# Detecting Euphemisms with Literal Descriptions and Visual Imagery

Ilker Kesen[1,2]     Aykut Erdem[1,2]     Erkut Erdem[1,3]     Iacer Calixto[4,5]

[1] Koç University, KUIS AI Center  [2] Koç University, Computer Engineering Department
[3] Hacettepe University, Computer Engineering Department
[4] Amsterdam UMC, University of Amsterdam, Department of Medical Informatics
[5] Amsterdam Public Health, Methodology & Mental Health, Amsterdam, The Netherlands

## Abstract

This paper describes our two-stage system[1] for the Euphemism Detection shared task hosted by the 3rd Workshop on Figurative Language Processing in conjunction with EMNLP 2022. Euphemisms tone down expressions about sensitive or unpleasant issues like addiction and death. The ambiguous nature of euphemistic words or expressions makes it challenging to detect their actual meaning within a context. In the first stage, we seek to mitigate this ambiguity by incorporating literal descriptions into input text prompts to our baseline model. It turns out that this kind of direct supervision yields remarkable performance improvement. In the second stage, we integrate visual supervision into our system using *visual imageries*, two sets of images generated by a text-to-image model by taking terms and descriptions as input. Our experiments demonstrate that visual supervision also gives a statistically significant performance boost. Our system achieved the second place with an F1 score of 87.2%, only about 0.9% worse than the best submission.

## 1 Introduction

Recent advances in large pretrained language models allowed the computational linguistics community to tackle more knowledge-intensive tasks which require commonsense reasoning (Talmor et al., 2019; Bisk et al., 2020; Lin et al., 2021), and figurative language understanding (Pedinotti et al., 2021; Liu et al., 2022). In this work, we focus on a figurative language understanding task called *euphemism detection*. Euphemisms attempt to smooth harsh, impolite, or blunt expressions about taboo or sensitive topics like death and unemployment (Holder, 2008). For instance, when we speak of older people we often refer to *senior citizens* instead of a direct expression that can be seen as offensive.

Identifying euphemisms is challenging due to their natural ambiguity, i.e., the meaning of the term shifts depending on the context: '*Over the hill*' could either mean someone or something is *physically* over some hill (*literal*), or someone or something is *old*, past one's prime (*figurative*) (Lee et al., 2022). One cannot distinguish these two different senses without sufficient context. Thus, these terms are referred as *potentially euphemistic terms* (PETs) (Gavidia et al., 2022). Here, we propose a two-stage method for the Euphemism Detection shared task hosted by the 3rd Workshop on Figurative Language Processing at EMNLP 2022.

In the first stage, we manually collect literal descriptions for each PET. We then incorporate these descriptions into input text prompts to help the model distinguish figurative from literal usage. We demonstrate that this kind of extraneous linguistic supervision improves a strong baseline by a large margin. In the second stage, we attempt to answer the question, *"Is visual supervision also useful to infer the meaning behind a PET?"* To answer this question, we use a text-to-image model which takes terms and descriptions as input, and we generate two sets of images, which we denote as *visual imageries*. Our experiments show that using visual imagery provides the best results. A paired t-test points out that the improvement is statistically significant. Our qualitative analysis also suggests visual imageries are beneficial for analyzing PETs.

The rest of this paper is organized as follows. Section 2 describes our proposed solution. In Section 3, we share the details of our evaluation setup and design choices. Section 4 reports our experimental results. In Section 5, we briefly review the relevant literature. Section 6 outlines our conclusions and discuss the limitations of our approach.

## 2 Approach

In this section, we first formulate the euphemism detection task by describing a simple baseline

---

[1]Code is available at github.com/ilkerkesen/euphemism

model, and then explain how we extend it with the literal term descriptions and visual imagery.

