Disentanglement or Entanglement, which is Better for TST

徐 勝 [1] 鈴木 良弥 [2] 福本 文代 [2]
山梨大学大学院 医工農学総合教育部 [1] 総合研究部工学域 [2]
{g22dts03,ysuzuki,fukumoto}@yamanashi.ac.jp

Abstract

With the continuous breakthroughs in the capabilities of Transformer-based models, NLP research focused on language style, such as Text Style Transfer (TST), has gradually attracted more attention. Approaches for handling TST tasks can generally be categorized into two main strategies: disentanglement and entanglement. This paper proposes a method to construct two prompting pipelines based on these two strategies, utilizing Chain of Thought (CoT) and Large Language Models (LLMs). We investigate the performance of these pipelines on four TST sub-tasks and analyze their improvements compared to the baseline.

1 Introduction

The text style is an intuitive notion involving how something is mentioned [1]. TST, a subset of text generation tasks, aims to alter the style of a given text (e.g., a sentence) while preserving its style-independent content. Depending on the type of style being considered, TST can be viewed as a collection of sub-tasks, such as sentiment style transfer (SST), and formality style transfer (FST).

Approaches employing disentanglement or entanglement strategies represent the dominant paradigm and an intuitive solution in prior research on TST tasks. Here, the disentanglement strategy assumes that the style and content information in the source sentence can be decoupled. It then integrates the separated content with the target style to produce the desired sentence. In contrast, the entanglement strategy leverages the target style directly to guide the model's generation process. These representative works include seq2seq models trained from scratch on non-parallel dataset [2, 3], fine-tuning pre-trained language models by parallel dataset [4, 5, 6], and LLM-based prompting techniques [7, 8]. In seq2seq and fine-tuned models, disentanglement/entanglement predominantly focuses on manipulating the hidden states of input sentences. For instance, the seq2seq model employing the disentanglement strategy is trained to learn disentangled representations in the latent space. Similarly, under the entanglement strategy, the decoder integrates controllable style features with the representations of the source sentences to generate the target sentence.

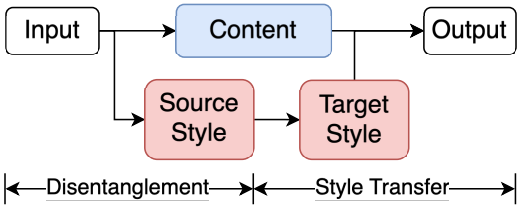
Although previous studies have demonstrated the effectiveness of their disentanglement or entanglement strategies through experimental results, a systematic investigation into which strategy is more effective remains an open problem. Furthermore, prior approaches have predominantly focused on employing a single strategy to develop specific methods, without capitalizing on the complementary advantages of integrating both strategies. Several innovative approaches have also been investigated, including methods leveraging Reinforcement Learning or attempts to examine the underlying transfer pattern from input to target [9, 10]. Nonetheless, these efforts have not emphasized disentanglement or entanglement strategies.

In this paper, to overcome the limitations mentioned above, we propose two CoT pipelines using LLMs, each of which is based on either the disentanglement or entanglement strategy. To comprehensively compare the performance and generalizability of each pipeline, we conduct experiments on four TST subtasks. The main contributions of our work are summarized as follows:

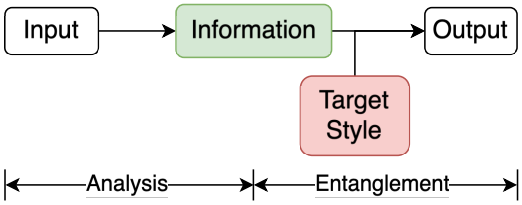
(1) We conducted a comparison of the performance of CoT prompting methods utilizing disentanglement and entanglement strategies.

(2) To fully harness the advantages of both strategies, we employ an LLM-based evaluation and reranking method, as proposed in [11], to ensemble the outputs from the two pipelines.

(3) Extensive experiments consistently demonstrate the effectiveness and generalizability of the various pipeline variants.



