Euphemism Detection by Transformers and Relational Graph Attention Network

Yuting Wang, Yiyi Liu, Ruqing Zhang, Yixing Fan, Jiafeng Guo
CAS Key Lab of Network Data Science and Technology, Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
University of Chinese Academy of Sciences, Beijing, China
{liuyiyi17s,wangyuting22g,zhangruqing,fanyixing,guojiafeng}@ict.ac.cn

Abstract

Euphemism is a type of figurative language broadly adopted in social media and daily conversations. People use euphemisms for politeness or to conceal what they are discussing. Euphemism detection is a challenging task because of its obscure and figurative nature. Even humans may not agree on if a word expresses euphemism. In this paper, we propose to employ bidirectional encoder representations transformers (BERT), and relational graph attention network in order to model the semantic and syntactic relations between the target words and the input sentence. The best performing method of ours reaches a macro F_1 score of 84.0 on the euphemism detection dataset of the third workshop on figurative language processing shared task 2022.

1 Introduction

Euphemism is a sophisticated language phenomenon in which one usually uses a polite word or expression instead of a more direct one to avoid shocking or upsetting someone[1]. For example, *"We are very sorry that he has passed away"*. Here, *"pass away"* does not mean dissipation intuitively, but death, which can make unpleasant things sound more polite. Due to its obscure and figurative nature, euphemism detection which aims to predict a text as euphemism or non-euphemism becomes a particularly challenging classification task. With the usage of euphemisms becoming prevalent on social media and in daily conversation, euphemism detection has received growing research attention to facilitate the understanding of natural language's sentiment and semantics.

Felt and Riloff (2020) make the first attempt to recognize euphemisms and dysphemisms. They identify synonym phrases of given seed euphemism-related phrases by a weakly supervised bootstrapping algorithm and then classify the phrases using sentiment cues and contextual sentiment analysis. With the advent of Pre-trained Language Models (PLMs), euphemism detection methods based on PLMs such as BERT (Devlin et al., 2019) have been proposed. Zhu and Bhat (2021) propose an automatic euphemistic phrase detection method without human effort. They first extract quality phrases and select euphemistic phrase candidates by computing embedding similarities. Then they use SpanBERT to rank and classify all candidates.

Despite existing work have achieved promising results, there are still several challenges to tackle. On the one hand, existing euphemism detection work mainly focus on mining characteristics of target words/phrases that triggered the euphemism phenomenon. They emphasize too much the euphemism of target words while ignoring the context circumstances where the target words sit. On the other hand, the first step of these methods is often to extract euphemism candidate words or phrases based on domain expertise or existing data annotations. If the first step is not done well, it will influence the subsequent classification and ranking, which may cause error propagation and lead to poor performance. We observe that euphemisms are essentially polysemy. In this sense, we argue that the meanings of euphemism target words/phrases are closely related to the context in which they are located semantically and syntactically.

Shed light on the great performance achieved by BERT and Graph Neural Network (Veličković et al., 2017) on the aspect-based sentiment analysis task, we propose to employ BERT and Relational Graph Attention Network (RGAT) (Wang et al., 2020) to deal with euphemism detection. Specifically, our model contains two isolated sub-models, BERT-Concat and RGAT-BERT. For BERT-Concat, the model's input is the concatenation of the input sentence and target words. We use BERT-Concat to enhance the information of target words and

[1] https://www.ldoceonline.com/dictionary/euphemism

Proceedings of the 3rd Workshop on Figurative Language Processing (FLP), pages 79 - 83

capture the sequential semantic knowledge of the input sentence and target words. RGAT-BERT is adopted mainly to capture the syntactic information between target words and their corresponding contexts. The graph is built on the dependency tree. To enhance the syntactic connections between target words and the essential contexts, RGAT reshapes the dependency tree in which target words are root. It also prunes the reshaped tree to avoid the noise that unimportant contexts bring. Finally, we design a voting mechanism to ensemble the results of the two sub-models, which can leverage the advantages of the two.

We conduct experiments on the euphemism detection dataset. Empirical experimental results demonstrate the effectiveness of our proposed method. We ended up fourth in the third workshop on figurative language processing shared task 2022.

2 Related Work

In this section, we briefly review the related work on euphemism detection.

