To appear in the Proceedings of the 42nd IEEE Symposium on Security and Privacy (Oakland 2021). San Francisco, CA. May 2021. © IEEE.

# Self-Supervised Euphemism Detection and Identification for Content Moderation

Wanzheng Zhu*, Hongyu Gong†*, Rohan Bansal‡, Zachary Weinberg§‡,
Nicolas Christin‡, Giulia Fanti‡, and Suma Bhat*
*University of Illinois at Urbana-Champaign, †Facebook,
‡Carnegie Mellon University, §University of Massachusetts, Amherst
*{wz6, spbhat2}@illinois.edu, †hygong@fb.com
‡{rohanb, nicolasc, gfanti}@andrew.cmu.edu, §zackw@cs.umass.edu

***Abstract*—Fringe groups and organizations have a long history of using *euphemisms*—ordinary-sounding words with a secret meaning—to conceal what they are discussing. Nowadays, one common use of euphemisms is to evade content moderation policies enforced by social media platforms. Existing tools for enforcing policy automatically rely on keyword searches for words on a "ban list", but these are notoriously imprecise: even when limited to swearwords, they can still cause embarrassing false positives [1]. When a commonly used ordinary word acquires a euphemistic meaning, adding it to a keyword-based ban list is hopeless: consider "pot" (storage container or marijuana?) or "heater" (household appliance or firearm?) The current generation of social media companies instead hire staff to check posts manually, but this is expensive, inhumane, and not much more effective. It is usually apparent to a human moderator that a word is being used euphemistically, but they may not know what the secret meaning is, and therefore whether the message violates policy. Also, when a euphemism is banned, the group that used it need only invent another one, leaving moderators one step behind.**

**This paper will demonstrate unsupervised algorithms that, by analyzing words in their sentence-level context, can both detect words being used euphemistically, and identify the secret meaning of each word. Compared to the existing state of the art, which uses context-free word embeddings, our algorithm for detecting euphemisms achieves 30–400% higher detection accuracies of *unlabeled* euphemisms in a text corpus. Our algorithm for revealing euphemistic meanings of words is the first of its kind, as far as we are aware. In the arms race between content moderators and policy evaders, our algorithms may help shift the balance in the direction of the moderators.**

***Index Terms*—Euphemism detection, Euphemism identification, Self-supervised learning, Masked Language Model (MLM), Coarse-to-fine-grained classification**

## I. Introduction

In recent years, large social media companies have been hiring content moderators to prevent conversations on their platforms that they deem to be inappropriate. Even though content moderation—the process of deciding what stays online and what gets taken down—often relies on organization-wide, centralized policies, the people who do this job often feel marginalized [2]. In 2019, The Verge reported on the emotional toll this work exacts, leading in some cases to post-traumatic stress disorder [3], [4].

† The work was done while Hongyu Gong was at UIUC.

Automation is an obvious way to assist content moderators. Ideally, they would be able to make a decision once and have it applied consistently to all similar content. One standard form of automated moderation is "ban-lists" of forbidden words. These are easy to implement, and define a clear-cut policy. However, they are also easy to evade: as soon as terms are added to a ban-list, the offenders will notice and adapt by inventing euphemisms to evade the filters [5]. Euphemisms are frequently words with other, innocuous meanings so they cannot be filtered unconditionally; they must be interpreted in context. To illustrate the problem, Table I gives many examples of euphemisms for a few terms that are frequently forbidden. Almost all of the euphemisms have innocuous meanings. Table II shows how a few of the euphemisms would be used in context, demonstrating that a human reader can often tell that a euphemistic meaning is intended even if they do not know exactly what the meaning is.

We present techniques for automated assistance with two tasks related to ban-list maintenance. Our algorithm for **euphemism detection** takes as input a set of *target keywords* referring to forbidden topics and produces a set of *candidate euphemisms* that may signify the same concept as one of the target keywords, without identifying which one. **Euphemism identification** takes a single euphemism as input and identifies its meaning. We envision these algorithms being used in a pipeline where moderators apply both in succession to detect new euphemisms and understand their meaning. For instance, if the target keywords are formal drug names (e.g., marijuana, heroin, cocaine), euphemism detection might find common slang names for these drugs (e.g., pot, coke, blow, dope) and euphemism identification could then associate each euphemism with the corresponding formal name (e.g., pot ⟶ marijuana, coke, blow ⟶ cocaine, dope ⟶ heroin).

In addition to their practical use in content moderation, our algorithms advance the state of the art in Natural Language Processing (NLP) by demonstrating the feasibility of self-supervised learning to process large corpora of unstructured, non-canonical text (e.g., underground forum posts), a challenging task of independent interest to the NLP community (e.g., [6]–[8]). Our algorithms require no manual annotation of text, and do not just rely on a "black box" pre-trained and fine-tuned model.

Table I
EXAMPLES OF THE VARIETY OF EUPHEMISMS ASSOCIATED WITH TARGET KEYWORDS IN COMMONLY FORBIDDEN CATEGORIES.

| Category | Target Keyword | Euphemisms |
|---|---|---|
| **Drugs** | Marijuana | blue jeans, blueberry, grass, gold, green, kush, popcorn, pot, root, shrimp, smoke, sweet lucy, weed |
| | Methamphetamine | clear, dunk, gifts, girls, glass, ice, nails, one pot, shaved ice, shiny girl, yellow cake |
| | Heroin | avocado, bad seed, ballot, beast, big H, cheese, chip, downtown, hard candy, mexican horse, pants |
| **Weapons** | Gun | bap, boom stick, burner, chopper, cuete, gat, gatt, hardware, heater, mac, nine, piece, roscoe, strap |
| | Bullet | ammo, cap, cop killer, lead, rounds |
| **Sex** | Breasts | bazooms, boobs, lungs, na-nas, puppies, tits, yabo |
| | Prostitution | call girl, girlfriend experience, hooker, poon, whore, working girl |

