# 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 process for many natural language processing (NLP) tasks. According to a survey, the average annotation cost per instance on crowdsourcing platforms is $0.11. This high annotation overhead has become a constraint for further research in many tasks. As large-scale language models (LMs) have achieved significant improvements in many small-sample learning tasks, researchers have begun to generate data samples by utilizing model hints to reduce the burden of data annotation. However, previous research has focused on surface language processing tasks and neglected in-depth studies of implicit semantic class tasks (e.g., metaphors), which require models to provide a deeper understanding of implicit meanings in text. Therefore, the goal of this paper is to explore the data generation capabilities of GPT-3 on processing metaphor tasks. We introduce two approaches, Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE). The experimental results show that the F1 scores of the two prompts we designed are significantly higher on the three datasets MOH-X, TroFi and VUAverb test set compared to the metaphor data generated directly using ChatGPT. Meanwhile, the samples generated using EPE and SPE have the same competitive performance compared to the manually labeled data.

## 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, the average labeling cost per instance on crowdsourcing platforms is as high as $0.11 (Wang et al., 2021a). Comparatively, generating samples using large language model (e.g., GPT3.5-turbo) APIs becomes a more cost-effective alternative, costing only 0.001 per 1K token input and 0.002 per 1K token output, respectively. However, this raises an interesting question: how to fully utilize GPT-3 to generate high-quality sample data at low cost?

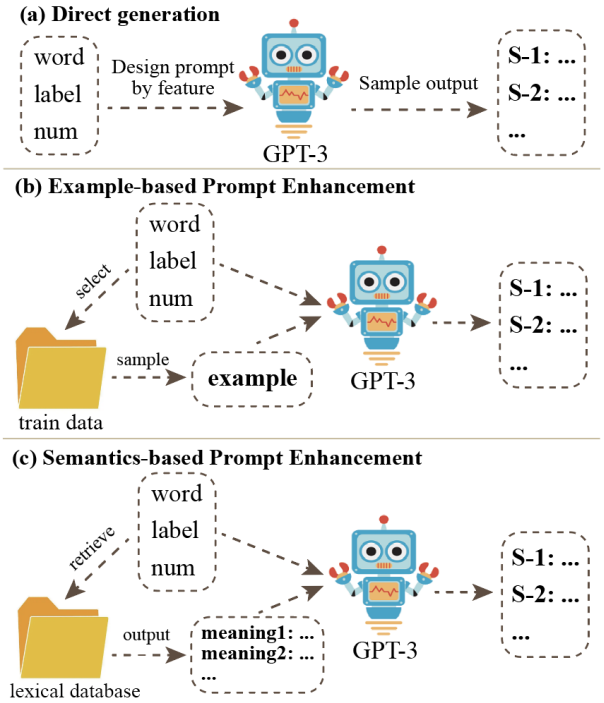


Figure 1: Data generation process. (a) is the direct sample generation using GPT-3. (b) and (c) are the prompts designed in this paper. (b) A sample is randomly sampled in the training set based on the target word "word" and the label "label" as an example. In (c), the lexical meaning of the target word is introduced into the prompt.

Initial research on Large Language Models (LLMs) was done mainly through fine-tuning. McCann et al. (2018); Rajani et al. (2019) used decoders (Radford et al., 2018) that were fed questions and context to generate correct responses. Another common approach is the use of manually

created completions (Trinh and Le, 2018; Petroni et al., 2019), by which the encoder (Devlin et al., 2018) is guided to generate content in a specified direction, e.g., "Donald Trump is [MASK]", where "[MASK]" can be: "former president" or "businessman".

With the continuous development of LLM, the emergence of GPT-3 (Brown et al., 2020) has significantly improved the ability of the model to generate data samples. However, the large number of parameters of GPT-3 also brings the problem of difficult fine-tuning. In contrast, prompt learning, with its non-invasive nature and no need for model fine-tuning or additional additional parameters, has become a new approach for many researchers to explore data generation. In this area, researchers have enabled 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) designed generic templates and provided examples to guide GPT-3 to generate similar data while adapting to multiple downstream tasks. And in (Meng et al., 2022), the generated data are filtered and they are then used for fine-tuning the sub-models.

