JudgerToken: A Single-Token Method for Reducing Repetition in Dialogue System

Qiang Xue, Tetsuya Takiguchi, Yasuo Ariki

Abstract Traditional approaches to enhancing language models in dialogue systems often involve complex modifications to tackle issues like sentence repetition and incoherence. Our research introduces 'JudgerToken', a streamlined solution that utilizes a single additional token, integrated into the language model's existing vocabulary. This novel token is designed to score sentences generated by the model, enabling the selection of outputs with the lowest repetition rates. By incorporating this token, JudgerToken effectively reduces repetitive patterns, enhancing text clarity and coherence without the need for architectural changes. Our results demonstrated that this method not only maintains the simplicity of the language model, but also significantly reduces repetition, thereby improving its overall performance.

1 Introduction

In recent years, dialogue systems have become increasingly sophisticated, evolving from simple rule-based engines to complex language models capable of generating human-like responses. Despite these advancements, one persistent challenge is the tendency of these systems to produce repetitive sentences [6]. This issue not only diminishes the user experience but also impacts the overall coherence and reliability of the dialogue system. For instance, a customer service chatbot might repeatedly offer the same solution, or a virtual assistant could echo similar phrases, leading to user frustration and disengagement.

Qiang Xue
Graduate School of System Informatics, Kobe University, e-mail: xueqiang@stu.kobe-u.ac.jp
Tetsuya Takiguchi
Graduate School of System Informatics, Kobe University, e-mail: takigu@kobe-u.ac.jp
Yasuo Ariki
Graduate School of System Informatics, Kobe University, e-mail: ariki@kobe-u.ac.jp

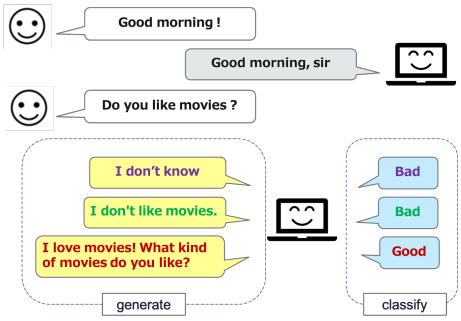

Fig. 1 The response generation process of the GCLM-based dialogue system.

Despite the advancements in dialogue systems, challenges like sentence repetition remain prevalent. Addressing these challenges often requires rethinking the foundational models used in dialogue systems. This leads us to explore two pivotal types of language models: Generative Language Models (GLMs) and Generative-Classification Language Models (GCLMs).

Generative Language Models (GLMs) [13, 1, 16], as the name suggests, primarily focus on generating text. These models, trained on vast corpora, excel in producing contextually relevant and syntactically correct responses. However, they can be prone to issues like repetitive sentence generation, which stems from their training on large-scale, diverse datasets without specific mechanisms to avoid repetition.

On the other hand, Generative-Classification Language Models (GCLMs) [2] represent an evolution in model design. They not only generate text but also classify or score it, typically using additional heads or mechanisms. This dual capability enables them to address some of the inherent limitations of GLMs, such as the tendency to produce repetitive or contextually irrelevant text. The response generation process of GCLM-based dialogue system is depicted in Figure 1.

In this context, our research introduces the SHGCLM (Sentence-Level Hybrid Generative-Classification Language Model) that leverages the strengths of both GLMs and GCLMs. The SHGCLM, while structurally akin to GLMs, incorporates a novel component—JudgerToken. This addition transforms the model into a GCLM by enabling it to perform classification tasks. Specifically, JudgerToken is used to evaluate and select responses with lower repetition rates, effectively tackling one of the key limitations of traditional GLMs.

The contributions of our research are multifaceted:

- **Simplicity and Efficiency**: The JudgerToken method maintains the inherent simplicity and efficiency of the existing language model, avoiding the complexities of architectural changes or additional training phases.
- **Effective Reduction of Repetition**: Our approach significantly lowers sentence repetition and simultaneously increases accuracy, as demonstrated by our experimental results.
- **Scalability and Flexibility**: The JudgerToken can be seamlessly integrated into existing models, providing a practical solution that can be easily adopted in various dialogue system applications.

