ReactGPT: Understanding of Chemical Reactions via In-Context Tuning

Zhe Chen[1,2], Zhe Fang[1,2], Wenhao Tian[1,2], Zhaoguang Long[1], Changzhi Sun[3], Yuefeng Chen[4], Hao Yuan[4], Honglin Li[2*], Man Lan[1*]

[1]School of Computer Science and Technology, East China Normal University, Shanghai, China
[2]Innovation Center for Artificial Intelligence and Drug Discovery, East China Normal University, Shanghai, China
[3]Institute of Artificial Intelligence (TeleAI), China Telecom
[4]Shenzhen Transsion Holdings CO.,LTD.
{zhechen666, zhefang, wenhao_tian, zglong}@stu.ecnu.edu.cn
czsun@chinatelecom.cn, yuefeng.chen@transsion.com, hao.yuan@transsion.com
hlli@hsc.ecnu.edu.cn, mlan@cs.ecnu.edu.cn

Abstract

The interdisciplinary field of chemistry and artificial intelligence (AI) is an active area of research aimed at accelerating scientific discovery. Large language Models (LLMs) have shown significant promise in biochemical tasks, especially the molecule caption translation, which aims to align between molecules and natural language texts. However, existing works mainly focus on single molecules, while alignment between chemical reactions and natural language text remains largely unexplored. Additionally, the description of reactions is an essential part in biochemical patents and literature, and research on this aspect not only can help better understand chemical reactions but also promote research on automating chemical synthesis and retrosynthesis. In this work, we propose **ReactGPT**, a framework aiming to bridge the gap between chemical reaction and text. ReactGPT allows a new task: reaction captioning, by adapting LLMs to learn reaction-text alignment from context examples via In-Context Tuning. Specifically, ReactGPT jointly leverages a Fingerprints-based Reaction Retrieval module, a Domain-Specific Prompt Design module, and a two-stage In-Context Tuning module. We evaluate the effectiveness of ReactGPT on reaction captioning and experimental procedure prediction, both of these tasks can reflect the understanding of chemical reactions. Experimental results show that compared to previous models, ReactGPT exhibits competitive capabilities in resolving chemical reactions and generating high-quality text with correct structure.

Introduction

With the prosperity of large language models(LLMS), artificial intelligence technology is increasingly applied in a variety of fields(Wu et al. 2023; Xie et al. 2023; Dan et al. 2023; Ahn et al. 2024; Azerbayev et al. 2024; Rozière et al. 2024; Luo et al. 2024). Especially in the field of chemical molecules, the development of large language models has facilitated a lot of meaningful work, as molecules can be represented as Simplified Molecular-Input Line-Entry System (SMILES) strings (Weininger 1988; Weininger, Weininger, and Weininger 1989), which can be comprehended and generated by LLMs in a similar manner to natural languages. Edwards et al. (2022) proposed the task of translation between molecules and natural language, and put forward MolT5 to solve the task.

However, most of the previous work focused on molecules and ignored chemical reactions, they focus solely on the interaction between individual molecules and text (Edwards, Zhai, and Ji 2021a; Edwards et al. 2022; Li et al. 2023; Lu et al. 2022; Li et al. 2024), lacking an engagement with text from the perspective of chemical reactions. Liu et al. (2024b) focuses on reaction-text modeling, proposing the ReactXT method for pretraining large language models specifically for chemical reactions, and applying it to a downstream task of experimental procedure prediction. Additionally, the detailed description of chemical reactions is an essential part in biochemical patents and scientific literature. Therefore, we are committed to bridging chemical reactions and natural language text by proposing a new task: reaction captioning. As shown in Figure 1, the goal of reaction captioning is to generate a text caption describing the reaction process. Specifically, the description of a chemical reaction is a detailed account of a series of chemical changes, including the exact names of compounds, their molecular weights, and molar amounts. Additionally, it includes specific information about the reaction conditions, such as temperature range, pressure, and stirring. Moreover, the characterization of the product is also an essential part of the reaction description, typically encompassing spectroscopic data, such as NMR and mass spectrometry. These standardized descriptions are common practice in patents and scientific literature, enabling other researchers to accurately replicate the experimental results. Our proposed task would promote research in the scientific field by enabling chemical domain researchers to generate the experimental steps of a chemical reaction and have a better understanding of chemical reactions.

Though our proposed reaction-caption task is similar to the molecular caption task in some aspects, we also face inherent significant challenges. First, the description of chemical reactions is very long and complex, including many details of reactants and reaction conditions, so it is difficult to generate an accurate description. Second, the same chemical reaction can be described in different ways, leading to a situation where current evaluation metrics that rely on refer-

