Are U a Joke Master? Pun Generation via Multi-Stage Curriculum Learning towards a Humor LLM

Yang Chen[1,2], Chong Yang[2], Tu Hu[1,2], Xinhao Chen[1,2], Man Lan[1,3]*, Li Cai[1,4], Xinling Zhuang[1], Xuan Lin[2], Xin Lu[2], Aiming Zhou[1,3]

[1]School of Computer Science and Technology, East China Normal University, Shanghai, China
[2]Ant Group, Shanghai, China
[3]Shanghai Institute of AI for Education, East China Normal University, Shanghai, China
[4]College of Computer Science and Technology, Guizhou University, Guiyang, China
{yangchen,thu,xinhaochen,lcai2020,xinlinzhuang}@stu.ecnu.edu.cn
{yangchong.yang,daxuan.lx,lx111333}@antgroup.com {mlan,amzhou}@cs.ecnu.edu.cn

Abstract

Although large language models (LLMs) acquire extensive world knowledge and some reasoning abilities, their proficiency in generating humorous sentences remains a challenge. Previous research has demonstrated that the humor generation capabilities of ChatGPT are confined to producing merely 25 unique jokes. In this work, we concentrate on endowing LLMs with the ability of generating puns, a particular category of humor by preference learning method. We propose a multi-stage curriculum preference learning framework to optimize both pun structure preferences and humor preferences. Specifically, we improve the Direct Preference Optimization (DPO) algorithm to address the challenge of multi-objective alignment problem. Besides, to facilitate further advancement in this field, we collect a Chinese Pun (ChinesePun) dataset, containing 2.1k puns and corresponding annotations. Experimental results on both Chinese and English benchmark datasets demonstrate that our method significantly outperforms all the baseline models.

1 Introduction

Humor serves various purposes and offers numerous benefits for humans, including the relief of anxiety, avoidance of painful emotions, and facilitation of learning (Buxman, 2008). However, it remains a challenge to let machines generate humorous sentences as humans (Jentzsch and Kersting, 2023). Empowering LLMs with a sense of humor has become an increasingly important topic in the field of natural language generation. As a particular category of humor, the creative utilization of puns, wordplay, and ambiguity plays a significant role in humor generation (Chiaro, 2006). Puns, utilizing identical or phonetically similar words, are categorized into homographic and homophonic types (Miller et al., 2017). Homographic puns play on different meanings of the same word; for instance, '*all right*' can signify '*satisfactory*' or '*not left*'. This word creates humor through contrasting interpretations within context. On the other hand, homophonic puns involve words with similar sounds but distinct meanings, like '*weak*' and '*week*'.

As pun generation aims to generate a pun to create humorous effects given a pair of pun word and alternative word as shown in Table 1, it substantially contains two targets: pun structure and humor effect. Generating puns requires vast world knowledge and common sense (Sun et al., 2022b). Due to the constrained model size, previous approaches (He et al., 2019; Tian et al., 2022; Sun et al., 2022b) primarily achieve limited performance improvement. With the rise of LLMs, the ability to generate text has been greatly improved. However, Jentzsch and Kersting (2023) reveal that LLMs' proficiency in generating humorous sentences and puns still remains a challenge. To further improve the capability of LLMs on an alternative aspect, preference learning/alignment has become one of the most commonly used approaches (Casper et al., 2023). Directly utilizing preference learning methods like Direct Preference Optimization (DPO) (Rafailov et al., 2023) in pun generation is not very efficient because we need to align both the pun structure preference and humor preference at the same time on a relatively small-scale dataset.

To address this problem, we propose a multi-stage curriculum learning framework to generate a sentence satisfying both the provided pun structure and high humorous level. Inspired by the paradigm of curriculum learning which learns samples from easy to hard, our model learns the two preference targets separately from easy to hard, i.e., first learns pun structure preference in stage 1, and then learns humor preference in stage 2. Specifically, to address the inherent catastrophic forgetting problem

* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2024, pages 878–890

Type	Dataset	Pun	pun word, alternative word
homophonic	ChinesePun	篮球队教练在开会时抱怨道：“我们球队的配置太差了，现在需要一个投篮的。”过了一会，助理教练领着机器猫走了进来：“教练，你看他的头蓝吗？” The coach of the basketball team was in a meeting and complained, "Our team is so poorly set up, we need a shooter now." A little while later, the assistant coach walks in with Robot Cat: "Coach, do you see his head in blue?"	头蓝, 投篮 the head is blue, shoot a basketball
	SemEval	I lift weights only on Saturday and Sunday because Monday to Friday are weak days.	weak, week
homographic	ChinesePun	路上看到消防车开过去,大人说:他们去救火. 孩子说:救火?应该是灭火,火是坏的,还要救它呀！ The adults said as a fire truck drove by on the road: 'They're going to put out a fire.' The child replied: 'Put out a fire? Shouldn't it be extinguishing the fire? Fire is bad; why would we save it?'	救, 救 put out the fire, save sb's life
	SemEval	Did you hear about the guy whose whole left side was cut off? He's all right now.	all right, all right

Table 1: Examples of homophonic and homographic puns in the ChinesePun and SemEval 2017 Task 7 dataset. Each annotation includes a pair consisting of the pun word and its alternative word. To facilitate better understanding for the readers, we have translated the Chinese examples in the table into English.

of these kinds of multi-stage preference learning, we introduce an improved humor preference aligning algorithm in stage 2. Different from standard DPO which learns from pairs of positive and negative samples, we make each training sample a triplet by adding an extra generation output from the model of stage 1 by the same input prompt. We propose a new loss function to adapt to this multi-objective alignment task and to alleviate the effects of catastrophic forgetting.

Additionally, previous researches have focused on the English pun dataset from SemEval 2017 task 7 (SemEval) (Miller et al., 2017) for evaluation. The dataset contains 1298 homographic puns and 1098 homophonic puns annotated with pun words and alternative words. To verify the broad applicability of our method in different languages, we manually annotated the first Chinese dataset in the field, which is called ChinesePun. Table 1 shows examples in our dataset and SemEval 2017 Task 7 dataset. This dataset comprises 1,049 homophonic puns and 1,057 homographic puns, where each pun is annotated with its corresponding pun words and alternative words. The experimental results on both datasets indicate that our proposed method effectively enhances the ability of LLMs to generate puns and significantly outperforms existing methods. Our contributions are summarized as follows:

- We introduce a multi-stage curriculum learning framework with a novel triplet preference learning method to enhance LLMs' ability to generate humorous puns.
- We release the very first dataset for the Chinese pun generation task, aiming to stimulate advancements in the field of Chinese humor understanding and generation[1].
- We conduct extensive experiments on both the English and Chinese datasets to demonstrate the superiority of our proposed method.