2 Related Work

2.1 Generative Language Models (GLMs)

Transformer-based GLMs typically adopt Encoder-Decoder or Decoder-only frameworks. In Encoder-Decoder systems, conversation history is processed by the Encoder, with the Decoder generating responses sequentially. Decoder-only models, like GPT-2 [13], directly feed dialogue history into the Decoder for response generation. Among these, knowledge-graph (KG)-based prompt systems utilize structured knowledge, enhancing response diversity and effective topic transitions [8]. Our research builds upon this particular subclass of generative dialogue system as its foundational baseline.

2.2 Contrastive Learning (CL)

Contrastive learning, widely applied in natural language processing, improves discrimination in dialogue systems by learning similarities and differences between samples. Models like Dicer [3] combine negative sampling loss with contrastive learning for effective representation learning. Similarly, DialAug [12] employs contrastive learning to create dialogue context representations resilient to perturbations introduced during data augmentation. Our approach parallels this, using positive and negative examples to train language models on similarities and differences.

2.3 Generative-Classification Language Models (GCLMs)

Generative-Classification Language Models (GCLMs) blend generative language modeling with classification tasks. Encoder-based models like BERT use Next Sentence Prediction for sentence-level classification [4], while decoder-based models such as GPT-2 employ response determination tasks to evaluate generated responses [17]. Our research adapts these concepts, introducing a sentence-level evaluation model that expands on the Double Heads approach by Arora et al. [2], shifting the focus from word-level to sentence-level assessments within the GCLM-based dialogue system.

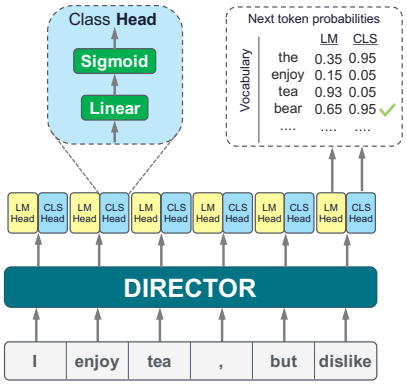

Fig. 2 DIRECTOR Model's architecture: Depicts the integration of the LM head and classifier in the DIRECTOR model, demonstrating how both heads collaboratively work to evaluate and generate dialogue responses.

3 Word-Level GCLM

DIRECTOR, a Word-Level Generative-Classification Language Model (GCLM), is designed to refine dialogue response quality by integrating a language model with a classification component.

Training Stage: The training of DIRECTOR is a two-step process. Initially, the model is trained on positive data samples, focusing on generating accurate and coherent responses. Once this generative training is complete, the model is used to produce responses based on the input from the training dataset. During this phase, repetitive words in these generated responses are identified and marked. Following this, the model undergoes further training, but this time exclusively for the classifier component. The classifier is fine-tuned to recognize and label the repetitiveness within the generated responses, thereby enhancing the model's ability to discriminate between suitable and unsuitable words.

Inference Stage: In the inference phase, as shown in Figure 2, DIRECTOR employs both the generator and the classifier in tandem. The generator predicts the likelihood of each word, while the classifier assesses its appropriateness in the given context. The final word selection is based on a combined score derived from both the language model head and the classifier. This score determines the most appropriate word for each position in the response, thus ensuring that the final output is contextually relevant and minimizes repetitiveness.

4 Sentence-Level GCLM

The GCLM's classifier evaluates the output generated from the GCLM's generator, categorizing each token as either <Good> (indicating an appropriate response) or <Bad> (indicating an inappropriate response).

4.1 Proposed Data Augmentation Strategy

In our approach to generate negative samples for sentence-level GCLM-based dialogue systems, we start with positive samples (target responses) from the training data.

Given a dialogue context $(x_1, \cdots, x_{H-1})$, which includes prompts and prior utterances, the corresponding positive response is denoted as $(x_H, \cdots, x_L)$. To construct a negative sample, we first extract a longer segment from the beginning of this positive response, represented as $(x_H, \cdots, x_R, H < R < L)$. For example, if the positive response is "I love going to the beach on sunny days", we might extract the segment "I love going to the beach on". Next, from within this extracted segment, we randomly select a smaller segment, such as "going to the beach", which corresponds to $(x_M, \cdots, x_N, H \leq M < N \leq R)$. This smaller segment is then appended to the end of the initial segment, forming a negative sample: "I love going to the beach on going to the beach", which represented as $(x_H, \cdots, x_R, x_M, \cdots, x_N)$.