*Corresponding author

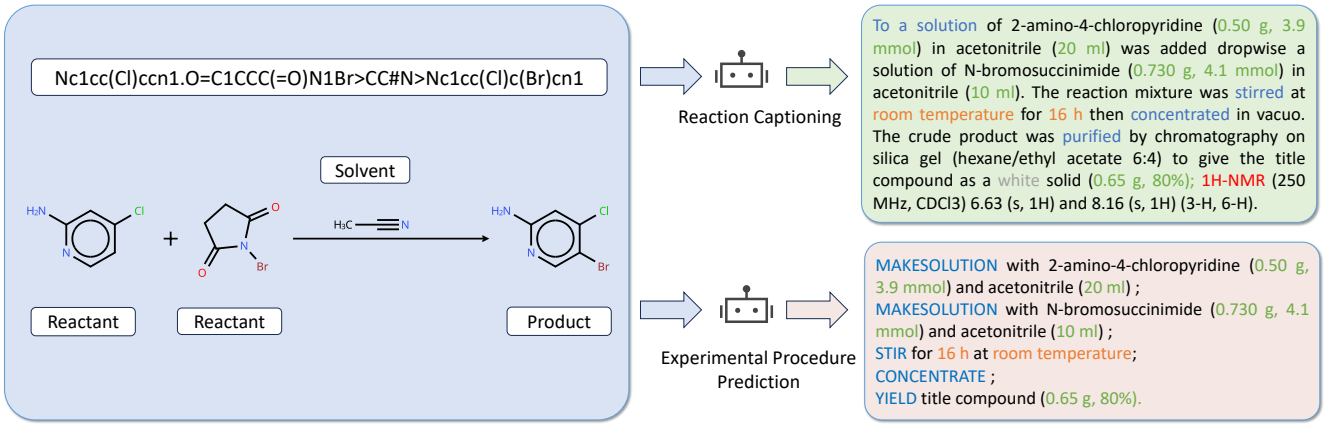

Figure 1: Illustration of Reaction Captioning and Experimental Procedure Prediction. The various elements are marked in different colors. For Experimental Procedure Prediction, each sequence begins with an action, and each action is predefined by Vaucher et al. (2021).

ence descriptions, such as BLEU (Papineni et al. 2002) and ROUGE (Lin 2004), are insufficient for properly assessing these tasks.

Motivated by the aforementioned factors and challenges, we propose ReactGPT, a framework aiming to bridge the gap between chemical reactions and natural language text by adapting LLMs. The purpose of ReactGPT is to leverage the domain knowledge from informative context examples via In-Context Tuning. Moreover, similar chemical reactions may share similar solvents, catalysts, or conditions, as indicated by the overlaps among reaction captions. Therefore, with ReactGPT, LLMs could leverage their inferential and in-context learning abilities to more effectively comprehend the correlation between chemical reactions and accompanying text captions as derived from contextual examples, thus delivering enhanced performance.

Specifically, ReactGPT includes the following components. First, ReactGPT adopts a Fingerprints-based Reaction Retrieval module, which can retrieve k similar reaction-caption pairs as context instances under the guidance of reaction similarity, that is, retrieval based on reaction fingerprint similarity. Second, ReactGPT integrates a Domain-Specific Prompt Design module, enabling strong prompt engineering capability from LLMs. Third, a two-stage In-Context Reaction Tuning is adopted to adapt the parameters of LLMs. We first fine-tune our model on the constructed prompts that contain contextual examples. Additionally, inspired by the paradigm of direct preference optimization (DPO), we construct negative samples to form the preference pair with the ground truth, applying the DPO algorithm to let our model learn the reaction caption structure, which is extremely important for the description of chemistry reaction.

In summary, our main contributions are as follows:

- We propose a new task: chemical reaction captioning, which needs to generate a description of the chemical reaction process, including a detailed account of a series of chemical changes.
- We propose ReactGPT, a framework that improves the performance of LLMs in chemical reaction captioning task. ReactGPT integrates three modules: Fingerprints-based Reaction Retrieval, Domain-Specific Prompt Design, and Two-stage In-Context Reaction Tuning.
- We conduct the experiments on two tasks: reaction captioning and experimental procedure prediction, and the results show that our method achieves state-of-the-art performances, which enables LLMs to better align between chemical reactions and texts.

Related Work

Large Language Models for Chemistry

The advent of large language models (LLMs) has unlocked novel opportunities across the scientific domain, accompanied by the creation of various new benchmarks (Lu et al. 2022; Chen et al. 2023). As an essential and challenging part of scientific domains, research in the chemistry domain is booming with the application of LLMs (Fang et al. 2023; Tang et al. 2024; Liao et al. 2024). Specifically, ChemDFM Zhao et al. (2024) proposes ChemDFM, which is considered as the first large language model towards Chemical General Intelligence and is trained on 34B tokens from chemical literature, textbooks, and instructions along with a variety of data from general domain. ChemCrow (Bran et al. 2023) integrates multiple existing tools for chemistry with LLMs to address a variety of downstream tasks. LLMs are also utilized to enhance the performance of specific chemical applications, such as drug editing (Liu et al. 2024a), reaction prediction (Qian et al. 2023; Zhong et al. 2023), and molecule-text translation (Edwards et al. 2022). Moreover, Guo et al. (2024) proposes MolTailer, which generates molecular representations for specific tasks via text prompts. Most recently, Liu et al. (2024b) proposes ReactXT, a method that explores reaction-text modeling, facilitating reaction-relevant tasks with a text interface and textual knowledge. Here we take one step further to propose a

new task, reaction captioning, to better align between chemical reactions and natural language, which can facilitate understanding of chemical reactions for domain-specific researchers.

Molecule-Text Translating

Inspired by the image captioning task, Edwards, Zhai, and Ji (2021b) proposed a new dataset, ChEBI-20, which contains 33,010 pairs of molecules and captions describing the molecular properties. After that, Edwards et al. (2022) proposed two innovative tasks: molecule captioning and text-guided de novo molecule generation. These tasks aim at translating between molecular representations and natural language texts. MolXPT (Liu et al. 2023c) pretrains a GPT model by leveraging literature annotations of molecules, which demonstrate better molecule-text alignment. Additionally, MolReGPT (Li et al. 2023) employs in-context learning that enables LLMs to learn the molecule-caption translation task from the context examples with a parameter-free scheme. Most recently, ICMA (Li et al. 2024) proposes the In-context Molecule Adaptation, which fine-tunes LLMs with informative contextual examples for better alignment between molecular representations and texts. Moreover, Chen et al. (2024) incorporate LLMs for low-resource molecule discovery, by proposing the first artificially-real dataset for molecule-caption translating task.