The above approaches provide new research ideas for introducing dataset generation tasks in large language models. However, these studies mainly focus on data generation for surface language tasks (Wang et al., 2021a; Yoo et al., 2021; Wang et al., 2021b; Meng et al., 2022), which often rely on information about lexical and syntactic structures. In contrast to these surface tasks, implicit semantic-type tasks (e.g., metaphor, sarcasm detection) are more complex and require an in-depth understanding of the implicit meanings in the text. This understanding does not only involve textual information in its surface form, but also requires consideration of contextual semantic relationships and potential inferences.

Therefore, the aim of this paper is to apply prompt learning methods to data generation for metaphor detection tasks. We design two prompt methods for the metaphor task, one of which is the fresh-shot, example-based Prompt Enhancement (EPE); and the other is the zero-shot, lexical meaning-based Prompt Enhancement (SPE). The latter does not rely on manually labeled samples at all and only requires the introduction of the WordNet (Miller, 1995; Fellbaum, 1998) knowledge base. Finally, our contribution is summarized as follows:

1. To the best of our knowledge, this is the first study to apply GPT-3 to metaphorical sample generation. GPT-3 will generate metaphorical or non-metaphorical samples based on the prompt we designed.

2. We are the first study to apply the sample less approach of Example-based Prompt Enhancement (EPE) to metaphorical sample generation. Compared to direct sample generation, EPE averages 30% higher F1 values on the three test datasets.

3. We also designed the zero-sample method of Prompt Enhancement (SPE) based on lexical meaning. This method does not require any manually labeled dataset and achieves similar performance to EPE, even with 10% higher F1 values than EPE on the MOH-X dataset.

## 2 Related Work

### 2.1 Large-scale 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. Remarkably, GPT-3 uses only a small number of hints, which include a small number of examples of a specific task as part of the input to train the model, and achieves state-of-the-art performance on a variety of tasks. Thus, we are witnessing another important shift in the paradigm of prompt-based natural language processing.

### 2.2 Prompt Learning

Unlike previous fine-tuning-based approaches, 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 based on prompt learning are usually categorized into two main groups: generating annotations and generating samples. Ye et al. (2022); Meng et al. (2022) used the method of adding polarity labels to prompts to guide the model to generate sample content related to a specified tendency. For example, a prompt can be constructed such as "Movie reviews 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. This study only explores the use of prompt learning to guide the LLM to generate samples, and therefore does not cover the data labeling part.

### 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); 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); 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); Zhang and Liu (2023) introduced the WordNet database (Fellbaum, 1998), where Gong et al. (2020); 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

This section describes in detail the prompts we designed, each of which will be fine-tuned to fit different metaphorical expression contexts based on the target word. The prompts we design fall into two types, namely example prompt and lexical prompt. In §3.1, we first describe the process of generating metaphorical data with GPT-3. Subsequently, in §3.2, we detail the Example-based Prompt Enhancement (EPE) approach, while in §3.3, the Semantics-based Prompt Enhancement (SPE) approach is presented.

### 3.1 GPT-3 Labeling

GPT-3 (Brown et al., 2020) is a huge pre-trained language model, and in this paper, we choose GPT3.5-turbo, released by OpenAI, for data tagging. GPT3.5-turbo (hereafter GPT-3) supports a 16K context window and is optimized for dialogue. Given a sequence, GPT-3 is able to generate output that naturally continues the input and ends with a special stop sign.

In the annotation process, we first designed the prompt template according to the metaphor detection task. Subsequently, we filled in the blanks 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:

$$\{x'_n\}_{n=1}^{N_k} = \text{GPT-3}(Prompt, w_i, y_i, n_i), \quad (1)$$

where $x'_n$ denotes the $n$th sample generated by GPT-3 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. To ensure fairness in testing the dataset, we direct GPT-3 to generate a specified number of samples (i.e., $N_k$) for each target word and ensure that the distribution of this sample is consistent with the manually labeled dataset (described in §4.1).

