Can ChatGPT understand the implicit meaning of language? Discussion of ChatGPT's ability to generate metaphorical samples

Anonymous ACL submission

Abstract

The effectiveness of large-scale language modeling (LLM) in generating data samples has been widely proven, especially in question answering and textual entailment tasks. However, these tasks are primarily concerned with surface semantics and usually require the model to learn only information about lexical and syntactic structures. In contrast, generating metaphorical samples requires LLMs to develop a deeper understanding of the implicit meanings in the text. Therefore, the aim of this paper is to explore the ability of ChatGPT to generate metaphorical samples. First, we propose two prompt enhancement methods based on definitions and multiple word meanings. The former introduces a metaphor definition, and the latter requires LLM to generate the corresponding metaphorical or literal sample content based on each word sense. Experimental results show that the SPE method performs slightly lower than manually labeled samples in terms of fine-tuning performance (3.5% lower than the average F1 value for the three metaphorical datasets), but at 1/250th the cost of the latter. Since most of work focuses on zero- or few-shot methods, we use it as a baseline. We provide an in-depth discussion of the differences between the four sample generation methods mentioned above through manual evaluation, automated evaluation, and example analysis. To enhance the reliability of the study, we introduce ChatGPT, LLaMA3, and Mixtral to further explore the differences in generating implicit semantic content across LLMs.

1 Introduction

Data annotation is a time-consuming and labor-intensive task. The average cost of labeling each instance on a crowdsourcing platform is \$0.11 (Wang et al., 2021a). This high cost has become a constraint for further development of many studies. In contrast, generating samples using ChatGPT API becomes a more cost-effective alternative, with a cost of \$0.05 per 1M tokens input and \$0.15 per 1M tokens output, respectively. Therefore, it is important and valuable to understand and guide ChatGPT to generate high-quality sample data. Specifically, (1) mitigating the labor and time overhead of manual annotation. (2) improving the performance of LLM in low-resource scenarios by transferring the rich world knowledge in LLM. (3) generating high-quality samples using LLM that can be used for fine-tuning of the lightweight model. (4) research on generating metaphorical samples that can allow LLM to better understand and generate content that contains complex semantics.

In previous studies, LLMs have been widely used to construct data for various NLP tasks, mainly including two categories, sample labeling and sample generation, each of which can be further categorized into zero- and few-shot methods. For example, sample labeling using simple instructions (Ollion et al., 2023; Laskar et al., 2023; Gilardi et al., 2023; Koptyra et al., 2023; Belal et al., 2023). On few-shot methods, Su et al. (2022), Liu et al. (2021) and Rubin et al. (2021) improve the quality of the model's annotation for new samples by filtering representative or content-diverse example samples. In addition, Wang et al. (2021a) and Alizadeh et al. (2023) explored ways to introduce LLM labeled data on top of manual labeling to minimize the manual labeling cost without significant performance degradation. For sample generation, past zero-shot approaches (Saha et al., 2024; Huang et al., 2023) only provide task descriptions and sample labels, and LLMs are required to generate specified sample contents. Few-shot studies (Li et al., 2024; Hartvigsen et al., 2022) use manually labeled samples as examples to guide LLMs to generate similar samples.

However, the above studies on LLM generation samples mainly focus on data generation for surface language tasks, which usually only require the model to learn information about lexical and

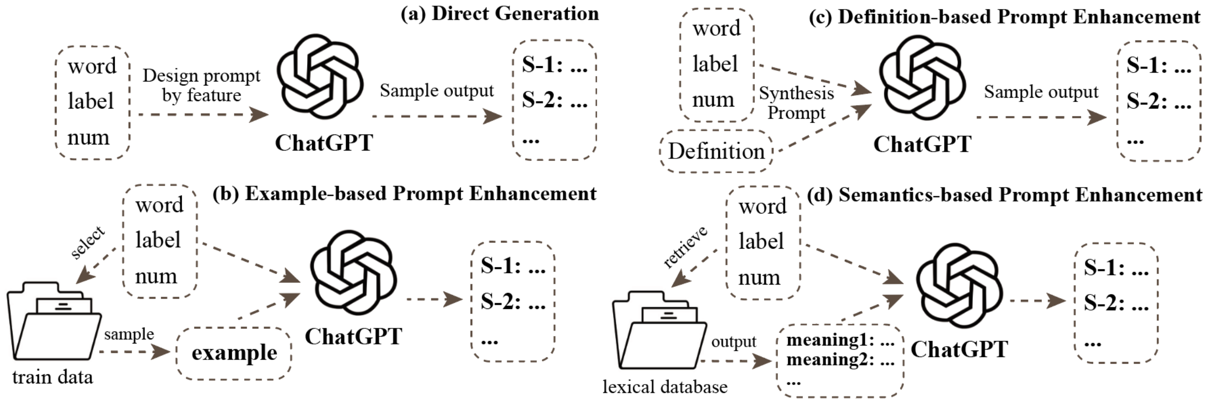

Figure 1: The four sample generation methods explored in this paper. DG: sample generation based on task formulation and labeling (metaphorical or not). EPE: metaphorical samples are used as reference examples in the generation process. **DPE (ours)**: enhancement of sample generation by adding metaphorical definitions. **SPE (ours)**: sample generation by using multi-word meanings of target words as knowledge.

syntactic structures. Metaphor is a high-level cognitive modality, and as an implicit semantic class of tasks, metaphor comprehension is very complex and requires in-depth understanding of the implicit meanings in the text. Therefore, the aim of this paper is to explore the performance of LLM in generating metaphor samples. We design two knowledge injection methods, definition-based prompt enhancement (DPE) and semantics-based prompt enhancement (SPE) methods. DPE only needs to give metaphor definitions, while SPE needs to introduce multi-meaning information from wordnet or oxford dictionary. We consider the first two meanings as literal and the rest as metaphorical (in order of frequency of use), and then ask LLM to generate corresponding literal and metaphorical samples based on different meanings. In addition, we introduce LLM direct generation (DG) and example-based prompt enhancement (EPE) methods as controls. We use three LLMs, ChatGPT, LLaMA, and Mixtral, to generate metaphor samples and verify the performance of our proposed scheme by fine-tuning the small model. Then, we analyze in-depth the similarities and differences between the LLM-generated samples and the manually labeled samples using both manual and automatic evaluation methods with case study. Overall, our contributions are as follows:

1. To the best of our knowledge, this is the first study to apply ChatGPT to metaphorical sample generation. We conducted manual and automatic evaluation of LLM-generated samples and manually labeled samples, and provided an in-depth analysis of the similarities and differences between the two.

2. We propose definition-based prompt enhancement (DPE) and semantics-based prompt enhancement (SPE) methods. Experimental results show that our proposed methods achieve the best performance when using different LLMs as sample generators.