The primary objective of this approach is to simulate responses that commonly exhibit repetitiveness. By training the model on such samples, we aim to enhance its ability to recognize and avoid generating responses with similar repetitive patterns, thereby improving the model's overall response quality.

4.2 Revised Model: DIRECTOR-Rev

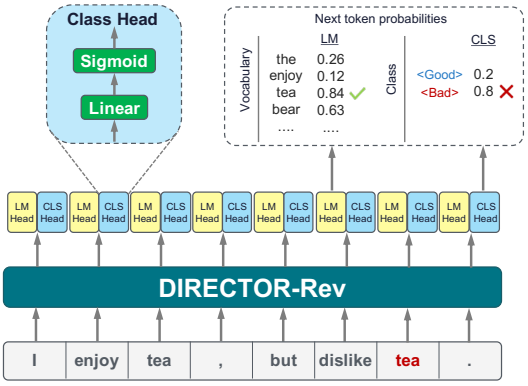

Fig. 3 DIRECTOR-Rev Model's architecture. This figure illustrates the revised version of the DIRECTOR model, the modified Class Head which evaluates multiple generated responses to determine the most contextually appropriate one based on the lowest <Bad> token score.

DIRECTOR-Rev is an evolved version of the original DIRECTOR model, specifically modified to align with the proposed data augmentation strategy at the sentence level. This revised model extends the capabilities of DIRECTOR, enabling it to effectively apply our sentence-level negative sample evaluation method.

The architecture of DIRECTOR-Rev, as shown in Figure 3, retains the core components of the original DIRECTOR model - the Class Head and the LM Head. The

Class Head, serving as a classifier, distinguishes between appropriate and inappropriate responses in the language model's output, similar to the original DIRECTOR. However, in DIRECTOR-Rev, it is further refined to handle sentence-level judgments, a key aspect of our data augmentation approach. The LM Head, as the generator, continues to produce the model's output, synthesizing language based on internal linguistic rules and input.

To accommodate the sentence-level data augmentation, the training and inference processes of DIRECTOR-Rev have been adjusted.

Positive Sample Training: In the training task for the LM Heads, the loss function is computed as:

$$L_{\text{LM}} = -\sum_{t=H}^{L} \log P(x_t \mid x_1, \cdots, x_{t-1}) \tag{1}$$

For the Class Heads training, the loss function is defined to maximize the probability of the <Good> token, indicating appropriate responses:

$$L_{\text{Class}} = -\log P(\textit{Good} \mid x_1, \cdots, x_L) \tag{2}$$

Negative Sample Training: The LM Heads' training for negative samples employs the following loss function:

$$L_{\text{LM}} = -\sum_{t=H}^{R} \log P(x_t \mid x_1, \cdots, x_{t-1}) \tag{3}$$

The Class Heads' training during negative sample processing is aimed at increasing the likelihood of the <Bad> token, to enhance the model's ability to identify inappropriate responses:

$$L_{\text{Class}} = -\sum_{t=M}^{N} \log P(\textit{Bad} \mid x_1, \cdots, x_t) \tag{4}$$

The overall training loss for the DIRECTOR-Rev model combines the LM and Class Heads' loss functions:

$$L_{\text{train}} = L_{\text{LM}} + L_{\text{Class}} \tag{5}$$

Inference Stage: As shown in Figure 3, the DIRECTOR-Rev model first generates 5 response candidates using the LM head through beam search (beam size = 5). Each response is then evaluated by the Class Head, which calculates the <Bad> token scores to assess response appropriateness. The final output is the response with the lowest <Bad> score, ensuring relevance and reducing repetitiveness.

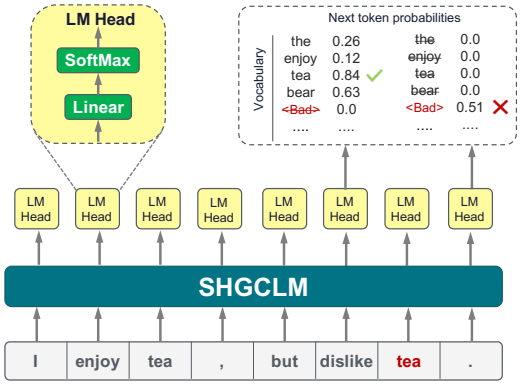

Fig. 4 SHGCLM Architecture: Illustrates the Single Head Model with the LM head using the JudgerToken <Bad> for response evaluation.

