DeRAGEC: Denoising Named Entity Candidates with Synthetic Rationale for ASR Error Correction

Solee Im[*1], **Wonjun Lee**[*2], **Jinmyeong An**[1,3],
Yunsu Kim[4], **Jungseul Ok**[1,2], **Gary Geunbae Lee**[1,2]
[1]Graduate School of Artificial Intelligence, POSTECH, Republic of Korea
[2]Department of Computer Science and Engineering, POSTECH, Republic of Korea
[3]Mobile eXperience Business, Samsung Electronics, Republic of Korea
[4]aiXplain Inc., Los Gatos, CA, USA
{solee0022, lee1jun, jungseul.ok, gblee}@postech.ac.kr, jinmyeong.an@samsung.com, yunsu.kim@aixplain.com

Abstract

We present DeRAGEC, a method for improving Named Entity (NE) correction in Automatic Speech Recognition (ASR) systems. By extending the Retrieval-Augmented Generative Error Correction (RAGEC) framework, DeRAGEC employs synthetic denoising rationales to filter out noisy NE candidates before correction. By leveraging phonetic similarity and augmented definitions, it refines noisy retrieved NEs using in-context learning, requiring no additional training. Experimental results on CommonVoice and STOP datasets show significant improvements in Word Error Rate (WER) and NE hit ratio, outperforming baseline ASR and RAGEC methods. Specifically, we achieved a 28% relative reduction in WER compared to ASR without postprocessing. Our source code is publicly available at: https://github.com/solee0022/deragec.

1 Introduction

Recent studies (Ma et al., 2023; Li et al., 2024; Chen et al., 2023) have demonstrated that post-processing speech recognition transcriptions with Large Language Models (LLMs) can significantly enhance the accuracy of Automatic Speech Recognition (ASR). In particular, Generative Error Correction (GEC) has been proposed to refine ASR outputs (Yang et al., 2023; Chen et al., 2023). Moreover, recent work has advanced GEC into a multi-modal paradigm (Radhakrishnan et al., 2023; Hu et al., 2024b) that jointly conditions on both the textual transcript and the corresponding audio to further improve correction quality. Under the GEC framework, an ASR model first processes the user's speech and outputs multiple hypotheses (top-k) through beam search decoding. An LLM then refines these hypotheses, either by re-ranking words across different hypothesis combinations or by predicting contextually appropriate replacements. This approach leverages the extensive linguistic knowledge and powerful generation capabilities of pre-trained LLMs, which are trained on large-scale text corpora (Brown et al., 2020).

Despite its effectiveness in improving ASR accuracy, GEC faces a critical limitation: its inability to reliably introduce words absent from the initial hypotheses, particularly Named Entities (NEs) (Wang et al., 2024; Pusateri et al., 2024). This challenge stems from inherent biases in LLMs, which, due to their exposure to large, diverse corpora, tend to favor high-frequency words and expressions (Gong et al., 2024; Gallegos et al., 2024). As a result, rare or out-of-vocabulary NEs—especially those missing from the ASR hypotheses—are difficult to recover with high accuracy (Ghosh et al., 2024; Pusateri et al., 2024). In response, retrieval-augmented methods (Wang et al., 2024; Ghosh et al., 2024) have recently gained traction as a way to incorporate external knowledge for more accurate NE correction. A notable example is Retrieval-Augmented Generative Error Correction (RAGEC), which augments GEC by retrieving relevant NEs from an external knowledge base and integrating them into the LLM's input context (Pusateri et al., 2024). Nonetheless, the presence of irrelevant or weakly related NEs often introduces noise, thereby undermining correction performance (Wang et al., 2024; Ghosh et al., 2024). Current solutions typically handle this noise implicitly, relying heavily on the LLM's internal GEC capabilities.

In this work, we propose an explicit method for denoising NE candidates, which filters out irrelevant NEs before the GEC process. Our approach combines phonetic score, augmented NE definitions, and synthetic rationales within an in-context learning (ICL) framework. By directly reducing noise in the retrieved NE candidates—without the need for additional training—we minimize the system's reliance on both ASR outputs and LLM inference. We tested our method on the CommonVoice

[*]Equally contributed

Findings of the Association for Computational Linguistics: ACL 2025, pages 15181–15193
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

and STOP speech datasets, achieving a 28% relative reduction in Word Error Rate (WER).

2 Related Works

2.1 Generative Error Correction in ASR

GEC (Chen et al., 2023; Yang et al., 2023; Ma et al., 2023) has become an effective post-processing method for ASR systems. However, GEC models struggle with novel or domain-specific NEs (Wang et al., 2024; Pusateri et al., 2024). While Retrieval-Augmented GEC (RAGEC) (Lei et al., 2024; Pusateri et al., 2024) improves NE correction by retrieving phonetically similar NE candidates, it faces challenges in determining the optimal number of retrieved NEs, leading to phonetic confusion (Wang et al., 2024) or insufficient NEs. To address this, we propose a novel, training-free RAGEC system with an explicit denoising mechanism that generates rationales to filter and select relevant NE from the retrieved list.