In-Context Learning

Recently, in-context learning has emerged as a promising approach to enhance the performance of large language models (LLMs), including a few input-output examples in the model's context as a precedent (Dong et al. 2022). LLMs can solve various tasks without updating any model's parameters by utilizing the capability of ICL. GPT-3 has shown this characteristic, since it can exhibit performance on unseen tasks that is comparable to that of fine-tuned models with few shot examples as input prefix (Brown et al. 2020). Moreover, an effective way to enhance the In-Context Learning (ICL) capabilities of pre-trained models is to fine-tune them with the addition of some labeled context examples before the target input. For instance, Chen et al. (2021) proposed a method of in-context tuning, which meta-trains the Language Model with a few examples to learn how to adapt to new tasks.

Task Definition

Inspired by image captioning and molecule captioning (Edwards et al. 2022), we propose a new task: chemical reaction captioning. For any given chemical reaction, the purpose of reaction captioning is to describe the chemical reaction process and details. However, reaction captioning can be a little more complicated than image captioning and molecule captioning, due to the description of the reaction includes quite a few technical terms and details such as reaction conditions. As shown in Figure 1, the description of chemical reaction at least contains the stoichiometric amount of compounds (*e.g.,* mass and molar amount), the action operation (*e.g.,* solution, stir, concentrated), the reaction condition (*e.g.,* temperature, time), the information of the product (*e.g.,* yield, conversion rate). A high-quality reaction text should at least contain the following elements: $(m_i, c_i, r_i), i \in D$, where m_i represents the compounds, c_i represents the conditions, such as temperature and time, r_i represents the results of the reaction, such as the yield and spectroscopic data of the products.

Generally speaking, molecules are represented as SMILES strings (Weininger 1988; Weininger, Weininger, and Weininger 1989), as it precisely translates molecules' chemical structures into a text string of atomic symbols and chemical bonds based on a set of rules. As for the input of our task, a chemical reaction is formed by the SMILES strings of the compounds. Specifically, the molecules within the same class were separated by dots ("."), while the reactants, catalysts/solvents, and product lists were separated by ">".

Methodology

In this section, we introduce ReactGPT as a novel framework to adapt LLMs to chemical reaction captioning. As shown in Figure 2, ReactGPT consists of three components, including Fingerprints-based Reaction Retrieval, Domain-Specific Prompt Design and Two-stage In-Context Tuning. Specifically, the Reaction Retrieval module first retrieves K examples from the training dataset D by calculating the similarity between the current query reaction and other reaction-caption pairs. After that, the Domain-Specific Prompt Design module aims to generate the domain-specific prompts and concatenate them with the input chemistry reactions to request responses from LLMs. Finally, the multi-stage In-Context Tuning first fine-tunes LLMs to learn the reaction-text alignment from the context examples, and then aligns structure preference with the Direct Preference Optimization (DPO) algorithm.

Fingerprints-based Reaction Retrieval

In order to make better use of the domain knowledge, we propose the Fingerprints-based Reaction Retrieval module, since there exists rich domain knowledge in a retrieval database, this module will retrieve the similar reaction and its corresponding description, and inject them into the text prompt. Generally speaking, the effective amount of information in context examples is closely related to the quality of retrieval. Thus, random examples may provide insufficient knowledge regarding the associations between reactions and natural language, as they fail to provide useful information for the detailed descriptions of reaction conditions and specific operations. Specifically, if the retrieval reactions are more similar to the current query chemical reaction, they may show more overlap in their respective captions, which allows for better alignment between reaction and texts. Therefore, we propose a retrieval method to select the context examples, which can help complement the lack of task-specific knowledge in LLMs.

In our tasks, ReactGPT adopts reaction fingerprints-based retrieval, which could better refine the quality of retrieval. However, traditional chemical reaction fingerprints are generated based on structural characteristics of chemi-

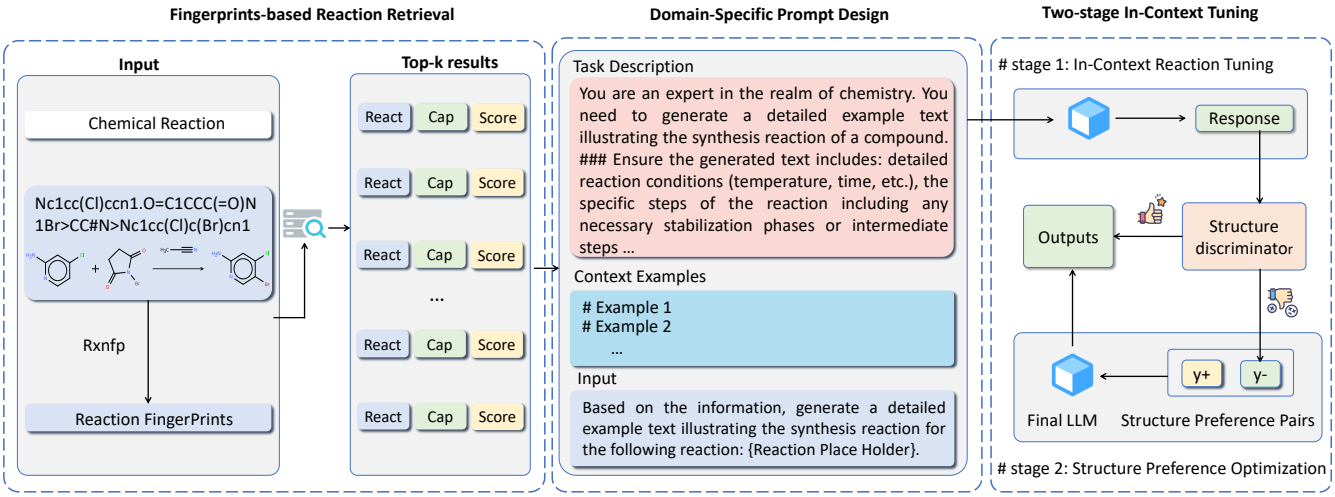

Figure 2: Framework of ReactGPT. Generally, ReactGPT consists of three components, Fingerprints-based Reaction Retrieval, Domain-Specific Prompt Design and Two-stage In-Context Tuning.