(a) Disentanglement pipeline



(b) Entanglement pipeline

Figure 1: Two overarching strategies for TST

2 Method

2.1 Constructing CoT Pipelines

To enhance the controllability and logical coherence of LLMs reasoning processes, we propose our two pipelines grounded in the CoT prompting [12], designed to ensure robust performance across a wide range of TST subtasks. Considering the prompting template can be directly constructed by Natural Language to define the expected transfer, each CoT pipeline consists of two steps of prompting which are designed by following the disentanglement and entanglement strategies shown in (a) and (b) of Figure 1, respectively. Let $\mathbf{X}$ indicate the input sentence with an original style $\mathbf{s}$ (e.g. "*negative*"). The target style is represented by $\mathbf{s}'$ (e.g. "*positive*"). The style to be transferred is referred to as $\mathbf{S}$ (e.g. "*sentiment*"). For the two steps of the disentanglement pipeline, we set the prompt templates as follows:

> **Disentanglement Prompt:** Here is a sentence "$\mathbf{X}$". Please analyze which part expresses $\mathbf{s}$, and which is $\mathbf{S}$-independent content.
>
> **Style Transfer Prompt:** Based on the analysis, please revise the sentence to transfer $\mathbf{s}$ content to $\mathbf{s}'$. while preserving the $\mathbf{S}$-independent content.

Similarly, the prompt templates for the entanglement pipeline are presented as follows:

> **Analysis Prompt:** Here is a sentence "$\mathbf{X}$". Please analyze the information conveyed in this sentence.
>
> **Entanglement Prompt:** Based on the analysis, please revise the sentence to express a more $\mathbf{s}'$.

To this end, the disentanglement and entanglement pipelines can be formalized as $P_{dis}(\mathbf{X})$ and $P_{ent}(\mathbf{X})$, respectively.

2.2 Ensembling Disentanglement and Entanglement

Considering the diversity of TST cases and the inherent flexibility of natural language, we assume that relying exclusively on either a disentanglement- or entanglement-based CoT pipeline may not be enough to handle all scenarios effectively. As depicted in Figure 2, the first input sentence can be easily decomposed into a content component, "*Ever since joes has changed hands it's just gotten* ", and a style component, "*worse and worse.*". However, the second input sentence presents challenges in explicitly separating content and style in natural language, as it expresses sentiment implicitly, and requires more advanced reasoning capabilities. In such cases, the entanglement-based pipeline may achieve better results.

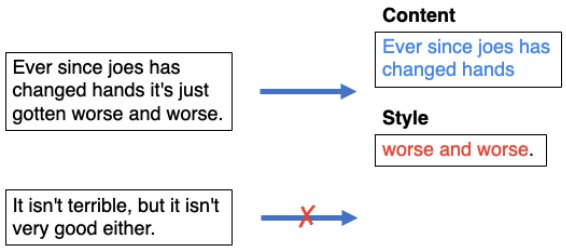


Figure 2: Two examples of SST.

To fully exploit the advantages of both CoT strategies, we adopt the re-ranking method proposed by [11]. Each generated candidate ($\mathbf{X}'$) will be evaluated with a score calculated by a specific function $\Phi(\mathbf{X}, \mathbf{X}')$. In our work, both CoT pipelines are applied for each input, and their outputs are subsequently evaluated with three scores, representing the strength of style transfer, content preservation, and fluency. All three scores are multiplied to get the $\Phi(\mathbf{X}, \mathbf{X}')$, as shown in Eq.(1). Different from [11], where these three scores are predicted by PLMs, we directly prompt LLM to assess each score on a regularized scale from 0 to 100 which is similar to [13].

$$\Phi(\mathbf{X}, \mathbf{X}') = \phi_s(\mathbf{X}, \mathbf{X}') \cdot \phi_c(\mathbf{X}, \mathbf{X}') \cdot \phi_f(\mathbf{X}, \mathbf{X}') \quad (1)$$

According to Eq.(2) the candidate with the higher

Table 1: Statistics of seven datasets for four TST subtasks

Task	Dataset	Size
SST	Yelp ($neg \rightarrow pos$)	500
	Yelp ($pos \rightarrow neg$)	500
	Amazon ($neg \rightarrow pos$)	500
	Amazon ($pos \rightarrow neg$)	500
FST	GYAFC	500
GST	JFLEG	747
AST	SHASP	599

$\Phi(\mathbf{X}, \mathbf{X}')$ is regarded as the final generation, $G(\mathbf{X})$.