Table II
EXAMPLE USAGE FOR A FEW OF THE EUPHEMISMS IN TABLE I.

| **Example Sentences** (euphemism in boldface) | **Euphemism means** |
|---|---|
| 1. I had to shut up: the dealers had **gats**, my boys didn't. | *machine pistol* |
| 2. For all vendors of **ice**, it seems pretty obvious that it is not as pure as they market it. | *methamphetamine* |
| 3. I feel really good and warm behind the eyes. It's not something I've felt before on **pot** alone to this degree. | *marijuana* |
| 4. You can get an ounce of this **blueberry kush** for like $300 and it's insane. | *variety of marijuana* |
| 5. I'm looking for the **girlfriend experience**, without having to deal with an actual girlfriend. | *form of prostitution* |

Table III
EXAMPLE INFORMATIVE AND UNINFORMATIVE CONTEXTS. The word "heroin" has been masked out of each sentence below. In cases 1–3 it is clear that the masked word must be the name of an addictive drug, while in cases 4–6 there are more possibilities.

| Context | Example Sentences |
|---|---|
| **Informative** | 1. This 22 year old former [redacted] addict who I did drugs with was caught this night. |
| | 2. I have xanax real roxi opana cole and [redacted] for sale. |
| | 3. Six [redacted] overdoses in seven hours in wooster two on life support. |
| **Uninformative** | 4. Why is it so hard to find [redacted]? |
| | 5. The quality of this [redacted] is amazing and for the price its unbelievable. |
| | 6. Could we in the future see [redacted] shampoo? |

### A. Euphemism Detection

The main challenge of automated euphemism detection is distinguishing the euphemistic meaning of a term from its innocuous "cover" meaning [9]. For example, in sentence 2 of Table II, "ice" *could* refer to frozen water. To human readers, this is unlikely in context, because the purity of frozen water is usually not a concern for purchasers. Previous attempts to automate this task [9]–[12] relied on *static word embeddings* (e.g., word2vec [13], [14]), which do not attempt to distinguish different senses of the same word. They can identify slang terms with only one meaning (e.g., "ammo" for bullets), but perform poorly on euphemisms. Continuing the "ice" example, sentences using it in its frozen-water sense crowd out the sentences using it as a euphemism and prevent the discovery of the euphemistic meaning.

A newer class of *context-aware* embeddings (e.g. BERT [15]) learns a different word representation for every context in which the word appears, so they do not conflate different senses of the same word. However, since there are now several vectors associated with each word, the similarity of two words is no longer well-defined. This means context-aware embeddings cannot be substituted for the static embeddings used in earlier euphemism detection papers, which relied on word similarity comparisons. Also, not all contexts are equal. For any given term, some sentences that use it will encode more information about its meaning than others do. Table III illustrates the problem: it is easier to deduce what the masked term probably was in sentences 1–3 than sentences 4–6. This can be addressed by manually labeling sentences as informative or uninformative, but our goal is to develop an algorithm that needs no manual labels.

In this paper, we design an end-to-end pipeline for detecting euphemisms by making *explicit* use of context. This is particularly important to help content moderation of text in forums. We formulate the problem as an unsupervised fill-in-the-mask problem [15], [16] and solve it by combining a masked language model (e.g., used in BERT [15]) with a novel self-supervised algorithm to filter out uninformative contexts. The salience of our approach, which sets itself apart from other work on euphemism detection, lies in its non-reliance on linguistic resources (e.g., a sentiment lexicon) [17], search-engine results, or a seed set of euphemisms. As such it is particularly relevant to our application case—online platforms with free-flowing discourse that may adopt their own vernacular over time. Evaluating on a variety of representative datasets of online posts we found that our approach yields top-$k$ detection accuracies that are 30–400% higher than state-of-

the-art baseline approaches on all of the datasets, with top-20 accuracies as high as 40–45%, which is high for this problem. A qualitative analysis reveals that our approach also discovers correct euphemisms that *were not on our ground truth lists*, i.e., it can detect previously unknown euphemisms. Again, this is highly valuable in the context of Internet communities, where memes and slang lead to rapidly evolving vocabulary.

### B. Euphemism Identification

Once the usage of euphemisms has been detected, it is important to *identify* what each euphemism refers to. Unlike the task of deciding whether a given word refers to *any* target keyword (euphemism detection), the task of euphemism identification maps a given euphemism to a *specific* target keyword. This involves not only using the nuance of contextual information but also aggregating this information from related instances across the collection to make the inference. Again, referring to the 2nd and 3rd examples in Table II, we want to identify that *ice* refers to *methamphetamine* and *pot* to *marijuana*. To the best of our knowledge, no prior work has explicitly captured the meaning of a euphemism except for a few peripheral works (e.g., [9]) that identify the broad category of a euphemism (e.g., sedative, narcotic, or stimulant for a drug euphemism).

Euphemism identification poses four main challenges:

1) The distinction in meaning between the target keywords (e.g., cocaine and marijuana) is often subtle and difficult to learn from raw text corpora alone. 2) A given euphemism can be used in a euphemistic or non-euphemistic sense, adding the extra layer of linguistic nuance (Table IV). 3) No curated datasets that are publicly available are adequate to exhaustively learn a growing list of mappings between euphemisms and their target keywords. 4) It is unclear what linguistic and ontological resources one would need to automate this task.

In this paper, we propose the first approach to identify the precise meaning of a euphemism (e.g., mapping *pot* to *marijuana* and *Adam* to *ecstasy*). We systematically address the challenges identified above via a self-supervised learning scheme, a classification formulation, and a coarse-to-fine-grained framework. The key novelty lies in how we formulate the problem and solve it without additional resources or supervision. Going beyond demonstrating the feasibility of the task on a variety of datasets, we observe improvements in top-$k$ accuracy between 25–80% compared to constructed baseline approaches.