### 3.2 Example-based Prompt Enhancement

The first prompt we designed was inspired by (Yoo et al., 2021; Wang et al., 2021b), approaches that provide one or more examples for each category of a given task, with category labeling to follow. Based on this intuition, this paper introduces an exemplar-driven generation approach, where a small amount of manually labeled data is used as the main component of the prompt, and the content

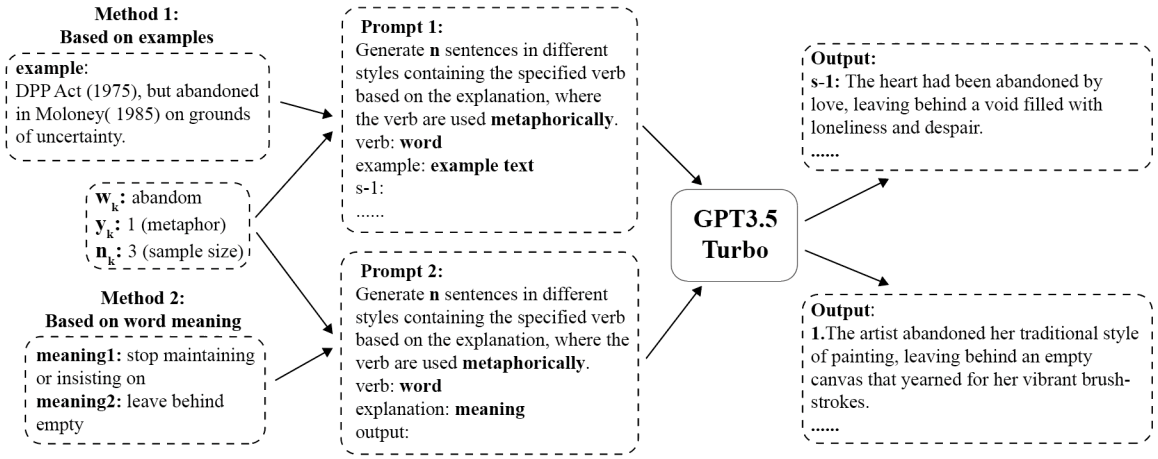


Figure 2: The prompt design diagram is shown below. Where $w_k$ denotes a specific target word, $y_k$ is its label, and when $y_k = 1$ indicates the metaphorical usage of the generated target word $w_k$. $n_k$ denotes the number of samples generated. In Method 1, "example" denotes the example usage of a certain category $(w_k, y_k)$, which will be used as the reference for GPT-3 to generate the metaphorical samples. For Method 2, "meaning x" denotes multiple word meanings of the target word $w_k$. GPT-3 will generate metaphor samples based on the specific word meanings of the target word.

of the exemplars guides the model in generating metaphorical or non-metaphorical samples of the specified type.

**Sample Selection.** For each target word $w_k$ and context $x_k$, there exists and only one label $y_k$ corresponding to it, where $y_k = 1$ denotes the metaphorical usage of the target word $w_k$ and $y_k = 0$ denotes the non-metaphorical usage. In this paper, we adopt VUAverb train set(Leong et al., 2018, 2020) as the sample set, denoted as $\mathcal{D} = (x_i, w_i, y_i)|1 \leq i \leq N$, where $x_i, w_i, y_i$ are the text, the target word, and the corresponding label, respectively. In the specific implementation, we divide $\mathcal{D}$ into subsets $\mathcal{D}_k$ based on the target word $w_k$ and the corresponding label $y_k$. For each category $\mathcal{D}_k$, we randomly select a sample $d_k$ as an example.

**Prompt construction.** For the category set $\mathcal{D}_k$, the number of categories $n_k$, and the category example $d_k = (x_k, w_k, y_k)$, we populate prompt 1 (see Method 1 in Figure 2). where **word** $= w_k$ specifies the target word, $y_k = 1, 0$ denotes the metaphorical or non-metaphorical usage of the target word to be generated, and $\mathbf{n} = n_k$ specifies the number of generation, i.e., GPT-3 is required to generate $n_k$ samples with different description styles. It is worth noting that the above number of samples setting is consistent with the distribution of the manually labeled dataset.

