A Prompt Based Approach for Euphemism Detection

Abulimiti Maimaitituoheti, Yang Yong, Fan Xiaochao
Xinjiang Normal University, China
{1149654712, 68523593, 37769630}@qq.com

Abstract

Euphemism is an indirect way to express sensitive topics. People can comfortably communicate with each other about sensitive topics or taboos by using euphemisms. The Euphemism Detection Shared Task in the Third Workshop on Figurative Language Processing co-located with EMNLP 2022 provided a euphemism detection dataset that was divided into the train set and the test set. We made euphemism detection experiments by prompt tuning pre-trained language models on the dataset. We used RoBERTa as the pre-trained language model and created suitable templates and verbalizers for the euphemism detection task. Our approach achieved the third-best score in the euphemism detection shared task. This paper describes our model participating in the task.

1 Introduction

Euphemism is a common linguistic phenomenon that is indispensable in everyday communication. In daily communication, people often encounter some sensitive topics or taboos that are inconvenient to express directly, such as death, age, income, etc., and need to borrow some synonymous sentences to express them tactfully or use words that are related to the original meaning. The social function of euphemism enables people to express their thoughts more freely and to communicate easily and happily.

Euphemism detection is a classification task in natural language processing and it can be divided into phrase-level euphemism detection and sentence-level euphemism detection. In English, euphemisms consist of one or more words, for example, indigent is a euphemistic word that refers to a poor or needy person; deceased means a person who has died; a sex worker means a prostitute euphemistically. Phrase-level euphemism detection refers to the task of identifying euphemistic phrases in a sentence or paragraph. A model needs to recognize the euphemistic phrases in the target sentences. Phrase-level euphemism detection can be modeled as a token classification task. Sentence-level euphemism detection refers to the task of determining whether a sentence or a paragraph contains a euphemism. For example, the sentence "Give me a moment, I just need to run to the powder room." Contains a euphemistic phrase "powder room" which means toilet, so the sentence could be classified as euphemistic. A model needs to determine whether target sentences contain any euphemistic phrases. Sentence-level euphemism detection can be modeled as a sequence classification task.

The Euphemism Detection Shared Task in the Third Workshop on Figurative Language Processing co-located with EMNLP 2022 is a sentence-level euphemism detection task, which provides a sentence-level euphemism detection dataset based on GloWbE (Davies et al., 2015) dataset. The euphemism detection dataset consists of a total of 1965 sentences, among them 1382 sentences containing euphemistic terms and 777 sentences that don't contain euphemistic terms. The dataset is divided into a train set and a test set, the train set contains 1572 sentences with target labels, and the test set contains 393 sentences without target labels. The task requires participants to develop a model which can determine whether the target sentence contains any euphemistic terms. A model can train on the train set and predict the labels of the test set.

Prompt learning adds additional prompt information to the original input sentence and transforms it into cloze forms to make the pre-trained language model better understand and

Proceedings of the 3rd Workshop on Figurative Language Processing (FLP), pages 8 - 12

process the sentence. Prompt learning unifies the training of downstream tasks with pre-training processes and avoids the pre-trained language models to forget some of the knowledge and tap more potential of them. By this, prompt learning improves the performance of the models in few-shot and zero-shot, or even full-shot tasks. So we developed a prompt learning-based euphemism detection model, using RoBERTa (Liu et al., 2019) as the pre-trained language model. We trained the model on the train set, then predict the labels of the test set by using it. Our model achieved an F1 score of 85.2% on the test set, the third highest among all participants.

2 Related Works

In recent years, people have made some explorations on euphemism detection, constructed some euphemism detection datasets, and made some efforts to recognize euphemistic phrases and sentences. In this section, we will introduce these works by phrase level and sentence level respectively.

2.1 Phrase-Level Euphemism Detection

Euphemisms consist of phrases, which means that recognizing euphemistic phrases is the direct way of euphemism detection research. So there has been relatively more work on phrase-level euphemism detection.