## 2.1 Vanilla Baseline

Given a textual context $C$ with a potentially euphemistic term (PET) $T$, the aim of euphemism detection is to decide whether the candidate term $T$ is euphemistic ($y = 1$) or not ($y = 0$). Here, we only pick a sentence $S = [w_1, w_2, ..., w_n]$ which contains a candidate term $T$, and ignore the rest of the context $C$ at first. We use a pretrained language model LM as our initial baseline as below.

$$e_i = \text{EMBED}(w_i), \quad \hat{y} = \begin{cases} 1 & \hat{p} \geq 0.5, \\ 0 & \text{otherwise.} \end{cases}$$
$$\hat{p} = \text{LM}(e_1, e_2, ..., e_n),$$

$e_i$ denotes the word embedding of the $i^{\text{th}}$ token $w_i$, $\hat{p}$ is the probability that the candidate term $T$ is euphemistic, and $\hat{y}$ is the predicted label. EMBED is the embedding layer and LM denotes the language model that produces the probability $\hat{p}$.

## 2.2 Literal Descriptions

We extend the baseline model by supplying extra supervision with literal descriptions $D$ for each candidate term $T$ (which we collect manually). To make use of the literal descriptions, we create a textual prompt $X = [x_1, x_2, ..., x_n]$ for each sentence $S$, term $T$ and description $D$ as below.

$$X = [\text{Term:} T, \ \text{Description:} D, \ \text{Sentence:} S].$$

Then, we change the formulation,

$$e_i = \text{EMBED}(x_i)$$
$$\hat{p} = \text{LM}(e_1, e_2, ..., e_n),$$

where $e_i$ is the embedding for the $i^{\text{th}}$ token of the input prompt $X$.

## 2.3 Visual Imagery

We subsequently move beyond the text-only baselines by integrating visual modality into the Literal Descriptions baseline in the form of *visual imagery*. To accomplish this, we generate two sets of images $I_T = [I_T^{(1)}, I_T^{(2)}, ..., I_T^{(k)}]$ and $I_D = [I_D^{(1)}, I_D^{(2)}, ..., I_D^{(k)}]$, for each term and description pair, respectively. We denote these set of images as visual imageries. To obtain the visual imageries, we feed a text-to-image model T2I with terms and descriptions as input language,

$$I_T^{(k)} \sim \text{T2I}(T), \qquad I_D^{(k)} \sim \text{T2I}(D).$$

Next, we use a pretrained visual encoder (VE) to embed visual imageries.

$$v_T = \frac{1}{K} \sum_{k=1}^{K} \text{VE}(I_T^{(k)}), \quad v_D = \frac{1}{K} \sum_{k=1}^{K} \text{VE}(I_D^{(k)})$$

where $v_T$ denotes the visual imagery embedding of the candidate term $T$ and $v_D$ denotes the visual imagery embedding of the corresponding literal description $D$. $K$ is the number of images per term $T$ and description $D$. Thus, we reformulate the literal descriptions baseline as follows,

$$e_i = \text{EMBED}(x_i)$$
$$\hat{p} = \text{LM}(f_p(v_T), f_p(v_D), e_1, e_2, ..., e_n)$$

We make sure visual imagery embeddings are compatible with the word embeddings and language model LM by applying a linear projection layer $f_p$. We train each baseline using the negative log-likelihood objective.

## 3 Data and Implementation

**Data.** The euphemism detection dataset consists of two separate splits for training and testing purposes with 1573 and 394 examples, respectively. The test split is unlabeled. The whole data includes 131 different PETs. Since there is no data supplied for validation, we reserve 20% of the training data for this purpose. We only select the sentences with PETs and remove repetitive patterns of punctuation "@ @ @ ..." to decrease computational requirements by shortening the input language. We manually collect literal descriptions within 6 hours, and try to avoid impolite expressions like insults or slang phrases.