### 3.3 Semantics-based Prompt Enhancement

**Lexical Meaning Search.** WordNet (Miller, 1995; Fellbaum, 1998) is a hierarchically structured lexical database in which each word forms links with other related words to represent semantic connections between them. In metaphor detection tasks, WordNet 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) considered the most commonly used meanings of target words in WordNet as literal meanings and introduced the subtask of Basic Sense Discrimination (BSD). Inspired by these studies, this paper also 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 refer to (Zhang and Liu, 2023) and consider the first two commonly used meanings as literal meanings, while considering the rest of them as metaphorical meanings. 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 detailed in Method 2 in Figure 2. Similar to §3.2, for the input message $(w_k, y_k, n_k)$, we first specify **word** $= w_k$, and then ask the model to generate $n_k$ metaphorical or non-metaphorical sentences based on the value of $y_k$ (0 or 1). However, the difference is that we consider the literal lexical set $\mathcal{V}_{k,l}$ and the metaphorical lexical 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:

$$n_{k,i} = \text{ceil}(n_k/|\mathcal{V}_{k,l}|), \tag{3}$$

where ceil is an upward rounding function, $|\mathcal{V}_{k,l}|$ denotes the number of word meanings, and $n_{k,i}$ denotes the number of samples that need to be generated using the $i$th word meaning. Then, we make $\mathbf{n} = n_{k,1}$, **meaning** $= v_1$ to generate $n_{k,1}$ samples of metaphors with different styles and with $v_1$ lexical meanings. Repeat until $n_k$ samples have been generated.

## 4 Experiment

In this section, we will conduct experiments on samples generated by GPT-3. We use three classical datasets from the metaphor detection task, namely VUAverb, TroFi, and MOH-X. where VUAverb training set will be used for training or extracting examples, while VUAverb test set, TroFi, and MOH-X will be used for testing. In §4.1, we will describe these three datasets in detail, with special treatment for VUAverb. §4.2 describes the experiments, including the classifiers used for fine-tuning the model and the experiments that need to be performed. Finally, in §4.3, we will detail the execution details of the experiments.

### 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 (BNC Consortium, 2007) (Edition et al.), using the MIPVU program for the annotation. Based on VUAMC, several different variants of the VUA corpus have emerged, among which VUAverb is the verb version of the VUA corpus. In this paper, we use the VUAverb dataset mentioned in the metaphor detection shared task (Leong et al., 2018, 2020). In this dataset, the VUAverb train set is used for example selection and generation of identically distributed samples, while the VUAverb test set is used for sample testing. The training set contains 15516 samples and the test set contains 5873 samples.

**VUAverb Cuts.** In the VUAverb train set, some verbs were over-sampled, e.g., "say" (509), "go" (506), while most of the verbs were extremely under-sampled. Statistically, among the 1875 verbs in the VUA train samples, 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 alleviate the problem of uneven distribution of the number of samples in the training set, we trimmed the VUAverb train set. Specifically, we first divided the data based on verbs and labeling information (metaphorical or non-metaphorical). Then, sample categories with numbers greater than 10 were identified, and 10 samples of the total number were randomly selected as the final samples of the category. After the cropping was completed, 7,900 pieces of sample data were obtained.

**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 used the (Choi et al., 2021; Zhang and Liu, 2023) version of the TroFi dataset, which contains a total of 3717 samples. These samples cover rich verb instances and provide diverse contextual information.

**MOH-X.** The MOH dataset was originally 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), which is referenced to mainstream anaphora detection systems (Choi et al., 2021; Zhang and Liu, 2023), excludes instances with pronouns, subordinate subjects or objects. In this paper, MOH-X is used as a test set.

