Is ChatGPT a Smart Data Generation Tool? Exploring ChatGPT for Generating Metaphorical Data

Anonymous ACL submission

Abstract

Data annotation is a time-consuming and labor-intensive task, with an average annotation cost of $0.11 per instance on crowdsourcing platforms. This high cost has become a constraint for further development of many researches. As large-scale language models (LLMs) have made significant progress in many tasks, researchers have begun to experiment with the use of prompt learning to generate samples. However, previous studies have mainly focused on surface semantic tasks and neglected in-depth studies of implicit semantic tasks (e.g., metaphors), which require LLMs to provide a deeper understanding of the implicit meanings in text. Therefore, the aim of this paper is to explore the data generation capabilities of ChatGPT in dealing with metaphorical tasks. In previous surface semantic tasks, researchers usually use direct generation of samples (DG) and example-based prompt enhancement (EPE) methods. We propose a sematic-based prompt enhancement (SPE) method. Experiments demonstrate that the SPE method has the best F1 performance on three datasets and exceeds the accuracy of crowdsourced annotations (CA) samples on two datasets. Finally, we provide an in-depth analysis and discussion of the three ChatGPT sample generation methods through extensive example analysis and experiments.

1 Introduction

Metaphors, as a unique way for people to understand the world, help understand vague and abstract concepts in the source domain by extracting familiar concepts in the target domain (Lakoff and Johnson, 2008). However, current metaphor detection systems often use supervised methods that rely on high-quality manually labeled data. According to a survey (Wang et al., 2021a), the average labeling cost per instance on crowdsourcing platforms is as high as $0.11. Comparatively, generating samples using large language model (e.g., GPT3.5-turbo) APIs becomes a more cost-effective alternative, costing only 0.05 per 1M token input and 0.15 per 1M token output, respectively. Therefore, this raises an interesting question: how can ChatGPT be effectively guided to generate high-quality sample data?

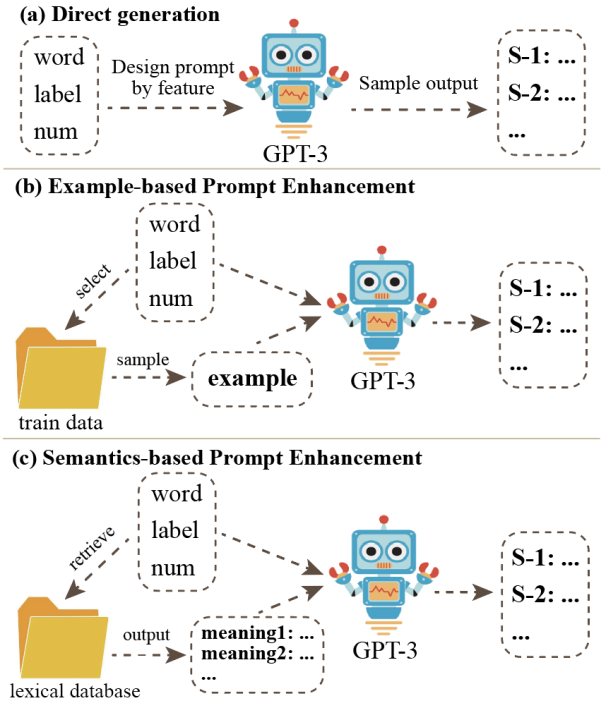

Figure 1: (a) is direct sample generation using ChatGPT. (b) is the example-based prompt enhancement (EPE) method, where examples are added to the prompt. (c) is the sematic-based prompt enhancement (SPE) method we proposed, which uses multiple word senses.