4.3 Proposed Model: SHGCLM

The Single Head model, instead of using a separate classifier, registers the JudgerToken, represented by <Bad>, as a token in the dictionary and employs the LM head to identify inappropriate responses. Since tokens other than <Bad> in the dictionary can be considered appropriate, the explicit inclusion of a <Good> token is not necessary. Figure 4 illustrates the architecture of the Single Head Model, highlighting the role of the JudgerToken in the process.

Positive Sample Training: In the training task for the LM head, the loss function is computed as per Equation 1.

Negative Sample Training: In the training task of the LM head, the loss function is defined as:

$$\begin{aligned} L_{\mathrm{LM}} = &- \sum_{t=H}^{R} \log P\left(x_t \mid x_1, \cdots, x_{t-1}\right) \\ &- \sum_{t=M}^{N} \log P\left(\mathit{Bad} \mid x_1, \cdots, x_t\right) \end{aligned} \tag{6}$$

Inference Stage: During inference, the SHGCLM model initially disregards the JudgerToken (<Bad>) and uses the LM head as a generator to produce 5 responses through beam search (beam size = 5). Following this, the model shifts focus, disregarding all tokens except the JudgerToken in the LM head's output to function as a classifier. This classifier evaluates and assigns a score to each generated sentence based on the likelihood of the JudgerToken. The final response is then selected based on the lowest JudgerToken score, ensuring that the output is both contextually appropriate and minimizes inappropriateness. This process, as depicted in Figure 4, demonstrates how the SHGCLM effectively generates and evaluates responses, utilizing the JudgerToken as a key factor in determining the most suitable reply.

5 Experiments and Evaluation

This chapter describes the experiments conducted to evaluate the performance of our proposed method from fowllowing perspectives:

- Performance comparison of different data augmentation strategy: See Section 5.2
- Performance comparison across different types of dialogue system: See Section 5.3
- Performance comparison of different number of generated negative sample : See Section 5.4

5.1 Experiment Setting

Our model was trained using the DialoGPT-small model from the Hugging Face Transformers library, featuring 117 million parameters for efficient yet effective dialogue tasks. We utilized the AdamW optimizer, with a learning rate set to 6.0e-5 and limited the dialogue history to three turns to balance computational efficiency and coherent response generation. The training, conducted over three epochs with a batch size of four, was performed on a machine equipped with an NVIDIA GeForce RTX 2070 GPU, demonstrating the model's improvements even on moderately equipped machines.

The experimental dataset, OpenDialKG [9], consists of 11,041 training sentences from dialogues about book and movie recommendations, incorporating structured knowledge entities. An equal number of negative training samples were generated and refreshed each epoch for robust learning. For experimental evaluation, response quality was measured in terms of repetitiveness, correctness, and diversity. Repetitiveness was assessed using Repeat@n [2], correctness through BLEU-n [11] and NIST-n [5], and diversity was evaluated with the DIST-n metric [7].

Additionally, in all our experiments, except for the DIRECTOR-Rev (section 4.2) and SHGCLM (section 4.3) models which utilized beam search as their decoding strategy to generate multiple sentences for selecting the highest quality response, all other models employed a greedy decoding strategy. This approach was chosen based on our observations that responses generated through greedy decoding were of higher quality in these models compared to those generated by beam search. The code is available at `https://github.com/asnower/SHGCLM`.

5.2 Data Augmentation Strategies: A Comparative Analysis

This experiment evaluates the performance of the proposed data augmentation strategy by comparing the following four approaches:

- **Word Detection [2]**(WD): This is the negative case generation method used in the DIRECTOR model. For details, see the learning phase in Chapter 3.

Table 1 Evaluation results of response sentences generated by different data augmentation strategy.

Model	**Method**	Repeat@5 ↓	DIST-2 ↑	BLEU-2 ↑	NIST-2 ↑
DIRECTOR	WD	20.44	**25.99**	10.73	1.67
DIRECTOR-Rev	RR	24.55	25.20	11.01	1.63
	RW	26.07	25.59	11.45	1.84
	CW	21.62	21.88	11.68	1.87
SHGCLM	RR	23.78	24.43	11.31	1.72
	RW	25.86	25.78	11.66	1.85
	CW	**12.28**	24.73	**11.89**	**1.90**

- **Random Response** (RR): In this data augmentation approach, other responses from the training data are utilized as negative samples. This approach generates an enriched and complex environment for the learning model to adapt to. For the negative sample target sentences, the <Bad> token is positioned at the end of the last word in the sentence.
- **Random Words** (RW): This data augmentation strategy generates negative samples by appending random segments from other positive samples to a given positive sample. This strategy introduces an element of randomness and ensures that the model is exposed to a diverse set of input contexts.
- **Copy Words** (CW, section 4.3): This is our proposed data augmentation strategy, which generates negative samples by appending a random segment from the same positive sample to itself. This technique is explicitly designed to deter the model from generating repetitive responses and its effectiveness is evaluated in comparison with RR and RW strategies.