**Implementation.** We use DeBERTa-v3 base and large as our language model (He et al., 2021a,b). We generate the visual imageries $I_T$ and $I_D$ by using an open-source DALL-E implementation (Ramesh et al., 2021; Dayma et al., 2021).[2] The number of images per visual imagery $K$ is set to 9. We extract visual imagery embeddings $v_T$ and $v_D$ using CLIP's ViT-L/14 as our visual encoder (Radford et al., 2021). $f_p$ is a single linear layer, and we randomly initialize its weights. We use Adam optimizer with weight decay (Kingma and Ba, 2015; Loshchilov and Hutter, 2018). The learning rate is set to $5e^{-6}$ and $3e^{-6}$ for the experiments

---

[2] https://github.com/kuprel/min-dalle

| Model | LM | *validation* | *test* |
|---|---|---|---|
| Vanilla Baseline | Base | 79.84 ±2.23 | - |
| + Desc. | Base | 86.39 ±1.05 | 83.58 |
| + Desc. | Large | <u>88.89</u> ±1.35 | <u>85.74</u> |
| + Desc. + Imag. | Large | **90.11** ±1.59 | **87.16** |

Table 1: Quantitative results on the labeled data using F1 as evaluation metric. The last two columns respectively show the average score over different *validation* splits, and the ensemble performance achieved on the *test* split.

with DeBERTa-v3-base and DeBERTa-v3-large, respectively. We train our models for a maximum of 50 epochs using Tesla V100s and mixed precision. A typical experiment takes less than one hour with a batch size of 16. Due to the small dataset size, we perform multiple experiments and reserve a different portion of the labeled data for validation in each experiment. We report mean and standard deviation over all experiments, and use ensembling to evaluate our system on the test set.

## 4 Experimental Analysis

### 4.1 Quantitative Results

Table 1 presents the quantitative results of our experiments as ablation studies. We perform several experiments in a curriculum, where each following experiment activates a different feature (e.g. literal descriptions). We first implement a vanilla baseline using DeBERTa-v3-base, which lacks descriptions and imagery.

**Using Literal Descriptions.** In our first ablative analysis, we incorporate the literal term descriptions into the vanilla baseline described in Section 2.2. Integrating this supervision results in substantial performance improvement, i.e. $\approx 6.5$ points using F1 as evaluation metric.

**Larger Language Model.** We implement the literal descriptions model using a larger language model which is the large architecture of the DeBERTa-v3 model. Using a bigger LM gives 2 points performance improvement.

**Visual Imagery.** We now report on the visual imagery model explained in Section 2.3. This model additionally uses two different visual embedding vectors, denoted as visual imageries, which are generated by a text-to-image model using terms and descriptions. By using this extra visual supervision, we obtain 1.22 and 1.42 F1 score increments in validation and testing phases. A paired t-test is applied

to determine the significance of the results: We obtained a p-value of 0.032, which points out that this improvement is statistically significant ($p < 0.05$).

### 4.2 Qualitative Analysis

Figure 1 wraps up our qualitative analysis, where we share the collected descriptions and the generated visual imageries for some euphemistic terms. The first two examples show that if a term has a dominant literal meaning, the text-to-image V2I model produces images conveying the literal meaning instead of the figurative one. V2I can also produce visuals based upon individual word meanings as a consequence of being completely unconscious to the figurative meaning. This can be seen on the third example, where the model generates *lunch* images instead of vomiting for phrase '*lose one's lunch*'. Moreover, V2I can generate unrelated images for some terms as one can see on the *pro-life* and *able-body* examples. On the other hand, the text-to-image model V2I is well aware of some euphemism candidates as in the case with the last two examples. This phenomenon arises when the term has just one single meaning which is euphemistic.

In summary, a text-to-image model can be a complementary tool for analyzing figurative language: one can observe how models process these expressions. By looking at the produced images, we can recognize the terms with dominant literal meanings (e.g. *late*) or single euphemistic meaning (e.g. *lavatory*).