cal molecules and known chemical rules to produce fixed-length bit vectors or binary strings, which may have limited capabilities in understanding complex and unseen reactions due to their constrained generalization ability. To address this challenge, we apply pre-trained language models, such as BERT (Devlin et al. 2018), to get reaction fingerprints. Reaction fingerprints are vector encodings of chemical reactions, which not only capture features of the atoms and functional groups involved in the reaction but also encode their changes. Moreover, reaction fingerprints can be used for various aspects, such as similarity searching, yield prediction, and reaction classification. In ReactGPT, we adopt Rxnfp (Schwaller et al. 2021) as the reaction encoder, which is pre-trained and fine-tuned on 2.6 million reactions.

After that, we use cosine similarity to evaluate the similarity between the current query reaction x and other reactions x_i in the training set D. Mathematically, the reaction similarity ranking function can be represented as:

$$R(x) = \underset{\{x_1,\ldots,x_N\}\subseteq D}{argmax} \sum_{i=1}^{N} cos(\mathbf{e}_x, \mathbf{e}_{x_i}) \quad (1)$$

where $\mathbf{e}_x$ and $\mathbf{e}_{x_i}$ denotes the embeddings of the given reactions, which is the reaction fingerprints.

Domain-Specific Prompt Design

ReactGPT is proposed to solve the challenging problem: improving the generalization capability of fine-tuned LLM for reaction captioning task. To address this problem, we propose the Domain-Specific Prompt Design module, since prompt design or prompt engineering (Liu et al. 2023a) has been proven to be an effective way to generalize well-trained LLMs to various NLP downstream tasks. Following the standard instruction tuning protocol, the reaction instruction set can be generated by the following text prompt template $T = \{P, C, I\}$, which consists of the following three parts:

- P**: Task Description** aims to help LLMs identify the role of experts in the chemical task and ensure that LLMs clearly understand the specific task they are required to complete by providing a thorough explanation of the task's content. Moreover, it contains essential explanations to illustrate specific terms or concepts unique to the task of reaction captioning.
- C**: Context Examples** provides several reaction-caption pairs which are similar to the query reaction, retrieving from the Fingerprints-based Reaction Retrieval module. These context examples enable LLMs to utilize the information in reaction-caption pairs to generate better results.
- I**: Input** contains not only the SMILES string of the query reaction but also the IUPAC name of each compound, hence LLMs could focus on learning details of chemical reaction, such as conditions, action operation, and yield.

Two-stage In-Context Tuning

In this part, we introduce our fine-tuned method to adapt LLMs for reaction captioning task, which consists of two stages: In-Context Tuning and Structure Preference Alignment.

In-Context Tuning Inspired by in-context learning, in-context tuning optimizes pre-trained LMs with the few-shot in-context learning objective (Brown et al. 2020; Chen et al. 2021; Li et al. 2024). LLMs are trained using in-context tuning, where context examples are concatenated with the input. ReactGPT first employs the Fingerprints-based Reaction Retrieval to obtain the context examples C, which contains k similar reaction-captioning pairs $\{(x_i, y_i)|1 \leq i \leq k\}$ from the training set. If the context examples are more similar to the current query chemical reaction, they may show more overlap in their respective captions, thus contributing the final prediction. For example, the reaction conditions and operations of the same or similar reactants

are similar. Given the prompt template T, it's easy for us to construct the In-Context Tuning set S_D by applying T on each query chemical reaction. Then, we fine-tune a pre-trained LLM by optimizing the following training loss:

$$\mathcal{L}_{ICT}(\theta) = \sum_{(x,y)\in S_D} [-\log f_\theta(y|x, C, P)] \quad (2)$$

Here, $\mathcal{L}_{ICT}(\theta)$ represents the overall loss. f_θ is a pre-trained LLM with parameter θ, and we initialize f_θ as LLaMA3-8B-Instruct[1]. ReactGPT enables LLMs to learn the alignment between reactions and natural language in a more comprehensible way by learning the context examples and their corresponding associations.

Structure Preference Alignment Since the captions of chemical reactions are typically lengthy and complex, it is crucial for them to possess the necessary components. In this stage, we optimize the structure preference of LLMs with the DPO algorithm. Given a chemical reaction x, the output y should at least contain the following elements: (m, c, r), where m represents the compounds including reactants, catalysts, solvents, and products; c represents the conditions, which at least contains temperature and time, r represents the results of the reaction, such as the yield and spectroscopic data of the products.

Direct Preference Optimization (DPO) is an offline preference optimization technique to align language models with human preferences. First, we introduce a structure discriminator to judge whether the caption satisfies the structure. The structure discriminator is composed of a set of rules that check if the generated text contains corresponding compounds and conditions like "heat," "cool," and "° C" as well as terms like "yield," "NMR," and so on to ensure structural requirements are met. Then, we consider the label captions in the training set that pass the discriminator as positive samples y_w. After that, for each selected reaction-caption pair above, we produce negative samples using SFT LLM from stage one, which has been performed by in-context tuning. If the generated caption does not satisfy the structural requirements, we accept it as the negative sample y_l. Therefore, we construct the structure preference pairs (y_w, y_l) to apply the DPO algorithm.