2.2 Retrieved Candidates Filtering

RAG (Lewis et al., 2020; Izacard et al., 2023; Guu et al., 2020) improves language models (LMs) accuracy in knowledge-intensive tasks but often retrieves noisy data (Li et al., 2022; Yoran et al., 2023). In standard RAG systems, including the RAGEC task (Pusateri et al., 2024; Ghosh et al., 2024), noise removal in retrieved data (e.g., NEs lists) is typically handled implicitly by training LMs to predict the correct answer (e.g., transcription) despite potentially noisy inputs. However, this approach is vulnerable to high noise ratios, lacks transparency, and heavily depends on the LMs (Cuconasu et al., 2024; Wu et al., 2024). Recent studies (Wei et al., 2024) introduce explicit denoising by generating rationales to filter noise without additional training. However, denoising retrieved phonetic information (e.g., NE lists) in RAGEC remains underexplored. To address this gap, we propose DeRAGEC, a training-free, rationale-driven approach that refines retrieved NE lists by explicitly denoising phonetic candidate information.

3 DeRAGEC

3.1 Preliminary

We begin with a pre-trained ASR model that generates a 5-best hypothesis set $H = \{h_1, h_2, h_3, h_4, h_5\}$ using beam search. From the top hypothesis h_1, a Named Entity Recognition (NER) model extracts NE candidates, which serve as a phonetic query q_p for retrieving the top-k phonetically similar NE candidates $N = \{n_1, \dots, n_k\}$ from an external NE database.

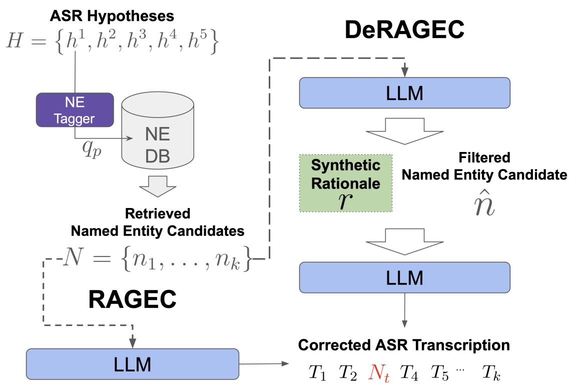

Figure 1: Comparison of RAGEC and DeRAGEC in handling retrieved NE candidates. For clarity, input features such as PS and Def are omitted to highlight the key differences between the two methods.

In the **ASR baseline**, the top hypothesis is directly used as the transcription $\hat{a} = h_1$ (Eq.1).

In the **GEC framework**, a language model $\mathcal{M}_\theta$ post-processes the hypothesis set H using T^{fs} number of few-shot examples $\mathcal{E}^{gec} = \{(H_i, a_i)\}$ sampled from dataset $\mathcal{D}$ (Eq. 2). Where $\mathcal{D} = \{(H_i, a_i, N_i), 1 \leq i \leq T\}$ contains T triplets of hypotheses (H), ground-truth transcription (a) and retrieved NE candidates (N).

The **RAGEC framework** enhances this by incorporating the retrieved NE candidates N during generation, with demonstrations $\mathcal{E}^{ragec} = \{(H_i, a_i, N_i), 1 \leq i \leq T^{fs}\} \in \mathcal{D}$ (Eq. 3) where T^{fs} is number of ICL examples.

$$\textbf{ASR} : \hat{a} = h_1 \tag{1}$$
$$\textbf{GEC} : \hat{a} = \mathcal{M}_\theta(a | H, \mathcal{E}^{gec}) \tag{2}$$
$$\textbf{RAGEC} : \hat{a} = \mathcal{M}_\theta(a | H, N, \mathcal{E}^{ragec}) \tag{3}$$

3.2 DeRAGEC Framework

DeRAGEC extends RAGEC with an explicit, training-free denoising gate. For each retrieved named-entity (NE) candidate we append (i) a phonetic similarity score (PS) and (ii) a one-line Wikipedia definition (Def) to every retrieved NE candidate. The denoising gate prunes irrelevant candidates, returning the selected entity $\hat{n}$ together with its rationale r. We then pass the tuple $(H, \hat{n}, r)$ to the GEC model, which outputs the final transcript $\hat{a}$. Figure 1 highlights the DeRAGEC extensions to the original RAGEC.

Phonetic & Semantic Enrichment: For every candidate $n_i \in N$ we attach its phonetic-similarity score $PS_i = \mathrm{sim}(q_p, n_i)$ and a one-sentence entity definition Def_i. The resulting triple is serialised as:
< n_i | `phonetic-score:`PS_i | `def:` Def_i >
and used for denoising gate.

Synthetic Rationale Generation: Inspired by prior work (Wei et al., 2024), we employ a separate rationale generation model $\mathcal{M}_r$ to synthesize denoising rationales r^{syn} that guide the selection of relevant NE candidates. As illustrated in Algorithm 1, for each training triplet $(h_1, a, N) \in \mathcal{D}$, along with PS and Def, the model generates a MCQ-style rationale r^{syn}, inspired by (Hu et al., 2024a), which explains which NE candidates contribute meaningfully to recovering the ground truth a. The NE list N is augmented by concatenating it with the NEs extracted from H, ensuring completeness. Consequently, this allows us to effectively augment the current dataset $\mathcal{D} \to \mathcal{D}^+ = \{(H_i, a_i, N_i, PS_i, Def_i, r_i), 1 \le i \le T\}$.