3. We give an example analysis of ChatGPT generated samples, summarizing the current problem into three categories: misinterpretation of conventional meaning (MCM), neglect of metaphorical evolution (NME), and polysemy confusion (PC).

2 Related Work

2.1 LLM Sample Generation

The zero-shot sample generation approach only requires the provision of a task description and sample labels to guide the LLM to generate samples of the specified type. e.g., "The movie review in positive sentiment is:" (Ye et al., 2022). Some of these studies (Ubani et al., 2023; Ye et al., 2022; Gao et al., 2022; Meng et al., 2022; Wang et al., 2022) were tested on multiple NLP-based tasks (e.g., SST-2 (Socher et al., 2013), IMDb (Maas et al., 2011)). Wang et al. (2022) adds a filtering mechanism to filter duplicate and low quality samples. Saha et al. (2024) and Huang et al. (2023) explore hate or counterfactual speech sample generation.

Another part of the research (Yoo et al., 2021; Wang et al., 2021b; Hartvigsen et al., 2022; Li et al., 2024) used an example-based approach, which takes a small amount of manually labeled data as an example and directs LLM to generate similar samples. Of these, Li et al. (2024) explored

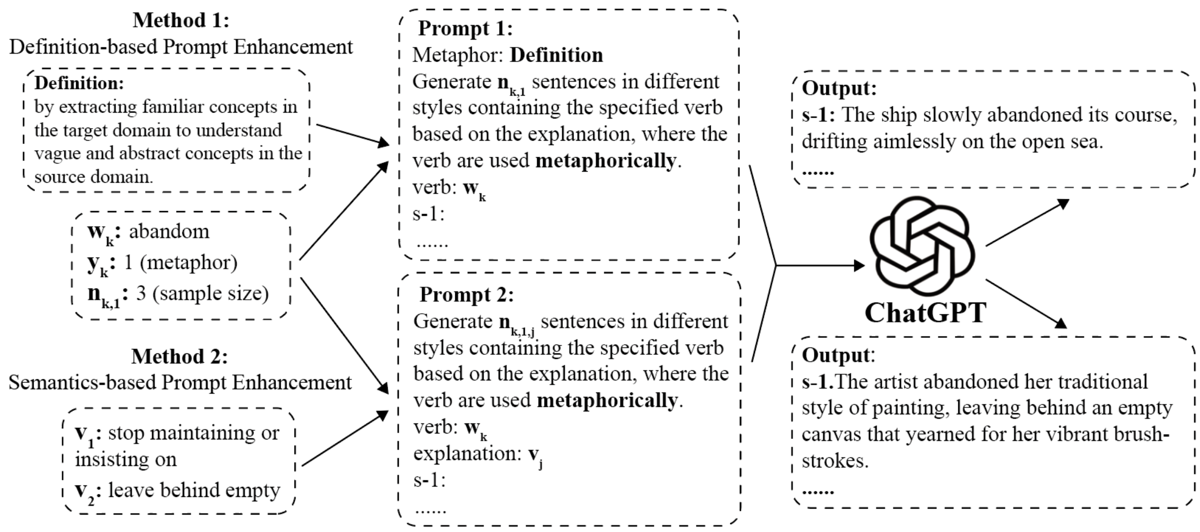

Figure 2: SPE and DPE methods prompt design. w_k denotes the target word and y_k is the label. In DPE, $n_{k,i}$ is the number of samples to be generated for the target word w_k, and $i = 0$ or 1 corresponds to the target word being used literally, metaphorically, respectively. For SPE, v_j denotes the jth lexical sense of the target word w_k. $n_{k,i,j}$ is the number of samples to be generated for the jth meaning of the target word w_k.

low-resource text generation and Hartvigsen et al. (2022) used LLM to generate hate speech datasets. Yoo et al. (2021) and Wang et al. (2021b) test the effectiveness of LLM's generation across multiple subtasks.

Furthermore, Xu et al. (2023) and Taori et al. (2023) devised a heuristic Instruction method that starts reasoning from the initial Instruction and iteratively generates a wider range of more complex Instruction. This work names the zero- or few-shot methods as direct generation (DG) and example-based prompt enhancement (EPE) methods.

2.2 Metaphor Detection

For the target words and corresponding contexts, metaphor detection aims to determine whether the words are used in a metaphorical manner. Compared to tasks such as sentiment labeling and question and answer, metaphor detection requires the model to have a deeper understanding of the implicit meaning of the text, a challenge that has typically been addressed in prior research by injecting domain knowledge. In prior work, researchers have used a variety of knowledge injection strategies. Among them, Le et al. (2020), Song et al. (2021) and Feng and Ma (2022) used dependency tree knowledge to direct the model to focus on specific syntactic structures. Mao and Li (2021), Choi et al. (2021) and Su et al. (2020) incorporate Part-Of-Speech tagging (POS), where Mao and Li (2021) treats POS as a separate subtask. In addition, Gong et al. (2020), Klebanov et al. (2016) and Zhang and Liu (2023) introduced the wordnet database (Fellbaum, 1998). Gong et al. (2020) and Klebanov et al. (2016) classified words into fifteen categories based on semantic features, while Zhang and Liu (2023) constructed a binary classification subtask by directly taking the most common definitions of words in wordnet as literal meanings.

3 Method

This section describes four sample generation methods: the DG, EPE, DPE, and SPE. prompt designs for the DG and EPE methods are shown in Appendix 12.1 and 12.2, respectively. the DPE and SPE methods will be described next.

Definition-based Prompt Enhancement. The DPE approach aims at injecting metaphorical definitions as knowledge into LLM. This paper uses the definition given by Lakoff and Johnson (2008): extracting familiar concepts in the target domain to understand vague and abstract concepts in the source domain.

Semantics-based Prompt Enhancement. The SPE approach aims to inject the lexical knowledge of the target word into the LLM. This paper use multiple word sense information from wordnet (Miller, 1995; Fellbaum, 1998) and the oxford dictionary. Among them, wordnet has been shown to help improve metaphor recognition performance (Gong et al., 2020; Klebanov et al., 2016; Zhang and Liu, 2023). For any target word w_k, as well as the verb meaning sets $\mathcal{V}_k$ retrieved from wordnet ($\mathcal{V}_k$ is sorted by frequency of use), we consider the first two common meanings as literal meanings, and the rest as metaphorical meanings. That is, for any lexical meaning $v_j \in \mathcal{V}_k$:

$$v_j \in \begin{cases} \mathcal{V}_{k,l} & 0 < j \leq 2 \text{ and } y_k = 0 \\ \mathcal{V}_{k,m} & j > 2 \text{ and } y_k = 1, \end{cases} \quad (1)$$

where $\mathcal{V}_{k,l}$ and $\mathcal{V}_{k,m}$ denote the literal and metaphorical lexical sense sets of the target word w_k, respectively. The label $y_k = 0$ indicates that w_k is used non-metaphorically, while $y_k = 1$ indicates that w_k is used metaphorically.