Initially, LLMs were studied mainly through fine-tuning. McCann et al. (2018) and Rajani et al. (2019) used decoders to generate correct responses with question and context. Trinh and Le (2018) and Petroni et al. (2019) used encoders that employ a completionist approach to guide the model in generating the required answers. For example, "Donald Trump is [MASK]", where "[MASK]" can be: "former president" or "businessman". With the

development of LLMs, the emergence of GPT-3 (Brown et al., 2020) has significantly improved the ability of sample generation. However, its large number of parameters also brings the problem of difficult fine-tuning. In contrast, prompt learning, with its non-invasive nature and no need for model fine-tuning, has become a new approach to explore sample generation. In this area, researchers have guided models to generate multiple samples of the same kind through prompts and labels (Ye et al., 2022; Meng et al., 2022). Yoo et al. (2021); Wang et al. (2021b) design generic templates and provide examples to guide GPT-3 to generate similar data while adapting to multiple downstream tasks.

The above approaches have brought new research ideas to sample generation tasks. However, these studies mainly focus on data generation for surface language tasks, which usually only require models to learn information about lexical and syntactic structures. In contrast, implicit semantic tasks (e.g., metaphors, sarcasm) are more complex and require in-depth understanding of the implicit meanings in the text. In past studies, Chakrabarty et al. (2022) attempted to generate metaphor samples using GPT-3, but manual checking is required.

Inspired by ChatGPT's excellent performance on zero- or few-sample NLP tasks, we consider utilizing ChatGPT's world knowledge to generate metaphor samples. Therefore, this paper aims to apply ChatGPT to metaphor sample generation. We design a sematic-based prompt enhancement (SPE) method based on word meanings, targeting the properties of metaphors. SPE does not rely on manually labeled samples, and only requires the introduction of the WordNet (Miller, 1995; Fellbaum, 1998). In addition, we introduce ChatGPT direct generation (DG) and example-based prompt enhancement (EPE), as well as crowdsourced annotations (CA) samples. Finally, we conduct extensive experiments and example analysis on these four samples. Overall, our contributions are summarized below:

1. To the best of our knowledge, this is the first study to apply ChatGPT to metaphorical sample generation. We conducted extensive experiments and analysis on samples generated by the three ChatGPT methods and manually labeled samples.

2. For the characteristics of metaphors, we design a sematic-based prompt enhancement (SPE) method. Experimental results show that SPE achieves the best performance on all three datasets compared to direct generation (DG) and example-based prompt enhancement (EPE) methods.

3. We give example analyses of samples generated by the three ChatGPT methods, and summarize the current problems of generating metaphor samples by ChatGPT into three categories: the misinterpretation of conventional meaning, the neglect of metaphorical evolution and polysemy confusion.

4. We provide automatic and manual evaluation of samples generated by the three ChatGPT methods and crowdsourced annotations (CA) samples, and provide an in-depth discussion of the results of several experiments and example analyses.

2 Related Work

2.1 Large Language Modeling

The core principle of large-scale language modeling (LLM) lies in revealing the tacit knowledge in the model by simulating task-specific linguistic environments. Since the introduction of the self-attention mechanism (Vaswani et al., 2017), the field of LLM has made a vigorous development. In the research, BERT (Devlin et al., 2018), which uses the Transformer encoder architecture, and GPT (Radford et al., 2018), which uses the Decoder architecture, have emerged. On the basis of BERT, many remarkable variants have emerged, such as RoBERTa (Liu et al., 2019) and DeBERTa (He et al., 2020). And the emergence of GPT-3 (Brown et al., 2020), a third-generation model based on the decoder structure with 175 billion parameters, 10 times more than any previous non-sparse language model, has changed the landscape of LLM.

2.2 Prompt Learning

The goal of prompt learning is to guide the LLM in a non-fine-tuned manner to generate specific content. In this task, the LLM plays the role of a sample less or zero sample learner. Past studies are usually categorized into two main groups: generating annotations and generating samples. Ye et al. (2022) and Meng et al. (2022) used the method of adding polarity labels to prompts to guide the model to a specified tendency. For example, a prompt can be constructed such as "Movie reviews