Learning Denoising Rationales: DeRAGEC uses ICL to learn from these synthetic rationales without further training. Given a instance (h_1, N, PS, Def), the model samples T^{fs} rationale-augmented examples $\mathcal{E}^{deragec} \in D^+$ and generates:

1. A filtered NE $\hat{n}$ and rationale r describing the reasoning process for selecting $\hat{n}$
2. The final corrected transcription $\hat{a}$, conditioned on H, $\hat{n}$, and r

DeRAGEC :

$$\begin{aligned} \hat{n}, r &\leftarrow \mathcal{M}_\theta(n | (h^1, N, PS, Def), \mathcal{E}^{deragec}) \\ \hat{a} &= \mathcal{M}_\theta(a | H, \hat{n}, r) \end{aligned} \tag{4}$$

For detailed prompts related to the rationale synthesis and GEC processes, please refer to Tables 6–9 in the Appendix G.

4 Experimental Result

4.1 Experimental Setting

For our evaluation, we use a subset of the CommonVoice (CV) dataset (Ardila et al., 2019), which includes 2,000 samples from (Chen et al., 2023), and a sub-sampled STOP test set (Tomasello et al., 2023) containing 5,000 samples. These datasets were chosen to evaluate the effectiveness of our method on both free-form speech (CV) and speech commands for spoken language understanding (STOP). We retrieved the top-10 phonetically similar NEs for RAGEC and DeRAGEC. Additionally, up to five ($T^{fs} = 5$) randomly selected samples from the CV and STOP training sets are used as ICL few-shot examples ($\mathcal{E}$).

Algorithm 1 Synthesize Rationales

Input: Phonetic Retrieval $\mathcal{R}$, Rationale Generator $\mathcal{M}_r$, Data $\mathcal{D}$, phonetic similarity set PS and textual definitions Def for NE candidates N.

```
/*Synthesized Rationales*/
for each {(H, a, N)} ∈ D do
    Extract named entities (NE) from H:
    N^hyp = {n_1^hyp, ..., n_5^hyp}
    Concatenate N^hyp and N:
    N ← N^hyp ⊕ N
    Synthesize denoising rationales:
    r^syn ← M_r(h_1, a, N, PS, Def)
end for
```

We employ Whisper-large-v3-turbo (0.8B) (Radford et al., 2022) with a beam search size of 5 as our **ASR** model. For **GEC**, we use Llama-3.1 (70B) (MetaAI, 2024), and GPT-4o-mini (`gpt-4o-mini-2024-07-18`) (OpenAI, 2023). Additionally, we use o1 (`o1-2024-12-17`) (Jaech et al., 2024) for rationale synthesis. Epitran (Mortensen et al., 2018) and Panphon (Mortensen et al., 2016) are utilized for phonemizing named entities (NEs) and computing phonetic similarity (PS) based on articulatory features. Specifically, phonetic similarity is measured between the tagged NE from the hypothesis (denoted as phonetic query q_p) and NEs retrieved from NE database ($N = \{n_1, \ldots, n_k\}$).

A total of 3,003,462 NEs are collected from CV training set, an open-source media entity dataset (Van Gysel et al., 2022), and Wikipedia to build NE database. For NE collection and GEC process, we use GLiNER-large-v2[1] model as the NE tagger.

4.2 DeRAGEC and Baselines

To demonstrate the effect of each feature, we conducted ablation experiments on the features MCQ format prompt for NE filtering, PS, Def, and rationale (Rat), as shown in Table 1 while NE candidates (N) is used for both RAGEC and DeRAGEC.

[1] `https://huggingface.co/urchade/gliner_large-v2`

	Denoising feature					WER (%) ↓ Llama-3.1 (70B)		gpt-4o-mini		NE Hit Ratio ↑ Llama-3.1 (70B)		gpt-4o-mini	
Model	N	MCQ	PS	Def	Rat	CV	STOP	CV	STOP	CV	STOP	CV	STOP
ASR only	-	-	-	-	-	7.7	8.9	7.7	8.9	0.751	0.787	0.751	0.787
GEC	-	-	-	-	-	6.8	7.8	6.9	7.4	0.782	0.805	0.784	0.804
RAGEC (+ Denoising)	✓	-	-	-	-	6.5	6.5	7.1	6.6	0.804	0.807	0.788	0.814
	✓	✓	-	-	-	6.5	6.5	7.1	6.5	0.804	0.807	0.788	0.816
	✓	✓	✓	-	-	6.6	6.0	7.1	6.0	0.796	0.828	0.785	0.827
	✓	✓	-	✓	-	6.5	7.2	7.0	6.9	0.807	0.697	0.795	0.752
	✓	✓	✓	✓	-	6.5	6.2	6.7	6.0	0.807	0.815	0.802	0.827
DeRAGEC (ours)	✓	✓	✓	✓	✓	**6.0**	**5.9**	**6.2**	**5.8**	**0.831**	**0.838**	**0.813**	**0.842**
ORACLE	A ground truth of NE					5.8	5.7	6.0	5.7	0.837	0.857	0.828	0.847

Table 1: Comparison of WER (%) and NE Hit Ratio on the CV and STOP datasets across different model and denoising settings. For RAGEC, using only N indicates that no additional denoising is applied, while other settings incorporate specific features to denoise NE candidates.