Magu et al. (2018) made research on euphemistic hate speech. Specifically, they collected 200000 tweets containing euphemistic code words and constructed a dataset, by using the Community detection analysis method they found some unknown euphemistic hate speech code words. Felt et al. (2020) constructed a dataset by manually annotating target phrases as euphemisms, dysphemism, or neutral and made x-phemism classification experiments by using sentiment lexicons and contextual sentiment analysis. As a result, experiments using contextual sentiment analysis achieved better results than experiments using sentiment lexicons. This illustrated that euphemisms are context-dependent and models relying on contexts are more efficient in euphemism detection. Zhu et al. (2021) investigated multi-word euphemistic phrase detection approaches automatically. They extracted suitable phrases from the original data and selected euphemistic phrase candidates utilizing word embedding similarities, then ranked candidate phrases by using Spanbert (Joshi et al., 2020). They achieved significant performance improvement on the euphemistic phrase detection task compared to baseline models. Lee et al. (2022) made an investigation to find potentially euphemistic terms by steps. Firstly, they extracted phrases from original data, then filtered phrases related to sensitive topics by calculating the cosine similarity between the phrases and a list of words representing sensitive topics. After that, they paraphrased quality phrases using the top 25 most similar words as output by word2vec. Finally, they ranked phrases by using a RoBERTa-base model trained on tweets for sentiment analysis and offensive language identification.

2.2 Sentence-Level Euphemism Detection

Sentence-level euphemism detection is another direction in euphemism detection research, which needs to determine whether a sentence contains any euphemism, but there is little research in this direction.

Gavidia et al. (2022) made a list of common euphemistic phrases, which contained a total of 184 euphemistic phrases. Then selected 1382 euphemistic sentences and another 583 not euphemistic sentences according to the list from GloWbE dataset, and constructed a sentence-level euphemism detection dataset. They made sentiment analysis using RoBERTa on the dataset and found that the sentiment and offensiveness of euphemistic phrases have some differences compared to their literal meanings.

3 System Overview

Prompt-tuning is a method of using pre-trained language models, which requires transforming downstream tasks into cloze forms so that the pre-training process of pre-trained language models and the training process of downstream tasks can be unified. The prompt-tuning method can unlock the potential of pre-trained language models and it is a better few-shot learning method compared with the fine-tuning method. Considering these advantages of prompt-tuning, we conducted our euphemism detection experiments using it.

3.1 System Architecture

The architecture of our euphemism detection system is shown in figure 1.

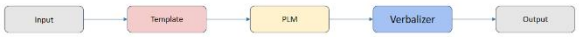

Figure 1: The architecture of the euphemism detection system based on prompt-tuning.

In figure 1, Input refers to the target text inputted into the model. The template is a module that transforms the input text into a cloze form by adding extra characters and <mask> tags to the head and end of the input text. PLM stands for pre-trained Language Model, in this paper we used RoBERTa as the pre-trained language model. Verbalizer is a module that maps predicted words to target labels, it is mainly composed of a word-label dictionary and mapping function. Output is the target label predicted by the model, in our euphemism detection experiment, the output is a label consisting of 0 or 1, 0 stands for not euphemistic, and 1 stands for euphemistic.

3.2 Template For Euphemism Detection

The template is an important module in prompt learning, the quality of the template has a great effect on the performance of the model. So people made a lot of explorations on the method of creating templates, including manually creating, automatically creating, and mixed approaches to the above two methods. We made many experiments using different templates, including manually created templates, prefix-tuning templates (Li et al. 2021), P-tuning templates (Liu et al., 2021), and ptr templates (Han et al., 2021). Through experiments, we found that ptr templates perform better than other templates on the euphemism detection task, so we selected ptr template as the template for our euphemism detection model. Ptr template is a hybrid template using the manually and automatically template-creating methods together. Specifically, a basic template should be created manually, and optimized by logic rules, so we created some candidate basic templates for the euphemism detection task as follows:

<text> Is this euphemistic? <mask>
<text> This is <mask> sentence.
<mask> - <text>
<text> This is <mask>.