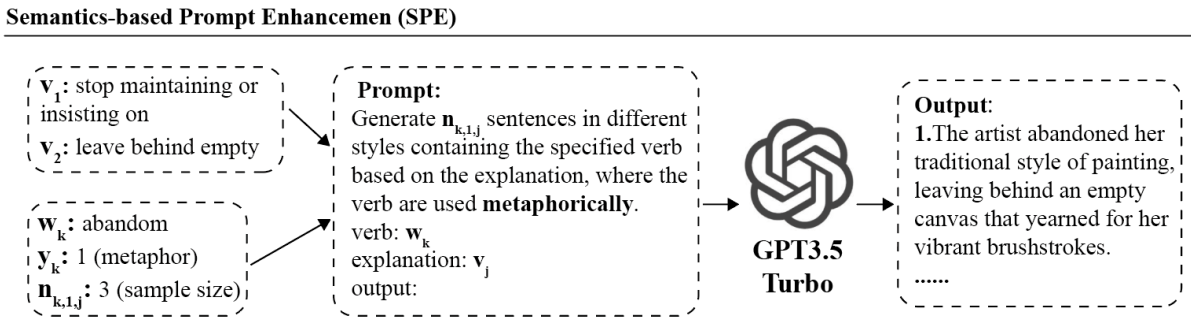

Figure 2: The prompt design of the SPE method. w_k denotes the target word, y_k is the label, and v_j denotes the jth meaning of the target word w_k. $n_{k,i,j}$ is the number of samples to be generated for the jth meaning of the target word w_k. $i = 0$ or 1 corresponds to $y_k = 0$, $y_k = 1$, respectively, which indicates that the target word is a literal, metaphorical usage.

with **positive** sentiment are". Wang et al. (2021a) proposed an approach that combines manual and LLM labeling to mitigate the cost. Yoo et al. (2021) designed a template to guide the model for sample annotation or sample generation by introducing instances of different tasks. Lang et al. (2022) designed a joint training framework of GPT-3 and BERT for the labeling of classification tasks.

2.3 Metaphor Detection

For the task of specifying target words and their corresponding contexts, metaphor detection aims to determine whether the target 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 dichotomous subtask by directly taking the most common definitions of words in WordNet as literal meanings.

3 Method

We investigate three ChatGPT sample generation methods: SPE, DG and EPE. our proposed SPE method is described in Section §3.2, and the prompt designs for the DG and EPE methods are described in Appendices 11.1 and 11.2, respectively.

3.1 GPT-3 Labeling

We choose GPT3.5-turbo-1106 (hereinafter referred to as Turbo3.5), which is released by OpenAI, for data labeling. In the labeling process, we first design the prompt according to the metaphor detection task. Subsequently, we filled in the gaps in the prompt according to different target words, labels and sample sizes. Specifically, for the target word w_k and label y_k, there are:

$$\{x'_n\}_{n=1}^{N_k} = \text{Turbo3.5}(prompt, w_i, y_i, n_i), \quad (1)$$

where x'_n denotes the nth sample generated by centering on the target word w_i, whose metaphoricity is related to the label y_i. Specifically, when $y_i = 1$ or $y_i = 0$, the target word w_i behaves as metaphorical or non-metaphorical in the sample set $\{x'_n\}_{n=1}^{N_k}$, respectively.

3.2 Semantics-based Prompt Enhancement

Lexical Meaning Search. In metaphor detection tasks, WordNet (Miller, 1995; Fellbaum, 1998) is a commonly used external knowledge base by researchers and has been shown to help improve metaphor detection performance (Gong et al., 2020; Klebanov et al., 2016; Zhang and Liu, 2023). Zhang and Liu (2023). Inspired by these studies, we utilizes WordNet to obtain multiple meanings of target words. 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 $v_j \in \mathcal{V}_k$:

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

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 construction method is illustrated in Figure 2. For the input (w_k, y_k), we first specify $word = w_k$. Then, depending 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 $y_k = 0$ and $y_k = 1$, respectively. Unlike the DG and EPE approaches, we consider the literal lexical sense set $\mathcal{V}_{k,l}$ and the metaphorical lexical sense set $\mathcal{V}_{k,m}$ of the target word w_k. Specifically, we first divide based on the number of samples to be generated, for $y_k = 1$ there are:

$$n_{k,1,j} = \text{ceil}(\frac{n_{k,1}}{|\mathcal{V}_{k,m}|}), \quad (3)$$

where ceil is an upward rounding function, $|\mathcal{V}_{k,m}|$ denotes the number of metaphorical lexemes, $n_{k,1,j}$ 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 Experiment

4.1 Dataset

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.