Based on the evaluation results presented in Table 1, it is evident that different negative example creation methods significantly impact the performance of the models in generating response sentences. The SHGCLM model, utilizing our proposed Copy Words (CW) data augmentation strategy, demonstrates a notable improvement in reducing repetitiveness (lowest Repeat@5 score) while simultaneously achieving the highest scores in BLEU-2 and NIST-2, indicating superior accuracy and relevance of the responses. This suggests that the CW method effectively trains the model to avoid repetitive patterns and generate contextually appropriate and diverse responses.

In comparison, the DIRECTOR model with Word Detection (WD) method shows a higher tendency towards repetitive responses, as indicated by its Repeat@5 score, albeit with a higher DIST-2 score, suggesting a degree of diversity in word usage. The Random Response (RR) and Random Words (RW) methods, applied in both DIRECTOR-Rev and SHGCLM models, exhibit varied performance, but neither approach matches the effectiveness of the CW method in overall response quality.

Table 2 Evaluation results of response sentences generated by different dialogue systems.

Dialogue System	Model	Repeat@5 ↓	DIST-2 ↑	BLEU-2 ↑	NIST-2 ↑
GLM-based	DialoGPT	21.96	22.23	10.68	1.61
	DialoGPT-ITF	35.42	**28.11**	10.78	1.83
CL-based	SelfCont	14.00	22.96	11.0	1.70
	R3F	25.20	18.06	10.67	1.64
GCLM-based	DIRECTOR	20.44	25.99	10.73	1.67
	DIRECTOR-Rev	21.62	21.88	11.68	1.87
	SHGCLM	**12.28**	24.73	**11.89**	**1.90**

5.3 Evaluating Dialogue Systems: A Type-Based Comparison

This experiment evaluates the performance of the proposed dialogue system by comparing the following three types of dialogue system:

- **GLM-based**: DialoGPT [19], a generative language model, is utilized in our experiments as an example of GLM-based dialogue systems. DialoGPT is known for its effectiveness in generating dialogue based on large-scale training data. Additionally, we evaluate DialoGPT-ITF, a variant of DialoGPT enhanced with the Inverse Token Frequency [10] method to improve response diversity by reducing the frequency of common phrases.
- **CL-based**: In the contrastive learning category, we employ SelfCont and R3F for comparison. SelfCont [6] is a self-contrastive learning method designed to mitigate repetitive text, while R3F [15] focuses on minimizing negative log-likelihood and symmetric KL divergence for robust feature learning.
- **GCLM-based**: For Generative-Classification Language Models, we include the DIRECTOR model (Section 3) and our propoesd SHGCLM model (Section 4.3).

The evaluation results in Table 2 indicate that the GCLM-based SHGCLM model outperforms others in reducing repetition (as shown by the lowest Repeat@5 score) and in the accuracy and relevance of responses (highest BLEU-2 and NIST-2 scores). The DialoGPT-ITF model demonstrates a significant improvement in diversity over the standard DialoGPT, as indicated by its higher DIST-2 score. This highlights the effectiveness of the ITF method in enhancing diversity in GLM-based models. However, while DialoGPT-ITF leads in diversity, GCLM-based systems still outperform in balancing contextual relevance and response accuracy. The CL-based models display mixed results, with SelfCont showing promise in reducing repetition but not as effectively as GCLM-based models. These outcomes highlight the advanced capabilities of GCLM-based systems in handling the complexities of dialogue generation, offering a balanced approach between creativity and coherence.

The dialogue examples in Table 3 clearly demonstrate the superiority of the SHGCLM model in generating coherent and varied responses. Unlike DialoGPT, which exhibits repetition, and DIRECTOR, which occasionally repeats words, SHGCLM consistently provides more diverse and contextually relevant replies, showcasing its advanced response generation capabilities in dialogue systems.