Comparing the results with our baselines, including pure ASR, GEC, and RAGEC, DeRAGEC consistently achieves the best performance in both WER and NE hit ratio. The NE hit ratio is a metric that measures the effectiveness of the methods in correcting NEs. It is calculated by dividing the number of correctly identified NEs by the total number of NEs in the final transcription. A higher NE hit ratio indicates better performance in NE correction.

From Table 1, we observe that DeRAGEC outperforms all other methods by a significant margin, achieving an average relative WER reduction of 28% over the pure ASR baseline and 5.9% over the best RAGEC configuration. Even when compared to the ORACLE setting, where ground truth NEs are provided to the GEC process, DeRAGEC shows a small gap in both WER and NE hit ratio, demonstrating its effectiveness even comparing with the upper bound setting.

4.3 Effect of Denoising

To evaluate the impact of denoising, we considered additional metrics beyond WER to assess the filtering effectiveness. Figure 2 demonstrates that our method maintains the candidate hit ratio (recall) on denoising step while improving precision, suggesting that the denoising step successfully selects the correct NE and eliminates unrelated candidates. The precision upper bound is 0.166 ($1/6$), assuming the correct NE is always included in one of the selected candidates n, along with the 5 existing NEs in ASR hypotheses. Although our method does not reach this upper bound, it achieves the highest precision of 0.139 compared to other baselines. Additionally, with the recall upper bound set at 0.841 in N only setting which dose not applied any denoising, our method maintains a recall of 0.839, demonstrating only a small gap.

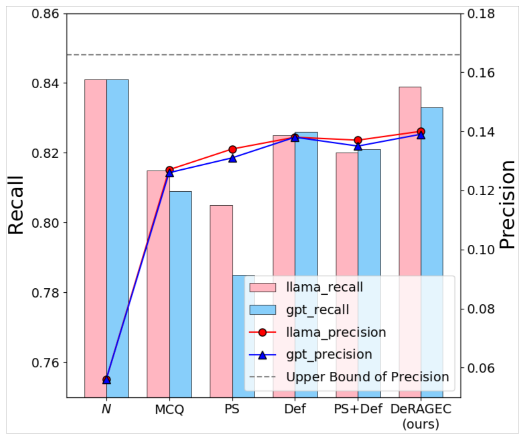

Figure 2: Recall and Precision of NE hit ratio of candidate denoising process. Different filtering methods are evaluated on CV dataset.

5 Conclusion

DeRAGEC is a novel, training-free approach that enhances Named Entity (NE) correction in ASR systems by filtering noisy NE candidates using phonetic similarity, augmented definitions, and synthetic rationales. Experiments on CommonVoice and STOP datasets show that DeRAGEC outperforms baseline ASR and RAGEC models, achieving a 28% relative reduction in WER and improved NE hit ratios. These results highlight DeRAGEC's effectiveness in NE correction, even compared to an ORACLE setting, with potential for broader ASR applications.

Limitation

Our study explored denoising retrieved named entities (NEs) using additional knowledge (phonetic and textual knowledge) and rationales within the RAGEC task, an area where denoising remains underexplored. However, our approach has some limitations. It may be necessary to investigate training methods that leverage our synthetic rationales. While this process can be costly and dependent on ASR or post-processing models, further research is needed to assess how well the model can internalize and apply the denoising instructions provided during training. In addition, it might be necessary to explore whether our approach generalizes across a wider range of ASR and post-processing models, ensuring its adaptability for broader applications. Moving forward, we aim to explore alternative denoising approaches and enhance the robustness of denoising techniques to further improve ASR post-processing effectiveness, as mentioned above.

Acknowledgments

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.RS-2019-II191906, Artificial Intelligence Graduate School Program(POSTECH)) (5%) and was supported by the IITP(Institute of Information & Coummunications Technology Planning & Evaluation)-ITRC(Information Technology Research Center) grant funded by the Korea government(Ministry of Science and ICT)(IITP-2025-RS-2024-00437866) (47.5%) and was supported by Smart HealthCare Program funded by the Korean National Police Agency(KNPA) (No. RS-2022-PT000186) (47.5%)

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng-Siong Chng. 2023. Hyporadise: An open baseline for generative speech recognition with large language models. *Advances in Neural Information Processing Systems*, 36:31665–31688.

Florin Cuconasu, Giovanni Trappolini, Federico Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 719–729.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. 2024. Bias and fairness in large language models: A survey. *Computational Linguistics*, pages 1–79.

Sreyan Ghosh, Mohammad Sadegh Rasooli, Michael Levit, Peidong Wang, Jian Xue, Dinesh Manocha, and Jinyu Li. 2024. Failing forward: Improving generative error correction for asr with synthetic data and retrieval augmentation. *arXiv preprint arXiv:2410.13198*.