## II. Related Work

Natural language processing (NLP) has been used effectively in various security and privacy problems, including clustering illicit online pharmacies [18], [19], identifying sensitive user inputs [20], [21], and detecting spam [22]–[25]. However, although euphemisms have been widely studied in linguistics and related disciplines [26]–[34], they have received relatively little attention from the NLP [17], or security and privacy communities. Next, we review relevant prior work, including: 1) euphemism detection, 2) euphemism identification, and 3) self-supervised learning.

### A. Euphemism Detection

Euphemism detection is broadly related to the tasks of set expansion [44]–[49] and lexicon construction and induction [50]–[56]. Set expansion aims to expand a small set of seed entities into a complete set of relevant entities, and its goal is to find other target keywords from the same category. Lexicon construction and induction focus on extracting relations and building the lexicon-based knowledge graph in a structured manner. Their goals are different from ours, which is to find euphemisms of target keywords.

The specific task of euphemism detection has been studied in the NLP literature under a number of frameworks, including supervised, semi-supervised, and unsupervised learning, summarized in Table V. For example, Yang et al. [39] build a Keyword Detection and Expansion System (KDES) and apply it to the search results of Baidu, China's top search engine. KDES aims to infer whether a search keyword should be blocked by inspecting the associated search results. This approach requires general domain information with distant-supervision (i.e., the Baidu search engine), and is therefore not suitable for our unsupervised setting. Even if assuming search engine access, euphemisms for sensitive keywords are often short and innocent-looking (e.g., blueberries), which may result in mainly legitimate search results. Another set of relevant articles [6], [7] generate high-level information to analyze underground forums via an automated, top-down approach that blends information extraction and named-entity recognition. They present a data annotation method and utilize the labeled data to train a supervised learning-based classifier. Yet, the results depend heavily on the quality of annotation, and as shown by several researchers [6], [9], the model does not perform as well in cross-domain datasets, where it is outperformed by standard semi-supervised learning techniques.

Our work is most closely related to four state-of-the-art approaches [9], [11], [17], [40]. CantReader [9] aims to automatically identify "dark jargon" from cybercrime marketplaces. CantReader employs a neural-network based embedding technique to analyze the semantics of words, and detects euphemism candidates whose contexts in the background corpus (e.g., Wikipedia) are significantly different from those in the target corpus. Therefore, it takes as input a "dark" corpus (e.g., Silk Road anonymous online marketplace [43] forum), a mixed corpus (e.g., Reddit), and a benign corpus (e.g., English Wikipedia). Different from CantReader, we assume only access to a single target corpus – although we do rely on context-aware embeddings that could be pre-trained from a reference corpus like Wikipedia, and then fine-tuned to the target corpus. More importantly, we find that our approach outperforms CantReader, presumably because we explicitly use context.

Another relevant baseline [17] detects euphemisms instead by using sentiment analysis. It identifies a set of euphemism candidates using a bootstrapping algorithm for semantic lexicon induction. Though the methodology seems reasonable and

Table IV
EXAMPLE USES OF WORDS IN BOTH EUPHEMISTIC AND NON-EUPHEMISTIC SENSES. ALL SENTENCES ARE FROM REDDIT.

| Word | Meaning | Sentences |
|---|---|---|
| Coke | Cocaine | We had already paid $70 for some shitty weed from a taxi driver but we were interested in some coke and the cubans.<br>Why are coke dealers the most nuttiest?<br>OK so we have one gram high quality coke between 2 people who have never done more than a bump. |
| | Coca-Cola | I love having coke with ice.<br>When I buy coke at the beverage shop in UK, I pay neither a transaction fee nor an exchange fee.<br>Never have tried mixing coke with sprite or 7up. |
| Pot | Marijuana | My cousin did the same and when the legalized pot in dc they really started cracking down in virginia and maryland.<br>As far as we know he was still smoking pot but that was it.<br>Age 17, every time I smoked pot, I felt out of place. |
| | Container | No one would resist a pot of soup.<br>There's plenty of cupboard space in the kitchen for all your pots and pans.<br>Most lilies grow well in pots. |

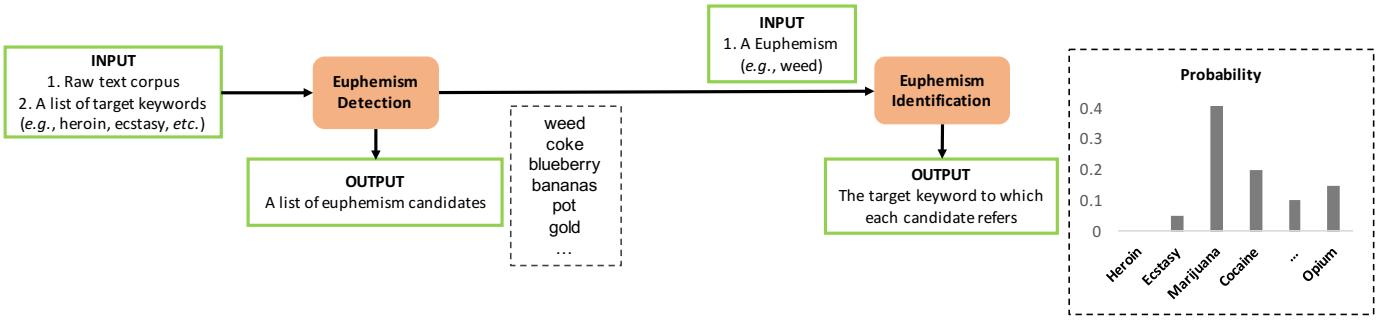


Figure 1. Euphemism detection and identification pipeline.