2 Related Work

2.1 Pun Generation

Previous researches have focused on the SemEval 2017 Task 7 dataset (Miller et al., 2017), which contains 1,298 homographic puns and 1,098 homophonic puns, annotated with pun words and alternative words. Prior works on pun generation primarily targeted phonological or syntactic patterns over semantic ones (Miller and Gurevych, 2015; Hong and Ong, 2009; Petrović and Matthews, 2013; Valitutti et al., 2013), sacrificing flexibility. He et al. (2019) used the local-global surprisal principle to create homophonic puns, whereas Yu et al. (2020) utilized constrained lexical rewriting for the same purpose. Hashimoto et al. (2018) employed a retrieve-edit approach for homographic puns, and Yu et al. (2018); Luo et al. (2019) proposed advanced neural models like constrained language models and GANs. Mittal et al. (2022) generated homographic puns from polysemes and sought to incorporate their multiple senses. Tian et al. (2022) proposed a unified framework for both homographic and homophonic puns, leveraging humor principles. The keyword-conditioned pun generation setup can also facilitate more engaging pun generation scenarios such as context-situated pun generation (Sun et al., 2022b). However, all the aforementioned methods are designed specifically for small-scale models. To the best of our knowledge, our work is the first to enhance the pun generation capability of LLMs.

[1]Code, data, and resources are publicly available for research purposes: https://github.com/cubenlp/PGCL.

2.2 Preference Learning

Preference learning/alignment, especially via Reinforcement Learning with Human Feedback (RLHF) (Ziegler et al., 2020), is a prominent trend catalyzed by advancements in LLMs like ChatGPT (Ouyang et al., 2022), LLaMA (Touvron et al., 2023), and Baichuan (Yang et al., 2023). However, RLHF faces challenges such as instability, inefficiency, and vulnerability to exploitation (Casper et al., 2023; Skalse et al., 2022). Addressing these issues, novel methods have emerged, notably Direct Preference Optimization (DPO) (Rafailov et al., 2023), which aligns models with reference ones using paired preferences, enhancing efficiency and stability. Statistical Rejection Sampling Optimization (RSO) (Liu et al., 2023), an advancement building on DPO and SLiC (Zhao et al., 2022), employs rejection sampling for more effective optimization. Additionally, Xu et al. (2023) introduces a contrastive post-training curriculum, progressively shifting from simpler to harder preference pairs to improve efficacy. In contrast to these works, we introduce a new multi-stage curriculum learning framework and a triplet preference learning loss aiming at enhancing the stability of multi-objective preference alignment in the task of pun generation.

3 Problem Definition

The input to our system consists of a *pun word pair*, which includes a pun word (p_w, e.g., *weak*) and a corresponding alternative word (a_w, e.g., *week*). p_w are defined as words that evoke humor by possessing conflicting and ambiguous meanings within jokes. In the case of homophonic puns, the pun words have a homonym a_w with the same or similar pronunciation but different meanings. For homographic puns where the pun word carries dual meanings that are logically coherent within the context, we adopt a same representation, setting $p_w = a_w$, following Tian et al. (2022). The objective is typically to generate a pun that contains p_w with high humorous level, e.g., *"I lift weights only on Saturday and Sunday because Monday to Friday are weak days."*. Thus, the generated pun should comply with two conditions: pun structure (e.g., containing pun word) and humorous.

4 Methodology

4.1 Multi-Stage Curriculum Learning

Different from direct preference alignment on LLMs, we present a multi-stage curriculum learning framework to offer a smoother and more effective preference learning trajectory on pun generation. We utilize the Direct Preference Optimization (DPO) method in two stages to steer the model toward optimizing two key preferences: ***pun structure*** and ***humor***. As illustrated in Figure 1, the framework, which we called ***P**un **G**eneration with **C**urriculum **L**earning* (**PGCL**), comprises two main components: (i) a structure preference optimization module (top) designed to enhance the LLM's ability to satisfy pun structures in the first stage, and (ii) a humor preference optimization module (bottom) aimed at aligning with the more challenging humor preference in the second stage.

Structure Preference Alignment In this stage, we optimize the structure preference of LLMs with the DPO algorithm. Given a pun word and alternative word pair (p_w, a_w), we transform it to a prompt x with the designed template (e.g., for English homophonic puns, the prompt template is *"Given the pun word 'p_w' and its homophonic alternative word 'a_w', please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word 'p_w' is present in the sentence."*. Other prompt templates can be found in Appendix A).

DPO is an offline preference optimization technique that takes a pre-computed pair of positive samples y^+ and negative samples y^-, both corresponding to the same prompt x. We make the labeled pun in the training dataset the positive sample y^+. Then, we produce negative samples using SFT LLMs (e.g., LLaMA2). Specifically, we randomly generate a pun with the prompt x, and introduce a structure discriminator to judge whether the pun satisfies the structure: 1) for English homographic and Chinese homographic puns, as $p_w = a_w$, we actually only need p_w present in the sentence. 2) For English homophonic puns, based on the settings of `SemEval` dataset, the sentence must have p_w without a_w. 3) For Chinese homophonic puns, both p_w and a_w must be presented in the sentence. If the generated pun does **NOT** comply with the above structure, we accept it as the negative sample y^-. Consequently, we construct the structure preference pairs (y^+, y_s^-) to do structure DPO.

Following the standard DPO algorithm which aims to increase the likelihood of the positive example while reducing that of the negative example. The loss function of the structure DPO is presented