Xun Gong, Anqi Lv, Zhiming Wang, and Yanmin Qian. 2024. Contextual biasing speech recognition in speech-enhanced large language model. *Proc. Interspeech. ISCA*, pages 257–261.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Yuchen Hu, Chen Chen, Chengwei Qin, Qiushi Zhu, Eng Siong Chng, and Ruizhe Li. 2024a. Listen again and choose the right answer: A new paradigm for automatic speech recognition with large language models. *arXiv preprint arXiv:2405.10025*.

Yuchen Hu, CHEN CHEN, Chao-Han Huck Yang, Ruizhe Li, Chao Zhang, Pin-Yu Chen, and EngSiong Chng. 2024b. Large language models are efficient learners of noise-robust speech recognition. In *The Twelfth International Conference on Learning Representations*.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Zhihong Lei, Xingyu Na, Mingbin Xu, Ernest Pusateri, Christophe Van Gysel, Yuanyuan Zhang, Shiyi Han, and Zhen Huang. 2024. Contextualization of asr with llm using phonetic retrieval-based augmentation. *arXiv preprint arXiv:2409.15353*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Daliang Li, Ankit Singh Rawat, Manzil Zaheer, Xin Wang, Michal Lukasik, Andreas Veit, Felix Yu, and Sanjiv Kumar. 2022. Large language models with controllable working memory. *arXiv preprint arXiv:2211.05110*.

Sheng Li, Chen Chen, Chin Yuen Kwok, Chenhui Chu, Eng Siong Chng, and Hisashi Kawai. 2024. Investigating asr error correction with large language model and multilingual 1-best hypotheses. In *Proc. Interspeech*, pages 1315–1319.

Rao Ma, Mengjie Qian, Potsawee Manakul, Mark Gales, and Kate Knill. 2023. Can generative large language models perform asr error correction? *arXiv preprint arXiv:2307.04172*.

MetaAI. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

David R. Mortensen, Siddharth Dalmia, and Patrick Littell. 2018. Epitran: Precision G2P for many languages. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

David R. Mortensen, Patrick Littell, Akash Bharadwaj, Kartik Goyal, Chris Dyer, and Lori Levin. 2016. PanPhon: A resource for mapping IPA segments to articulatory feature vectors. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3475–3484, Osaka, Japan. The COLING 2016 Organizing Committee.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Ernest Pusateri, Anmol Walia, Anirudh Kashi, Bortik Bandyopadhyay, Nadia Hyder, Sayantan Mahinder, Raviteja Anantha, Daben Liu, and Sashank Gondala. 2024. Retrieval augmented correction of named entity speech recognition errors. *arXiv preprint arXiv:2409.06062*.

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*.

Srijith Radhakrishnan, Chao-Han Yang, Sumeer Khan, Rohit Kumar, Narsis Kiani, David Gomez-Cabrero, and Jesper Tegnér. 2023. Whispering llama: A cross-modal generative error correction framework for speech recognition. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10007–10016.

Paden Tomasello, Akshat Shrivastava, Daniel Lazar, Po-Chun Hsu, Duc Le, Adithya Sagar, Ali Elkahky, Jade Copet, Wei-Ning Hsu, Yossi Adi, et al. 2023. Stop: A dataset for spoken task oriented semantic parsing. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pages 991–998. IEEE.

Christophe Van Gysel, Mirko Hannemann, Ernest Pusateri, Youssef Oualil, and Ilya Oparin. 2022. Space-efficient representation of entity-centric query language models. In *Proc. Interspeech 2022*, pages 679–683.

Yi-Cheng Wang, Hsin-Wei Wang, Bi-Cheng Yan, Chi-Han Lin, and Berlin Chen. 2024. Dancer: Entity description augmented named entity corrector for automatic speech recognition. *arXiv preprint arXiv:2403.17645*.

Zhepei Wei, Wei-Lin Chen, and Yu Meng. 2024. Instructrag: Instructing retrieval-augmented generation with explicit denoising. *arXiv preprint arXiv:2406.13629*.

Kevin Wu, Eric Wu, and James Zou. 2024. How faithful are rag models? quantifying the tug-of-war between rag and llms' internal prior. *arXiv e-prints*, pages arXiv–2404.

Chao-Han Huck Yang, Yile Gu, Yi-Chieh Liu, Shalini Ghosh, Ivan Bulyko, and Andreas Stolcke. 2023. Generative speech recognition error correction with large language models and task-activating prompting. In *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 1–8. IEEE.

Ori Yoran, Tomer Wolfson, Ori Ram, and Jonathan Berant. 2023. Making retrieval-augmented language models robust to irrelevant context. *arXiv preprint arXiv:2310.01558*.

Appendix

A Effects of static threshold filtering.

Table 2 and 3 present three existing methods to reduce the candidate set: reducing the K value (number of retrieved NEs, $N = \{n_1, \ldots, n_k\}$) during retrieval, setting a static distance threshold (θ_d), and applying a statistical threshold based on the standard deviation (STD) of the retrieved candidates. However, the static threshold methods, including K and θ_d, have a limitation: even correct candidates may be excluded, leading to a decrease in recall.