intuitive at first, it requires additional manual filtering process to refine the candidates and thus, fails to meet the requirement of automatic, large-scale detection that online content moderators desire. In yet another approach, Magu et al. [11] and Taylor et al. [40] propose two algorithms that leverage word embeddings and community detection algorithms. Magu et al. [11] generates a cluster of euphemisms by the ranking metric of eigenvector centralities [57], [58]. Due to the intrinsic nature of the algorithm, this approach requires a starting euphemism seed to find others. Taylor et al. [40] creates neural embedding models that capture the word similarities, uses graph expansion and the PageRank scores [59] to bootstrap initial seed words, and finally enriches the bootstrapped words to learn out-of-dictionary terms that behave like euphemisms. However, the approaches of Magu et al. [11] and Taylor et al. [40] were tested on one single dataset. Unfortunately, we do not find their performance to be as strong on the multiple datasets we evaluate.

### B. Euphemism Identification

To the best of our knowledge, no work has explicitly attempted to infer euphemism meaning. Yuan et al. [9] tackles a related problem by identifying the hypernym of euphemisms (e.g., whether it refers to a drug or a person). In a more general sense, the task of euphemism identification is also related to sense discovery of unknown words [60], [61] and word sense disambiguation [62]–[65]. While sense discovery aims to understand the meaning of an unknown word by generating a definition sentence, word sense disambiguation focuses on identifying which sense of a word is used in a sentence, given a set of candidate senses and relies heavily on a sense-tagged reference corpus, created by linguists and lexicographers. However, neither of these are able to capture nuanced differences between a group of semantically-similar target keywords in the same category.

### C. Self-supervised Learning

The technical innovations in our work rely heavily on *self-supervision*, a form of unsupervised learning where the data itself provides the supervision [66]. Self-supervision was designed to make use of vast amounts of unlabelled data (e.g., free text, images) by constructing a supervised learning task from the data itself to predict some attribute of the data. For example, to train a text prediction model, one can take a corpus of text, mask part of the sentence, and train the model to predict the masked part; this workflow creates a supervised learning task from unlabelled data. Self-supervision has been widely used in language modeling [15], [67]–[71],

Table V
RELATED WORK ON EUPHEMISM DETECTION.

| System | Learning Type | Categories (Platform) | Required Input | Approach Keywords |
|---|---|---|---|---|
| **Durrett et al. (2017) [6]** | Supervised & semi-supervised | Cybercriminal wares (Darkode), cybersecurity (Hack Forums), search engine optimization techniques (Blackhat), data stealing tools and services (Nulled) | A fully labelled dataset with annotated euphemisms | Support Vector Machine (SVM), Conditional Random Field (CRF) |
| **Pei et al. (2019) [35]** | Supervised | General topics (Online Slang Dictionary) | Slang-less corpus (Penn Treebank) as the negative examples, Slang-specific corpus (Online Slang Dictionary) as the positive examples | Linguistic features, bidirectional LSTM [36] , Conditional Random Field (CRF) [37], multilayer perceptron (MLP) [38] |
| **Zhao et al. (2016) [12]** | Unsupervised | Cybersecurity (QQ) | Target keywords, online search service | Unsupervised learning, word embedding (i.e., word2vec), Latent Dirichlet Allocation (LDA) |
| **Yang et al. (2017) [39]** | Unsupervised | Sex, gambling, dangerous goods, surrogacy, drug, faked sites (Baidu) | Target keywords, online search service | Web analysis, keywords expansion, candidate filtering |
| **Hada et al. (2020) [10]** | Unsupervised | Drug trafficking and enjo kosai (Twitter) | A clean background corpus, a bad corpus related to illegal transactions, a set of euphemism seeds | Word embedding (word2vec), cosine similarity |
| **Felt et al. (2020) [17]** | Unsupervised | Firing, lying and stealing (The English Gigaword corpus) | Category name, a lexicon dictionary (i.e., Gigaword) | Sentiment analysis, bootstrapping, semantic lexicon induction |
| **Taylor et al. (2017) [40]** | Unsupervised | Hate speech (Twitter) | The text corpus, category name | Word embedding (fasttext [41] and dependency2vec [42]), community detection, bootstrapping |
| **Magu et al. (2018) [11]** | Unsupervised | Hate speech (Twitter) | The text corpus, a euphemism seed | Word embedding (word2vec), network analysis, centrality measures |
| **Yuan et al. (2018) [9]** | Unsupervised | Sale and trade of hacking services and tools (Darkode), blackhat hacking (Hack Forums), data stealing tool and service (Nulled), illegal drug (Silk Road) | A background corpus (e.g., Wikipedia), A dark corpus (e.g., Silk Road [43]), A mixed corpus (e.g., Reddit) | Word embedding, semantic comparison across corpora |
| **Our algorithm** | Unsupervised | Drug (Reddit), weapon (Gab, SlangPedia, [6], [7]), sexuality (Gab) | The text corpus, target keywords | Contextual information, masked language model, BERT |

representation learning [72]–[75], robotics [76]–[78], computer vision [79]–[82] and reinforcement learning [83]–[85]. One of our contributions is to generalize and extend the idea of self-supervision to the task of euphemism identification.

## III. PROBLEM DESCRIPTION

In this study, we assume a content moderator has access to a textual corpus (e.g., a set of posts from an online forum), and is required to moderate content related to a given list of target keywords. In practice, forum users may use *euphemisms*—words that are used as substitutes for one of the target keywords. We have two goals, euphemism detection and euphemism identification, defined as follows: 1) *Euphemism detection:* Learn which words are being used as euphemisms for target keywords. A moderator can use this to filter content that may need to be moderated. 2) *Euphemism identification:* Learn the meaning of euphemisms. This can be used by the moderator to understand context, and individually review content that uses euphemisms.