### 4.2 Experimental Setup

The ChatGPT inference used in this paper is accessed via the OpenAI API. Unless otherwise stated, we chose GPT3.5-turbo as the generator tool, and the temperature was kept constant at 0.7.

| Dataset | | RoBERTa-base | | | | RoBERTa-large | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | DG | EPE* | SPE* | $\overline{\Delta}$ gain | DG | EPE* | SPE* | $\overline{\Delta}$ gain |
| MOH-X | **Acc.** | 0.585 | 0.589 | 0.718 | 0.069 | 0.611 | 0.601 | 0.766 | 0.073 |
| | **F1** | 0.385 | 0.536 | 0.681 | 0.224 | 0.453 | 0.574 | 0.722 | 0.195 |
| | **Pre.** | 0.774 | 0.625 | 0.798 | -0.063 | 0.785 | 0.625 | 0.906 | -0.020 |
| | **Rec.** | 0.256 | 0.469 | 0.594 | 0.276 | 0.319 | 0.531 | 0.6 | 0.247 |
| TroFi | **Acc.** | 0.579 | 0.607 | 0.637 | 0.043 | 0.597 | 0.571 | 0.641 | 0.009 |
| | **F1** | 0.225 | 0.578 | 0.564 | 0.346 | 0.37 | 0.583 | 0.583 | 0.213 |
| | **Pre.** | 0.567 | 0.543 | 0.592 | 0.001 | 0.579 | 0.506 | 0.59 | -0.031 |
| | **Rec.** | 0.14 | 0.618 | 0.538 | 0.438 | 0.271 | 0.688 | 0.576 | 0.361 |
| VUAverb-tr | **Acc.** | 0.723 | 0.701 | 0.716 | -0.015 | 0.711 | 0.697 | 0.726 | 0.001 |
| | **F1** | 0.318 | 0.559 | 0.475 | 0.199 | 0.283 | 0.53 | 0.527 | 0.246 |
| | **Pre.** | 0.613 | 0.503 | 0.537 | -0.093 | 0.562 | 0.498 | 0.55 | -0.038 |
| | **Rec.** | 0.215 | 0.629 | 0.425 | 0.312 | 0.189 | 0.567 | 0.506 | 0.348 |

Table 1: Experimental results are shown. We trained GPT-3 on samples generated according to different strategies, covering three datasets, MOH-X, TroFi and VUAverb test set. We used three different strategies, including Direct Generation of Samples (DG), Example-based Prompt Enhancement (EPE) strategy, and Semantics-based Prompt Enhancement (SPE) strategy. Among them, EPE and SPE are new methods that we propose. In the evaluation process, we use four metrics, namely Accuracy (Acc.), F1 Score (F1), Precision (Pre.) and Recall (Rec.), with F1 as the core evaluation metric. $\overline{\Delta}$ denotes the difference between the mean of EPE and SPE and DG

| Dataset | Tokens | Sentences | % Met. |
|---|---|---|---|
| VUAverb_tr | 15,516 | 7,479 | 27.9% |
| VUAverb_te | 5,873 | 2,694 | 29.9% |
| MOH-X | 647 | 647 | 48.7% |
| TroFi | 3,737 | 3,737 | 43.5% |

Table 2: Dataset statistics. tr: training 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

For the choice of classifiers, we used RoBERTa (Liu et al., 2019), considering both base and large versions, and imported the weight parameters from the Huggingface library (Wolf et al., 2019), respectively. For the output of the model, we borrowed some of the model ideas designed in (Choi et al., 2021) and used the output of the hidden layer corresponding to the target word for classification. The classification layer consists of a fully connected layer with weights initialized using random seeds.

This time, we designed two experiments. For Experiment 1, we first obtain the Direct Generation (DG) data samples, and the two sample sets we designed based on Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE) generation. We fine-tuned the three sample sets using RoBERTa, where the DG sample served as a control group. For Experiment 2, we compare the manually labeled data (AN) with the samples from our proposed EPE and SPE, and the generated sample data will use RoBERTa-large as a classifier.

The samples generated in Experiment 1 will be tested on MOH-X, TroFi, and VUAverb test set, while for Experiment 2, we only use MOH-X for testing. Due to the lack of validation sets, we divide the above three datasets according to a 1:1 ratio of word types (e.g., "go", "get") and labels (0 or 1). Eventually, the number of validation and test sets for TroFi is 1869 and 1868, respectively, for MOH-X is 316 and 331, and for VUAverb test set is 2934 and 2939, respectively.