Instead of using a fixed constant threshold (K, θ_d) for all cases, we also conducted an experiment using STD to set a more sample-specific threshold. The statistical filtering works as follows: when the distance STD of the candidates is large (indicating a skewed distribution), fewer candidates remain after filtering, specifically those within $\sigma * STD$ of the mean. Conversely, when the STD is small (indicating a more uniform distribution), more candidates are retained. However, the statistical threshold approach did not perform well with low recall.

B ORACLE comparison of different retrieval settings.

In Table 4, we demonstrate the denoising effect under both real retrieval situations and an upper-bound retrieval scenario, where the correct answer NE is unconditionally retrieved. For this experiment, we used the CommonVoice dataset and Llama-v3.1. In this setup, ORACLE-retr represents a scenario where the actually retrieved NE (with recall@10 of 0.841) is fed to the GEC, with noisy NEs removed. ORACLE represents a scenario where only the correct answer NE is provided to the GEC, while ORACLE+noise includes phonetically similar noisy NEs. This setup allowed us to observe the performance degradation caused by NE noise on the GEC. These results suggest that applying our methodology to various retrieval scenarios can lead to performance improvements.

C Performance of DeRAGEC w/o MCQ.

We attempted to concatenate all processes, reasoning, selecting the NE, and performing error correction. However, as you can see in Table 5, DeRAGEC w/o MCQ show performance degradation. When DeRAGEC was performed as a single step, the generated rationale averaged 583 words, which was too long and made the process overly complicated. In contrast, the MCQ step generated an average of 211 words of rationale and performed better. Consequently, our experiments demonstrate that separating the steps for NE selection and error correction outperforms combining them into a single inference step.

D Case study

Figure 3 and 4 compare the generated answer in the MCQ step of RAGEC and DeRAGEC. RAGEC generates only the NE, while DeRAGEC first generates a rationale to select the NE and then generates the NE itself. This study demonstrates that our model effectively denoises irrelevant NEs using PS, Def and its reasoning skills, which involve identifying the relevant information, removing irrelevant NEs, and selecting the best candidate NE.

E Effect of Number of few-shots

Figure 5 shows that GEC and RAGEC has no improvement with an increasing number of few-shots. In contrast, DeRAGEC exhibits enhanced performance in generating rationales and selecting the correct NE as the number of few-shots grows. These results demonstrate that DeRAGEC effectively leverages the additional fewshots to refine its reasoning process and improve task-specific outcomes.

F NER performance on ASR hypothesis.

Figure 6 shows the NER performance on the ASR 1-best hypothesis, using the NER results from the Ground-Truth transcription as pseudo-labels. The NER performance decreases as the ASR hypothesis WER increases, highlighting the bottleneck in extracting phonetic queries from h_1 and retrieving potential NE candidates.

G Prompt templates

We modularize the framework to streamline the overall process and provide a dedicated prompt template for each module. Module 1 performs Synthetic Rationale Generation, producing intermediate explanations that serve as few-shot examples to guide the model's reasoning. Module 2 performs NE filtering by reformulating the input into a MCQ format. Module 3 conducts GEC. The corresponding prompt templates for Modules 1, 2, and 3 are detailed in Tables 6, 7, 8, and 9, respectively.

Cloze sentence: The party is led by [Blank]
Options:
A: Samraj Singh (1.00 | Maharaja (also spelled Maharajah or Maharaj; lit. 'great ruler'; feminine: Maharani) is a South Asian royal title of Sanskrit origin.)
B: Amrit Singh (0.70 | Amrit Singh may refer to: Amrit Singh (cyclist) (born 1991), Indian track cyclist Amrit Singh, daughter of former Indian Prime)
C: Samurai Gun (0.64 | Samurai Gun (Japanese: サムライガン, Hepburn: Samurai Gan) is a Japanese manga series written and illustrated by Kazuhiro Kumagai.)
D: Samresh Singh (0.64 | Samresh Singh (1940/1941 – 1 December 2022) was an Indian politician from the state of Jharkhand.)
E: Yuvraj Singh (0.64 | Yuvraj Singh (born 12 December 1981) is a former Indian international cricketer who played in all formats of the game.)
F: samurai prince (0.62 | Samurai Jack is an American science fantasy action-adventure animated television series created by Genndy Tartakovsky for Cartoon Network and Adult)
G: Cambyses (0.60 | Cambyses II (Old Persian: romanized: Ka^{m}būjiya) was the second King of Kings of the Achaemenid Empire from 530)
H: Sam Rice (0.60 | Edgar Charles "Sam" Rice (February 20, 1890 – October 13, 1974) was an American pitcher and outfielder in Major League)
I: Preissing (0.60 | Thomas Joseph Preissing (born December 3, 1978) is an American former professional ice hockey player who played six seasons)
J: Samboy Lim (0.60 | Avelino "Samboy" Borromeo Lim Jr. (April 1, 1962 – December 23, 2023), nicknamed "The Skywalker", was a Filipino professional basketball)
K: Sandeep Singh (0.60 | Sandeep Singh (born 27 February 1986) is an Indian professional field hockey player from Haryana and an ex-captain)