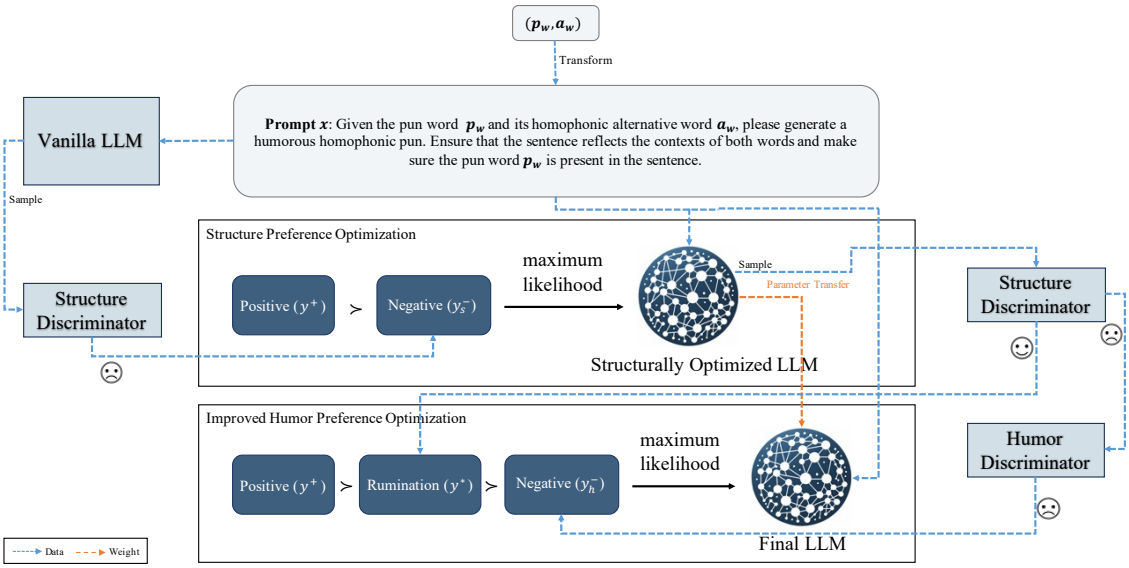


Figure 1: Overall framework of our method. "Vanilla LLM" denotes the initial SFT model (e.g., LLaMA2). Blue dotted lines represent the flow of data, while orange dotted lines indicate the flow of parameters. The "Prompt x" we gave in the figure originates from English homophonic puns.

as follows:

$$r^+(\theta) = \beta(\log \pi_\theta(y^+|x) - \log \pi_{\text{sft}}(y^+|x)) \quad (1)$$

$$r_s^-(\theta) = \beta(\log \pi_\theta(y_s^-|x) - \log \pi_{\text{sft}}(y_s^-|x)) \quad (2)$$

$$\mathcal{L}_{\text{structure-dpo}}(\theta) = -\log \sigma(r^+(\theta) - r_s^-(\theta)) \quad (3)$$

where β is a temperature hyperparameter; π_θ is the language model to be optimized and π_{sft} is the SFT language model; σ is the sigmoid function; r^+ and r_s^- are the two pseudo-rewards that resemble the reward function in RLHF. DPO requires the logits from the frozen reference model (i.e., the SFT model) for both the positive and negative sequences. It enhances the performance by increasing the discrepancy between the r^+ and r_s^-.

Humor Preference Alignment In the second stage, we attempt to use the similar method to further optimize the humor preference. We use the same prompts x and positive samples y^+ in the training dataset with the first stage. Then, we generate the candidate sentences from the LLMs which have been structurally optimized in the first stage. Similarly, we train a humor discriminator to judge the humor level. Specifically, for the English humor discriminator, we follow Mittal et al. (2022) to fine-tune RoBERTa-large (Liu et al., 2019) using the ColBERT dataset (Annamoradnejad and Zoghi, 2022), which comprises 200,000 examples of jokes and non-jokes for humor detection. For the Chinese humor discriminator, we use the humor recognition dataset from Chen et al. (2023), which comprises 18709 examples of jokes and 7709 examples of non-jokes, and fine-tune RoBERTa-large (Cui et al., 2020, 2021) model on the dataset. The output probability from the classification model is then employed to select negative samples y_h^-. Consequently, we can get the humor preference pairs (y^+, y_h^-).

Subsequently, we employ the following loss function to optimize the humor preference on the structure-optimized LLM of stage 1:

$$r_\phi^+(\theta) = \beta(\log \pi_\theta(y^+|x) - \log \pi_\phi(y^+|x)) \quad (4)$$

$$r_h^-(\theta) = \beta(\log \pi_\theta(y_h^-|x) - \log \pi_\phi(y_h^-|x)) \quad (5)$$

$$\mathcal{L}_{\text{humor-dpo}}(\theta) = -\log \sigma(r_\phi^+(\theta) - r_h^-(\theta)) \quad (6)$$

where $r_\phi+$ and r_h^- are the corresponding pseudo-rewards; π_ϕ is the language model that has been structurally optimized in stage 1.

4.2 Improved Humor DPO

Due to the limited training data, multi-stage preference alignment faces the issue of catastrophic forgetting, meaning that the effect of structure preference alignment decreases in stage 2. To mitigate the impact of catastrophic forgetting, we propose an improved triplet humor alignment DPO in the second stage.

Specifically, in order to preserve the model's preference objectives of the previous stage, we initially sample candidate sentences from the structurally optimized LLMs of stage 1 using the same

prompts x. The sentences that satisfy the pun structure by our structure discriminator described in Section 4.1 are retained, which we call "rumination" sample y^*. Then, we select negative samples $y^-_{h^*}$ among the candidate sentences that lack humor and do not satisfy pun structures with both the structure discriminator and humor discriminator. We use the same positive samples y^+ with the standard humor DPO we described in the last section to construct the humor preference triplets $(y^+, y^-_{h^*}, y^*)$.

Then, we optimize the humor preference of the LLM with a new triplet loss:

$$r^-_{h^*}(\theta) = \beta(\log \pi_\theta(y^-_{h^*}|x) - \log \pi_\phi(y^-_{h^*}|x)) \quad (7)$$

$$r^*(\theta) = \beta(\log \pi_\theta(y^*|x) - \log \pi_\phi(y^*|x)) \quad (8)$$

$$\begin{aligned}\mathcal{L}_{\mathrm{I-humor-dpo}}(\theta) = &- \log \sigma(r^+_\phi(\theta) - r^*(\theta)) \\ &- \log \sigma(r^*(\theta) - r^-_{h^*}(\theta))\end{aligned} \quad (9)$$

where r^* means the pseudo-reward of rumination samples; r^+_ϕ is same with Equation 4. The objective is to maximize the difference in reward values between positive and rumination sequences, as well as the difference between the reward values of rumination and negative sequences. We can obtain the LLM which is well aligned with both structure preference and humor preference through the above loss function.

5 ChinesePun Dataset

5.1 Data Preparation

To construct a Chinese Pun dataset, we first gathered original jokes from various sources, including Github[2] and humor websites [3], and obtained a dataset of 28,269 jokes. Some of these jokes have been labeled as homographic or homophonic types, while others have yet to be classified. Subsequently, we developed an automatic approach for labeling the remaining unclassified data.