Prompt Construction. The prompt design of DPE and SPE is shown in Fig.2. For the input (w_k, y_k), we first specify the target word $word = w_k$. Then, based on the value of y_k, the model is asked to generate $n_{k,i}$ literal or metaphorical sentences, where $i = 0$ or 1 corresponds to $y_k = 0$, $y_k = 1$, respectively. For DPE, we added the metaphorical definition at the beginning. For SPE, we consider the literal lexical sense set $\mathcal{V}_{k,l}$ and the metaphorical lexical sense set $\mathcal{V}_{k,m}$ for the target word w_k. Specifically, we first divide based on the number of samples to be generated, for $y_k = 1$ there are:

$$n_{k,1,j} = \text{ceil}(\frac{n_{k,1}}{|\mathcal{V}_{k,m}|}), \quad (2)$$

where ceil is an upward rounding function, $|\mathcal{V}_{k,m}|$ denotes the number of metaphorical lexical sense, $n_{k,1,j}$ denotes the target word of the kth metaphorical usage, and the number of samples to be generated for the jth lexical meaning. For example, for the first metaphorical lexical meaning $v_3 \in \mathcal{V}_{k,m}$ and its required number of generated samples $n_{k,1}$. We specify the values of the variables in the prompt: $n = n_{k,1,j}, meaning = v_3$, bootstrap ChatGPT to generate the metaphor samples. The next metaphorical meaning v_4 is then given until $n_{k,1}$ samples have been generated.

4 Fine-tuning Model Experiments

4.1 Experimental Setup

Experiment 1. The experiment was designed to fine-tune the mini-model using the LLM-generated samples as a training set and to test it on three metaphorical datasets, VUAverb, TroFi, and MOH-X (see Appendix 11 for a detailed description of the datasets). The purpose of the experiment was: (1) verify whether the samples generated by LLM contain sufficient metaphorical knowledge. Higher quality samples tend to allow the fine-tuned model to achieve higher performance on the metaphor test set. (2) Compare the differences in the samples generated by different LLMs. (3) Discuss how different metaphorical knowledge injection methods affect the quality of sample data generated by LLM. The experiments include four types of DG (no external knowledge), EPE (metaphorical example knowledge), DPE (metaphorical definition knowledge), and SPE (metaphorical knowledge with multiple word meanings). We used three LLMs for sample generation:

- **Mixtral**: Mixtral is an open source generative sparse expert mixture model provided by Mistral AI[1]. We use Mixtral-8x7B-Instruct-v0.1 version, whose weight parameters are derived from huggingface[2].
- **LLaMA3**: LLaMA3 is a parametric large language model released by Meta AI on April 18, 2024, including 8B and 70B. We use the version Llama-3-70B-Instruct, its weights can be obtained from the official website[3].
- **ChatGPT**: ChatGPT is a closed-source model developed by OpenAI, which is available for paid use through API[4]. This paper, the version gpt-3.5-turbo-0125 is used.

For fine-tuning, we used RoBERTa (Liu et al., 2019), initialized by the weight parameters of Huggingface (Wolf et al., 2019). The output of the model adopts part of the model idea devised in Choi et al. (2021), i.e., the hidden layer output corresponding to the CLS and the target word is used for classification. Specifically, we first let RoBERTa be trained on DG, EPE, DPE, and SPE samples, respectively, and then validated on the test set. We perform the test set on the entire VUAverb test, TroFi and MOH-X with samples 5875, 3739 and 649, respectively.

Experiment 2. Experiment 2 compares the SPE method for generating samples with manually labeled samples (i.e., the VUAverb training set) in terms of fine-tuning performance and cost. Since

[1]https://mistral.ai/news/mixtral-of-experts/

[2]https://huggingface.co/mistralai/Mixtral-8x7B-Instruct-v0.1/tree/main

[3]https://llama.meta.com/llama-downloads

[4]https://platform.openai.com/

Method	VUAverb			TroFi			MOH-X		
	P	R	F1	P	R	F1	P	R	F1
Mixtral-DG	0.516	0.261	0.347	0.531	0.175	0.263	1.000	0.038	0.073
Mixtral-EPE	0.412	0.039	0.071	0.586	0.036	0.067	0.529	0.057	0.103
Mixtral-DPE	0.448	0.375	**0.408**	0.538	0.297	0.383	0.813	0.289	**0.426**
Mixtral-SPE	0.348	0.454	0.394	0.461	0.342	**0.392**	0.551	0.311	0.398
LLaMA3-DG	0.547	0.166	0.254	0.558	0.146	0.231	0.900	0.171	0.288
LLaMA3-EPE	0.440	0.086	0.144	0.442	0.101	0.164	0.545	0.038	0.071
LLaMA3-DPE	0.552	0.277	**0.368**	0.565	0.242	**0.338**	0.858	0.384	**0.531**
LLaMA3-SPE	0.325	0.338	0.332	0.420	0.248	0.312	0.559	0.359	0.446
ChatGPT-DG	0.541	0.136	0.217	0.506	0.084	0.144	0.870	0.171	0.286
ChatGPT-EPE	0.450	0.294	0.356	0.564	0.266	0.361	0.516	0.316	0.392
ChatGPT-DPE	0.507	0.298	0.376	0.549	0.237	0.330	0.836	0.324	0.467
ChatGPT-SPE	0.302	0.794	**0.438**	0.439	0.910	**0.593**	0.497	0.470	**0.483**

Table 1: LLM generated samples on three metaphorical datasets to fine-tune performance. Experiments is binary classification. **F1** scores are core metrics indicating the weighted average of precision (**P**) and recall (**R**). **DG**: LLM direct generation method; **EPE**: example-based prompt enhancement method; **DPE**: definition-based prompt enhancement method; **SPE**: Lexical semantics-based prompt enhancement method.