As shown in Figure 1, these two tasks are complementary and

form, together, a content moderation pipeline. The euphemism detection task takes as input (a) the raw text corpus, and (b) a list of target keywords (e.g., heroin, marijuana, ecstasy, etc.). The expected output is an ordered ranked list of euphemism candidates, sorted by model confidence. The euphemism identification module takes as input a euphemism (e.g., weed) and outputs a probability distribution over the target keywords in the list. For example, if we feed the euphemism *weed* into this module, the output should be a probability distribution over keywords, with most of the mass on *marijuana*.

**Remark**: We use the term "category" to denote a topic (i.e., drug, weapon, sexuality). We use "target keyword" to refer to the specific keyword in each category users might be trying to use euphemisms for (e.g., "marijuana" and "heroin" are examples of target keywords in the drug category).

## IV. Proposed Approach

We next discuss in detail our proposed euphemism detection approach in Section IV-A and the proposed euphemism identification approach in Section IV-B.

### A. Euphemism Detection

We formulate the euphemism detection problem as an unsupervised fill-in-the-mask problem and solve it by combining self-supervision with a Masked Language Model (MLM), an important modeling idea behind BERT [15]. Our proposed approach to euphemism detection has three stages (represented in Figure 2): 1) Extracting contextual information, 2) Filtering out uninformative contexts, and 3) Generating euphemism candidates.

**Contextual information extraction.** Taking as input all the target keywords, this stage first extracts the masked sentences of all the keywords. Here, a masked sentence refers to a sentence excluding the target keyword. Taking the first example sentence in Table III as an example, the corresponding masked sentence is "This 22 year old former [MASK] addict who i did drugs with was caught this night.". A collection of all masked sentences of the target keywords serves as the source of the relevant and crucial contextual information.

**Denoising contextual information.** Not all masked sentences are equally informative. There may be instances where the mask token (i.e., "[MASK]") can be filled by more than one target term, or words unrelated to the target terms, without affecting the quality of the sentence. The fourth example sentence in Table III is one such case, where the masked sentence "Why is it so hard to find [MASK]?" is not specific to a drug; the mask token can be filled by many words, including nouns such as "jobs", "gold" and even pronouns such as "him". Such masked sentences (example sentences 4–6 in Table III) are generic and lack relevant context for disambiguating a polysemous word.

To filter such generic masked sentences, we propose a self-supervised approach that makes use of the Masked Language Model (MLM) proposed in BERT [15]. Recall that self-supervision involves creating a new learning task from unlabeled data. An MLM aims to find suitable replacements of the masked token, and outputs a ranked list of potential replacement terms. We start by fine-tuning the "bert-base-uncased" pre-trained model[1] to the language of the of domain-specific body of text (for instance, a collection of Reddit posts for identifying drug-related euphemisms).

Empirically, we find that if a masked sentence is specific to a target category (e.g., drug names), words related to the target category will be ranked highly in the replacement list. In contrast, if the masked sentence is generic, the highly ranked replacements are more likely to be random words unrelated to the target category (e.g., "jobs", "gold", "him"). Therefore, we set an MLM threshold $t$ to filter out the generic masked sentences. Considering the ranked list of replacements for the mask token, if any target keyword appears in the top $t$ replacement candidates for the masked sentence, we consider the masked sentence to be a valid instance of a context. Otherwise, it is considered to be a generic one and filtered out. We set the threshold $t$ to $t = 5$ in our experiments and discuss its sensitivity in Section VI-B.

**Candidate euphemism generation.** Once we have (a) a pre-trained language model that is fine-tuned to the text corpus of interest, and (b) a filtered list of masked sentences, we are ready to generate euphemism candidates. For each masked sentence $m$, and for each word candidate $c$ in the vocabulary (i.e., all words available in the BERT pre-trained model), we compute its MLM probability (the probability of the word occurring in $m$ as predicted by the language model) $h_{c,m}$ by a pre-trained BERT model. Therefore, given a set of masked sentences, the weight $w_c$ of a word candidate $c$ is calculated as: $w_c = \sum_{m'} h_{c,m'}$. The final generation stage simply ranks all word candidates by their weights.

To clarify, we use the masked language model twice—once for filtering the masked sentences and a second time for generating the euphemism candidates from the masked sentences.

### B. Euphemism Identification

Once the euphemisms are detected, we aim to identify what target keyword each euphemism refers to. Taking the second and third example sentences in Table II, we want to identify that "ice" refers to "methamphetamine" and "pot" to "marijuana". Euphemism identification has been acknowledged as a highly challenging task [9], due to two problems:

- *Resource challenge*: No publicly-available, curated datasets are adequate to exhaustively learn a growing list of mappings between euphemisms and their target keywords. Moreover, it is unclear what linguistic and ontological resources one would need to automate this task.
- *Linguistic challenge*: The distinction in meaning between the target keywords (e.g., cocaine and marijuana) is often subtle and difficult to learn from raw text corpora alone. Even human experts are often unable to accurately identify what a euphemism refers to by looking at a single sentence. A second linguistic challenge is related to the ambiguity