The standard DPO algorithm aims to increase the likelihood of the positive example while reducing that of the negative example. Therefore, we optimize the structure preference by applying the loss function as follows:

$$r_w(\theta) = \beta(\log \pi_\theta(y_w|x) - \log \pi_{\text{sft}}(y_w|x)) \quad (3)$$

$$r_l(\theta) = \beta(\log \pi_\theta(y_l|x) - \log \pi_{\text{sft}}(y_l|x)) \quad (4)$$

$$\mathcal{L}_{structure}(\theta) = -\log \sigma(r_w(\theta) - r_l(\theta)) \quad (5)$$

where π_θ is the language model to be optimized, π_{sft} is the SFT language model that has been In-Context Tuned in the stage above, and β is the temperature hyperparameter. σ is the sigmoid function, the difference between r_w and r_l is considered the reward that needs to be optimized.

[1]https://github.com/meta-llama/llama3

Experiments

To evaluate the effectiveness of ReactGPT, we conduct comprehensive experiments comparing our proposed approaches with existing methods on two tasks: reaction captioning and experimental procedure prediction. After that, we include ablation studies demonstrating the contributions of individual components.

Experimental Setting

Data We employ the OpenExp dataset (Liu et al. 2024b) for fine-tuning and evaluation. It consists of 274,439 chemical reactions with the corresponding captions and experimental procedures, which have been filtered and processed from chemical reaction databases of USPTO-Applications (Lowe 2017) and ORD (Kearnes et al. 2021). For our evaluation, we focus on the test split while using the training set as the local database to retrieve k-shot context examples for In-Context Tuning.

Evaluation Metrics To evaluate the understanding of chemical reaction, we employ BLEU (Papineni et al. 2002), ROUGE (Lin 2004), METEOR (Banerjee and Lavie 2005), and the normalized Levenshtein similarity (Levenshtein et al. 1966) for assessing the quality of generations. However, there are many non-overlapping ways to describe a reaction, which makes these metrics less effective to a certain extent. Therefore, we take the structure success of generated texts into account. To measure the model's ability of generating the correct structure of caption, we utilize the structure success rate as one of the evaluation metrics.

Implementation Details The LLaMA3-8B-Instruct is acquired from huggingface Transformers[2]. To efficiently fine-tune the LLaMA3-8B-Instruct, we employ the LoRA approach. To enhance memory utilization and speed up the training process, we incorporated Deepspeed ZeRO stage 2. The entire project is based on the LLaMA-Factory[3]. We adopt AdamW optimizer, set the learning rate as 5e-5, batch size as 4, and the maximum input length to 4096 tokens. 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. All our experiments are performed on 2 NVIDIA A100-80G.

Baselines Specifically, we select the following baselines for performance evaluation.

- **MolT5 (Edwards et al. 2022)**. MolT5 models are pre-trained with MLM on molecule string representations and natural language text, and then fine-tuned on downstream datasets.
- **Galactica (Taylor et al. 2022)**. Galactica is a large language model pre-trained on unstructured scientific corpus, and the model has acquired molecular knowledge since the pretraining stage.
- **ChemDFM (Zhao et al. 2024)**. ChemDFM is pre-trained and fine-tuned based on LLaMA2-13B, and it is considered as the first large language model towards Chemical General Intelligence. ChemDFM-13B

[2]https://github.com/huggingface/transformers

[3]https://github.com/hiyouga/LLaMA-Factory

Methods	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	Structure Succ.
Nearest Neighbor	43.19	29.91	48.77	23.12	36.80	41.99	38.53
MolT5-Base	40.64	30.39	50.82	27.44	40.53	44.12	70.77
MolT5-Large	41.05	31.15	53.19	27.65	41.39	44.32	73.21
GPT-4o-mini	31.59	21.36	44.28	19.48	30.44	42.97	37.82
Galactica	33.58	24.19	46.04	21.43	33.34	37.35	67.70
ChemDFM	45.08	33.05	52.00	26.93	40.52	44.23	83.23
LLaMA3	43.54	32.23	51.29	26.52	40.05	44.37	76.56
ReactGPT (ours)	**48.12**	**36.41**	**53.92**	**28.88**	**42.23**	**47.54**	**89.53**

Table 1: Results of different models for Reaction Captioning task(%). The best scores are marked in bold. For models larger than 6B, we utilize the LoRA (Low-Rank Adaptation) method for low-rank fine-tuning to save running memory. For GPT-4o-mini, we provide it with a context example to learn reaction-text alignment.

is trained on 34B tokens from chemical literature, textbooks, and instructions along with a variety of data from general domain.

- **ReactXT (Liu et al. 2024b)**. ReactXT employs MolCA(Liu et al. 2023b) as the primary LM backbone, which is based on Galactica-1.3B, integrating three types of input contexts to incrementally pre-train an LM. These contexts are tailored to improve LMs' comprehension of individual molecules and chemical reactions.