Dataset	Tokens	Sentences	% Met.
DG_tr	106833	7921	34.2%
EPE_tr	140143	7720	34.8%
SPE_tr	168003	8027	37.4%
VUA_tr	245706	7900	34.1%
VUA_de	83660	2935	30.1%
VUA_te	83915	2940	29.8%
TroFi_de	60763	1870	43.5%
TroFi_te	60539	1869	43.5%
MOH-X_de	2722	317	50.5%
MOH-X_te	2880	332	46.7%

Table 1: Dataset statistics. tr: training set. de: dev set. te: test set. tokens: number of vocabulary units or samples to be tested. sent.: total number of sentences, %Met.: proportion of metaphor samples to total samples

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.

4.2 Experimental Setup

Experiment 1. Two pre-trained models, BERT (Devlin et al., 2018) and RoBERTa (Liu et al., 2019), were considered and initialized with weight parameters from the Huggingface library (Wolf et al., 2019). The output of the model adopts part of the model idea designed in Choi et al. (2021), i.e., the hidden layer output corresponding to the target word is used for classification. In the experimental part, we first trained BERT and RoBERTa on DG, EPE, SPE, and CA samples, respectively, and then validated them on the test set. We chose three datasets, VUAverb test, TroFi and MOH-X, as test sets. Due to the lack of validation sets, we divided the above three datasets according to a 1:1 ratio of lexical types (e.g., "go", "get") and labels (0 or 1). Eventually, the number of validation and test sets for TroFi is 1870 and 1869, for MOH-X is 317 and 332, respectively, and for VUAverb-test is 2935 and 2940, respectively.The final samples used for training are shown in Table 1.

Experiment 2. Experiment 2 demonstrates the cost required for the three ChatGPT sample generation methods, DG, EPE, and SPE, and the crowdsourced annotations (CA) samples. For CA, we use the manual labeling cost recorded in Wang et al. (2021a), which is $0.11 per sample. For DG, EPE and 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. For the cost, we use the token price given in the official OpenAI website as the auto-labeling cost [1]. The input is $0.5 per 1M tokens and the output is $1.5 per 1M tokens.

Experiment 3. Experiment 3 investigates the effects of the three methods, DG, EPE and SPE, on the performance of the test set after the gradual introduction of CA samples. We designed six experiments that examined different combinations of generation samples and CA samples with different percentages: generation samples 100% + CA samples 0%, generation samples 80% + CA samples 20%, generation samples 60% + CA samples 40%, generation samples 40% + CA samples 60%, generation samples 20% + CA samples 80%, and generation samples 0% + CA samples 100%. In the experiments, we randomized the percentage of the target word category (target word + label), and if the number of group samples was smaller than the number of samples required to be extracted, repeated extraction was used. Please refer to Appendix 11.3 for detailed analysis of the experimental results.

[1]OpenAI cost link: https://openai.com/pricing. The model version is GPT3.5-turbo-1106 with a record date of 2024.3.

4.3 Implementation Details

All experiments in this paper use the Adam optimizer (Kingma and Ba, 2014), initialized with a learning rate of 3e-5 and a dropout rate of 0.2. The batch sizes for training, validation, and testing were set to 100. the maximum length of a sentence was 150 tokens, the metaphor weights were set to 5, and the maximum epoch was set to 25. to prevent the model from being underfitted at the beginning of the training period, we only save the epoch$\leq$14 or the model when overall loss is less than 2. We used the model weights that reached the maximum F1 value in the validation set for testing. In addition, all experiments were run on a cloud server equipped with a single card A100 80G GPU.