Method	Fine-tuning performance									Labeling costs		
	VUAverb			TroFi			MOH-X					
	P	R	F1	P	R	F1	P	R	F1	Input	Output	Avg
SPE-w	0.302	0.794	0.438	0.439	0.910	0.593	0.497	0.470	0.483	0087$	**0.252$**	**0.339$**
SPE-o	0.356	0.842	0.501	0.453	0.895	**0.601**	0.609	0.806	0.694	**0.068$**	0.280$	0.348$
GT	0.479	0.646	**0.550**	0.509	0.731	0.600	0.738	0.768	**0.753**	-	-	869$

Table 2: Comparison of the SPE method with manually labeled samples in terms of fine-tuning performance (left) and labeling cost (right). SPE-w and SPE-o use wordnet and oxford dictionary's multi lexical sense knowledge, respectively, and both use ChatGPT to generate the samples. **Input**: cost of input for the prompt design; **Output**: cost of ChatGPT output data; **Avg**: average of inputs and outputs

SPE introduces metaphorical knowledge of multi-word meanings, in addition to wordnet, oxford dictionary also contains multi-word meanings. We denote the wordnet- and oxford dictionary-based SPE methods as SPE-w and SPE-o, respectively. On fine-tuning, the model fine-tuning method is the same as that of Experiment 1. For cost analysis, we use the manual annotation cost recorded in Wang et al. (2021a) (i.e., $0.11/per sample). For SPE-generated samples, we tokenize them using the methods provided by RoBERTa (Liu et al., 2019) and record the total number of sample tokens for each method separately. We record the token price given in the official OpenAI website as the automatic labeling cost. The input is $0.5 per 1M tokens and the output is $1.5 per 1M tokens.

4.2 Results

Experiment 1. The experimental results are presented in Table 1. Our proposed methods achieve the best F1 performance on all three LLMs and all three datasets (e.g., on VUAverb, SPE 0.438 vs. EPE 0.356 on ChatGPT and DPE 0.368 vs. DG 0.254 on LLaMA3 and DPE 0.408 vs. DG 0.347 on Mixtral). This shows that the DPE and SPE methods somewhat balance the accuracy of detecting metaphorical and literal samples.

For the EPE method introduced as an example, while its performance is better for samples generated using ChatGPT, we observe a larger performance degradation when using the open-source Mixtral and LLaMA3 models (e.g., on F1, Chat-

Method	Clarity			Relevance			Diversity		
	Literal	Metaphor	Avg	Literal	Metaphor	Avg	Literal	Metaphor	Avg
GT	4.054	3.828	3.941	4.075	3.387	3.731	**4.086**	**3.785**	**3.935**
DG	4.677	4.355	4.516	**4.151**	**3.699**	**3.925**	3.753	3.419	3.586
EPE	4.505	4.312	4.409	3.430	3.344	3.387	3.796	3.505	3.651
DPE	**4.710**	**4.473**	**4.591**	4.108	3.237	3.672	3.892	3.376	3.634
SPE	4.602	4.333	4.468	4.097	3.301	3.699	3.946	3.634	3.790

Table 3: Results of manual evaluation of ChatGPT generated samples and manually labeled samples. Clarity, relevance, and diversity are formulated in Appendix 13.1. **Literal**: literal sample scores; **Metaphor**: metaphorical sample scores; **Avg**: average of Literal and Metaphor samples.

GPT 0.356 vs. LLaMA3 0.144 on VUAverb and ChatGPT 0.361 vs. LLaMA3 0.164 on TroFi and ChatGPT 0.392 vs. Mixtral 0.103 on MOH-X). On the one hand, it shows that example knowledge can be counterproductive if the model is unable to understand or misinterprets the introduced example information. On the other hand, it also shows that compared to closed-source ChatGPT, current open-source LLM models often do not understand the metaphorical information in the examples well, which leads to a drastic decrease in the recall of the EPE method (e.g., on VUAverb, EPE 0.039 on Mixtral and EPE 0.086 on LLaMA3).

For our proposed DPE method, the low recall of DG or EPE is improved without decreasing precision (e.g., on VUAverb, DPE 0.277 vs. EPE 0.086 on LLaMA and DPE 0.375 vs. EPE 0.039 on Mixtral). This suggests that introducing metaphor definitions works better than introducing metaphor examples when modeling capabilities are weak. For ChatGPT with some degree of metaphor comprehension, the difference in recall between the two is not significant when definitions or examples are introduced (e.g., DPE 0.298 vs. EPE 0.294 on VUAverb and DPE 0.237 vs. EPE 0.266 on TroFi and DPE 0.324 vs. EPE 0.316 on MOH-X).

In addition, DG, EPE and DPE tend to have higher precision than recall. It shows a stronger ability to recognize non-metaphorical samples. In particular, the DG method is the most prominent among the three (e.g., on MOH-X, DG 1 on Mixtral and DG 0.9 on LLaMA3 and DG 0.870 on ChatGPT). Since DG tend to use simple instruction descriptions, whereas EPE and DPE methods introduce partial external knowledge. This suggests that unguided LLM output knowledge tends to be non-metaphorical. This is manifested in the fine-tuning model with its low recall (i.e., weak recognition of metaphorical samples). In contrast, our proposed alternative SPE approach based on multiple lexical meanings has a more balanced precision and recall on all three LLM models, and even higher recall (e.g., on ChatGPT, precision 0.302 vs. recall 0.794 on VUAverb and precision 0.439 vs. recall 0.910 on TroFi and precision 0.497 vs. recall 0.470 on MOH-X). This suggests that injecting metaphorical knowledge in the form of multiple lexical meanings is superior to introducing metaphorical examples or definitions directly.

Experiment 2. As can be seen from the results in Table 2, the SPE-oxford method outperforms the SPE-wordnet method in fine-tuning on all three metaphor datasets. Compared to wordnet, oxford dictionary tend to contain richer and more current lexical information. As a result, the SPE-oxford method produces higher quality, as evidenced by further improvements in precision and recall (e.g., VUAverb SPE-oxford 0.356 vs. SPE-wordnet 0.302 on precision and SPE-oxford 0.842 vs. SPE-wordnet 0.794 on recall). While SPE-oxford is lower than GT (manually labeled samples) on VUAverb and MOH-X (i.e., on F1, -0.049 on VUAverb and -0.059 on MOH-X), it is slightly higher on TroFi (i.e., on F1, + 0.001 on TroFi). Overall, although the SPE-oxford method slightly underperforms the real samples in terms of fine-tuning performance, it requires only about 1/250th of the cost of manual labeling. This demonstrates the superiority of our proposed method.

5 Manual Evaluation

The manual evaluation was designed to compare the differences between the samples generated using ChatGPT, and the manually labeled real samples. The manual evaluation is done on a group basis. For example, a sample group (target word "abandon" and label "1"). We invited three volunteers to assess this sample group, using clarity, relevance, and diversity as the three metrics for evaluation, and redefining them for the characteristics of the metaphor task (see Appendix 13.1 for specific definitions). These metrics were scored on a scale of 1 to 5, and the final results were averaged across the three ratings.