Reaction Captioning

Table 1 displays the comparison results between our model and other baselines. For the Nearest Neighbor method, we select the reaction text from the training set with the reaction most similar to the query one. We can observe that ReactGPT consistently outperforms all previous models and achieves state-of-the-art across all metrics. Specifically, as for the text generation metrics, ReactGPT shows improvements of 4.18% BLEU-4 and 2.18% ROUGE-L scores compared to LLaMA3-8B-Instruct with naive SFT, while the latter merely obtains a performance that is slightly better than MolT5-base. Moreover, the surpassing performance compared to the domain-specific large language models, such as Galactica-6.7B and ChemDFM-13B, reveals that our framework successfully improves the understanding of chemical reactions and generates high-quality texts that better align with reactions by enriching the context. In the aspect of the structure success rate, we can observe that most of the baselines perform poorly, though they learn the reaction-text pairs during training, they lack training specifically for the structure of the reaction text, which may result in the text lacking the necessary information. To address this challenge, our framework utilizes structure preference optimization via the DPO algorithm and achieves an improvement of 6.27%. In conclusion, these improvements show our framework combines Fingerprints-based Reaction Retrieval, Domain-Specific Prompt Design, and two-stage In-Context Tuning, excelling in the reaction captioning task.

Experimental Procedure Prediction

This task is to predict step-by-step actions of conducting chemical experiments and every action is predefined by Vaucher et al. (2021). We employ the evaluation metrics following Liu et al. (2024b) and Vaucher et al. (2021) and Table 2 shows the performances. Validity examines the syntactical correctness of the predicted action sequences. 90%LEV represents the ratio of predictions with a normalized Levenshtein similarity larger than 90%. Specifically, compared to the original foundation models with naive SFT, ReactGPT achieves an improvement of 4.45% BLEU-4 and 10.20% 50%LEV, demonstrating its generalization performance on different tasks. Moreover, ReactGPT achieves 100% for the validity and can perform as well as ReactXT (Liu et al. 2024b), which is pre-trained and fully fine-tuned base on Galactica-1.3B for 20 epochs.

Ablation Study

Study of k context examples The number of context examples is a factor that affects model performance, since different quantities of context examples denote to the varying lengths of prompts and different amounts of domain knowledge infusion. We set the cutoff length of input to 4096 to avoid affecting the input length. Figure 3 shows that ReactGPT's performance with a different number of context examples. We can observe an obvious difference in performance between 0-shot and k-shot settings. The presence of context examples leads to improved performance, which indicates the effectiveness of the In-Context Tuning. Specifically, we can observe that either a small ($k = 1$) or large number ($k = 3$) of context examples could not reach the optimal results. This may be due to the insufficient domain knowledge injection and the model's limitation of handling a long context of input. In conclusion, when the number of context examples is set to 2, our model is capable of learning sufficiently from domain knowledge without being affected by the limitations of context length.

Study of different components To evaluate the effectiveness of different components, we compare ReactGPT with it

Methods	BLEU-2	BLEU-4	100%LEV	90%LEV	75%LEV	50%LEV	ROUGE-1	ROUGE-2	ROUGE-L	Validity
Nearest Neighbor	45.00	30.70	0.60	6.50	13.00	38.40	55.70	29.20	47.00	76.00
MolT5-Base	54.04	40.31	0.30	4.27	13.22	60.34	61.56	39.43	55.30	99.10
MolT5-Large	54.50	41.00	0.60	6.60	16.60	63.70	62.50	40.90	57.20	99.60
Galactica	53.50	39.50	0.40	5.70	13.40	60.50	60.90	38.60	55.20	99.90
MolCA	54.90	41.50	1.00	9.20	18.90	65.30	62.50	40.40	57.00	99.90
ReactXT	57.40	**44.00**	1.00	**9.50**	**22.60**	70.20	64.40	42.70	58.90	**100.00**
LLaMA3	51.73	39.28	1.80	7.70	18.50	62.50	62.52	39.50	56.19	99.80
ReactGPT (ours)	**57.89**	43.73	**1.89**	8.00	21.96	**72.70**	**66.10**	**42.95**	**59.98**	**100.00**

Table 2: Results of different models for Experimental Procedure Prediction task(%). The best scores are marked in bold. For models larger than 6B, we utilize the LoRA (Low-Rank Adaptation) method for low-rank fine-tuning to save running memory.

Methods	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	Structure Succ.
ReactGPT (ours)	48.12	36.41	53.92	28.88	42.23	47.54	89.53
w/o retrieval	45.69	33.91	52.29	26.93	41.39	45.32	80.56
w/o In-Context Tuning	43.54	32.23	51.29	26.52	40.05	44.37	76.56

Table 3: Ablation Study of ReactGPT across different components.

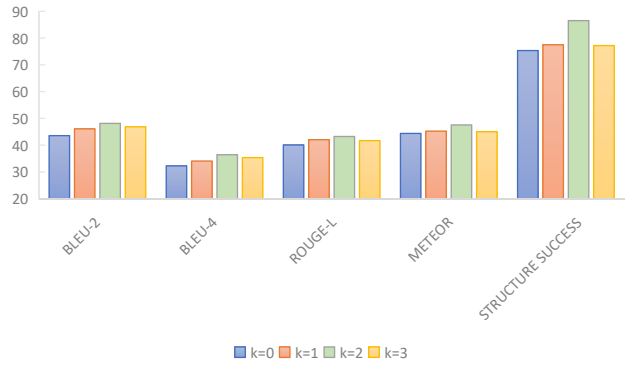

Figure 3: The model performance with a different number of context examples.

variants on the reaction captioning task: 1) w/o Fingerprints-based Reaction Retrieval. In this variant, we have eliminated the retrieval component and directly utilized random context examples to provide domain knowledge. 2) w/o In-Context Tuning. In this variant, we remove the In-Context Tuning stage.

From Table 3, we can observe that 1) ReactGPT exhibits the best performance, indicating that its effectiveness is a cumulative contribution of all its components. 2) The absence of retrieval module results in the model's inability to learn from similar context examples and only obtains limited improvements from random context examples, therefore performing worse than ReactGPT. 3) Whether the model incorporates random context examples or selects the most similar ones, the outcome is superior to that of a model without In-Context Tuning, underscoring the necessity and effectiveness of In-Context Tuning.