We introduced a classification model to assist us in selecting homographic and homophonic puns. To achieve this, we trained the RoBERTa-Large (Liu et al., 2019) model on the humor classification dataset from CCL2018 Task4[4], which includes homographic puns, homophonic puns and reverse jokes that have already been classified. We used the model's output to extract candidate sentences that may contain homographic or homophonic puns. Upon processing the unclassified data, we combined the classified puns with the candidate sentences to obtain 2,394 pseudo-homophonic puns and 4,147 pseudo-homographic puns, which are used for the pun words annotation.

Type	sentences	words	pun words		
			min	max	mean
homophonic	1049	96212	1	4	1.11
homographic	1057	91100	1	3	1.05

Table 2: Statistics of the ChinesePun dataset

5.2 Dataset Annotation

To begin with, we utilized few-shot prompting on ChatGPT (Ouyang et al., 2022) for pre-annotation to decrease the amount of manual annotation required. Specifically, we first randomly selected three sentences each from homophonic puns and homographic puns and manually annotated them. Then, we used the original text and corresponding annotation results as examples and concatenated them with an instruction as a prompt, which was inputted into ChatGPT. We processed each sentence using ChatGPT and obtained shallow annotations.

We then revised the original text and corresponding annotation results manually. Three postgraduate students worked together to complete the annotation of the pun dataset. To ensure agreement among the annotation results, any disagreements were discussed by the entire group, and the final result was determined by the option receiving the majority of votes. To check inter-annotator agreement (IAA), we collected multiple annotations for 150 instances and measured agreement using Fleiss' kappa (Fleiss and Cohen, 1973) ($\kappa = 0.68$), suggesting high agreement.

During the annotation process, we carefully reviewed each sentence to determine whether it contains a pun. If a sentence is found not including a pun, we removed it from the list of sentences to be annotated. This step ensured that we focus our attention solely on the sentences that require annotation and helped to improve the efficiency and accuracy of the overall annotation process.

5.3 Data Statistics

Upon completion of all the annotation processes, we report details of the dataset in Table 2. The

[2] https://github.com/liuhuanyong/ChineseHumorSentiment

[3] http://www.jokeji.cn; http://www.yiyixh.com/a/wenzixiaohua; http://m.jokedw.com/joke

[4] http://www.cips-cl.org/static/CCL2018

Dataset	SemEval		ChinesePun	
	phonic	graphic	phonic	graphic
Train Examples	879	1039	839	845
Test Examples	219	259	210	212
Total Examples	1098	1298	1049	1057

Table 3: Data statistics. “phonic” and “graphic” denote homophonic and homographic puns.

dataset contains 1,049 homophonic puns, 1,057 homographic puns, and 187,315 words in total. Upon analyzing the count of pun words in our dataset, we discovered that the majority of puns include only one *pun word pair*. Consequently, our study narrows its focus exclusively to this scenario.

6 Experiments

6.1 Dataset

To prove the effectiveness of our method, we conduct experiments on ChinesePun and SemEval 2017 Task 7 dataset. Table 3 shows the data statistics of both datasets. For the ChinesePun dataset, we partition it into 1684 samples for training and 422 for testing. Regarding the SemEval 2017 Task 7 dataset, we allocate 1918 samples for the training set and 478 for the testing set. Both training and testing sets maintain an equal distribution of homophonic and homographic data.

6.2 Metrics

Automatic Evaluation To measure the model's ability to incorporate *pun word pairs* in the final generation, we utilize the structure success rate (**Structure Succ.**) as our primary automatic evaluation metric, following Sun et al. (2022a,b). To evaluate diversity, we follow Luo et al. (2019); Yu et al. (2018); Mittal et al. (2022) to calculate distinct unigrams (**Dist-1**) and bigrams (**Dist-2**) in terms of sentence level and corpus level. Average sentence length (**Avg-Length**) is also reported.

Human Evaluation 50 *pun word pairs* are sampled randomly from the test dataset for human evaluation. To evaluate humor capabilities, we conduct a **Human A/B-test** to compare our models with ChatGPT. We pair the sentences generated by models and ask annotators to choose the better sentence based on the humor. If the difference is not significant, a tie is allowed. To evaluate the success rate of pun generation (**Pun Succ.**), following Mittal et al. (2022), we invite evaluators to classify each sentence into one of two categories: "pun" or "non-pun". We use the pairwise kappa coefficient to measure the inter-annotator agreement (IAA). The average inter-annotator agreement of all evaluators for humor capability and pun success are 0.47 and 0.58, meaning that annotators moderately agree with each other. Details of the human evaluation are provided in Appendix D.

6.3 Baselines

AmbiPun AmbiPun (Sun et al., 2022b) is the state-of-the-art pun generation model fine-tuned on T5 (Raffel et al., 2023), which utilizes "generate sentence: $\{p_w\}$, $\{a_w\}$" for homophonic puns and "generate sentence: $\{p_w\}$" for homographic puns as the prompt. To adapt to the Chinese pun generation, we fine-tune it on ChinesePun dataset.

LLMs **LLaMA2-7B** and **Baichuan2-7B** are used for generating English and Chinese puns respectively. We fine-tune these models through the standard DPO, denoted by $\mathbf{LLaMA2_{dpo}}$ and $\mathbf{Baichuan2_{dpo}}$ and instruction tuning denoted by $\mathbf{LLaMA2_{sft}}$ and $\mathbf{Baichuan2_{sft}}$ as the baseline models. For the standard DPO method, we select the negative samples with humor discriminator and structure discriminator simultaneously as in Section 4, and choose the labeled puns from pun datasets as the positive samples. Besides, we also choose **ChatGPT** as a baseline, which is used to generate puns through a few-shot prompt as shown in Appendix A.

6.4 Experiment Setup

The ChatGPT model we use is *gpt-3.5-turbo*.The LLaMA2-7B and Baichuan2-7B are acquired from huggingface Transformers[5]. We adopt AdamW optimizer (Loshchilov and Hutter, 2017), set learning rate as 5e-5 and batch size as 4. We employ the LoRA strategy (Hu et al., 2021) for fine-tuning the LLMs. For the hyperparameter in DPO training, we set $\beta = 0.5$. The temperature is set to 0.95, the top-p is set to 0.95, and the top-k is set to 5 in the decoding strategy. For the preference data sampling, we require the number of preference data pairs or triplets to reach 10,000. The details of preference data sampling can be found in Appendix B. The entire project is based on the LLaMA-Factory[6], and all other settings are default parameters. All our experiments are performed on 2 RTX 3090.