Results. The results of the manual evaluation are shown in Table 3. Compared to the real sample (GT), the clarity scores of the samples generated using ChatGPT were higher (e.g., on avg, DG +0.623 and EPE +0.451 and DPE +0.656 and SPE +0.548). This suggests that the generated samples are easier to understand. Similarly, DG performs best on relevance (e.g., +0.194 on GT and +0.226 on SPE). This suggests that prompt without external knowledge makes LLM less disturbed compared to the introduction of metaphorical knowledge generation methods, thus ensuring to some extent that LLM generates samples with better accuracy.

However, the understandability and accuracy of the generated samples do not enhance the performance of the fine-tuned model (comp. GT, DG +0.575 on clarity and DG +0.194 on relevance, but DG -0.333 on VUAverb-F1). Instead, there was a correlation between diversity and model fine-tuning performance (e.g., on ChatGPT and VUAverb, GT-avg 3.935 vs. GT-F1 0.550 and SPE-avg 3.790 vs. SPE-F1 0.438 and DG-avg 3.586 vs. DG-F1 0.217). This suggests that the richness of metaphorical usage can inject more metaphorical knowledge into the fine-tuned model, thus improving the quality of the metaphorical samples. In addition, we notice that EPE scores on relevance are relatively weak (e.g., on avg, -0.538 on EPE and -0.285 on DPE). This suggests that LLMs have difficulty understanding the metaphorical knowledge in the examples. Finally, the overall ratings of the different method-generated samples on non-metaphor were always higher than those of the metaphor samples (e.g., on GT, Literal 4.054 vs. Metaphor 3.828 on Clarity and Literal 4.075 vs. Metaphor 3.387 on Relevance). This also reflects the relative weakness of ChatGPT in its ability to generate metaphor samples.

6 Automatic Evaluation

This experiment uses automatic evaluation to explore the similarity between ChatGPT-generated samples and manually labeled samples. We used three automatic evaluation metrics, Bleu, Rouge, and Meteor, to measure the degree of similarity between LLM-generated and manually labeled samples (see Appendix 13.2 for a detailed description).

Method	Automatic Evaluation			
	Bleu	Rouge	Meteor	Avg
DG to GT	0.111	0.149	0.305	0.188
EPE to GT	**0.194**	**0.212**	**0.348**	**0.251**
DPE to GT	0.115	0.156	0.313	0.195
SPE to GT	0.131	0.142	0.275	0.183

Table 4: The result of the automatic evaluation. Bleu and Rouge are Bleu-1 and Rouge-1, respectively. The automatic evaluations are all referenced to the manually labeled samples.

Result. The results of the experiments are presented in Table 4. The EPE method reached its maximum values on three metrics (e.g., EPE 0.194 vs. SPE 0.131 on Bleu and EPE 0.212 vs. DPE 0.156 on Rouge and EPE 0.251 vs. DPE 0.195 on Meteor). This suggests that the method of introducing the examples was able to guide ChatGPT to generate samples similar to the examples, but similarity does not mean that the metaphor was understood (see the manual evaluation analysis). Additionally, we observed that direct generation was more similar to using the defined DPE approach on three metrics (i.e., DPE 0.115 vs. DG 0.111 on Bleu and DPE 0.156 vs. DG 0.149 on Rouge and DPE 0.313 vs. DG 0.305 on Meteor). This suggests that the direct definition-giving approach minimizes the disturbance of external knowledge while improving the metaphor comprehension of ChatGPT. Comparatively, the EPE and SPE methods are more variable.

7 Case Study

Based on the above experimental analysis, despite the huge cost advantage of the ChatGPT method, there are still some problems with the samples it generates, which can be summarized into three categories: the misinterpretation of conventional meaning (MCM), the neglect of metaphorical evolution (NME) and polysemy confusion (PC). Examples of problems in these three categories are listed in Table 5.

MCM states that ChatGPT incorrectly interprets the conventional meaning as a literal use. For ex-

Types	DG	EPE	DPE	SPE
MCM	The account manager was responsible for maintaining relationships ···	Taking into account the increasing number of car accidents ···	I will need to account for all the expenses before submitting the budget report.	The meticulous accountant carefully accounted for every penny ···
NME	The sun rose, painting the sky with yellow, as if expecting a glorious day ahead.	The sunflower, reaching for the sky, expects a warm embrace from the sun.	She found that exceeding expectations was not as difficult as she had anticipated.	It's natural to expect professionalism and competence from our employees ···
PC	Being the winner entitled him to a cash prize.	··· as the ancient philosophers entitled them.	The painting was entitled "Starry Night" by Vincent ···	··· entitles you to receive a certificate of achievement.

Table 5: Common Errors Showcase. **MCM** stands for misinterpretation of conventional meaning. **NME** stands for neglect of metaphorical evolution. **PC** stands for polysemy confusion. the example of MCM requires ChatGPT to generate the literal usage of "account", and the examples of NWE and PC require the metaphorical usage of "expect" and the literal usage of "entitle", respectively.

ample, the literal use of "account", which originally meant "counting", evolved into "customer or client having an account" or "statement answering for conduct". However, due to the customized meaning of "having an account", ChatGPT misinterprets it as literal. In the MCM example, the DPE generated accurately, interpreting it as "counting"; DG and EPE misinterpreted "having an account" as literal, and SPE directly generated "accounting".

NME stated that ChatGPT often creates metaphors by anthropomorphizing elements of nature, while ignoring the evolution of metaphors. Take the metaphorical usage of "expect" as an example, its initial meaning is "long for, anticipate", which is later extended to mean "the expected changes in the economy and stock market". In the NME example, DG and EPE ignore the evolution of metaphors and construct inappropriate metaphors (e.g., "sun expects", "sunflower expects") through anthropomorphism. There are many such examples generated by the DG method. Differently, DPE and SPE did not find metaphorical meanings, and misidentified "long for, anticipate" as metaphorical.

PC indicated that too many lexical variations led to confusion in the understanding of metaphors in ChatGPT. Take the literal usage of "entitle" as an example, its original meaning is "to give a title to a chapter, book" or "give a title or name to". which is later extended to "bestow an office" or "give (someone) property". Entitle obviously has more literal and derived meanings than other words. In the PC example, DG and SPE generate the wrong interpretation of "have the right to", while EPE correctly translates it as "give a title or name to" due to the use of manually labeled samples as examples. DPE was also correctly interpreted as 'give a title or name to'.

8 Conclusion

This work investigate how to generate metaphor samples using ChatGPT. We propose definition-based prompt enhancement (DPE) and semantics-based prompt enhancement (SPE) methods. Experimental results show that our proposed methods achieve the best performance when using different LLMs as sample generators. Moreover, in the case where we used the oxford dictionary as an information source for multi-lexical knowledge, the fine-tuning performance of the SPE method is close to the manually labeled sample case at only 1/250th of the cost of the latter. We then extensively compare the similarities and differences between the different generative methods and the manually labeled real samples using manual evaluation, automatic evaluation, and case study.