In these templates, <text> stands for the original input text, and <mask> is the token that should be predicted by the model. We made experiments using these candidate basic templates and found that template "<text> This is <mask> sentence." achieved the best performance among them, so we selected it as the final basic template for our euphemism detection model.

3.3 Verbalizer For Euphemism Detection

Another important module of prompt learning is the verbalizer, the function of the verbalizer is mapping the predicted words to target labels, in our euphemism detection experiments the verbalizer will map the predicted words into one of the euphemistic or not euphemistic labels. Like the template, there are a lot of ways to create verbalizers, including manually created verbalizer, knowledgeable verbalizer (Hu et al., 2022), and ptr verbalizer (Han et al., 2021). We made euphemism detection experiments using different verbalizers and found that the ptr verbalizer performed better than other verbalizers, so we selected ptr verbalizer as our verbalizer.

In our euphemism detection task, we only need to detect the euphemistic sentences and don't need to detect not euphemistic sentences, so we made some changes to the ptr verbalizer, and determine whether the predicted word is in the words list of the euphemistic label before mapping the predicted words to the target label if it is, then maps the predicted word to euphemistic label, else then maps the predicted word to not euphemistic label. By this, we reduced the burden of the model and we only need to create a word list for the euphemistic labels.

After selecting the ptr verbalizer and making some changes to it, we created some candidate verbalizers as follows:

euphemistic, tactful, periphrastic
yes, yeah

The verbalizer "yes, yeah" is for the template "<text> Is this euphemistic? <mask>" and other templates use the verbalizer "euphemistic, tactful, periphrastic". For example, when we use the template "<text> This is <mask> sentence." and the verbalizer "euphemistic, tactful, periphrastic", if the predicted word is one of the words in the list "euphemistic, tactful, periphrastic", the predicted word will be mapped to euphemistic label, else the predicted word will be mapped to not euphemistic label.

4 Experiments

In this section, we will introduce the specifics of the experiments, including dataset statistics,

coding framework, hyperparameters of experiments, results of experiments and analysis, etc.

4.1 Dataset Statistics

There are some hyperparameters like the maximum sentence length inputted the pre-trained language model that has much relations with the dataset features. So we made some statistical analysis to the dataset. The details of the statistical analysis are shown in table 1.

Table 1: Statistics of text length in the dataset

Dataset	Avg len	Max len	len>192	Total	Percentage
Train	83.13	381	20	1572	1.27%
Test	83.36	247	3	393	0.76%

In table 1, avg len stands for the average number of words in the sentence, max len stands for the number of words in the longest sentence, len>192 stands for the number of sentences that contain words more than 192, and total stands for the total number of the sentence in the dataset, the percentage stands for the ratio of the number of sentences longer than 192 among all the sentences.

4.2 Coding Framework

OpenPrompt (Ding et al., 2022) is an open-source prompt learning framework based on PyTorch (Paszke et al., 2019), which provided the implementation of many templates and verbalizers, including manually created templates, prefix-tuning templates, p-tuning templates, ptr templates, manually created verbalizers, knowledgeable verbalizers, ptr verbalizers, etc. Relying on OpenPrompt, we could implement our prompt learning experiments by conveniently switching among the different templates and verbalizers. So we selected OpenPrompt as our prompt learning framework and coded our model using it.

4.3 Hyperparameters

Table 2: Hyperparameters of the model

From table 1, we can see that the proportion of the sentences containing more than 192 words is about 1%, this means that 192 is a proper value for the maximum length of the sentence inputted to the pre-trained language model, so we set it as 192. There are some other hyperparameters like learning rate, weight decay, etc. The details are shown in table 2.

Hyperparameter	Value
Max sequence length	192
Learning rate	0.00003
Weight decay	0.01
Number warmup steps	500
Number of epochs	5

4.4 Results And Analysis

Using RoBERTa-base as our pre-trained language model, we implemented euphemism detection experiments on the euphemism detection dataset. We trained the prompt-based euphemism detection model on the train dataset and let it predict the labels of the test data. By training the model 5 epochs, achieved relatively good results on the test data, and ranked third among all participants. The details of the results are shown in Table 3.