## 5 Related Work

**Euphemisms.** Recently, euphemisms have attracted the attention of the natural language processing community. Zhu et al. (2021) and Zhu and Bhat (2021) extract euphemistic phrases by using masked language modeling. A few work practices sentiment-oriented methods to recognize candidate euphemism phrases (Felt and Riloff, 2020; Gavidia et al., 2022; Lee et al., 2022). Most notably, Gavidia et al. (2022) replace PETs with their literal meanings and observe how the sentiment scores change. They demonstrate that using literal meanings produces higher scores for offensive speech and negative sentiment. Similarly, we also put literal meanings to use, but differently, by creating a textual input prompt. In this work, we also use the euphemism dataset they created.
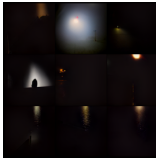
**Knowledge-augmented Language Understand-**
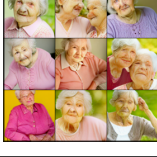
| Term | Description | $I_T$ | $I_D$ |
|------|-------------|-------|-------|
| late | old person, elderly | | |
| pass on | death, dying | | |
| lose one's lunch | vomit, vomiting, throwing up | | |
| pro-life | a person opposes abortion | | |
| able-body | not disabled | | |
| lavatory | restroom, toilet | | |
| senior citizen | old person, elderly | | |



Figure 1: Examples of collected literal descriptions for euphemistic terms and their visual imageries.

**ing.** External knowledge[3] can be either unstructured (i.e. text) or structured (i.e. graph). To benefit from unstructured knowledge, a text retriever collects related entries from an external corpus (Karpukhin et al., 2020; Guu et al., 2020). Conversely, structured knowledge integration may happen in two ways: explicit methods prefer to use knowledge in their input (Weijie Liu, 2020; Zhang et al., 2019), and implicit methods try to learn knowledge in their objective (Xiong et al., 2019; Shen et al., 2020). Some exceptions (Yu et al., 2022a; Shangwen Lv and Hu, 2020) combines both: they learn to predict graph embeddings and use these embeddings as input in their model concurrently. Similar to us, Yu et al. (2022b); Xu et al. (2021); Chakrabarty et al. (2021) also insert descriptions into their textual inputs.

**Visually-aided Language Understanding.** Several methods have been proposed to aid language learning with external visual knowledge. Most of these methods experiment on machine translation (MT). Calixto et al. (2019) propose a latent variable model for multi-modal MT, to learn an association between an image and its target language description. Long et al. (2021); Li et al. (2022) first synthesize an image conditioned on the source sentence, then use both the source sentence and the synthesized image to produce translation. Caglayan et al. (2020) obtain a lower latency in simultaneous MT by supplying visual context. Differently, Vokenization (Tan and Bansal, 2020) extend BERT (Devlin et al., 2019) by implementing visual token prediction objective to learn a mapping between tokens and associated images. Most relevantly, Lu et al. (2022) improve text-only language understanding performance in low-resource settings by using generated imagination as visual supervision.

## 6 Conclusion

In this paper, we described our two-stage method for the euphemism detection task. We first collected literal descriptions for PETs, inserted these descriptions into the model input, and showed that such linguistic supervision greatly boosts performance. We then supplied extra visual supervision using a text-to-image model, where we denote this kind of supervision as visual imageries. We achieved a statistically significant performance increase by using visual imageries in addition to the term descriptions. Our qualitative analysis on visual imageries also suggests that a text-to-image model can be a functional tool to break down how models interpret figures of speech.

**Limitations.** Due to working with a small-scale dataset, we were able to manually collect descriptions for the PETs. Collecting these descriptions using an automatic retrieval system would be more sophisticated. We also did not perform a detailed analyses of the results, which could help shed light on the contribution of each model component.

---

[3]Please check Zhu et al. (2022) for a comprehensive review of the related literature.

## References

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Ozan Caglayan, Julia Ive, Veneta Haralampieva, Pranava Madhyastha, Loïc Barrault, and Lucia Specia. 2020. Simultaneous machine translation with visual context. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2350–2361, Online. Association for Computational Linguistics.

Iacer Calixto, Miguel Rios, and Wilker Aziz. 2019. Latent variable model for multi-modal translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6392–6405, Florence, Italy. Association for Computational Linguistics.