9 Limitations

This paper investigate the problem of how to generate a metaphorical dataset using ChatGPT and propose a semantics-based prompt enhancement (SPE). The method relies on the knowledge of word meanings in wordnet, which brings some overhead. Example analysis reveals that there are still a number of problems with the current samples generated using ChatGPT, which are broadly classified into three categories: the misinterpretation of conventional meaning (MCM), the neglect of metaphorical evolution (NME), and the polysemy confusion (PC). Addressing these issues still requires improvements in generating sources (ChatGPT) as well as prompt design methods. In future work, we will aim to explore ways to minimize the reliance on manual annotation or the use of external databases, and to ensure the quality of metaphorical sample generation.

10 Ethics Statement

In this paper, we detail how ChatGPT was utilized to generate the metaphorical dataset. The datasets used and the research papers cited were obtained from publicly available sources, and we strictly adhere to academic and research ethics guidelines to ensure the legitimacy and transparency of the research process. We place particular emphasis on transparency and openness of information, and are committed to providing clear methodological descriptions and experimental details so that other researchers can understand and reproduce our research. We encourage other researchers in our academic community to conduct responsible research and adhere to best practices in knowledge sharing to advance the continued development of the field. Through open information sharing, we expect to foster broader collaboration and deeper understanding of the metaphor detection task.

References

Meysam Alizadeh, Maël Kubli, Zeynab Samei, Shirin Dehghani, Juan Diego Bermeo, Maria Korobeynikova, and Fabrizio Gilardi. 2023. Open-source large language models outperform crowd workers and approach chatgpt in text-annotation tasks. *arXiv preprint arXiv:2307.02179*.

Mohammad Belal, James She, and Simon Wong. 2023. Leveraging chatgpt as text annotation tool for sentiment analysis. *arXiv preprint arXiv:2306.17177*.

Julia Birke and Anoop Sarkar. 2006. A clustering approach for nearly unsupervised recognition of nonliteral language. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 329–336.

Eugene Charniak, Don Blaheta, Niyu Ge, Keith Hall, John Hale, and Mark Johnson. 2000. Bllip 1987-89 wsj corpus release 1. *Linguistic Data Consortium, Philadelphia*, 36.

Minjin Choi, Sunkyung Lee, Eunseong Choi, Heesoo Park, Junhyuk Lee, Dongwon Lee, and Jongwuk Lee. 2021. Melbert: Metaphor detection via contextualized late interaction using metaphorical identification theories. *arXiv preprint arXiv:2104.13615*.

B Edition, BNC Baby, and BNC Sampler. British national corpus.

Christiane Fellbaum. 1998. *WordNet: An electronic lexical database*. MIT press.

Huawen Feng and Qianli Ma. 2022. It's better to teach fishing than giving a fish: An auto-augmented structure-aware generative model for metaphor detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 656–667.

Jiahui Gao, Renjie Pi, Yong Lin, Hang Xu, Jiacheng Ye, Zhiyong Wu, Weizhong Zhang, Xiaodan Liang, Zhenguo Li, and Lingpeng Kong. 2022. Self-guided noise-free data generation for efficient zero-shot learning. *arXiv preprint arXiv:2205.12679*.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. Chatgpt outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120.

Hongyu Gong, Kshitij Gupta, Akriti Jain, and Suma Bhat. 2020. Illinimet: Illinois system for metaphor detection with contextual and linguistic information. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 146–153.

Thomas Hartvigsen, Saadia Gabriel, Hamid Palangi, Maarten Sap, Dipankar Ray, and Ece Kamar. 2022. Toxigen: A large-scale machine-generated dataset for adversarial and implicit hate speech detection. *arXiv preprint arXiv:2203.09509*.

Fan Huang, Haewoon Kwak, and Jisun An. 2023. Is chatgpt better than human annotators? potential and limitations of chatgpt in explaining implicit hate speech. In *Companion proceedings of the ACM web conference 2023*, pages 294–297.

Beata Beigman Klebanov, Chee Wee Leong, E Dario Gutierrez, Ekaterina Shutova, and Michael Flor. 2016. Semantic classifications for detection of verb metaphors. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–106.

Bartłomiej Koptyra, Anh Ngo, Łukasz Radliński, and Jan Kocoń. 2023. Clarin-emo: Training emotion recognition models using human annotation and chatgpt. In *International Conference on Computational Science*, pages 365–379. Springer.

George Lakoff and Mark Johnson. 2008. *Metaphors we live by*. University of Chicago press.

Md Tahmid Rahman Laskar, Mizanur Rahman, Israt Jahan, Enamul Hoque, and Jimmy Huang. 2023. Cqsumdp: a chatgpt-annotated resource for query-focused abstractive summarization based on debatepedia. *arXiv preprint arXiv:2305.06147*.

Duong Le, My Thai, and Thien Nguyen. 2020. Multi-task learning for metaphor detection with graph convolutional neural networks and word sense disambiguation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8139–8146.

Chee Wee Leong, Beata Beigman Klebanov, Chris Hamill, Egon Stemle, Rutuja Ubale, and Xianyang Chen. 2020. A report on the 2020 vua and toefl metaphor detection shared task. In *Proceedings of the second workshop on figurative language processing*, pages 18–29.

Chee Wee Leong, Beata Beigman Klebanov, and Ekaterina Shutova. 2018. A report on the 2018 vua metaphor detection shared task. In *Proceedings of the Workshop on Figurative Language Processing*, pages 56–66.

Zhuang Li, Levon Haroutunian, Raj Tumuluri, Philip Cohen, and Gholamreza Haffari. 2024. Improving cross-domain low-resource text generation through llm post-editing: A programmer-interpreter approach. *arXiv preprint arXiv:2402.04609*.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

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. *arXiv preprint arXiv:1907.11692*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Rui Mao and Xiao Li. 2021. Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 13534–13542.

Yu Meng, Jiaxin Huang, Yu Zhang, and Jiawei Han. 2022. Generating training data with language models: Towards zero-shot language understanding. *Advances in Neural Information Processing Systems*, 35:462–477.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Saif Mohammad, Ekaterina Shutova, and Peter Turney. 2016. Metaphor as a medium for emotion: An empirical study. In *Proceedings of the Fifth Joint Conference on Lexical and Computational Semantics*, pages 23–33.

Etienne Ollion, Rubing Shen, Ana Macanovic, and Arnault Chatelain. 2023. Chatgpt for text annotation? mind the hype! *SocArXiv. October*, 4.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Punyajoy Saha, Aalok Agrawal, Abhik Jana, Chris Biemann, and Animesh Mukherjee. 2024. On zero-shot counterspeech generation by llms. *arXiv preprint arXiv:2403.14938*.