Table 3 Examples of dialogues using each response generation method. The words in red are problem words in the response.

Input 1	Prompt	Mark Osborne starred actors Kung Fu Panda.
	User A	Could you recommend some movies that were directed by Mark Osborne?
DialoGPT		Sure! He starred in Kung Fu Panda 2 and Kung Fu Panda 2.
DIRECTOR		Mark Osbourne directed Kung Fu Panda. Have you seen that movie?
SHGCLM		Sure! He starred in Kung Fu Panda. Have you seen that movie?
Input 2	Prompt	Andrew Delbanco is-a Professor
	User A	Do you know the author Andrew Delbanco?
DialoGPT		Andrew Delbanco is a Professor. He is also a Professor.
DIRECTOR		Andrew Delbanco is a Professor, Professor, and Lawyer. Are you familiar with his work?
SHGCLM		Yes, Andrew Delbanco is a Professor. He is also an author. Do you like Science Fiction?
Input 3	Prompt	Douglas Adams written by The Restaurant at the End of the Universe
	User A	I really liked the book The Ultimate Hitchhiker's Guide: Five Complete Novels and One Story. Could you recommend something similar?
	User B	Would you like another story written by Douglas Adams?
DialoGPT		Yes, I would. I really liked him in The Restaurant at the End of the Universe.
DIRECTOR		Douglas Adams also wrote The Restaurant at the End of the Universe. Have you read that one?
SHGCLM		Yes, please!
Input 4	Prompt	Do you know who starred in the film This Is the End?
	User A	It appears that its Evan Goldberg. Did you know that he wrote The Watch?
	User B	I did not. When was The Watch released?
DialoGPT		It was released in 2010.
DIRECTOR		The Watch was released in 2001.
SHGCLM		The Watch was released in 2008.

However, the response in Input 3 highlights a limitation of the SHGCLM model: its tendency to overlook the context provided in the input prompt. Instead of leveraging the specific information about Douglas Adams' work, the model responds generically, missing an opportunity to provide a more informed and tailored response. Input 4 presents a broader issue affecting all models, where each generates factually incorrect information. This phenomenon, known as hallucination, underscores the need for dialogue systems to ensure factual accuracy and contextual alignment. For the SHGCLM model, addressing these challenges will involve enhancing its ability to integrate and utilize provided knowledge effectively, ensuring that responses are not only contextually appropriate but also factually accurate.

Table 4 Comparative Performance Metrics of SHGCLM on Various Datasets with Inclusion of Negative Samples.

Dataset	**Negative Data**	Repeat@5 ↓	DIST-2 ↑	BLEU-2 ↑	NIST-2 ↑
OD	0%	21.96	22.23	10.68	1.61
	40%	12.89	**25.68**	11.62	1.83
	80%	**10.96**	23.85	11.44	1.75
	100%	12.28	24.73	**11.89**	**1.90**
ED	0%	24.34	20.4	5.7	0.80
	40%	13.09	**23.31**	6.23	0.95
	80%	15.28	19.52	**6.68**	**1.15**
	100%	**11.10**	20.17	5.39	0.68
PC	0%	14.21	**16.59**	6.35	0.97
	40%	8.72	16.10	2.08	0.87
	80%	**8.42**	12.77	5.57	0.79
	100%	13.22	16.20	**6.58**	**1.04**

5.4 Analyzing Performance Across Datasets with Varied Negative Sample Sizes

In our experiments, the SHGCLM model's performance was assessed on three distinct datasets: OpenDialKG (OD), Empathetic Dialogues [14], and PersonaChat (PC) [18]. These datasets, each with unique dialogue characteristics, were augmented with varying proportions of negative samples (0%, 40%, 80%, and 100%).

As illustrated in Table 4, the inclusion of negative samples generally improved the performance metrics across all datasets. Notably, a 100% negative data inclusion in OD and PC showed significant improvements in BLEU-2 and NIST-2 scores, suggesting enhanced response accuracy and relevance. However, the same level of negative sample inclusion in ED resulted in a decrease in these metrics, indicating that the optimal proportion of negative samples may vary depending on the dataset's characteristics. The DIST-2 scores were the highest at a 40% inclusion rate for OD and ED, implying that a balanced mix of positive and negative samples is crucial for maintaining diversity in responses. These findings highlight the importance of carefully calibrating the proportion of negative samples to achieve the best performance in different dialogue environments.