### 4.3 Implementation Details

In this experiment, we use the Adam (Kingma and Ba, 2014) optimizer with an initialized learning rate of 3e-5. The learning rate is controlled by a linear warm-up scheduler, during which the learning rate is gradually increased, where the length of the warm-up period is set to 3 epochs. we set a dropout rate of 0.2, and keep the size of the hidden layer at 768. The batch size for training, validation, and testing was set to 200, and the maximum number of training rounds was set to 30. The maximum length of sentences was 150 tokens, the metaphori-

cal weights are set to 3. All the experiments were run on cloud servers with a single A100 80G GPU. To ensure that the model adequately learns the distribution of the dataset, we set the epoch to be greater than 20, and the model parameters that reach the maximum F1 value on the validation samples are used for testing on the test set.

## 5 Result

### 5.1 Evaluation Metric

Four evaluation metrics are commonly used for metaphor detection tasks. Among them, accuracy indicates the number of correctly categorized samples as a proportion of the total number of samples; precision measures the extent to which the model correctly predicts, focusing on the proportion of samples that the model determines to be in the positive category that are truly in the positive category; recall measures the model's ability to correctly identify positively categorized samples (true instances); and the F1 score is a metric that combines precision and recall and is used to balance the model's accuracy and recall. In this experiment, we also use accuracy, precision, recall and F1 score as evaluation metrics, which are denoted as "Acc.", "Pre.", "Rec." and "F1" respectively.

### 5.2 Results

This subsection analyzes in detail the experimental results of the two experiments designed in §4.2. Experiment 1 aims at exploring the performance of directly generated (DG) samples with samples generated by the two methods we designed, i.e., Example-based Prompt Enhancement (EPE) and Semantic-based Prompt Enhancement (SPE), on three test sets, namely, the MOH-X, TroFi, and VUAverb test set. In contrast, Experiment 2 focuses on comparing the performance of manually labeled (AN) data with EPE and SPE on the MOH-X test set.

**Experiment 1.** The experimental results are detailed in Table 1. The results show that the recall of Directly Generated Data (DG) is low on all the three datasets (25.6%, 15.0% and 21.5% in the base model, and 31.9%, 27.1% and 18.9% in the large model, respectively). This suggests that generating metaphor samples directly using GPT-3 is quite challenging, and the large model does not understand the difference between metaphors and non-metaphors well.

Compared to DG, in the Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE) approaches, both on the RoBERTa base and large models, the performance of the models on the three datasets can be significantly improved. Taking EPE as an example, the F1 metrics of its base model on the three datasets are 0.536 (+15.1%), 0.578 (+35.3%) and 0.559 (+24.1%), respectively. While the F1 metrics for the SPE base model were 0.681 (+29.6%), 0.564 (+33.9%) and 0.475 (+15.7%). On the MOH-X dataset, SPE even outperforms EPE (+14.5% and +14.8% on the base and large models, respectively), suggesting that the zero-sample approach that we designed outperforms the less-sample approach in some aspects.

However, on the VUAverb test set, the performance gap of SPE is larger (-8.4% on the base model) compared to EPE. This may be due to the fact that EPE uses samples with similar distribution as the test set as a result of the example. Finally, we observe that on the VUAverb test set, the F1 performance of DG and EPE on the large model is instead lower than that of the base model, which may be due to the overfitting phenomenon caused by the increase in model parameters. Instead, SPE was able to reduce the risk of fitting by providing data diversity by introducing external lexical meanings. Compared to the base model, the F1 value of the large model is 0.527 (+5.2%).

| Dataset | Methods | | | |
|---|---|---|---|---|
| | EPE | SPE | AN | $\Delta$ gain |
| **MOH-X** | 0.574 | 0.722 | 0.789 | 0.067 |
| **TroFi** | 0.583 | 0.583 | 0.641 | 0.058 |
| **VUA** | 0.53 | 0.527 | 0.697 | 0.167 |

Table 3: We compare samples from Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE) with manually labeled data (AN) from the VUAverb train set. We only consider the F1 evaluation metrics on the three datasets MOH-X, TroFi and VUAverb test set. $\Delta$ denotes the difference between the maximum value of EPE and SPE and the AN.