$$G(\mathbf{X}) = \begin{cases} P_{dis}(\mathbf{X}), & \alpha \leq 0 \\ P_{ent}(\mathbf{X}), & \alpha > 0 \end{cases} \tag{2}$$

$$\alpha = \Phi(\mathbf{X}, P_{dis}(\mathbf{X})) - \Phi(\mathbf{X}, P_{ent}(\mathbf{X}))$$

3 Experiments

3.1 Experimental Setup

We conducted experiments on four TST subtasks, i.e., SST, FST, grammar style transfer (GST), and authorship style transfer (AST). The datasets, which have been cleaned by [11], used for these tasks are briefly explained as follows:

(1) **SST**. We choose the annotated Yelp and Amazon test datasets for the SST task [14], where both datasets include two subsets for transfer from negative to positive ($neg \rightarrow pos$) and vice versa ($pos \rightarrow neg$).

(2) **FST**. Following most of the related works, we use the GYAFC dataset collected to evaluate the performance of each variant for the FST task [15]. We focus on the transfer direction from informality to formality.

(3) **GST**. The last dataset we selected is JFLEG for the automatic grammatical error correction task [16]. We conducted the transfer from ungrammatical sentences to their grammatical counterparts.

(4) **AST**. For the AST task, we leverage a small subset of the dataset, proposed to translate the plays of Shakespeare to their counterparts written in modern English [17]. For convenience, the subset is named "SHASP".

Since Yelp and Amazon contain two subsets for $neg \rightarrow pos$ and $pos \rightarrow neg$ tasks, respectively, all other datasets involve single-directional transfer. In total, seven TST datasets are used across all experiments. The statistics of these datasets are shown in Tabel 1.

We explore four prompting variants: a straightforward prompt serving as the baseline, disentanglement CoT, entanglement CoT, and their ensembled configuration. The experiments for each variant on the above seven datasets are conducted by leveraging LLaMA3.2 as the backbone. The prompt templates, designed for interacting with LLMs to address each specific task, are detailed in our code[1)]. To obtain the most accurate scores, we select LLaMA3.3 with 70 billion parameters as the scoring evaluator which is prompted with three templates to implement the ϕ_s, ϕ_c, and ϕ_f, respectively. The scoring prompt examples are listed in Figures 3, 4, and 5. To focus on investigating the disentanglement and entanglement CoTs, all inferences are conducted in a zero-shot context. During each inference step, the main hyperparameters are the same as the default settings shown in Appendix, Table 3.

Five evaluation metrics are utilized to evaluate the performance of each prompt pipeline, including accuracy (Acc), reference-SacreBLEU score (r-sB), self-SacreBLEU score (s-sB), token-level perplexity (t-PPL), and sentence-level perplexity (s-PPL). Acc is the rate of the output with the target style and is used to measure the style transfer strength. Following previous work, we fine-tuned a standard BERT-base model with the style labels of sentences in each dataset to serve as a specific style classifier for every transfer subtask. s-sB and r-sB indicate the SacreBLEU scores between generation with the input and annotated reference, respectively, which are calculated by a tool[2)]. Here, s-sB evaluates the ability to preserve style-independent content, and r-sB measures the overall transfer performance. t-PPL and s-PPL represent the perplexities of the next token predicted by a specific language model (GPT2-large) to assess the fluency of the generated sentences. t-PPL is averaged over the number of tokens, while s-PPL is averaged over the number of sentences across the dataset. Instead of relying on human evaluation, we use the same LLaMA3.3 model to evaluate the performance on these three aspects. The pre-trained parameters of BERT-base and GPT2-large are downloaded from Huggingface[3)]. Likewise, all LLMs are set up by utilizing the Ollama[4)].

3.2 Results

Table 2 presents the performance of LLaMA3.2 across seven TST datasets. Comparing the results of the five

1) https://github.com/codesedoc/CoT4TST
2) https://github.com/mjpost/sacrebleu
3) https://huggingface.co/models
4) https://ollama.com

Table 2: Results of each pipeline across seven TST datasets by leveraging LLaMA3.2 as the backbone model. The **bold** font indicates the best scores among each subgroup.