[1]https://huggingface.co/transformers/model_doc/bert.html#bertformaskedlm

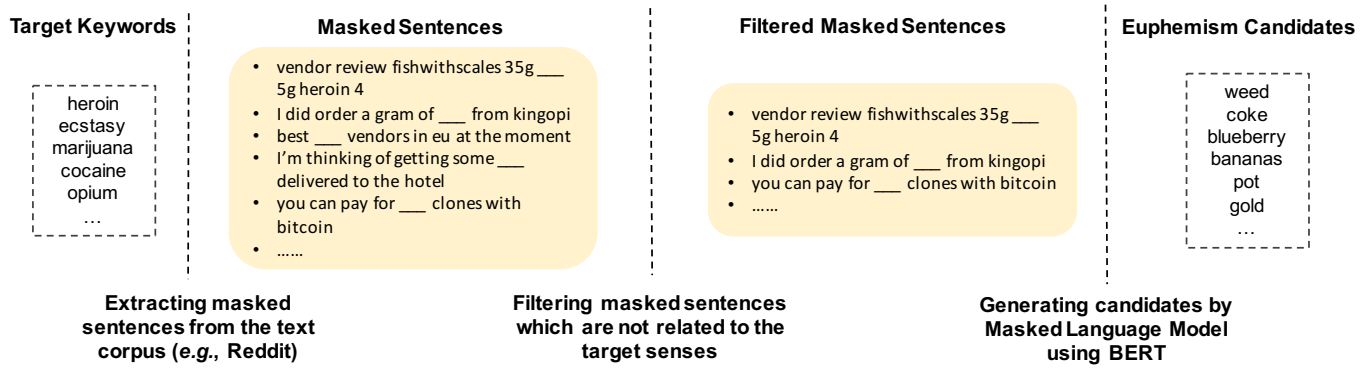


Figure 2. An overview of the euphemism detection framework.

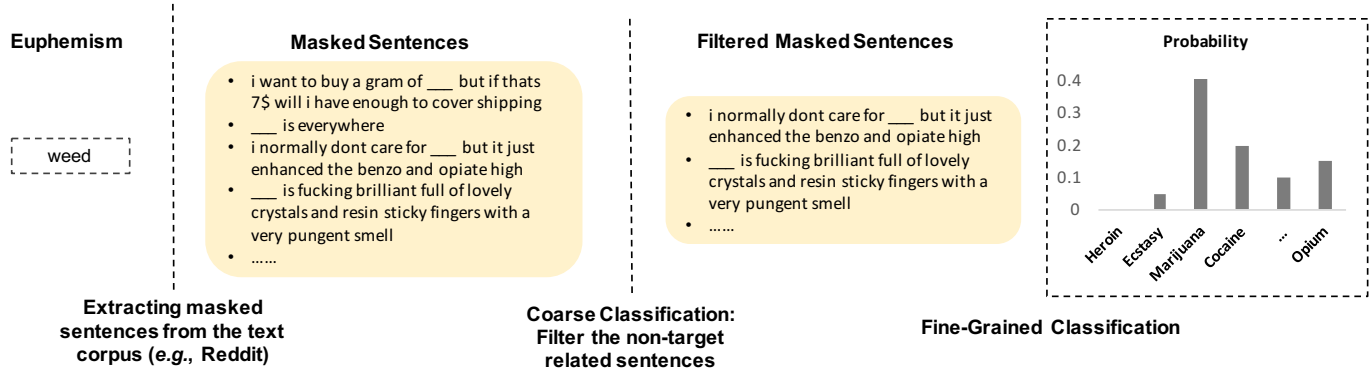


Figure 3. An overview of the euphemism identification framework.

of the euphemism itself. A given euphemism can be used in a euphemistic or non-euphemistic sense, adding the extra layer of linguistic nuance (Table IV).

We tackle the *resource challenge* by designing a self-supervised learning scheme. We extract all sentences that include the target keywords (e.g., cocaine, marijuana, heroin), mask the target keywords, and consider the masked sentences as training samples. This allows us to automatically construct a labeled dataset, where the input samples are the masked sentences, and their respective target keywords are labels.

To address the *linguistic challenge*, we adopt a coarse-to-fine-grained classification scheme. Such hierarchical schemes have shown better discriminative performance in various tasks [86]–[88]. The coarse classifier is a binary classifier that outputs whether a sentence is related to a specific category (e.g., drug) or not. It aims to filter out sentences where the euphemism candidates do not occur in a euphemistic sense. The fine-grained classifier is a multi-class classifier trained on the curated dataset from the self-supervised learning scheme; this aims to learn a specific mapping from the masked sentence to the target keyword. We discuss the details of these classifiers below; first, we step through an example of the end-to-end pipeline.

**Example**: Suppose our euphemism detection pipeline outputs the term "weed". We aim to generate a probability distribution over target keywords, with most of the mass on marijuana (Figure 3). Assume that we already have a trained coarse classifier and a trained fine-grained classifier (training details will be discussed below in IV-B1). We first extract all masked sentences that previously contained "weed" from the text corpus. Second, using the coarse classifier, we filter out the masked sentences that are unrelated to the target category (i.e., all masked sentences that do not discuss something drug-related). Then, we use the filtered masked sentences as inputs to the fine-grained multi-class classifier, and obtain the target keyword label for each masked sentence. We now have a list of labels for the euphemism "weed" (e.g., 36,100 "marijuana" labels, 4,200 "ecstasy" labels, etc.) and the final output for a euphemism is a probability distribution by the number of labels for each target keyword.

*1) Training Details:* As discussed above, two classifiers need to be trained: 1) A coarse classifier to filter out the masked sentences of the euphemism words not associated with their euphemistic sense and, 2) A multi-class classifier to determine the target keyword to which the euphemism refers.

