. Transformer-based online ctc/attention end-to-end speech recognition architecture. In *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6084–6088.

Josef R. Novak, Nobuaki Minematsu, and Keikichi Hirose. 2012. Dynamic grammars with lookahead composition for wfst-based speech recognition. In *Interspeech*.

Motoi Omachi, Yuya Fujita, Shinji Watanabe, and Matthew Wiesner. 2021. End-to-end ASR to jointly predict transcriptions and linguistic annotations. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1861–1871.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

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*.

Kanishka Rao, Haşim Sak, and Rohit Prabhavalkar. 2017. Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 193–199.

Anthony Rousseau, Paul Deléglise, and Yannick Estève. 2012. TED-LIUM: an automatic speech recognition dedicated corpus. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 125–129, Istanbul, Turkey. European Language Resources Association (ELRA).

Kanthashree Mysore Sathyendra, Thejaswi Muniyappa, Feng-Ju Chang, Jing Liu, Jinru Su, Grant P Strimel, Athanasios Mouchtaris, and Siegfried Kunzmann. 2022. Contextual adapters for personalized speech recognition in neural transducers. In *2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8537–8541. IEEE.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, volume 28.

Tongtong Wu, Guitao Wang, Jinming Zhao, Zhaoran Liu, Guilin Qi, Yuan-Fang Li, and Gholamreza Haffari. 2022. Towards relation extraction from speech. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10751–10762.

Song Xu, Haoran Li, Peng Yuan, Youzheng Wu, Xiaodong He, and Bowen Zhou. 2020. Self-attention guided copy mechanism for abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1355–1362.

Hemant Yadav, Sreyan Ghosh, Yi Yu, and Rajiv Ratn Shah. 2020. End-to-end named entity recognition from english speech. In *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, pages 4268–4272. ISCA.

Zhengyi Zhang and Pan Zhou. 2022. End-to-end contextual asr based on posterior distribution adaptation for hybrid ctc/attention system. *arXiv preprint arXiv:2202.09003*.

Ziqiang Zhang, Long Zhou, Junyi Ao, Shujie Liu, Lirong Dai, Jinyu Li, and Furu Wei. 2022. SpeechUT: Bridging speech and text with hidden-unit for encoder-decoder based speech-text pre-training. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1663–1676.

Ding Zhao, Tara N Sainath, David Rybach, Pat Rondon, Deepti Bhatia, Bo Li, and Ruoming Pang. 2019a. Shallow-fusion end-to-end contextual biasing. In *2019 Conference of the International Speech Communication Association (Interspeech)*, pages 1418–1422.

Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, and Jingming Liu. 2019b. Improving grammatical error correction via pre-training a copy-augmented architecture with unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 156–165.

Appendices

A Datasets

Aishell (Bu et al., 2017) and ST-cmds[6] are two widely used Chinese Mandarin datasets. Aishell contains about 150 hours of speech. ST-cmds was built based on commonly used online chatting and user command speeches, which contains about 110 hours of speech. For the English dataset, we utilize the portion of data that has been manually annotated with entities by Yadav et al. (2020), which comprises approximately 150 hours. The Eng dataset is built by extracting content from well-known English datasets, including Librispeech (Panayotov et al., 2015), CommonVoice[7], Tedlium (Rousseau et al., 2012), and Voxforge[8].

Table 6 shows the detailed statistics of the datasets used in our experiments. “Sent” means the number of instances. “NE” is the number of different named entities in the dataset and also the size of the contextual entity dictionary used during inference.

Dataset	Train		Dev		Test	
	Sent	NE	Sent	NE	Sent	NE
Aishell	119919	14241	14326	2194	7176	1186
Aishell-NE	119919	14241	4949	2194	2244	1186
ST-cmds	82080	17376	10260	3029	10260	3124
ST-cmds-NE	82080	17376	3285	3029	3241	3124
Eng	64570	11858	3100	2568	3100	2508
Eng-NE	64570	11858	2677	2568	2690	2508

Table 6: Statistics of the used datasets.

B Parameter Settings

We use 80-dimensional log-mel acoustic features with 25ms frame window and 10ms frame shift. The log-mel features are first fed into a 2D convolutional layer for downsampling and mapped to 256 dimensions before being inputted into the Audio Encoder. Both the audio encoder and decoder consist of 6 Transformer layers with 4 attention heads each. The NE Encoder is composed of three LSTM layers, with the input being a randomly initialized 256-dimensional embedding vector and the hidden size being 512. The relative weight λ in Equation 13 is set to 0.7. The experiments are conducted on two NVIDIA A100 GPUs.

In addition, for the experiments on Whisper, we use Whisper-small model[9], which includes 12 transformer layers in both its encoder and decoder, and is pre-trained on a total of 680,000 hours of multi-lingual and multi-task data. We replace our audio encoder and decoder with the Whisper model and fine-tune the parameters of the entire model on our training set for a maximum of 20 epochs. During fine-tuning, the initial learning rate for the Whisper model's parameters is set to 1e-5, while the learning rate for other parameters is set to 1e-3 with 10,000 warm-up steps. During inference, we used beam search with a beam size of 5 and 10 for models with and without Whisper, respectively.

[6] https://www.openslr.org/38/

[7] https://en.wikipedia.org/wiki/Common_Voice

[8] https://en.wikipedia.org/wiki/VoxForge

[9] https://huggingface.co/openai/whisper-small

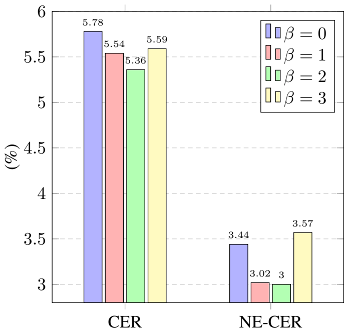

Figure 5: Effect of β.

C Influence of Negative Entities in Training

During training, we construct an NE dictionary for each batch. To enhance the model's ability of copying correct entities, we sample additional negative examples.

Suppose the dictionary already contains m entities, either real entities or pseudo sub-string entities. We sample $\beta \cdot m$ entities as negative examples from the training set. We utilize the parameter β to control the number of negative examples. Thus, we get the final dictionary for this batch which contains a total of $(\beta + 1) \cdot m$ entities. As shown in Figure 5, adding 1 or 2 times the number of negative samples can reduce transcription errors. Specifically, when $\beta = 2$, the CER and NE-CER decreased by 0.42% and 0.44% compared to $\beta = 0$. However, as β continues to increase, the error rate started to rise. We think that this is due to the presence of excessive noise. This causes the model to excessively focus on the negative samples, thus affecting its ability to accurately copy entities. Therefore, we set β to 2 during training.