[5] https://github.com/huggingface/transformers
[6] https://github.com/hiyouga/LLaMA-Factory

Dataset	Model	Avg-Length	Corpus-Div %		Sentence-Div %		Humor A/B-test vs. ChatGPT %		Structure Succ. %	Pun Succ. %
			Dist-1	Dist-2	Dist-1	Dist-2	Win	Lose		
ChinesePun	ChatGPT	53.94	**5.51**	**41.88**	**77.87**	**95.17**	—	—	**93.07**	28.00
	AmbiPun	79.31	2.68	15.50	64.50	85.06	28.00	42.00	53.84	16.00
	Baichuan2-7B	91.12	3.00	26.92	70.84	92.36	18.00	44.00	78.69	14.00
	$\text{Baichuan2}_{\text{sft}}$	85.57	3.16	30.94	62.61	85.27	20.00	46.00	74.30	16.00
	$\text{Baichuan2}_{\text{dpo}}$	79.26	3.22	30.25	60.04	82.74	30.00	52.00	80.41	20.00
	PGCL (ours)	85.20	2.78	27.52	58.73	81.39	**64.00**	**22.00**	89.10	**44.00**
	Human	88.94	1.85	30.19	65.92	89.13	—	—	87.01	—
SemEval	ChatGPT	16.35	24.29	68.69	92.64	99.78	—	—	91.42	34.00
	AmbiPun	14.0	21.50	64.96	91.15	98.75	45.00	48.00	94.12	30.00
	LLaMA2-7B	59.95	9.56	32.72	79.99	98.46	32.00	36.00	89.12	22.00
	$\text{LLaMA2}_{\text{sft}}$	11.71	24.84	66.49	**95.84**	99.76	32.00	38.00	84.52	22.00
	$\text{LLaMA2}_{\text{dpo}}$	43.65	13.90	49.31	89.36	98.09	42.00	34.00	94.56	28.00
	PGCL (ours)	22.51	20.09	61.64	94.26	99.19	**68.00**	**14.00**	**98.95**	**56.00**
	Human	11.66	**31.76**	**79.05**	95.36	**99.86**	—	—	86.82	—

Table 4: The results of pun generation. The boldface denotes the best performance and the underline denotes the second-best performance among systems. The paired t-test shows that the difference between our model and the baseline methods is statistically significant ($p < 0.05$).

6.5 Results

Table 4 displays a comparison of results between our model and baselines, from which we can get the following conclusions. First, our method achieves the new SOTA performance with substantial improvements on both pun datasets, which proves the superiority and generalization of our approach (We show the examples of generated puns in Appendix E). Specifically, when considering the pun success rate, it obtains 16% and 22% improvements over the best results of baselines on ChinesePun and SemEval datasets. In terms of humor degree and structure success rate, it also performs consistently better than most other models. Nevertheless, when considering the diversity of the generated sentences, ChatGPT exhibits a notable increase in diversity compared to other models. This can be attributed to the fact that, unlike ChatGPT, the other models have been fine-tuned on the specific dataset. Second, it can be observed that $\text{Baichuan2}_{\text{dpo}}$ and $\text{LLaMA2}_{\text{dpo}}$ outperforms $\text{Baichuan2}_{\text{sft}}$ and $\text{LLaMA2}_{\text{sft}}$ respectively, which proves that preference alignment supports the LLMs to obtain the specific capability (e.g., structure or humor) more effectively than instruction tuning. Third, compared with the $\text{Baichuan2}_{\text{dpo}}$ and $\text{LLaMA2}_{\text{dpo}}$, the most significant improvement of our approach is utilizing multi-stage curriculum learning to optimize the two preference targets separately from easy to hard, which indicates the effectiveness of our method.

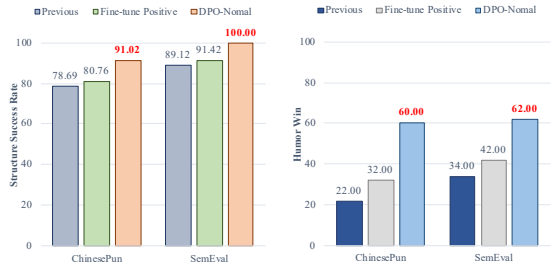
6.6 Ablation Study

To evaluate the effects of different components, we compare PGCL with its variants: 1) w/o Improved Humor DPO. In this variant, we use the standard DPO algorithm instead of the improved humor DPO algorithm in the humor preference alignment process; 2) w/o Humor. In this variant, we remove the humor preference alignment stage, i.e., only learn the structure preference. We intend to explore whether using the new triplet alignment loss for humor preference optimization contributes to alleviating the effect of catastrophic forgetting.

From Table 5, we can observe that our PGCL method consistently exhibits better performance than their corresponding variants across both ChinesePun and SemEval datasets. Specifically, the structure success rate decreases by 11.02% and 8.36% respectively compared with LLMs tuned by the standard DPO algorithm in the second stage, which demonstrates that the standard DPO algorithm can not maintain good structure preferences while performing humor preference alignment. Moreover, the novel triplet loss brings significant performance improvements (i.e., PGCL vs. PGCL w/o Improved Humor DPO), with about 8.64% and 7.31% gains in structure success rate on both datasets. The improvement in satisfying pun structure further increases the pun success rate by about 6% and 10%. It proves that the new triplet alignment loss can guide LLMs to keep a balance between structure and humor preference.

Dataset	Model	Avg-Length	Corpus-Div %		Sentence-Div %		Humor A/B-test vs. ChatGPT %		Structure Succ. %	Pun Succ. %
			Dist-1	Dist-2	Dist-1	Dist-2	Win	Lose		
ChinesePun	PGCL (ours)	85.20	2.78	27.52	58.73	81.39	**64.00**	**22.00**	89.10	**44.00**
	w/o Improved Humor DPO	86.94	2.30	22.96	55.51	78.39	60.00	32.00	80.46	38.00
	w/o Humor	88.28	**2.82**	**27.73**	**59.94**	**83.06**	22.00	48.00	**91.02**	26.00
SemEval	PGCL (ours)	22.51	20.09	**61.64**	94.26	**99.19**	**68.00**	**14.00**	98.95	**56.00**
	w/o Improved Humor DPO	29.15	20.68	59.55	93.02	97.64	62.00	24.00	91.64	46.00
	w/o Humor	17.39	**24.11**	61.39	**95.45**	98.76	34.00	44.00	**100.00**	26.00

Table 5: Architecture ablation analysis on SemEval and ChinesePun dataset



(a) Structure Preference (b) Humor Preference

Figure 2: The impact of using DPO or SFT method in two preference alignment processes on ChinesePun and SemEval datasets.