Conclusion

A wealth of chemical reactions and their text descriptions exist in patents and scientific literature. By aligning chemical reactions and natural language texts, it will be beneficial to quickly understand chemical reactions, obtain information such as reaction conditions and reaction yields, and strengthen research in the field of chemistry.

In this work, we propose ReactGPT, a novel approach that adapts LLMs to align between chemical reactions and natural language texts. ReactGPT enables LLMs to utilize the relevant domain knowledge and the ability of in-context learning to understand chemical reactions and generate their textual representations. Moreover, we propose a new task: reaction captioning, which aims to generate a detailed description of chemical reactions. In the two tasks of reaction captioning and experimental procedure prediction, ReactGPT is able to achieve a good performance, which indicates the effectiveness of our methods.

Acknowledgments

We would like to thank the anonymous reviewers for helpful questions and comments. We appreciate the support from National Natural Science Foundation of China with the Main Research Project on Machine Behavior and Human Machine Collaborated Decision Making Methodol-

ogy(72192820 &72192824) and Pudong New Area Science &Technology Development Fund (PKX2021-R05).

References

Ahn, J.; Verma, R.; Lou, R.; Liu, D.; Zhang, R.; and Yin, W. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. arXiv:2402.00157.

Azerbayev, Z.; Schoelkopf, H.; Paster, K.; Santos, M. D.; McAleer, S.; Jiang, A. Q.; Deng, J.; Biderman, S.; and Welleck, S. 2024. Llemma: An Open Language Model For Mathematics. arXiv:2310.10631.

Banerjee, S.; and Lavie, A. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 65–72.

Bran, A. M.; Cox, S.; White, A. D.; and Schwaller, P. 2023. ChemCrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*.

Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J. D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901.

Chen, W.; Ming, Y.; Max, K.; Elaine, W.; Ma, X.; Xu, J.; Xia, T.; Wang, X.; and Lu, P. 2023. TheoremQA: A Theorem-driven Question Answering dataset. *arXiv preprint arXiv:2305.12524*.

Chen, Y.; Xi, N.; Du, Y.; Wang, H.; Chen, J.; Zhao, S.; and Qin, B. 2024. From Artificially Real to Real: Leveraging Pseudo Data from Large Language Models for Low-Resource Molecule Discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 21958–21966.

Chen, Y.; Zhong, R.; Zha, S.; Karypis, G.; and He, H. 2021. Meta-learning via language model in-context tuning. *arXiv preprint arXiv:2110.07814*.

Dan, Y.; Lei, Z.; Gu, Y.; Li, Y.; Yin, J.; Lin, J.; Ye, L.; Tie, Z.; Zhou, Y.; Wang, Y.; Zhou, A.; Zhou, Z.; Chen, Q.; Zhou, J.; He, L.; and Qiu, X. 2023. EduChat: A Large-Scale Language Model-Based Chatbot System for Intelligent Education.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; and Sui, Z. 2022. A Survey for In-context Learning. *arXiv preprint arXiv:2301.00234*.

Edwards, C.; Lai, T.; Ros, K.; Honke, G.; Cho, K.; and Ji, H. 2022. Translation between Molecules and Natural Language. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 375–413. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Edwards, C.; Zhai, C.; and Ji, H. 2021a. Text2Mol: Cross-Modal Molecule Retrieval with Natural Language Queries. In Moens, M.-F.; Huang, X.; Specia, L.; and Yih, S. W.-t., eds., *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 595–607. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Edwards, C.; Zhai, C.; and Ji, H. 2021b. Text2mol: Cross-modal molecule retrieval with natural language queries. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 595–607.

Fang, Y.; Liang, X.; Zhang, N.; Liu, K.; Huang, R.; Chen, Z.; Fan, X.; and Chen, H. 2023. Mol-Instructions: A Large-Scale Biomolecular Instruction Dataset for Large Language Models. *arXiv preprint arXiv:2306.08018*.

Guo, H.; Zhao, S.; Wang, H.; Du, Y.; and Qin, B. 2024. Moltailor: Tailoring chemical molecular representation to specific tasks via text prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 18144–18152.

Kearnes, S. M.; Maser, M. R.; Wleklinski, M.; Kast, A.; Doyle, A. G.; Dreher, S. D.; Hawkins, J. M.; Jensen, K. F.; and Coley, C. W. 2021. The open reaction database. *Journal of the American Chemical Society*, 143(45): 18820–18826.

Levenshtein, V. I.; et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, 707–710. Soviet Union.

Li, J.; Liu, W.; Ding, Z.; Fan, W.; Li, Y.; and Li, Q. 2024. Large Language Models are In-Context Molecule Learners. *arXiv preprint arXiv:2403.04197*.

Li, J.; Liu, Y.; Fan, W.; Wei, X.-Y.; Liu, H.; Tang, J.; and Li, Q. 2023. Empowering Molecule Discovery for Molecule-Caption Translation with Large Language Models: A ChatGPT Perspective. *arXiv preprint arXiv:2306.06615*.

Liao, C.; Yu, Y.; Mei, Y.; and Wei, Y. 2024. From Words to Molecules: A Survey of Large Language Models in Chemistry. *arXiv preprint arXiv:2402.01439*.

Lin, C.-Y. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, 74–81.

Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; and Neubig, G. 2023a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9): 1–35.