**Coarse Classifier**: The coarse classifier is a binary classifier that decides whether a masked sentence is related to the target keywords or not. Obtaining positive instances is easy: we collect all the masked sentences of the target keywords (e.g., we obtain the masked sentences from Table III). To obtain the negative instances, we adopt a negative sampling approach [13]; we randomly choose a sentence in the whole text corpus and randomly mask a token. Since the corpus is large and diverse,

we assume the randomly chosen masked sentence is unrelated to the target keyword. With high probability, this assumption is correct. To create a balanced dataset, we select as many negative instances as there are positive ones. This set of positive and negative instances constitutes the training set, with masked sentences and their respective labels to indicate whether a masked sentence is related to the target keywords or not. We use 70% of the data instances for training, 10% for validation, and 20% for testing. We select an LSTM recurrent neural network model [89] with an attention mechanism [90] for its ability to learn to pay attention to the correct segments of an input sequence. We obtain 98.8% training accuracy and 90.1% testing accuracy. Our experiments also include other classification models—we discuss our selection in Section VI-A1.

**Multi-class Classifier**: As presented above, we use as inputs the masked sentences and as labels the target keywords. Empirically, we obtained good performance from a multinomial logistic regression classifier [91]. We first represent each word as a one-hot vector[2]. We then represented each sentence as the average of its member words' encodings. By using the same data splitting ratio as the coarse classifier, we obtain a training accuracy of 55% and a testing accuracy of 24% for the drug dataset (described in Section V). As a point of comparison, with 33 target names in the drug dataset a random guess would yield an accuracy of 3.3%. We discuss the results for other classification models in Section VI-A2.

## V. Empirical Evaluation

In this section, we empirically evaluate the performance of our proposed approach and compare with that a set of baseline models on both euphemism detection (in Section V-B) and euphemism identification (in Section V-C).

### A. Experimental Setup

We implemented all models in Python 3.7 and conducted all the experiments on a computer with twenty 2.9 GHz Intel Core i7 CPUs and one GeForce GTX 1080 Ti GPU.

**Datasets**: We empirically validate our proposed model on three separate datasets related to three broad areas of euphemism usage: drugs, weapons, and sexuality. For the algorithm to be applicable to a dataset, we require two kinds of inputs: 1) the raw text corpus from which we extract the euphemisms and their masked sentences, and 2) a list of target keywords (e.g., heroin, marijuana, ecstasy, etc.). For the purpose of carrying out a quantitative evaluation of the euphemism detection and identification approaches and comparing them with prior art, we rely on a ground truth list of euphemisms and their target keywords. Ideally, such a list should contain all euphemisms for the evaluation of euphemism detection, and a one-to-one mapping from each euphemism to its actual meaning, for the evaluation of euphemism identification.

- *Drug dataset*: From a publicly available data repository [92], we extracted 1,271,907 posts from 46 distinct "subreddits"[3] related to drugs and dark web markets, including the largest ones—"Bitcoin" (565,614 posts), "Drugs" (373,465 posts), "DarkNetMarkets" (125,300 posts), "SilkRoad" (22,989 posts), "DarkNetMarketsNoobs" (22,699 posts). A number of these subreddits were banned from the platform in early 2018 [93]. As a result, the posts collected were authored between February 9, 2008 and December 31, 2017. While online drug trade dates back (at least) to USENET groups in the 1990s, it truly picked up mainstream traction with the emergence of the Silk Road in 2011. Our data corpus captures these early days, as well as the more mature ecosystem that followed [94].
  For ground truth, we use a list of drug names and corresponding euphemisms compiled by the (USA) Drug Enforcement Administration [95]. This list is intended as a practical reference for law enforcement personnel. Due to the rapidly evolving language used in the drug-use subculture, it cannot be comprehensive or error-free, but it is the most reliable ground truth available to us.
- *Weapon dataset*: The raw text corpus comes from a combination of the corpora collected by Zanettou et al. [96], Durrett et al. [6], Portnoff et al. [7] and the examples in Slangpedia[4]. The combined corpus has 310,898 posts. Both the ground truth list of weapon target keywords and the respective euphemisms are obtained from The Online Slang Dictionary[5] (one of the most comprehensive slang thesaurus available), Slangpedia, and The Urban Thesaurus[6].
- *Sexuality dataset*: The raw text corpus comes from the Gab social networking services[7]. We use 2,894,869 processed posts, collected from Jan 2018 to Oct 2018 by PushShift.[8] Both the ground truth list of sexuality target keywords and the euphemisms are obtained from The Online Slang Dictionary.

### B. Euphemism Detection

We evaluate the performance of euphemism detection in this section.

**Evaluation Metric**: For each dataset, the input is an unordered list of target keywords and the output is an ordered ranked list of euphemism candidates. Given the nature of the output, we evaluate the output using the metric precision at $k$ ($P@k$), which is commonly used in information retrieval to evaluate how well the search results corresponded to a query [97]. $P@k$, ranging from 0 to 1, measures the proportion of the top $k$ generated results that are correct (in our case, valid euphemisms), which we calculate with respect to the ground truth list for

[2]One-hot encoding is used to represent a categorical variable whose values do not have an ordinal relationship. The one-hot encoding of a word $v_i \in V$, where $V$ denotes the vocabulary, is a $|V|$-dimensional vector of all zeros except for a 1 at the $i$th index.

[3]Forums hosted on the Reddit website, and associated with a specific topic.
[4]https://slangpedia.org/
[5]http://onlineslangdictionary.com/
[6]https://urbanthesaurus.org/
[7]https://gab.com/
[8]Available at https://files.pushshift.io/gab/

each dataset. In cases where an algorithm recovers only one word of a multi-word euphemism (e.g., “Chinese" instead of “Chinese tobacco"), we treat the candidate as incorrect. Because of the known shortcoming that $P@k$ fails to take into account the positions of the relevant documents [98], we report $P@k$ for multiple values of $k$ ($k = 10, 20, 30, 40, 50, 60, 80, 100$) to resolve the issue.

We are unable to measure recall for the following two reasons: 1) Some euphemisms in the ground truth list do not appear in the text corpus at all and using recall as a measure can result in a misrepresentation of the performance of the approaches; 2) Those euphemisms that indeed appear in the text corpus, may not have been used in the euphemistic sense. For example, “chicken" is a euphemism for “methamphetamine,” but it could have been used only in the animal sense in the corpus.