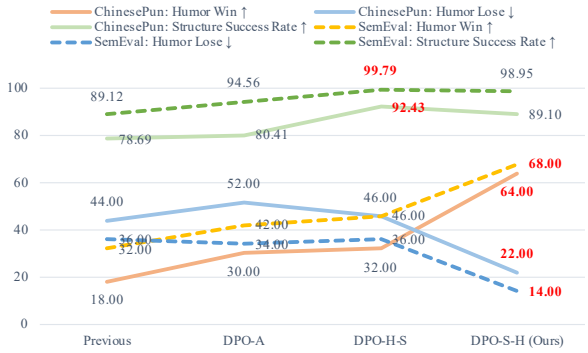


Figure 3: The performance of different strategies of multi-objective preference alignment.

6.7 Further Discussions

At the very beginning, we set up a "**Previous**" category in the following discussions to represent the model from the previous stage, which serves as the basis for the comparison experiments.

DPO vs. instruction tuning Figure 2 illustrates the performance difference between continuing to use instruction tuning and standard DPO method based on the previous stage model. The standard DPO utilizes the (positive, negative) data pairs, and the instruction tuning only employs positive samples from the data pairs. Both Figure 2(a) and Figure 2(b) prove that the DPO method significantly improves the performance of LLMs compared to instruction tuning.

Is Curriculum Learning Effective? Figure 3 illustrates the performance of different strategies to enhance a model's ability of multi-objective preference alignment on ChinesePun and SemEval datasets. DPO-A involves simultaneous optimization of the model's structural and humorous capabilities. DPO-H-S signifies prioritizing the optimization of the model's humorous capability, followed by enhancing its ability to satisfy pun structures. DPO-S-H (Ours) follows the opposite sequence. Both DPO-H-S and DPO-S-H use the new triplet alignment loss in the second stage. The details of DPO-H-S can be found in Appendix C. We can see that the success rate of the pun structure, denoted by green lines, peaks at DPO-H-S, while the humor degree, illustrated by blue and orange lines, attains its optimum at DPO-S-H. It indicates that LLMs tend to prioritize the final stage in multi-stage preference learning. However, DPO-H-S shows only slight improvements in pun structure success rates over DPO-S-H, with gains of 3.33% and 0.84% on the ChinesePun and SemEval datasets, respectively. In contrast, switching from DPO-H-S to DPO-S-H results in a substantial increase in humor win rates by 32% and 22% along with a significant decrease in humor loss rates, by 24% and 22%, respectively. It indicates that multi-stage curriculum learning enables LLMs to concentrate on hard tasks without compromising their performance. DPO-A does not enhance the model's performance significantly, as the inherent difficulty of directly generating puns makes it challenging for the model to optimize in that direction immediately.

7 Conclusion

In this work, we propose a multi-stage curriculum learning approach with the Direct Preference Optimization technique to effectively endow LLMs with the capability of generating humorous puns.

Specifically, to alleviate the effects of catastrophic forgetting in the multi-objective preference alignment, we present a novel triplets preference learning schema, which is quite different from standard DPO. To verify the broad applicability of our method in different languages, we construct a new benchmark dataset, called ChinesePun, which is the very first dataset on the Chinese pun generation task. The evaluation shows that our method significantly outperforms all the existing methods in Chinese and English, even for ChatGPT.

8 Limitations

In our research, we concentrate on pun generation, a niche within creative language and humor production. However, our method still fails in recognizing humor's subjective nature. Due to diverse backgrounds and experiences, what is funny to one individual might not be to another. In our future work, we will aim at offering broader insights into humor's variability, influenced by contextual subtleties and personal interpretations.

9 Acknowledgement

The authors thank all the anonymous reviewers for their valuable comments and constructive feedback. This work was supported by Ant Group Research Fund & the Science and Technology Commission of Shanghai Municipality Grant (No. 22511105901).

References

Issa Annamoradnejad and Gohar Zoghi. 2022. Colbert: Using bert sentence embedding in parallel neural networks for computational humor.

Karyn Buxman. 2008. Humor in the or: a stitch in time? *AORN journal, 88(1), 67–77*.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.

Yuyan Chen, Zhixu Li, Jiaqing Liang, Yanghua Xiao, Bang Liu, and Yunwen Chen. 2023. Can pre-trained language models understand chinese humor? In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, WSDM '23, page 465–480, New York, NY, USA. Association for Computing Machinery.

Delia Chiaro. 2006. *The language of jokes: Analyzing verbal play*. Routledge.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668, Online. Association for Computational Linguistics.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, and Ziqing Yang. 2021. Pre-training with whole word masking for chinese bert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3504–3514.

Joseph L. Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and Psychological Measurement*, 33:613 – 619.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.

He He, Nanyun Peng, and Percy Liang. 2019. Pun generation with surprise. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1734–1744, Minneapolis, Minnesota. Association for Computational Linguistics.

Bryan Anthony Hong and Ethel Ong. 2009. Automatically extracting word relationships as templates for pun generation. In *Proceedings of the Workshop on Computational Approaches to Linguistic Creativity*, pages 24–31, Boulder, Colorado. Association for Computational Linguistics.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Sophie Jentzsch and Kristian Kersting. 2023. Chatgpt is fun, but it is not funny! humor is still challenging large language models.

Tianqi Liu, Yao Zhao, Rishabh Joshi, Misha Khalman, Mohammad Saleh, Peter J Liu, and Jialu Liu. 2023. Statistical rejection sampling improves preference optimization. *arXiv preprint arXiv:2309.06657*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Fuli Luo, Shunyao Li, Pengcheng Yang, Lei Li, Baobao Chang, Zhifang Sui, and Xu Sun. 2019. Pun-GAN: Generative adversarial network for pun generation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3388–3393, Hong Kong, China. Association for Computational Linguistics.

Tristan Miller and Iryna Gurevych. 2015. Automatic disambiguation of English puns. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 719–729, Beijing, China. Association for Computational Linguistics.

Tristan Miller, Christian Hempelmann, and Iryna Gurevych. 2017. SemEval-2017 task 7: Detection and interpretation of English puns. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 58–68, Vancouver, Canada. Association for Computational Linguistics.

Anirudh Mittal, Yufei Tian, and Nanyun Peng. 2022. AmbiPun: Generating humorous puns with ambiguous context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1053–1062, Seattle, United States. Association for Computational Linguistics.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Saša Petrović and David Matthews. 2013. Unsupervised joke generation from big data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 228–232, Sofia, Bulgaria. Association for Computational Linguistics.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Alessandra Cervone, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022a. ExPUNations: Augmenting puns with keywords and explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4590–4605, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiao Sun, Anjali Narayan-Chen, Shereen Oraby, Shuyang Gao, Tagyoung Chung, Jing Huang, Yang Liu, and Nanyun Peng. 2022b. Context-situated pun generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4635–4648, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yufei Tian, Divyanshu Sheth, and Nanyun Peng. 2022. A unified framework for pun generation with humor principles. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3253–3261, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models.