Liu, S.; Wang, J.; Yang, Y.; Wang, C.; Liu, L.; Guo, H.; and Xiao, C. 2024a. Conversational drug editing using retrieval and domain feedback. In *The Twelfth International Conference on Learning Representations*.

Liu, Z.; Li, S.; Luo, Y.; Fei, H.; Cao, Y.; Kawaguchi, K.; Wang, X.; and Chua, T.-S. 2023b. MolCA: Molecular Graph-Language Modeling with Cross-Modal Projector and Uni-Modal Adapter. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 15623–15638. Singapore: Association for Computational Linguistics.

Liu, Z.; Shi, Y.; Zhang, A.; Li, S.; Zhang, E.; Wang, X.; Kawaguchi, K.; and Chua, T.-S. 2024b. ReactXT:

Understanding Molecular" Reaction-ship" via Reaction-Contextualized Molecule-Text Pretraining. *arXiv preprint arXiv:2405.14225*.

Liu, Z.; Zhang, W.; Xia, Y.; Wu, L.; Xie, S.; Qin, T.; Zhang, M.; and Liu, T.-Y. 2023c. Molxpt: Wrapping molecules with text for generative pre-training. *arXiv preprint arXiv:2305.10688*.

Lowe, D. 2017. Chemical reactions from US patents (1976-Sep2016).

Lu, P.; Mishra, S.; Xia, T.; Qiu, L.; Chang, K.-W.; Zhu, S.-C.; Tafjord, O.; Clark, P.; and Kalyan, A. 2022. Learn to Explain: Multimodal Reasoning via Thought Chains for Science Question Answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*.

Luo, L.; Ning, J.; Zhao, Y.; Wang, Z.; Ding, Z.; Chen, P.; Fu, W.; Han, Q.; Xu, G.; Qiu, Y.; Pan, D.; Li, J.; Li, H.; Feng, W.; Tu, S.; Liu, Y.; Yang, Z.; Wang, J.; Sun, Y.; and Lin, H. 2024. Taiyi: A Bilingual Fine-Tuned Large Language Model for Diverse Biomedical Tasks. *Journal of the American Medical Informatics Association*, ocae037.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, 311–318.

Qian, Y.; Li, Z.; Tu, Z.; Coley, C.; and Barzilay, R. 2023. Predictive Chemistry Augmented with Text Retrieval. In Bouamor, H.; Pino, J.; and Bali, K., eds., *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12731–12745. Singapore: Association for Computational Linguistics.

Rozière, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; Rapin, J.; Kozhevnikov, A.; Evtimov, I.; Bitton, J.; Bhatt, M.; Ferrer, C. C.; Grattafiori, A.; Xiong, W.; Défossez, A.; Copet, J.; Azhar, F.; Touvron, H.; Martin, L.; Usunier, N.; Scialom, T.; and Synnaeve, G. 2024. Code Llama: Open Foundation Models for Code. arXiv:2308.12950.

Schwaller, P.; Probst, D.; Vaucher, A. C.; Nair, V. H.; Kreutter, D.; Laino, T.; and Reymond, J.-L. 2021. Mapping the space of chemical reactions using attention-based neural networks. *Nature machine intelligence*, 3(2): 144–152.

Tang, X.; Jin, Q. J.; Zhu, K.; Yuan, T.; Zhang, Y.; Zhou, W.; Qu, M.; Zhao, Y.; Tang, J.; Zhang, Z.; Cohan, A.; Lu, Z.; and Gerstein, M. 2024. Prioritizing Safeguarding Over Autonomy: Risks of LLM Agents for Science. *arXiv preprint arXiv:2402.04247*.

Taylor, R.; Kardas, M.; Cucurull, G.; Scialom, T.; Hartshorn, A.; Saravia, E.; Poulton, A.; Kerkez, V.; and Stojnic, R. 2022. Galactica: A large language model for science. *arXiv preprint arXiv:2211.09085*.

Vaucher, A. C.; Schwaller, P.; Geluykens, J.; Nair, V. H.; Iuliano, A.; and Laino, T. 2021. Inferring experimental procedures from text-based representations of chemical reactions. *Nature communications*, 12(1): 2573.

Weininger, D. 1988. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1): 31–36.

Weininger, D.; Weininger, A.; and Weininger, J. L. 1989. SMILES. 2. Algorithm for generation of unique SMILES notation. *Journal of chemical information and computer sciences*, 29(2): 97–101.

Wu, S.; Irsoy, O.; Lu, S.; Dabravolski, V.; Dredze, M.; Gehrmann, S.; Kambadur, P.; Rosenberg, D.; and Mann, G. 2023. BloombergGPT: A Large Language Model for Finance. arXiv:2303.17564.

Xie, Q.; Han, W.; Zhang, X.; Lai, Y.; Peng, M.; Lopez-Lira, A.; and Huang, J. 2023. PIXIU: A Large Language Model, Instruction Data and Evaluation Benchmark for Finance. arXiv:2306.05443.

Zhao, Z.; Ma, D.; Chen, L.; Sun, L.; Li, Z.; Xu, H.; Zhu, Z.; Zhu, S.; Fan, S.; Shen, G.; et al. 2024. Chemdfm: Dialogue foundation model for chemistry. *arXiv preprint arXiv:2401.14818*.

Zhong, M.; Ouyang, S.; Jiang, M.; Hu, V.; Jiao, Y.; Wang, X.; and Han, J. 2023. ReactIE: Enhancing Chemical Reaction Extraction with Weak Supervision. In Rogers, A.; Boyd-Graber, J.; and Okazaki, N., eds., *Findings of the Association for Computational Linguistics: ACL 2023*, 12120–12130. Toronto, Canada: Association for Computational Linguistics.