Existing work mainly focus on identifying euphemistic words. Magu and Luo (2018) provide an unsupervised word embedding's similarity method to identify euphemisms (code words) in hate speech. Felt and Riloff (2020) use sentiment analysis to recognize the euphemistic and dysphemistic language. They adopt a bootstrapping algorithm for finding near-synonym phrases and then classify the collected phrases as euphemistic, dysphemistic, or neutral using lexical sentiment cues and contextual sentiment analysis.

With the advent of pre-trained language models, a lot of euphemism detection methods based on PLMs have been proposed. Zhu et al. (2021) propose a self-supervised euphemistic detection method. They first extract candidate phrases from a base corpus and then filter out ones associated with euphemistic seed phrases through embedding similarity computing. Finally, they use pre-trained language models to classify these phrases. Similar to (Felt and Riloff, 2020), Zhu et al. (2021) rely on a set of predefined seed phrases, which may not be generalized to different datasets. Zhu and Bhat (2021) improve Zhu et al. (2021)'s approach by adding an automatic paraphraser. Kapron-King and Xu (2021) investigate gender differences in euphemism usage and they find that women do not use euphemisms more than men through empirical

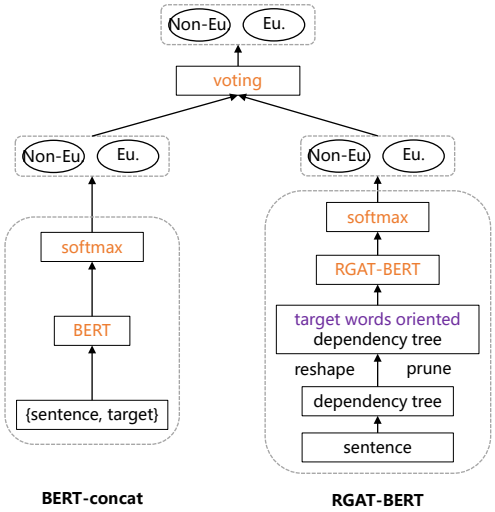

Figure 1: Structure of our model, which contains BERT-Concat(left) and RGAT-BERT(right). Eu. and Non-Eu. denote euphemism and non-euphemism classes respectively.

analysis. Gavidia et al. (2022) present a corpus of potentially euphemistic terms, which promotes the development of euphemism detection. We observe that most work on euphemism detection focus on euphemistic terms. They pay less attention to the contexts and the connections between euphemistic terms and their corresponding contexts in a sentence, which may lose important information.

3 Model

In this section, we introduce our method for euphemism detection in detail. The overview of our proposed method is shown in Figure 1. We first introduce the pre-processing of the dataset, and then the BERT model and the RGAT-BERT model. Finally we elucidate the model ensembling process.

3.1 Data Pre-processing

The original data includes text IDs, utterances, and euphemistic labels. We pre-process the text to (1) extract target words and their position, (2) remove the unexpected punctuation. Since the target is marked with "<>" symbols, for the convenience of subsequent model implementation, we extract the target and mark the position of the left character start point and the right character endpoint. Then we remove the unexpected punctuation marks "@@@@@@@@@@", "<" and ">". "@@@@@@@@@@" is a feature of GloWbE corpus that obscures spans of text. The removal of the above marks will not affect the meaning of the input utterance. The input sentence is denoted as $s = \{w_1^s, w_2^s, ..., w_n^s\}$ and the corresponding target

words is represented as $t = \{w_i^t, w_{i+1}^t, ..., w_k^t\}$. n is the length of the input sentence. k is the length of target words.

3.2 BERT-Concat

We design the BERT-Concat model to enhance the information of target words and capture the sequential semantic knowledge of the input sentence and target words. The input of BERT-Concat is $\{s, t\}$. Note that the concatenation happens at the sequence length, not the hidden dimension. We also try to concatenate the input sentence and target words at the hidden dimension, but the experimental results are not good. The reason may be that the hidden size of the new representation is too large after concatenating, which may increase the complexity of the model and introduce irrelevant noisy information.

The training objective is to minimize the cross-entropy loss of the euphemism label probability distribution.

$$L_{CE}(\theta) = \sum \text{cross-entropy}(y, P(\hat{y})),$$

where y is the ground-truth of the euphemism label, and $P(\hat{y})$ is the predicted score. θ is the parameter set of the model.

3.3 RGAT-BERT

The syntactic structure is an important tool for understanding natural language. The relationships between words can be denoted with directed edges and labels. Sometimes the context that is important to understand target words may not be found in the sequence structure but in the syntactic structure. Therefore, the use of graph neural networks and syntactic trees can solve the mistakes caused by sequential attention mechanisms. We leverage RGAT-BERT to capture the syntactic information between target words and their corresponding contexts.