5 Analysis of results

Experiment 1. The experimental results are presented in Table 2. Compared to DG and EPE, our proposed SPE method achieves the best performance on F1 for all three datasets (e.g., on RoBERTa, SPE 0.488 vs. EPE 0.454 on VUAverb and SPE 0.518 vs. EPE 0.498 on TroFi and SPE 0.723 vs. 0.441 on MOH-X) with a p-values of 0.039 (<0.05). This proves the superiority of SPE. However, SPE still falls short compared to CA (e.g., on RoBERTa, -0.067 on VUAverb and -0.1 on TroFi and -0.054 on MOH-X) with p-values of 0.017 (<0.05). This is demonstrated by the fact that the ChatGPT generation method is much lower than CA in the Rec metric. indicating that the ChatGPT method generates a poor diversity of metaphor samples, which prevents the model from learning enough metaphor information.

However, the SPE method differs very little from CA on Acc and even slightly exceeds it (e.g. on RoBERTa, SPE 0.698 vs. CA 0.658 on VUAverb and SPE 0.589 vs. CA 0.563 on TroFi), with

Dataset		BERT-base				RoBERTa-base			
		DG	**EPE**	**SPE**	**CA**	**DG**	**EPE**	**SPE**	**CA**
VUAverb	**Acc.**	0.701	0.666	0.71	0.732*	0.694	0.66	0.698*	0.658
	F1	0.283	0.434	0.458	0.591*	0.303	0.454	0.488	0.555*
	Pre.	0.496	0.439	0.518	0.542*	0.474	0.436	0.493*	0.454
	Rec.	0.198	0.429	0.41	0.649*	0.222	0.473	0.482	0.713*
TroFi	**Acc.**	0.578	0.565	0.589	0.601*	0.582	0.581	0.589*	0.563
	F1	0.236	0.445	0.466	0.612*	0.27	0.498	0.518	0.618*
	Pre.	0.555*	0.501	0.53	0.53	0.565*	0.509	0.53	0.499
	Rec.	0.15	0.401	0.411	0.723*	0.177	0.49	0.507	0.811*
MOH-X	**Acc.**	0.622	0.52	0.728*	0.713	0.628	0.526	0.77	0.789*
	F1	0.346	0.34	0.648	0.709*	0.376	0.441	0.723	0.777*
	Pre.	0.917*	0.477	0.822	0.674	0.88*	0.492	0.832	0.767
	Rec.	0.213	0.265	0.535	0.748*	0.239	0.4	0.639	0.787*

Table 2: The performance of the samples generated by the three ChatGPT methods was evaluated against manually labeled samples on a test dataset. First, the four samples were fine-tuned using the BERT or RoBERTa models and then evaluated on the VUAverb test, TroFi and MOH-X, respectively.

Mth.	CA	ChatGPT		
	total	**input**	**output**	**total**
CA	**869$**	-	-	-
DG	-	0.060$	0.16$	0.220$
EPE	-	**0.114$**	0.21$	0.324$
SPE	-	0.087$	**0.252$**	**0.339$**

Table 3: Cost statistics. CA stands for crowdsourced annotations and the annotation cost is $0.11 per sample. While ChatGPT method generates cost of $0.5 per 1M tokens input and $1.5 per 1M tokens output.

a p-value of 0.018 (<0.05). This suggests that using SPE as a training sample in a supervised metaphor detection task is not inferior to CA. although the Rec of the SPE method is slightly lower than that of CA, it improves the accuracy of the non-metaphorical samples. Specifically, the SPE method has higher Pre than CA (e.g., on RoBERTa, SPE 0.493 vs. CA 0.454 on VUAverb and SPE 0.53 vs. CA 0.499 on TroFi and SPE 0.832 vs. CA 0.767 on MOH-X), with a p-value of 0.017 (<0.05).