Dataset	Pipeline	Acc ↑	r-sB ↑	s-sB ↑	t-PPL ↓	s-PPL ↓	Style ↑	Content ↑	Fluency ↑
Yelp ($neg \rightarrow pos$)	baseline	78.2	7.81	13.64	37	69	64.24	59.04	67.73
	disentanglement	76.2	**16.48**	**31.4**	47	99	67.84	62.85	70.42
	entanglement	76.2	8.81	14.85	**32**	**54**	63.38	57.43	66.19
	ensemble	**82.2**	11.83	20.62	38	79	**73.24**	**66.81**	**74.79**
Yelp ($pos \rightarrow neg$)	baseline	81.4	10.56	20.64	50	**98**	65.88	64.34	67.86
	disentanglement	76.6	**18.19**	**38.25**	65	165	60.3	60.16	63.1
	entanglement	84.8	10.89	21.01	**46**	**98**	64.28	62.14	66.75
	ensemble	**91.4**	15.62	29.65	55	116	**74.23**	**72.08**	**75.08**
Amazon ($neg \rightarrow pos$)	baseline	74.4	11.42	16.37	38	62	61.98	59.56	65.9
	disentanglement	74.6	**22.12**	**34.39**	51	105	64.54	62.43	68.66
	entanglement	77.8	9.81	15.14	**32**	**61**	65.57	57.91	67.59
	ensemble	**81.0**	14.11	21.66	37	65	**70.37**	**64.83**	**72.67**
Amazon ($pos \rightarrow neg$)	baseline	70.6	17.38	24.28	47	87	51.93	55.12	56.31
	disentanglement	62.8	**27.26**	**40.21**	61	127	45.52	50.63	50.77
	entanglement	**90.4**	14.82	21.21	**40**	**76**	62.23	56.17	63.37
	ensemble	84.6	19.79	27.81	46	81	**65.3**	**61.52**	**66.93**
GYAFC	baseline	**98.8**	7.48	4.65	30	50	81.73	**76.27**	**80.5**
	disentanglement	92.0	**13.31**	**15.18**	35	59	78.03	74.96	77.52
	entanglement	**98.8**	3.1	2.63	**25**	**39**	73.56	60.0	69.31
	ensemble	96.4	7.2	7.6	30	50	**81.75**	74.3	79.51
JFLEG	baseline	94.24	41.02	34.28	32	47	**79.87**	**79.73**	**85.57**
	disentanglement	87.68	**46.38**	**44.05**	40	77	71.77	73.41	77.33
	entanglement	**95.18**	23.57	19.78	**28**	**46**	64.89	62.8	74.83
	ensemble	92.1	41.75	37.74	33	53	77.62	78.16	84.52
SHASP	baseline	97.83	4.95	4.64	39	54	59.81	64.65	65.91
	disentanglement	88.15	**11.05**	**15.45**	60	95	61.8	67.42	67.45
	entanglement	**98.0**	4.32	4.39	**34**	**51**	49.15	53.54	57.18
	ensemble	94.82	8.72	10.51	47	72	**63.89**	**69.28**	**70.43**

automatic metrics across various pipeline variants reveals that the disentanglement strategy consistently achieves the highest r-sB and s-sB scores across all tasks. However, Acc scores are consistently lower than those of the baseline. In contrast, the entanglement strategy consistently surpasses the baseline in Acc scores and achieves the best t-PPL and s-PPL scores, although it performs less favorably in r-sB and s-sB. These findings suggest that the disentanglement CoT is particularly adept at decomposing sentence components and generating target sentences, while the entanglement CoT is more logically intuitive and excels at generating more natural sentences that align with the target style.

However, based on the LLM's scoring of the generated sentences, the variants generally exhibit consistent performance, either strong or weak, across the three dimensions of style, content, and fluency. Notably, aside from the results on the Amazon dataset, neither the disentanglement nor the entanglement strategy consistently outperforms the baseline. This discrepancy between the LLM-based evaluations and the automatic metric results requires further investigation.

A noteworthy finding is that, across all eight evaluation metrics, the ensemble variant achieves a more balanced trade-off between the disentanglement and entanglement pipelines. This results in improved performance over the baseline in terms of Acc, r-sB, and s-sB, while maintaining comparable perplexity scores. From the perspective of LLM-based evaluation, the ensemble variant even surpasses both CoT pipelines, demonstrating superior overall effectiveness and outperforming the baseline in most tasks.