Firstly, we extract the original dependency graph by syntax parsing tools. Note that the root of the current dependency graph may not be target words. Then the structure of the dependency tree is rooted in the euphemism target words by reshaping and pruning the ordinary dependency analysis tree. The new dependency tree is encoded by the relational graph attention network(RGAT) model.

The reconstructed tree can be represented by a graph G with N nodes, where each node is a word in the utterance, and the edges of the graph represent the dependencies between words. The neighborhood nodes of node i are N_i. The graph attention network(Veličković et al., 2017) iteratively updates each node by aggregating the representation of neighborhood nodes with multiple heads of attention. Training the BERT model can obtain the hidden layers. The whole RGAT formula comes from (Wang et al., 2020). The attention formula is as follows:

$$h_{att_i}^{l+1} = ||_{k=1}^{K} \sum_{j \epsilon N_i} \alpha_{ij}^{lk} W_k^l h_j^l \quad (1)$$

$$\alpha_{ij}^{lk} = attention(i, j), \quad (2)$$

where l means the number of the layer and i and j mean the number of the node. And $h_{att_i}^{l+1}$ means the attention head, $||_{k=1}^{K} x_i$ is the concatenation of vectors from x_1 to x_k, α_{ij}^{lk} is a dot-product attention which comes from $attention(i, j)$ computed by the k-th attention at layer l, W_k^l is an input transformation matrix. K means the number of attention headers.

The graph attention mechanism aggregates the representations of neighborhood nodes along the dependency path. However, neighborhood nodes with different dependencies should have different effects. Therefore, RGAT uses additional relationship headers to expand the original network. The dependency relationship is mapped into a vector representation to calculate a relationship header. RGAT contains M relationship headers. The calculation formula is as follows:

$$h_{rel_i}^{l+1} = ||_{m=1}^{M} \sum_{j \epsilon N_i} \beta_{ij}^{lm} W_m^l h_j^l \quad (3)$$

$$g_{ij}^{lm} = \sigma(relu(r_{ij} W_{m1} + b_{m1}) W_{m2} + b_{m2}) \quad (4)$$

$$\beta_{ij}^{lm} = \frac{exp(g_{ij}^{lm})}{\sum_{j=1}^{N_i} exp(g_{ij}^{lm})}, \quad (5)$$

where r_{ij} is the relation embedding between nodes i and j. The final representation of each node is as follows:

$$x_i^{l+1} = h_{att_i}^{l+1} || h_{rel_i}^{l+1} \quad (6)$$

$$h_i^{l+1} = relu(W_{l=1} x_i^{l+1} + b_{l+1}). \quad (7)$$

The hidden representation is then passed through a fully connected softmax layer and mapped to probabilities over the euphemistic labels. BERT is used as a basic encoder in the RGAT-BERT model. The training objective of RGAT-BERT is also to minimize cross-entropy loss. For a more detailed description of RGAT, please refer to the original paper (Wang et al., 2020).

Dataset	Eu.	Non-Eu.	Total	Avg ℓ
Train	1106	466	1572	65.7
Test	/	/	393	65.8

Table 1: The detailed statistics of the dataset. Eu. and non-Eu. mean the number of euphemism and non-euphemism samples respectively. Avg ℓ denotes the average length of texts in the number of tokens.

3.4 Model Ensembling

We adopt a voting strategy for ensembling the results of BERT-Concat and RGAT-BERT. Specifically, there is a set of predicted labels by different models. For each sample, if more than half models saying that the sample belongs to the euphemistic class, then the voting result is euphemism. On the contrary, if more than half models saying that the sample belongs to the non-euphemistic class, then the voting result is non-euphemism.

4 Experiment

In this section, we will introduce the dataset and experimental settings, and then analyze the results.

4.1 Dataset

We use the official euphemism dataset provided by the third workshop on figurative language processing shared task 2022. The statistics are shown in Table 1. We observe that the training dataset is unbalanced. The number of euphemistic samples is more than twice as large as the number of non-euphemistic samples. The original dataset does not contain a validation set. We randomly choose 200 samples from the training set as a validation set to fine-tune the parameters. In the validation set, there are 133 euphemism and 67 non-euphemism.