Experiment 2. The cost of generating the samples is shown in Table 3. In total, CA cost $869, which is much higher than $0.22 for DG, $0.324 for EPE, and $0.339 for SPE. This indicates that using ChatGPT to generate metaphor samples has a huge advantage in terms of cost. In connection with the results of Experiment 1, we find that with an increase of only $0.015, SPE achieves the best performance among the three methods, DG, EPE, and SPE, and even achieves a huge F1 improvement on the MOH-X dataset (e.g., SPE 0.648 vs. EPE 0.34 on BERT and SPE 0.723 vs. EPE 0.441 on RoBERTa) with p-value less than 0.001. This proves the superiority of our proposed method. In addition, we observe that the output spend of the three ChatGPT methods correlates with their F1 performance. Since the output spend depends on the number of tokens of the generated samples, this suggests that increasing the length of the generated samples can improve the sample quality.

6 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 4.

MCM states that ChatGPT incorrectly interprets the conventional meaning as a literal use. For example, the literal use of "account", which originally meant "counting", later evolved into "customer or client having an account" or "statement answering for conduct". However, due to the customized

Types	CA	DG	EPE	SPE
MCM	· · · natural hazards account for up to 4 per cent of total deaths · · ·	The account manager was responsible for maintaining relationships · · ·	Taking into account the increasing number of car accidents · · ·	The meticulous accountant carefully accounted for every penny · · ·
NME	The City had been expecting bad figures and the shares rose 15p to 239p.	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.	It's natural to expect professionalism and competence from our employees · · ·
PC	In the fifth group session entitled Focus on the Individual, · · ·	Being the winner entitled him to a cash prize.	· · · as the ancient philosophers entitled them.	· · · entitles you to receive a certificate of achievement.

Table 4: Common errors showcase. CA: crowdsourced annotations samples. DG: ChatGPT direct generation. EPE: example-based prompt enhancement method. SPE: sematic-based prompt enhancement method. CM denotes misinterpretation of conventional meaning. NME denotes neglect of metaphorical evolution. PC denotes polysemy confusion. The MCM example requires ChatGPT to generate the literal usage of "account", while the NME and PC examples require the metaphorical usage of "expect" and the literal usage of "entitle", respectively.

meaning of "having an account", ChatGPT misinterprets it as literal. In the MCM example, the CA is accurately labeled and interpreted as "counting", while the samples generated by DG, EPE, and SPE all contain errors. DG and EPE misinterpreted "having an account" as literal, while SPE directly generated the word "accountant".

NME suggests 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, which initially means "long for, anticipate", and was later extended to mean "the expected changes in the economy and stock market". In the NME example, the CA is accurate, interpreting it as "expected changes in the stock market". However, DG and EPE ignore the evolutionary pattern of metaphors and construct inappropriate metaphors through anthropomorphism (e.g., "sun expects", "sunflower expects"). Such examples abound in other samples generated by the DG method. On the contrary, SPE was influenced by the pre-positioned common meanings, reducing the occurrence of NME.

PC indicated that the ChatGPT's understanding of metaphors is confused due to too many lexical variations. Take the literal usage of "entitle" as an example, its initial meaning is "to give a title to a chapter, book" or "give a title or name to". Later extended to "to bestow an office" or "to give (someone) property". Entitle obviously has more literal and derived meanings than other words. In the PC example, CA is correctly labeled as "give a title or name to". DG and SPE generate an incorrect interpretation as "have the right to". But EPE, which uses the correct usage of CA as an example, also correctly translates it as "give a title or name to".

7 Integrated Assessment

Automatic Evaluation. We use three evaluation metrics, BLEU, METEOR and ROUGE, to measure the degree of similarity between the three ChatGPT methods and CA samples. All three metrics employ the n-gram matching mechanism, but differ slightly in the factors considered. Specifically, BLEU and ROUGE focus on precision and recall, respectively, while METEOR additionally considers information such as synonyms and stems. For each generated sample, we first compute its evaluation value (e.g., BLEU) with the same target words and labeled samples in CA. Then, we select the maximum value from multiple evaluation values as the final evaluation value of this generated sample with respect to the original sample.