6 Conclusion

In conclusion, we proposed the SHGCLM model with the innovative use of JudgerToken, a streamlined approach to reduce repetition in dialogue systems. Our experiments across various datasets confirmed the model's effectiveness in enhancing response quality. Future efforts will concentrate on fine-tuning negative sample ratios and

broadening the model's adaptability to further linguistic contexts, aiming to enrich both coherence and diversity in generated dialogues.

References

1. Adiwardana, D., Luong, M.T., So, D.R., Hall, J., Fiedel, N., Thoppilan, R., Yang, Z., Kulshreshtha, A., Nemade, G., Lu, Y., et al.: Towards a human-like open-domain chatbot. arXiv preprint arXiv:2001.09977 (2020)
2. Arora, K., Shuster, K., Sukhbaatar, S., Weston, J.: Director: Generator-classifiers for supervised language modeling. arXiv preprint arXiv:2206.07694 (2022)
3. Cho, J., Ko, Y.: Dicer: Dialogue-centric representation for knowledge-grounded dialogue through contrastive learning. Pattern Recognition Letters **172**, 151–157 (2023). DOI https://doi.org/10.1016/j.patrec.2023.05.034. URL https://www.sciencedirect.com/science/article/pii/S0167865523001691
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
5. Doddington, G.: Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on Human Language Technology Research, pp. 138–145 (2002)
6. Guan, J., Huang, M.: Mitigating the learning bias towards repetition by self-contrastive training for open-ended generation. In: A. Rogers, J. Boyd-Graber, N. Okazaki (eds.) Findings of the Association for Computational Linguistics: ACL 2023, pp. 6897–6909. Association for Computational Linguistics, Toronto, Canada (2023). DOI 10.18653/v1/2023.findings-acl.431. URL https://aclanthology.org/2023.findings-acl.431
7. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, B.: A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055 (2015)
8. Liu, S., Chen, H., Ren, Z., Feng, Y., Liu, Q., Yin, D.: Knowledge diffusion for neural dialogue generation. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1489–1498 (2018)
9. Moon, S., Shah, P., Kumar, A., Subba, R.: Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 845–854 (2019)
10. Nakamura, R., Sudoh, K., Yoshino, K., Nakamura, S.: Another diversity-promoting objective function for neural dialogue generation. CoRR **abs/1811.08100** (2018). URL http://arxiv.org/abs/1811.08100
11. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
12. Poddar, L., Wang, P., Reinspach, J.: DialAug: Mixing up dialogue contexts in contrastive learning for robust conversational modeling. In: N. Calzolari, C.R. Huang, H. Kim, J. Pustejovsky, L. Wanner, K.S. Choi, P.M. Ryu, H.H. Chen, L. Donatelli, H. Ji, S. Kurohashi, P. Paggio, N. Xue, S. Kim, Y. Hahm, Z. He, T.K. Lee, E. Santus, F. Bond, S.H. Na (eds.) Proceedings of the 29th International Conference on Computational Linguistics, pp. 441–450. International Committee on Computational Linguistics, Gyeongju, Republic of Korea (2022). URL https://aclanthology.org/2022.coling-1.35
13. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019)
14. Rashkin, H., Smith, E.M., Li, M., Boureau, Y.L.: Towards empathetic open-domain conversation models: A new benchmark and dataset. arXiv preprint arXiv:1811.00207 (2018)
15. Tong, S., Dong, Q., Dai, D., Song, Y., Liu, T., Chang, B., Sui, Z.: Robust fine-tuning via perturbation and interpolation from in-batch instances. In: International Joint Conference on Ar-

tificial Intelligence (2022). URL https://api.semanticscholar.org/CorpusID:248495890
16. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
17. Wolf, T.: How to build a state-of-the-art conversational ai with transfer learning (2019). https://medium.com/huggingface/
18. Zhang, S., Dinan, E., Urbanek, J., Szlam, A., Kiela, D., Weston, J.: Personalizing dialogue agents: I have a dog, do you have pets too? arXiv preprint arXiv:1801.07243 (2018)
19. Zhang, Y., Sun, S., Galley, M., Chen, Y.C., Brockett, C., Gao, X., Gao, J., Liu, J., Dolan, B.: Dialogpt: Large-scale generative pre-training for conversational response generation. arXiv preprint arXiv:1911.00536 (2019)