4 Conclusion

In this paper, we focused on investigating the performance of the CoT prompting pipelines based on disentanglement and entanglement in comparison to the baseline. Inspired by the algorithm proposed by [11], we proposed an ensemble operation to trade off the performance of these two pipelines. The experimental results demonstrate the ensemble variant can achieve consistently better metrics results on different TST tasks.

Acknowledgements

This work is supported by JKA (2023M-401) and the Support Center for Advanced Telecommunications Technology Research (SCAT). The first author is supported by JST SPRING, Grant Number JPMJSP2133.

References

[1] David D. McDonald and James D. Pustejovsky. A computational theory of prose style for natural language generation. In Maghi King, editor, **Second Conference of the European Chapter of the Association for Computational Linguistics**, Geneva, Switzerland, March 1985. Association for Computational Linguistics.

[2] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 6008–6019, Florence, Italy, July 2019. Association for Computational Linguistics.

[3] Ning Dai, Jianze Liang, Xipeng Qiu, and Xuanjing Huang. Style transformer: Unpaired text style transfer without disentangled latent representation. In Anna Korhonen, David Traum, and Lluís Màrquez, editors, **Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics**, pp. 5997–6007, Florence, Italy, July 2019. Association for Computational Linguistics.

[4] Léo Laugier, John Pavlopoulos, Jeffrey Sorensen, and Lucas Dixon. Civil rephrases of toxic texts with self-supervised transformers. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, **Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume**, pp. 1442–1461, Online, April 2021. Association for Computational Linguistics.

[5] Xu Sheng, Fumiyo Fukumoto, Jiyi Li, Go Kentaro, and Yoshimi Suzuki. Learning disentangled meaning and style representations for positive text reframing. In C. Maria Keet, Hung-Yi Lee, and Sina Zarrieß, editors, **Proceedings of the 16th International Natural Language Generation Conference**, pp. 424–430, Prague, Czechia, September 2023. Association for Computational Linguistics.

[6] Sheng Xu, Yoshimi Suzuki, Jiyi Li, and Fumiyo Fukumoto. Decoupling style from contents for positive text reframing. In Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li, and Chaojie Li, editors, **Neural Information Processing**, pp. 73–84, Singapore, 2024. Springer Nature Singapore.

[7] Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. A recipe for arbitrary text style transfer with large language models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)**, pp. 837–848, Dublin, Ireland, May 2022. Association for Computational Linguistics.

[8] Jingxuan Han, Quan Wang, Zikang Guo, Benfeng Xu, Licheng Zhang, and Zhendong Mao. Disentangled learning with synthetic parallel data for text style transfer. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, **Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 15187–15201, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[9] Huiyuan Lai, Antonio Toral, and Malvina Nissim. Thank you BART! rewarding pre-trained models improves formality style transfer. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, **Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)**, pp. 484–494, Online, August 2021. Association for Computational Linguistics.

[10] Jingxuan Han, Quan Wang, Licheng Zhang, Weidong Chen, Yan Song, and Zhendong Mao. Text style transfer with contrastive transfer pattern mining. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, **Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)**, pp. 7914–7927, Toronto, Canada, July 2023. Association for Computational Linguistics.

[11] Mirac Suzgun, Luke Melas-Kyriazi, and Dan Jurafsky. Prompt-and-rerank: A method for zero-shot and few-shot arbitrary textual style transfer with small language models. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, **Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing**, pp. 2195–2222, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[12] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In **Proceedings of the 36th International Conference on Neural Information Processing Systems**, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[13] Phil Sidney Ostheimer, Mayank Kumar Nagda, Marius Kloft, and Sophie Fellenz. Text style transfer evaluation using large language models. In Nicoletta Calzolari, Min-Yen Kan, Veronique Hoste, Alessandro Lenci, Sakriani Sakti, and Nianwen Xue, editors, **Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)**, pp. 15802–15822, Torino, Italia, May 2024. ELRA and ICCL.

[14] Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[15] Sudha Rao and Joel Tetreault. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, **Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)**, pp. 129–140, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[16] Courtney Napoles, Keisuke Sakaguchi, and Joel Tetreault. JFLEG: A fluency corpus and benchmark for grammatical error correction. In Mirella Lapata, Phil Blunsom, and Alexander Koller, editors, **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**, pp. 229–234, Valencia, Spain, April 2017. Association for Computational Linguistics.