Manual Evaluation. The manual evaluation is performed on a group basis, e.g., for samples of

Methods	Automatic Evaluation				Manual Evaluation			
	BLEU	METEOR	ROUGE	avg	Clarity	Relevance	Diversity	avg
CA	-	-	-	-	3.946	3.73	**3.93**	3.869
DG	0.103	0.146	0.303	0.184	**4.519**	**3.93**	3.584	**4.011**
EPE	**0.19**	**0.207**	**0.34**	**0.246**	4.411	3.389	3.643	3.814
SPE	0.123	0.134	0.264	0.174	4.47	3.708	3.784	3.987

Table 5: BLEU, METEOR and ROUGE were used as automated assessment indicators using 1-gram matching. While clarity, relevance and diversity were used as manual assessment methods.

the target word "go" and the label "1". The assessment metrics include clarity, relevance, and diversity, each of which is rated on a scale of 1 to 5. Clarity indicates the comprehensibility of the sample, including whether the text is easy to understand and whether the metaphors are easy to determine. Relevance indicates whether the labeled categories match actual usage. The greater the number of accurate annotations in the same group, the higher the relevance score. Diversity indicates whether the same panel sample contains more and more diverse information, e.g., whether the text descriptions cover different domains (e.g., economics, politics). Based on the above three metrics, three volunteers were invited to evaluate the samples using the CA, DG, EPE, and SPE methods, respectively, and the final results were averaged across the three ratings.

Results. The experimental results show that the EPE method reaches the maximum values on BLUE, METEOR, and ROUGE metrics (e.g., 0.19, 0.207, and 0.246) in the automatic evaluation, indicating that the introduction of the examples is effective in guiding ChatGPT to generate content that is similar to CA samples. In manual evaluation, the ChatGPT method far exceeds the CA samples in terms of clarity. Specifically, the DG, EPE, and SPE methods outperform the CA in terms of clarity by 0.57, 0.46, and 0.52, respectively. suggesting that ChatGPT-generated samples are more comprehensible compared to CA samples. However, in conjunction with the sample analysis, we found that the DG samples often used a "shortcut" approach to create metaphors by anthropomorphizing elements from nature. While this makes the generated metaphors easier to understand (maximum clarity of 4.519 and maximum relevance of 3.93). However, it also significantly reduces the richness of the content of the metaphor samples (minimum diversity value of 3.584).

The results of linkage Experiment 1 show that although DG, EPE, and SPE far outperform CA in terms of clarity, these three ChatGPT methods have relatively low performance on the test set. In addition, in terms of relevance metrics, EPE performs better than DG on the test set, even though EPE is lower than DG (e.g., EPE 3.389 vs. DG 3.93). This suggests that metaphor comprehensibility or labeling accuracy is not sufficient to determine the quality of a metaphor sample. Furthermore, current metaphor detection methods seem to learn only a certain distribution (possibly similar subject-predicate collocations) at the expense of understanding the nature of the metaphor (e.g., whether derivations are detected, etc.).

Finally, there is a correlation between the F1 performance and diversity. This suggests that the richness of the sample content is an important factor affecting performance, and that richer samples can be generated by introducing exemplars or multiple word meanings. We designed our SPE method to improve clarity and relevance while maintaining high diversity (e.g., SPE 3.784 vs. EPE 3.643).

8 Conclusion

This paper investigate how to generate a metaphorical dataset using ChatGPT. We propose a sematic-based prompt enhancement (SPE) method. Experimental results show that the SPE method achieves the best F1 performance on the three datasets, but still falls short of crowdsourced annotations (CA) samples. In addition, we introduce the direct generation method (DG) and the exemplar-based prompt enhancement method (EPE). We provide insights into the advantages and disadvantages of the three ChatGPT sample generation methods by means of example analysis, automatic evaluation and manual evaluation.