RAGEC

Answer: **A: samraj singh** ❌

DeRAGEC

<think>
The sentence is "The party is led by [BLANK]." We need to identify the most suitable named entity to fill the blank.
• Option A (Samraj Singh) has the highest phonetic similarity score of 1.00. However, the definition provided does not directly relate to a person but rather to a royal title.
• Options B (Amrit Singh), C (Samurai Gun), D (Samresh Singh), and E (Yuvraj Singh) all have lower phonetic similarity scores but are actual names of individuals, which could fit the context of leading a party.
• Among these, D (Samresh Singh) stands out because the definition mentions that Samresh Singh was a politician from the state of Jharkhand, indicating that he could be associated with leading a political party.
Hence, D: Samresh Singh is the most appropriate choice given the context of the sentence and the definitions provided. </think>
<answer> **D: Samresh Singh** </answer> ✅

Figure 3: Case study between REGEC and DeRAGEC in CommonVoice dataset. The process of denoising irrelevant NE candidates is highlighted in pink, while the selection of the correct NE is in blue.

Cloze sentence: How many miles from [Blank] to brinkley arkansas
Options:
A: kentucky tennessee (1.00 | The Kentucky–Tennessee football rivalry is an American college football rivalry between the Kentucky Wildcats and Tennessee Volunteers.)
B: kenton tennessee (0.91 | Kenton is a town in Gibson and Obion counties, Tennessee.)
C: Memphis Tennessee (0.69 | Memphis is a city in the U.S. state of Tennessee. It is the county seat of Shelby County)
D: Phanton menace (0.69 | I Want to Die in New Orleans is the debut studio album by American hip hop duo Suicideboys)
E: Captain Kennedy (0.69 | Captain Kennedy may refer to: Archibald Kennedy, 11th Earl of Cassilis (bef. 1736–died 1794), Scottish peer who lived in the)
F: Austin Hennessey (0.69 | Hennessey Performance Engineering (HPE) is an American hypercar manufacturer and high-performance vehicle creator. In addition to building the Venom F5)
G: Kendra Hennessy (0.69 | Lynnie Greene (born May 21, 1954) is an actress, writer, director and producer in the television industry.)
H: launch Tennessee (0.69 | Tennessee (, locally), is a landlocked state in the Southeastern region of the United States.)
I: Kieran Hennessy (0.69 | The 1957 Croke Cup was the sixth staging of the Croke Cup since its establishment by the Gaelic Athletic Association)
J: Kenton Kelley (0.69 | Stan Kenton Presents Gabe Baltazar is an album by Gabe Baltazar. It was the last recording by Stan Kenton's Creative)
K: twenty ten Tennessee (0.65 | Tennessee (, locally), is a landlocked state in the Southeastern region of the United States.)

RAGEC

Answer: **A: kentucky tennessee** ❌

DeRAGEC

<think> The sentence \"how many miles from [Blank] to Brinkley, Arkansas\" is asking for a location that can be measured in distance to Brinkley. Among the options provided, \"Kenton, Tennessee\" (Option B) is the only recognized geographical location that fits this context. \n\n1. **Phonetic Match**: \"Kenton Tennessee\" has a high phonetic score of 0.90. \n\n2. **Geographical Relevance**: Kenton, Tennessee is a real place, making it a suitable candidate for the blank, as it can be measured in miles to Brinkley, Arkansas.\n\n3. **Other Options**: The other options either do not refer to geographical locations or are not relevant to the context of measuring distance.\n\nGiven this reasoning, the optimal choice to fill in the blank is \"Kenton, Tennessee.\" </think>
<answer> **B: Kenton Tennessee** </answer> ✅

Figure 4: Case study between REGEC and DeRAGEC in STOP dataset. The process of selecting the correct NE is highlighted in blue.

		Recall	Precision	WER
TopK	1	0.823	0.137	7.2
	5	0.835	0.084	6.6
	10	0.841	0.056	6.5
	15	0.843	0.032	6.7
PS (θ_d)	0.6	0.697	0.078	6.5
	0.7	0.457	0.117	6.6
	0.8	0.251	0.147	6.9
	0.9	0.126	0.163	7.2
STD (σ)	0.5	0.830	0.124	7.0
	1	0.827	0.131	7.3
	1.5	0.822	0.137	7.4
	2	0.815	0.138	7.4

Table 2: Effect of static phonetic filtering in WER at CommonVoice with Llama-v3.1.

		Recall	Precision	WER
TopK	1	0.887	0.148	6.7
	5	0.895	0.089	6.7
	10	0.896	0.060	6.5
	15	0.897	0.043	6.6
PS (θ_d)	0.6	0.857	0.065	6.5
	0.7	0.549	0.105	6.7
	0.8	0.244	0.143	6.7
	0.9	0.035	0.165	7.2
STD (σ)	0.5	0.841	0.107	6.8
	1	0.836	0.124	7.5
	1.5	0.822	0.134	7.6
	2	0.821	0.146	7.6

Table 3: Effect of static phonetic filtering in WER at STOP with Llama-v3.1.