4.2 Baselines

We adopt LSTM (Hochreiter and Schmidhuber, 1997), RGAT (Wang et al., 2020), and BERT (Devlin et al., 2019) as the baseline methods for comparison. Each utterance in the given dataset contains only one euphemism, there is no case of multiple euphemisms mixed in one utterance. So we take the sentences as input directly for the above baseline models.

4.3 Experimental Settings

We train our models on Nvidia Telsa V100-16GB GPUs. For the BERT-Concat model, we set the learning rate to $5e-5$, the batch size to 16, and the maximum sequence length to 512. We implement RGAT-BERT for euphemism detection based on the released source code [2] in their paper. For the RGAT-BERT model, the learning rate is set to $5e-5$, the batch size is 8, and the dropout is 0.3. For other parameters of RGAT-BERT, we use the default settings in the source code. For each method, we train them with five seeds among $\{2022, 2021, 2019, 142, 42\}$. The difference between macro F_1 scores of different seeds is within 2%. For model ensembling, we selected 7 highest results of the two models and vote on the final labels. We use BERT-base as the backbone model.

Method	Precision	Recall	Macro F_1
LSTM	73.4	71.0	71.7
RGAT	77.6	73.5	73.9
BERT	78.4	76.9	77.5
BERT-Concat	76.7	81.4	78.4
RGAT-BERT	81.1	83.4	82.1
Ensembled	**84.2**	**83.8**	**84.0**

Table 2: The precision, recall, and macro F_1 (%) on the test set. Best results as bold.

4.4 Experimental Results

The overall experimental results are shown in Table 2. We observe that: (1) RGAT model outperforms LSTM model, which shows that involving syntactic information is more effective than relying solely on sequential information intra-sentence. (2) Fine-tuning with pre-trained language models performs better than traditional deep neural models. By using only BERT model, the macro F_1 score reaches 78.4. It demonstrates the power of large-scale pre-trained language models. This indicates that though euphemisms are obscure, they are commonly used, so euphemism detection tasks can make better use of the knowledge in the pre-trained language models. (3) There is a slight improvement using BERT-Cocat compared to the basic BERT model. RGAT-BERT outperforms BERT-Concat with a large margin of 3.7 on the macro F_1 score. This demonstrates that syntactic connections between target words and their corresponding contexts can better understand the meaning of euphemism. (4) Ensembling the two models achieves the best performance since model ensembling can leverage the advantages of the two models.

5 Conclusion

In this paper, we have proposed to leverage transformers and relational graph attention networks to detect euphemisms. Specifically, on the one hand,

[2] https://github.com/shenwzh3/RGAT-ABSA

we utilize BERT-Concat to capture sequential semantic information between target words and their corresponding contexts. On the other hand, we adopt RGAT-BERT to learn the syntactic connections between target words and essential contexts. Experimental results show that ensembling the two sub-models can achieve promising performance on the euphemism detection shared task of the third workshop on figurative language processing.

Limitations

At present, we view euphemism detection from the perspective of the task itself and specific datasets. Our model is not much integrated with the euphemistic theory linguistically. Later, we will explore the different meanings between original target words and their euphemistic usage by text matching strategies.

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Christian Felt and Ellen Riloff. 2020. Recognizing euphemisms and dysphemisms using sentiment analysis. In *Proceedings of the Second Workshop on Figurative Language Processing*, pages 136–145, Online. Association for Computational Linguistics.

Martha Gavidia, Patrick Lee, Anna Feldman, and Jing Peng. 2022. CATs are Fuzzy PETs: A Corpus and Analysis of Potentially Euphemistic Terms. *arXiv e-prints*, page arXiv:2205.02728.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Anna Kapron-King and Yang Xu. 2021. A diachronic evaluation of gender asymmetry in euphemism. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 28–38, Online. Association for Computational Linguistics.

Rijul Magu and Jiebo Luo. 2018. Determining code words in euphemistic hate speech using word embedding networks. In *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, pages 93–100, Brussels, Belgium. Association for Computational Linguistics.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *arXiv e-prints*, page arXiv:1710.10903.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Wanzheng Zhu and Suma Bhat. 2021. Euphemistic phrase detection by masked language model. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 163–168, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wanzheng Zhu, Hongyu Gong, Rohan Bansal, Zachary Weinberg, Nicolas Christin, Giulia Fanti, and Suma Bhat. 2021. Self-supervised euphemism detection and identification for content moderation. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 229–246.