9 Limitations

This paper investigate the problem of how to generate a metaphorical dataset using ChatGPT and propose a sematic-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

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.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tuhin Chakrabarty, Arkadiy Saakyan, Debanjan Ghosh, and Smaranda Muresan. 2022. Flute: Figurative language understanding through textual explanations. *arXiv preprint arXiv:2205.12404*.

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*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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.

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.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

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

Hunter Lang, Monica N Agrawal, Yoon Kim, and David Sontag. 2022. Co-training improves prompt-based learning for large language models. In *International Conference on Machine Learning*, pages 11985–12003. PMLR.

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.

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*.

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.

Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *arXiv preprint arXiv:1806.08730*.

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.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Nazneen Fatema Rajani, Bryan McCann, Caiming Xiong, and Richard Socher. 2019. Explain yourself! leveraging language models for commonsense reasoning. *arXiv preprint arXiv:1906.02361*.

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.

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.

Trieu H Trinh and Quoc V Le. 2018. A simple method for commonsense reasoning. *arXiv preprint arXiv:1806.02847*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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*.

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*.

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 Appendix A

11.1 direct generation

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 Direct Sample Generation (DG) approach aims to direct ChatGPT to generate samples of a specified type without using external knowledge content (e.g., metaphorical examples). For input information , $w_k, y_k, n_{k,i}$ 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.

11.2 example-based prompt enhancement

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.

11.3 Sample Fusion Experiment

On both VUAverb and TroFi (see Figure 3 a,b), the introduction of the original sample at the beginning leads to a decrease in Acc. 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., DG100% + CA0% 0.299 vs. DG80% + CA20% 0.465 on VUAverb and DG100% + CA0% 0.272 vs. DG80% + CA20% 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 de-

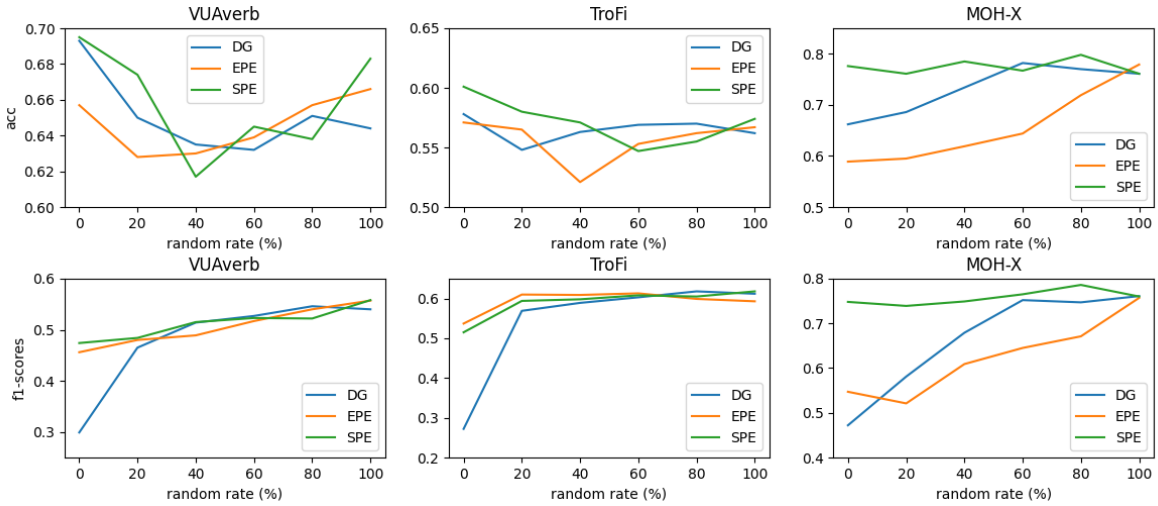

Figure 3: The prompt design diagram is shown below. w_k denotes a specific target word, y_k is its label, and when $y_k = 1$ indicates the metaphorical usage of the generated target word w_k. n_k denotes the number of samples generated.

sign 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.