	Recall	WER	NE hit ratio
RAGEC	0.841	6.5	0.804
DeRAGEC		6.0	0.831
ORACLE-retr		5.8	0.837
ORACLE	1.000	4.1	0.922
ORACLE+noise		5.4	0.887

Table 4: ORACLE comparison of different retrieval settings at CommonVoice with Llama-v3.1.

	RAGEC	DeRAGEC w/o MCQ	DeRAGEC
WER	6.5	6.8	**6.0**
NE hit ratio	0.804	0.780	**0.831**

Table 5: DeRAGEC w/o MCQ step at CommonVoice with Llama-v3.1.

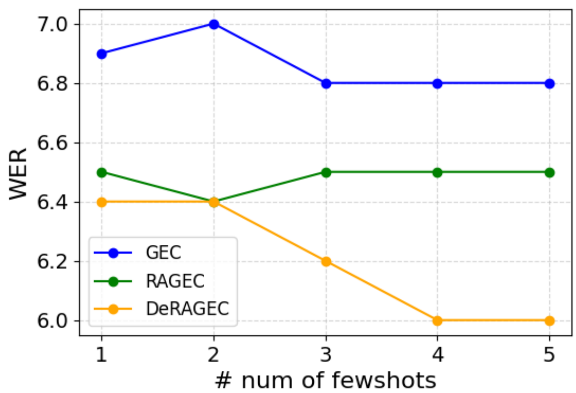

Figure 5: Effect of number of few-shot examples in WER at CommonVoice with Llama-v3.1.

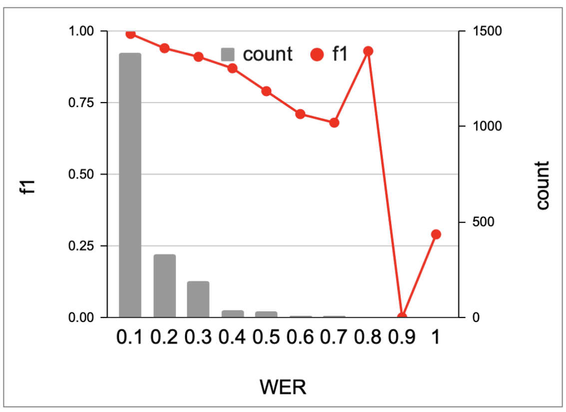

Figure 6: NER performance (F1 score) on different WER.

Module 1: Synthesize rationales for fewshots

You are given a cloze sentence, candidate named entities, their phonetic similarity scores, and their definitions. Each named entity is provided with several options indicated by ID letters A, B, C, etc. Each option follows the format: [ID letter]: [Named-Entity] ([Phonetic similarity score] | [Definition])

Your task is to identify the most appropriate Named-Entity for [BLANK].
Answer with the "letter: Named-Entity" corresponding to your choice A, B, C, etc.

{input}

Explain in detail how the input results in the answer: {answer}
Answer should not be said at first. The rationale and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> rationale here </think> <answer> answer here </answer>.

Table 6: Synthesize rationales for fewshots.

Module 2 -(1): Filtering NEs without reasoning process

[Task Description]
You are given a cloze sentence, candidate named entities, their phonetic similarity scores, and their definitions. The sentence are formatted as a cloze test, where the blanks to fill are indicated by [Blank]. Each named entity is provided with several options indicated by ID letters A, B, C, etc. Each option follows the format: [ID letter]: [Named-Entity] ([Phonetic similarity score] | [Definition])

Your task is to identify the most appropriate Named-Entity for [BLANK]. Please provide the answer with [letter]: [Named-Entity] format. REMEMBER you should return only the answer, not return any explanation.

I will give you few-shot examples.
{fewshot_examples}

[Test Case]
<input>
Cloze sentence: {cloze_sentence}
Options: {options}
<output>
Answer:

Table 7: Filtering NEs without reasoning process.

Module 2 -(2): Filtering NEs with reasoning process

[Task Description]
You are given a cloze sentence, candidate named entities, their phonetic similarity scores, and their definitions. The sentence are formatted as a cloze test, where the blanks to fill are indicated by [Blank]. Each named entity is provided with several options indicated by ID letters A, B, C, etc. Each option follows the format: [ID letter]: [Named-Entity] ([Phonetic similarity score] | [Definition])

Your task is to identify the most appropriate Named-Entity for [BLANK]. Please generate a brief explanation how the given input lead to your answer. You should not generate the answer before reasoning process. The rationale and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> rationale here </think> <answer> answer here </answer>.

I will give you few-shot examples.
{fewshot_examples}

[Test Case]
<input>
Cloze sentence: {cloze_sentence}
Options: {options}
<output>

Table 8: Filtering NEs with reasoning process.

Module 3: GEC

[Task Description]
You are given the 5-best hypotheses from a speech recognition system and a list of named entity candidates. Your task is to generate a corrected transcription using the 5-best hypotheses and named entities.

I will give you few-shot examples.
{fewshot_examples}

REMEMBER you should return only the corrected transcription, not return any explanation.

[Test Case]
<input>
5-best: {hypotheses (H)}
Named-Entities: {filterd_named_entities_with_generated_rationale $(\hat{n}, r)$}
<output>
Corrected:

Table 9: Prompt for GEC.