**Experiment 2.** The experimental results are detailed in Table 3. It can be observed that the manually labeled data performs better on all three test sets, which indicates that there is still a gap in the current use of GPT-3 as a data generation tool. Specifically, the F1 scores of AN on MOH-X, TroFi and VUAverb test set reach 78.9%, 64.1% and

69.7%, which are 6.7%, 5.8% and 16.7% higher than the maximum values of EPE and SPE, respectively. Since the AN data was composed using the VUAverb train set trimming, the performance on the VUAverb test set should be better. For MOH-X and TroFi, competitive performance was achieved using our designed prompts, confirming the effectiveness of EPE and SPE in generating metaphorical data.

## 6 Ablation Experiment

| Methods | MOH-X | | | |
|---|---|---|---|---|
| | Acc. | F1 | Pre. | Rec. |
| **DG** | 0.585 | 0.385 | 0.774 | 0.256 |
| **SPE w/o** | 0.560 | 0.653 | 0.544 | 0.819 |
| **SPE** | 0.718 | 0.681 | 0.798 | 0.594 |

Table 4: Ablation experiments targeting Semantics-based Prompt Enhancement (SPE). DG: Direct Generation, SPE w/o denotes SPE without the use of metaphorical or non-metaphorical descriptions. samples generated by the three methods were tested only on the MOH-X dataset.

This section discusses the impact of the additional addition of metaphorical descriptions to the Semantics-based Prompt enhancement (SPE) on the quality of the GPT-3 generated dataset. Specifically, the direct use of word meanings (SPE w/o) did not add metaphorical or non-metaphorical descriptions, as shown in Figure 2: "where the verb are used metaphorically / non-metaphorically". We used RoBERTa-base to train the model on the generated dataset and then tested it on MOH-X, TroFi and VUAverb test set respectively.

Table 4 compares the performance results of directly generated samples (DG), SPE without metaphorical descriptions (SPE w/o), and SPE with metaphorical descriptions on the three datasets. The results show that SPE w/o still outperforms the directly generated sample using GPT-3. Its F1 value on the MOH-X dataset is 65.3% (+26.8% increase). However, the SPE with metaphorical descriptions achieved the best performance on all three test datasets, at 68.1% (+2.8% increase). This suggests that the metaphorical and non-metaphorical nature of generating samples through word sense segmentation alone is slightly insufficient. Adding qualification of metaphorical or non-metaphorical in the prompts could help to further improve the quality of GPT-3 generated data.

## 7 Conclusion

Manual labeling of datasets is usually time- and money-consuming, especially in metaphor detection tasks, where the quality of the dataset depends on the educational level of the subjects. To address this problem, this paper investigates how to generate metaphor datasets using GPT-3. We propose two approaches, namely Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE). For EPE, we need a certain number of manually labeled samples, i.e., one example for the target word and label. Then, GPT-3 generates the specified number of samples based on the given example. For SPE, we sort the word meanings given in WordNet in order of frequency, considering the first two as literal meanings and the rest as metaphorical meanings. Then, we ask GPT-3 to generate metaphorical or non-metaphorical samples based on the specified word meanings. The experimental results show that the data generated using the EPE and SPE methods can significantly improve the performance of the downstream model on the metaphor detection fine-tuning task compared to generating samples directly using GPT-3.

## 8 Limitations

In this paper, we investigate the problem of how to generate a metaphorical dataset using GPT-3. We propose two methods, namely Example-based Prompt Enhancement (EPE) and Semantics-based Prompt Enhancement (SPE). First of all, the data generated by the above two methods are consistent with the distribution of the VUAverb dataset, which may lead to the discrepancy between the generated metaphor dataset and the actual distribution. In addition, the EPE method obtains examples from VUAverb based on the target words and labels (whether they are metaphorical usage or not), i.e., the number of instances obtained is proportional to the number of target word types, which may lead to a higher number of instances required. In practice, manually labeling these numbers of samples would likewise require a larger cost. In future work, we will aim to explore ways to ensure the quality of metaphor sample generation while minimizing the number of instances.

## 9 Ethics Statement

In this paper, we detail how GPT-3 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.

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*.