Table 3: Details of the results of the euphemism detection experiment

Type	Precision	Recall	F1
Euphemistic	0.904	0.924	0.914
Not euphemistic	0.811	0.769	0.789
macro avg	0.858	0.847	0.852

From table 3 we can see that the precision, recall, and f1 score of the euphemistic label is better than the not euphemistic label. We think that because the dataset is an unbalanced dataset, contains more euphemistic sentences and fewer not euphemistic sentences, model learning is biased toward euphemistic sentences.

5 Conclusions

In this paper, we described the euphemism detection model based on prompt learning and the details of euphemism detection experiments based on this model. We selected RoBERTa as our pre-trained language model, selected the ptr template and ptr verbalizer, made some reasonable changes to the ptr verbalizer, implemented euphemism detection experiments using this model, and achieved better results on the euphemism detection dataset.

References

Davies, M. and Fuchs, R. 2015. *Expanding horizons in the study of world englishes with the 1.9 billion word global web-based english corpus (glowbe).* English World-Wide, 36(1):1–28.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, Veselin Stoyanov. 2019. *Roberta: A Robustly Optimized BERT Pretraining Approach.* arXiv:1907.11692 [cs.CL].

Rijul Magu, Jiebo Luo. 2018. *Determining Code Words in Euphemistic Hate Speech Using Word Embedding Networks.* Proceedings of the 2nd Workshop on Abusive Language Online (ALW2), pages: 93–100.

Christian Felt, Ellen Riloff . 2020. *Recognizing Euphemisms and Dysphemisms Using Sentiment Analysis.* Proceedings of the Second Workshop on Figurative Language Processing, pages: 136–145.

Wanzheng Zhu, Suma Bhat. 2021. *Euphemistic Phrase Detection by Masked Language Model.* Findings of the Association for Computational Linguistics: EMNLP 2021, pages: 163–168.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. *Spanbert:Improving pre-training by representing and predicting spans.* Transactions of the Association for Computational Linguistics, 8:64–77.

Patrick Lee, Martha Gavidia, Anna Feldman, Jing Peng. 2022. *Searching for PETs: Using Distributional and Sentiment-Based Methods to Find Potentially Euphemistic Terms.* Proceedings of the Second Workshop on Understanding Implicit and Underspecified Language, pages: 22–32.

Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. 2022. *Cats are fuzzy pets: A corpus and analysis of potentially euphemistic terms.* arXiv preprint arXiv:2205.02728.

Xiang Lisa Li, Percy Liang. 2021. *Prefix-Tuning: Optimizing Continuous Prompts for Generation.* 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: 4582–4597.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, Jie Tang. 2021. *GPT Understands, Too.* arXiv:2103.10385 [cs.CL].

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, Maosong Sun. 2021. *PTR: Prompt Tuning with Rules for Text Classification.* arXiv:2105.11259 [cs.CL].

Shengding Hu, Ning Ding, Huadong Wang, Zhiyuan Liu, Jingang Wang, Juanzi Li, Wei Wu, Maosong Sun. 2022. *Knowledgeable Prompt-tuning: Incorporating Knowledge into Prompt Verbalizer for Text Classification.* Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages: 2225–2240.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Haitao Zheng, Maosong Sun. 2022. *OpenPrompt: An Open-source Framework for Prompt-learning.* Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pages: 105–113.

Paszke, Adam and Gross, Sam and Massa, Francisco and Lerer, Adam and Bradbury, James and Chanan, Gregory and Killeen, Trevor and Lin, Zeming and Gimelshein, Natalia and Antiga, Luca and Desmaison, Alban and Kopf, Andreas and Yang, Edward and DeVito, Zachary and Raison, Martin and Tejani, Alykhan and Chilamkurthy, Sasank and Steiner, Benoit and Fang, Lu and Bai, Junjie and Chintala, Soumith. 2019. *PyTorch: An Imperative Style, High-Performance Deep Learning Library.* Advances in Neural Information Processing Systems 32, pages: 8024—8035.