[17] Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. Paraphrasing for style. In Martin Kay and Christian Boitet, editors, **Proceedings of COLING 2012**, pp. 2899–2914, Mumbai, India, December 2012. The COLING 2012 Organizing Committee.

A Experiment Setting

Figures 3, 4, and 5 illustrate the prompt templates for LLM-based evaluation using the SST task ($neg \rightarrow pos$) as an example. The $[input]$, $[reference]$, and $[generation]$ in each figure represent the placeholders for each input sentence, its corresponding annotated reference, and the output generated by LLM, respectively. It is important to note that reference-related content is excluded during the ensemble operation, as the $[reference]$ is unavailable.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.
user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates "no sentiment transferred" while a score of 100 denotes "perfect sentiment transferred".
input sentence: $[input]$
human reference: $[reference]$
revised sentence: $[generation]$
Please only reply me the score.

Figure 3: Prompt template for evaluating the sentiment transfer strength.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.
user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates "no preservation of sentiment-independent content" while a score of 100 denotes "perfect preservation of sentiment-independent content".
input sentence: $[input]$
human reference: $[reference]$
revised sentence: $[generation]$
Please only reply me the score.

Figure 4: Prompt template for evaluating the capacity of preserving content.

system: You are a helpful assistant for evaluating the sentiment style transfer task. The definition of this task is to revise the input sentence to transfer negative content to positive while preserving the sentiment-independent content.
user: Evaluate the following transfer case relative to the human reference on a continuous scale ranging from 0 to 100 points. A score of 0 indicates "not fluent" while a score of 100 denotes "quite fluent".
input sentence: $[input]$
human reference: $[reference]$
revised sentence: $[generation]$
Please only reply me the score.

Figure 5: Prompt template for evaluating the fluency.

Table 3 presents the major hyperparameters used to configure the LLM for each prompting and evaluation process.

B Performance with other LLMs

To compare the performance of the two CoT pipelines and the baseline under different LLM settings, we conducted experiments on the Yelp dataset, focusing on the transfer from negative to positive, using six distinct LLMs including Gemma, Gemma2, LLaMA2, LLaMA3, LLaMA3.1, and LLaMA3.2. The results of these experiments on five automatic metrics are shown in Table 4. Similar to the finding in Table 2, the disentanglement strategy performs optimally in terms of r-sB and s-sB across different LLMs, while the entanglement strategy significantly achieves the best Acc, t-PPL, and s-PPL scores. This confirms that, compared to the baseline, the respective strengths and weaknesses of the disentanglement and entanglement strategies exhibit generalizability across different LLMs.

Table 3: Main hyperparameters for setting each LLM

Name	Range	Value
temperature	[0, 1]	0.8
top_p	[0, 1]	0.9
seed	-	42

Table 4: Results of each pipeline on Yelp ($neg \rightarrow pos$) dataset, by using six different LLMs. The **bold** font indicates the best scores among each subgroup.

Pipeline	Acc ↑	r-sB ↑	s-sB ↑	t-PPL ↓	s-PPL ↓
Gemma					
baseline	80.6	4.67	7.3	37	61
disentanglement	84.6	**9.06**	**15.23**	39	74
entanglement	**91.4**	5.32	7.16	**29**	**42**
Gemma2					
baseline	74.8	6.46	10.28	46	78
disentanglement	82.8	**15.04**	**28.12**	50	100
entanglement	**87.8**	3.82	6.71	**30**	**50**
LLaMA2					
baseline	83.2	7.56	12.73	35	**64**
disentanglement	74.2	**13.93**	**26.45**	49	92
entanglement	**86.2**	8.61	14.67	**32**	**64**
LLaMA3					
baseline	87.6	7.94	12.48	48	85
disentanglement	85.0	**17.62**	**31.59**	54	111
entanglement	**90.4**	7.47	12.23	**35**	**70**
LLaMA3.1					
baseline	78.0	9.86	17.26	45	87
disentanglement	78.4	**18.7**	**34.59**	57	117
entanglement	**80.0**	8.88	14.43	**33**	**64**
LLaMA3.2					
baseline	**78.2**	7.81	13.64	37	69
disentanglement	76.2	**16.48**	**31.4**	47	99
entanglement	76.2	8.81	14.85	**32**	**54**