Alessandro Valitutti, Hannu Toivonen, Antoine Doucet, and Jukka M. Toivanen. 2013. "let everything turn well in your wife": Generation of adult humor using lexical constraints. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 243–248, Sofia, Bulgaria. Association for Computational Linguistics.

Canwen Xu, Corby Rosset, Luciano Del Corro, Shweti Mahajan, Julian McAuley, Jennifer Neville, Ahmed Hassan Awadallah, and Nikhil Rao. 2023. Contrastive post-training large language models on data curriculum. *arXiv preprint arXiv:2310.02263*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao,

Hang Xu, Haoze Sun, Hongda Zhang, Hui Liu, Jiaming Ji, Jian Xie, JunTao Dai, Kun Fang, Lei Su, Liang Song, Lifeng Liu, Liyun Ru, Luyao Ma, Mang Wang, Mickel Liu, MingAn Lin, Nuolan Nie, Peidong Guo, Ruiyang Sun, Tao Zhang, Tianpeng Li, Tianyu Li, Wei Cheng, Weipeng Chen, Xiangrong Zeng, Xiaochuan Wang, Xiaoxi Chen, Xin Men, Xin Yu, Xuehai Pan, Yanjun Shen, Yiding Wang, Yiyu Li, Youxin Jiang, Yuchen Gao, Yupeng Zhang, Zenan Zhou, and Zhiying Wu. 2023. Baichuan 2: Open large-scale language models.

Zhiwei Yu, Jiwei Tan, and Xiaojun Wan. 2018. A neural approach to pun generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1650–1660, Melbourne, Australia. Association for Computational Linguistics.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876, Online. Association for Computational Linguistics.

Yao Zhao, Misha Khalman, Rishabh Joshi, Shashi Narayan, Mohammad Saleh, and Peter J Liu. 2022. Calibrating sequence likelihood improves conditional language generation. *arXiv preprint arXiv:2210.00045*.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences.

A Prompt Template

Homophonic:
Given the pun word " p_w" and its homophonic alternative word " a_w", please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word " p_w" is present in the sentence.
Homographic:
Given the pun word " p_w", please generate a humorous homographic pun. Ensure that the sentence reflects the contexts of different meanings and make sure the pun word " p_w" is present in the sentence.

Figure 4: The prompt template used in `SemEval` dataset.

The prompt templats we used in SFT stage, structure preference optimization and humor preference optimization are as Figure 4 and Figure 5. For the few-shot ChatGPT, Figure 6 utilized identical prompts and incorporated three data examples as context.

B Preference Data Construction

Structure Preference Data As outlined in Section 4.1, we initially sample outputs from the SFT model using the same specifically designed prompt employed in the SFT step. Subsequently, these outputs are classified by a structural discriminator into groups of structurally compliant and non-compliant sentences. We selected 10,000 structure preference pairs randomly to do structural preference aligning.

Homophonic:
请为我编写一个幽默的句子，其中包含词语"p_w"和"a_w"。这个句子应该能够引发笑声或娱乐，并展示"p_w"和"a_w"这两个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。
Homographic:
根据上述词义，请为我编写一个幽默的句子，并使用"p_w"这个词在一个句子中一次或多次。这个句子应该能够引发笑声或娱乐，并展示"p_w"这个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。

Figure 5: The prompt template used in `ChinesePun` dataset.

Humor Preference Data We sampled outputs from the structurally optimized model and then categorized them into rumination and negative samples using the humor discriminator and structure discriminator. Concurrently, humor preference triplets are constructed based on the pun data corresponding to the used prompts, as well as the rumination and negative samples. We randomly selected 10,000 triplets for humor preference alignment.

C Details in Experiment of DPO-H-S

As the settings of our main experiment, DPO-H-S uses the standard DPO loss at stage 1 to optimize the humor preference. We generate candidate sentences from the SFT model and use the humor discriminator to select the negative samples as we described in Section 4.1. The positive samples we used are puns from pun datasets. we use these positive samples and negative samples to do the standard humor DPO.

Then, DPO-H-S uses the new triplet alignment loss at stage 2 to optimize the structure preference. We also generate candidate sentences from the humorous optimized LLM which in the first stage. Then, we use the humor discriminator to identify the humorous sentence from candidate sentences. These sentences are used as rumination samples. The humorless sentences from candidate sentences are further fed to the structure discriminator to choose the samples that do not fit the pun structure. These samples are used as negative samples. The