Ekaterina Shutova, Douwe Kiela, and Jean Maillard. 2016. Black holes and white rabbits: Metaphor identification with visual features. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: Human language technologies*, pages 160–170.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Wei Song, Shuhui Zhou, Ruiji Fu, Ting Liu, and Lizhen Liu. 2021. Verb metaphor detection via contextual relation learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4240–4251.

Gerard Steen, Aletta G Dorst, J Berenike Herrmann, Anna Kaal, Tina Krennmayr, Trijntje Pasma, et al. 2010. A method for linguistic metaphor identification. *Amsterdam: Benjamins*.

Chuandong Su, Fumiyo Fukumoto, Xiaoxi Huang, Jiyi Li, Rongbo Wang, and Zhiqun Chen. 2020. Deepmet: A reading comprehension paradigm for token-level metaphor detection. In *Proceedings of the second workshop on figurative language processing*, pages 30–39.

Hongjin Su, Jungo Kasai, Chen Henry Wu, Weijia Shi, Tianlu Wang, Jiayi Xin, Rui Zhang, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, et al. 2022. Selective annotation makes language models better few-shot learners. *arXiv preprint arXiv:2209.01975*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. Stanford alpaca: An instruction-following llama model.

Solomon Ubani, Suleyman Olcay Polat, and Rodney Nielsen. 2023. Zeroshotdataaug: Generating and augmenting training data with chatgpt. *arXiv preprint arXiv:2304.14334*.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. 2021a. Want to reduce labeling cost? gpt-3 can help. *arXiv preprint arXiv:2108.13487*.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2022. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*.

Zirui Wang, Adams Wei Yu, Orhan Firat, and Yuan Cao. 2021b. Towards zero-label language learning. *arXiv preprint arXiv:2109.09193*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, M Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. arxiv. *arXiv preprint arXiv:1910.03771*.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. Zerogen: Efficient zero-shot learning via dataset generation. *arXiv preprint arXiv:2202.07922*.

Kang Min Yoo, Dongju Park, Jaewook Kang, Sang-Woo Lee, and Woomyeong Park. 2021. Gpt3mix: Leveraging large-scale language models for text augmentation. *arXiv preprint arXiv:2104.08826*.

Shenglong Zhang and Ying Liu. 2023. Adversarial multi-task learning for end-to-end metaphor detection. *arXiv preprint arXiv:2305.16638*.

11 Fine-tuning Datasets

Among the fine-tuning experiments, we use the metaphor samples generated by LLM as the training set to fine-tune RoBERTa. and then test them on three metaphor datasets, VUAverb, TroFi and MOH-X, respectively.

VUAverb. The VU Amsterdam Metaphor Corpus (VUAMC) (Steen et al., 2010) metaphorically annotates each lexical unit in a subset of the British National Corpus (Edition et al.), and the annotation was done using the MIPVU program. Based on VUAMC, several different variants of the VUA corpus have emerged, among which VUAverb is the verb version of the VUA corpus. This paper uses the VUAverb dataset mentioned in the metaphor detection shared task (Leong et al., 2018, 2020), which contains 15516 training samples and 5873 test samples.

VUAverb Cuts. VUAverb has the problem of long-tailed distribution. for example, the target words "say" and "go" contain 509 and 506 samples respectively, while the number of most verbs is very small. According to statistics, among the 1875 verbs in the VUAverb training set, there are only 257 verbs with number greater than 10 (13.7% of the total), while there are 781 verbs with number equal to 1 (41.7% of the total). To mitigate the long-tailed distribution, we trimmed the VUAverb train. Specifically, we first filtered out the target word categories with sample sizes larger than 10, and then randomly selected 10 of them as the final samples of the category. After such processing, we finally obtained 7,900 pieces of data, which will be used as crowdsourced annotations (CA) data for subsequent experiments.

TroFi. TroFi (Birke and Sarkar, 2006) is a verb-target focused dataset containing the literal and metaphorical usage of 50 English verbs from the 1987-1989 Wall Street Journal corpus (Charniak et al., 2000). We use the same version of TroFi as Choi et al. (2021) and Zhang and Liu (2023), which contains a total of 3739 samples. These samples cover rich verb instances and provide diverse contextual information.

MOH-X. The MOH dataset was created by Mohammad et al. (2016), and its construction methodology involves first extracting polysemous verb samples from wordnet, and then metaphorically labeling the sentences via a crowdsourcing platform. To ensure the quality of the dataset annotation, Mohammad et al. (2016) adopted a 70% annotation consistency criterion. A subset of MOH, MOH-X (Shutova et al., 2016), contains 649 samples and is a commonly used dataset in mainstream metaphor detection systems (Choi et al., 2021; Zhang and Liu, 2023). This subset excludes instances with pronouns, dependent subjects or objects. Therefore, we use MOH-X for model evaluation.

12 Prompt Designs

12.1 Direct Generation Method

Prompt: Generate $\mathbf{n_{k,i}}$ sentences in different styles containing the specified verb based on the explanation, where the verb are used $\mathbf{metaphorically}$. word: $\mathbf{w_k}$ s-1: ......

Table 6: DG prompt.

The DG approach aims to direct ChatGPT to generate samples of a specified type without using external knowledge content. For input, $w_k, y_k, n_{k,i}$ represent the target word, label, and the number of samples to be generated, respectively. ($n_{k,i}$ is the same as the number of samples in the same group in VUAverb cut). $i = 0$ or 1 corresponds to $y_k = 0$, $y_k = 1$, respectively, indicating that the target word is literal, metaphorical usage. The specific prompt design is shown in Table 6.

12.2 Example-based Prompt Enhancement Method

Prompt: Generate $\mathbf{n_{k,i}}$ sentences in different styles containing the specified verb based on the explanation, where the verb are used $\mathbf{metaphorically}$. word: $\mathbf{w_k}$ example: $\mathbf{d_{k,i}}$ s-1: ......

Table 7: EPE prompt.

Example-based prompt enhancement (EPE) methods are commonly used techniques for prompt learning. For example, Yoo et al. (2021); Wang et al. (2021b) provide one or more examples and category labels for each category of a particular task. Inspired by the above, this paper introduces the EPE method and adapts it for metaphorical features. First, we notate the sample set of all available examples (i.e., the VUAverb cut) as $\mathcal{D} = (x_i, w_i, y_i)|1 \leq i \leq N$, where x_i, w_i, and y_i are the text, the target word, and the corresponding labels, respectively. In then, we classify $\mathcal{D}$ into subsets $\mathcal{D}_{ki}$ based on the target word w_k and the corresponding label y_k, where $i = 0$ or 1 denotes the literal, metaphorical usage, respectively. For each category $\mathcal{D}_{k,i}$, we randomly select a sample $d_{k,i}$ as an example. Finally, $d_{k,i}$ will be used as a prompt message in the prompt.