Tuhin Chakrabarty, Debanjan Ghosh, Adam Poliak, and Smaranda Muresan. 2021. Figurative Language in Recognizing Textual Entailment. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3354–3361, Online. Association for Computational Linguistics.

Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Lê Khc, Luke Melas, and Ritobrata Ghosh. 2021. Dall·e mini.

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 preprint arXiv:2205.02728*.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Robert W Holder. 2008. *Dictionary of euphemisms*. Oxford University Press.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

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

Yi Li, Rameswar Panda, Yoon Kim, Chun-Fu (Richard) Chen, Rogerio S. Feris, David Cox, and Nuno Vasconcelos. 2022. VALHALLA: Visual Hallucination for Machine Translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5216–5226.

Bill Yuchen Lin, Haitian Sun, Bhuwan Dhingra, Manzil Zaheer, Xiang Ren, and William Cohen. 2021. Differentiable open-ended commonsense reasoning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4611–4625, Online. Association for Computational Linguistics.

Emmy Liu, Chenxuan Cui, Kenneth Zheng, and Graham Neubig. 2022. Testing the ability of language models to interpret figurative language. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4437–4452, Seattle, United States. Association for Computational Linguistics.

Quanyu Long, Mingxuan Wang, and Lei Li. 2021. Generative imagination elevates machine translation. In *Proceedings of the 2021 Conference of the North*

*American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5738–5748, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yujie Lu, Wanrong Zhu, Xin Wang, Miguel Eckstein, and William Yang Wang. 2022. Imagination-Augmented Natural Language Understanding. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4392–4402, Seattle, United States. Association for Computational Linguistics.

Paolo Pedinotti, Eliana Di Palma, Ludovica Cerini, and Alessandro Lenci. 2021. A howling success or a working sea? testing what BERT knows about metaphors. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 192–204, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. 2021. Zero-shot text-to-image generation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8821–8831. PMLR.

Jingjing Xu Duyu Tang Nan Duan Ming Gong Linjun Shou Daxin Jiang Guihong Cao Shangwen Lv, Daya Guo and Songlin Hu. 2020. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, USA*, pages 8449–8456. AAAI Press.

Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. 2020. Exploiting structured knowledge in text via graph-guided representation learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8980–8994, Online. Association for Computational Linguistics.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. CommonsenseQA: A question answering challenge targeting commonsense knowledge. 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 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.

Hao Tan and Mohit Bansal. 2020. Vokenization: Improving language understanding with contextualized, visual-grounded supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2066–2080, Online. Association for Computational Linguistics.

Zhe Zhao Zhiruo Wang Qi Ju Haotang Deng Ping Wang Weijie Liu, Peng Zhou. 2020. K-BERT: Enabling language representation with knowledge graph. In *Proceedings of AAAI 2020*.

Wenhan Xiong, Jingfei Du, William Yang Wang, and Veselin Stoyanov. 2019. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. In *International Conference on Learning Representations*.

Yichong Xu, Chenguang Zhu, Ruochen Xu, Yang Liu, Michael Zeng, and Xuedong Huang. 2021. Fusing context into knowledge graph for commonsense question answering. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1201–1207, Online. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yiming Yang, and Michael Zeng. 2022a. Jaket: Joint pre-training of knowledge graph and language understanding. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11630–11638.

Wenhao Yu, Chenguang Zhu, Yuwei Fang, Donghan Yu, Shuohang Wang, Yichong Xu, Michael Zeng, and Meng Jiang. 2022b. Dict-BERT: Enhancing language model pre-training with dictionary. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1907–1918, Dublin, Ireland. Association for Computational Linguistics.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL 2019*.

Chenguang Zhu, Yichong Xu, Xiang Ren, Bill Yuchen Lin, Meng Jiang, and Wenhao Yu. 2022. Knowledge-augmented methods for natural language processing. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Tutorial Abstracts*, pages 12–20, Dublin, Ireland. 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 *42nd IEEE Symposium on Security and Privacy*.