SemEval Homophonic:
Q: Given the pun word "staring" and its homophonic alternative word "stair", please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word "staring" is present in the sentence. A: \'Boy, I wish the elevator were working,\' said Tom, staring up to the top. Q: Given the pun word "doggedly" and its homophonic alternative word "dog", please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word "doggedly" is present in the sentence. A: "I\'ll never give up my hounds!" Tom said doggedly. Q: Given the pun word "handy" and its homophonic alternative word "hand", please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word "handy" is present in the sentence. A: I\'m glad I know sign language, it\'s pretty handy. Q: Given the pun word " p_w" and its homophonic alternative word " a_w", please generate a humorous homophonic pun. Ensure that the sentence reflects the contexts of both words and make sure the pun word " p_w" is present in the sentence. A:
SemEval Homographic:
Q: Given the pun word "landing", please generate a humorous homographic pun. Ensure that the sentence reflects the contexts of different meanings and make sure the pun word "landing" is present in the sentence. A: Careless stair dancers are heading for a heavy landing. Q: Given the pun word "register", please generate a humorous homographic pun. Ensure that the sentence reflects the contexts of different meanings and make sure the pun word "register" is present in the sentence. A: With certain cashiers, things are slow to register. Q: Given the pun word "date", please generate a humorous homographic pun. Ensure that the sentence reflects the contexts of different meanings and make sure the pun word "date" is present in the sentence. A: One palm tree said to another, \'Let\'s have a date.\' Q: Given the pun word " p_w", please generate a humorous homographic pun. Ensure that the sentence reflects the contexts of different meanings and make sure the pun word " p_w" is present in the sentence. A:
ChinesePun Homophonic:
Q：请为我编写一个幽默的句子，其中包含词语“父女”和“妇女”。这个句子应该能够引发笑声或娱乐，并展示“父女”和“妇女”这两个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：三月八日早上起床，女儿跑过来说：爸爸，节日快乐！ 我大惊：你个小丫头片子，搞什么鬼？ 女儿淡定地说：今天不是我俩过节吗？父女节呀！Q：请为我编写一个幽默的句子，其中包含词语“咬”和“摇”。这个句子应该能够引发笑声或娱乐，并展示“咬”和“摇”这两个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：生物课上，老师给学生讲解怎样区分狼与狗：狗会摇尾巴，而狼不会摇尾巴。 讲完后，老师走下讲台随手翻开一名学生的课堂笔记，只见上面写着：“狗会咬尾巴，而狼不会咬尾巴。”Q：请为我编写一个幽默的句子，其中包含词语“缺点”和“缺碘”。这个句子应该能够引发笑声或娱乐，并展示“缺点”和“缺碘”这两个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：爸爸要去逛街，儿子说：“爸爸，帮我买一根海带回来。”爸爸说：“你有什么缺碘？”儿子说：“我的游泳衣有缺点，太宽松，我想买一根海带绑住。”Q：请为我编写一个幽默的句子，其中包含词语“ p_w”和“ a_w”。这个句子应该能够引发笑声或娱乐，并展示“ p_w”和“ a_w”这两个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：
ChinesePun Homographic:
Q：请根据上述词义，请为我编写一个幽默的句子，并使用“死”这个词在一个句子中一次或多次。这个句子应该能够引发笑声或娱乐，并展示“死”这个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：很久以前上电脑课，有一排同学的电脑死机了。 一位同学站起来说：“老师，电脑死机了，我们这排全死了。”这时，许多同学都说：“我们也死了。”老师问：“还有谁没死？”只有一位同学站起来：“我还没死！”老师奇怪地说：“全班都死了，你为什么不死？Q：请根据上述词义，请为我编写一个幽默的句子，并使用“西北风”这个词在一个句子中一次或多次。这个句子应该能够引发笑声或娱乐，并展示“西北风”这个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：老师：夏天刮东南风，冬天刮西北风，请记住。 学生：不对，我妈说跟我爸结了婚，一年四季都喝西北风。Q：请根据上述词义，请为我编写一个幽默的句子，并使用“一块”这个词在一个句子中一次或多次。这个句子应该能够引发笑声或娱乐，并展示“一块”这个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：有个豆腐摊，旁边竖着块牌子：每斤八角。来了个顾客：“一块卖不卖？”卖主以为这人不识字，不禁有些高兴：“卖。”然后提起秤问道：”要几斤？”那人指着一小块：“要这一块。”Q：根据上述词义，请为我编写一个幽默的句子，并使用“ p_w”这个词在一个句子中一次或多次。这个句子应该能够引发笑声或娱乐，并展示“ p_w”这个词语在幽默语境下的创意用法。你可以尝试在句子中加入意想不到的情节或转折，以增加幽默效果。请确保句子仍然通顺、自然，并具有良好的语法和逻辑结构。A：

Figure 6: The few-shot prompt template of ChatGPT on the ChinesePun and SemEval dataset.

positive samples are puns from the pun datasets. we use these positive samples, rumination samples, and negative samples to do the structure DPO with the new triplet loss.

D Detailed Human Annotation and Evaluation Process

The evaluators are the same as in the dataset annotation because they have experience with puns. Our annotation process was carried out by a team composed of three postgraduates from language-related fields. Before the actual annotation process, the team underwent a training session to familiarize themselves with the tasks. The tasks included defining pun words, defining pun sentences, and locating the pun word from the pun sentence. Each sentence was annotated by all of them and any disagreements were discussed by the entire group. The final result was determined by the option of receiving the majority of votes. This process was aimed at maintaining the quality and consistency of the annotations. In human evaluation, we invited them to evaluate our experiments. In the human A/B-test for humor evaluation, we paired the sentences generated by models and asked annotators to choose the better sentence based on the humor. If the difference is not significant, a tie is allowed. In the evaluation of the success rate of pun generation, we asked them to classify each sentence into one of two categories: "pun" or "non-pun". The workers were paid 60 yuan per hour.

Dataset	p_w/a_w	Methods	Generated Puns
SemEval	cannily/cannes	AmbiPun	She cannily invested in the stock market.
		ChatGPT	The sly fox cannily prepared for the Cannes Film Festival.
		Ours	I like movies about film festivals because they're cannily made.
		Human	"I've been to a film festival in the south of France," said Tom cannily.
	faculty/faculty	AmbiPun	The university's faculty is renowned for its expertise in various fields of study.
		ChatGPT	The professor's faculties were sharpened after drinking a cup of strong coffee.
		Ours	Old deans never die; they just lose their faculties.
		Human	Old principles never die; they just lose their faculties.
ChinesePun	庙/喵	AmbiPun	喵在庙前悠然自得地打了个滚，仿佛在享受这片宁静的时光。
		ChatGPT	我在庙里点燃了香火，结果吸引来一只喵，它竟然跳上了神坛，像是在向神明祈福一样，我真是见证了喵生的奇迹啊！
		Ours	一日，一男青年到女子大学校门口一宾馆找服务员问："姑娘，你们这有房间吗？"姑娘说："我们这没有房间，但有个猫可以给你玩。"男青年说："行，那给我个喵吧。"姑娘把男青年带到庙里说："你到里边自己玩吧。"
		Human	美国人：我家的狗会说英语哩！ 中国人：它会说英语？ 美国人：是的，我问它，房屋最上层是什么？它会说'ROOF！'中国人：那不稀奇，我有小猫会说中文， 美国人：它会说中文？ 中国人：是的，每当我问它，'我们中国在哪里拜拜呢？'它会回答'庙！庙'
	共同语言/共同语言	AmbiPun	老师:你爸爸妈妈怎么一直吵架呢? 学生:我爸爸和我妈妈根本就没有共同语言。
		ChatGPT	尽管我们来自不同的文化背景，但足球成了我们之间的共同语言，让我们在比赛中建立了深厚的友谊。
		Ours	某男跟某女谈了两年多，快要谈婚论嫁时。女说："未来老公要跟我有'共同语言'才行啊！"某男想了想，说："那好办，我跟你讲英语吧！"
		Human	法官，我坚决要求离婚，我同我妻子根本没有共同语言。 那没关系，你们可以一同去找个翻译嘛。

Table 6: Example outputs of different models. The p_w and a_w are randomly selected.

E Examples of Generated Puns

A comparison between our model and the baselines is in Table 6. Our method is capable of generating humorous puns that adhere to the defined structure, akin to human creativity. In contrast, it proves challenging for other models to produce even basic humorous sentences, let alone high-quality humorous puns.