13 Evaluation Metrics

13.1 Manual Evaluation Metrics

In the manual evaluation experiments on ChatGPT generated samples, we used clarity, relevance, and diversity as evaluation metrics, and their specific meanings are:

- **Clarity**: the ease with which a metaphor can be understood. The greater the number of samples in the same sample set where it is easier to judge the metaphor, the higher the clarity.
- **Relevance**: whether the category (metaphorical or literal) in which the sample is labeled matches the actual usage of the sample. The greater the number of matching samples in the same sample group, the greater the correlation.
- **Diversity**: whether the usage of the sample metaphors (often expressed in different word meanings) is diverse within the same group. For example, "catch" is "to win someone's affection or love" in "catch someone's heart" and "to attract someone's attention" in "catch someone's eye".

13.2 Automatic Evaluation Metrics

In the automated evaluation experiments on ChatGPT generated samples, we used Bleu, Rouge and Meteor as evaluation metrics, and their specific meanings are:

- **Bleu**: Bleu calculates how well the LLM output matches the real samples on n-grams of different lengths. We use the nltk (Loper and

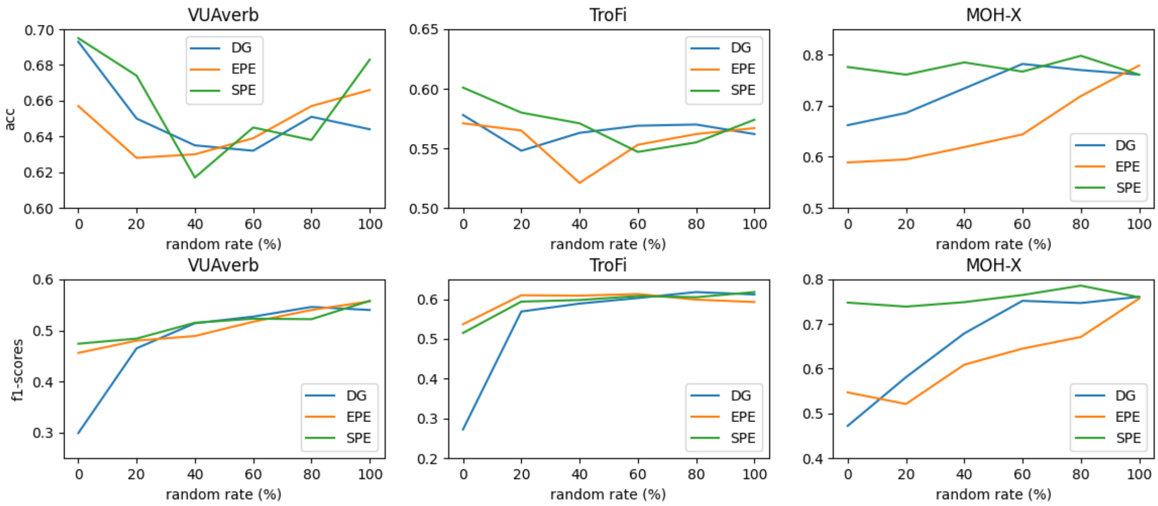

Figure 3: Plot of the results of the sample fusion experiment. The experiment aims to investigate the impact of the performance of the three methods DG, EPE and SPE on the test set after the gradual introduction of manually labeled samples. The top, bottom graphs show the relationship between accuracy, F1 score and the percentage of manually labeled samples, respectively.

Bird, 2002) tool to calculate Bleu-1 for generated samples and manually labeled samples separately.

- **Rouge**: Rouge is similar to Bleu and also uses the n-gram computation method, but turns precision into recall. In this paper, we use the ROUGE_score library function in python to calculate ROUGE-1.

- **Meteor**: Meteor is an improved version of Bleu, which performs finer-grained matching by taking into account lexical variations (e.g., roots, synonyms) and word order. Again the nltk (Loper and Bird, 2002) tool was used for the computation.

14 Sample Fusion Experiment

This experiment explores the effects of three methods, DG, EPE and SPE, on the performance of the test set after gradually introducing manually labeled samples (GT). We designed six experiments to examine different combinations of generated and GT samples with different percentages: 100% generated samples + 0% GT samples, 80% generated samples + 20% GT samples, 60% generated samples + 40% GT samples, 40% generated samples + 60% GT samples, 20% generated samples + 80% GT samples, and 0% generated samples + 100% GT samples. In the experiments, we randomly selected percentages in terms of target word categories (target word + label), and if the number of group samples was less than the number of samples to be extracted, the method of repeated extraction was used.

results. On both VUAverb and TroFi (see Figure 3 a,b), the introduction of the original sample at the beginning leads to a decrease in accuracy. This suggests that the difference in the distribution of the generated samples and the original samples affects the model's ability to learn metaphorical information, which leads to the opposite effect. In contrast, compared to DG and SPE, EPE has an early turning point in the decline of VUAverb-Acc, and its performance starts to increase after 20%. This is due to the fact that the examples of the EPE method are derived from VUAverb. However, Acc is also able to improve as the original data share continues to increase. Moreover, the F1 values of the three methods in each dataset also show a general upward trend (see Figure 3 d,e,f). This indicates that the introduction of the original sample can improve the ability of the model model to capture metaphorical information.

In addition, since the DG method has a low performance, the introduction of a small number of proto-samples can achieve a high F1 performance improvement (e.g., 100% DG + 0% GT 0.299 vs. 80% DG + 20% GT 0.465 on VUAverb and 100% DG + 0% GT 0.272 vs. 80% DG + 20% GT 0.569 on TroFi). The EPE and SPE originally had not-so-low F1 values, so the introduction of a small number of original samples yielded little in terms of performance improvement.

Overall, the introduction of manually labeled data on top of the ChatGPT generated data is related to the performance of the generated data on the test set. On the one hand, researchers may not be able to construct prompts that are suitable for certain general tasks. therefore, they often generate samples directly using ChatGPT. This situation makes it possible to introduce partially manually labeled data, and by paying a small portion of the cost of manual labeling, the samples can quickly catch up in performance with the performance of the samples generated by the customized prompt. On the other hand, if the researcher is able to design a reasonable prompt based on a specific task (e.g., the SPE method proposed in this paper). As it performs well on the test set. Therefore, the introduction of some of the original sample data may lead to performance degradation due to factors such as distribution mismatch, or yield little results. In this regard, the second case is not used to introduce manually labeled samples.