**Baselines**: We compare our proposed approach with the following competitive baseline models:

- **Word2vec**: We use the word2vec algorithm [13], [14] to learn the word embeddings (100-dimensional) for all the words separately for the Drug, Weapon and Sexuality datasets. We then choose as euphemism candidates those words that are most similar to the input target keywords, in terms of cosine similarity (average similarity between the word and all input target keywords). This approach relates words by implicitly accounting for the context in which they occur.
- **TF-IDF + word2vec**: Instead of treating all the words in the dataset equally, this method first ranks the words by their potential to be euphemisms. Toward this, we calculate the TF-IDF weights of the words [97] with respect to a background corpus (i.e., Wikipedia[9]), which captures a combination of the frequency of a word and its distinct usage in a given corpus. The idea is inspired by the assumption that words ranked higher based on TF-IDF in the target corpus have a greater chance of being euphemisms than those ranked lower [11]. After the pre-selection by TF-IDF, we then generate the euphemism candidates by following the Word2vec approach above.
- **CantReader**[10] [9] employs a neural-network based embedding technique to analyze the semantics of words, detecting the euphemism candidates whose contexts in the background corpus (e.g., Wikipedia) are significantly different from those in the dark corpus.
- **SentEuph** [17] recognizes euphemisms by the use of sentiment analysis. It lists a set of euphemism candidates using a bootstrapping algorithm for semantic lexicon induction. For a fair comparison with our approach, we do not include the manual filtering stage of the algorithm proposed by Felt and Riloff [17].
- **EigenEuph** [11] leverages word embeddings and a community detection algorithm, to generate a cluster of euphemisms by the ranking metric of eigenvector centralities.
- **GraphEuph**[11] [40] also identifies euphemisms using word embeddings and a community detection algorithm. Specifically, it creates neural embedding models that capture word similarities, uses graph expansion and the PageRank scores [59] to bootstrap an initial set of seed words, and finally enriches the bootstrapped words to learn out-of-dictionary terms that behave like euphemisms.
- **MLM-no-filtering** is a simpler version of our proposed approach and shares its architecture. The key difference from our proposed approach is that instead of filtering the noisy masked sentences, it uses them *all* to generate the euphemism candidates. In effect, this baseline serves as an ablation to understand the effect of the filtering stage.

For a fair comparison of the baselines, we experimented with different combinations of parameters and report the best performance for each baseline method.

**Results**: Table VI summarizes the euphemism detection results. Our proposed approach outperforms all the baselines by a wide margin for the different settings of the evaluation measure on all the three datasets we studied.

The most robust baselines over the three datasets are TF-IDF + word2vec, EigenEuph and MLM-no-filtering. When compared with Word2vec, the superior performance of TF-IDF + word2vec lies in its ability to select a set of potential euphemisms by calculating the TF-IDF with a background corpus (i.e., Wikipedia). While this pre-selection step works well (relative to Word2vec) on the Drug and Sexuality datasets, it does not impact the performance on the Weapon dataset. A plausible explanation for this is that the euphemisms do not occur very frequently in comparison with the other words in the Weapons corpus and therefore, are not ranked highly by the TF-IDF scores.

SentEuph [17]’s comparatively poor performance is explained by the absence of the required additional manual filtering stage to refine the results. As mentioned before, this was done to compare the approaches based on their automatic performance alone. GraphEuph [40] shows a reasonable performance on the Drug dataset, but fails to detect weapon- and sexuality-related euphemisms. This limits the generalization of the approach that was tested only on a hate speech dataset by Taylor et al. [40]. The approach of CantReader [9] seems to be ineffective because not only does it require additional corpora to make semantic comparisons—a requirement that is ill-defined because the nature of the additional corpora needed for a given dataset is not specified—but also because the results of CantReader are quite sensitive to parameter tuning. We were unable to reproduce the competitive results reported by Yuan et al. [9], even after multiple personal communication attempts with the authors. By comparing the performance of our approach and that of the ablation MLM-no-filtering, we conclude that the proposed filtering step is effective in eliminating the noisy masked sentences and is indispensable for reliable results.

[9]https://dumps.wikimedia.org/enwiki/

[10]https://sites.google.com/view/cantreader

[11]https://github.com/JherezTaylor/hatespeech_codewords

Table VI
RESULTS ON EUPHEMISM DETECTION. BEST RESULTS ARE IN BOLD.

| | | $P@10$ | $P@20$ | $P@30$ | $P@40$ | $P@50$ | $P@60$ | $P@80$ | $P@100$ |
|---|---|---|---|---|---|---|---|---|---|
| **Drug** | **Word2vec** | 0.10 | 0.10 | 0.09 | 0.09 | 0.08 | 0.09 | 0.08 | 0.09 |
| | **TF-IDF + word2vec** | 0.30 | 0.25 | 0.20 | 0.20 | 0.16 | 0.17 | 0.16 | 0.18 |
| | **CantReader [9]** | 0.00 | 0.00 | 0.07 | 0.10 | 0.08 | 0.12 | 0.12 | 0.10 |
| | **SentEuph [17]** | 0.10 | 0.10 | 0.07 | 0.05 | 0.08 | 0.07 | 0.09 | 0.07 |
| | **EigenEuph [11]** | 0.30 | 0.30 | 0.30 | 0.25 | 0.22 | 0.22 | 0.20 | 0.19 |
| | **GraphEuph [40]** | 0.20 | 0.15 | 0.13 | 0.13 | 0.14 | 0.17 | 0.14 | 0.11 |
| | **MLM-no-filtering** | 0.30 | 0.30 | 0.28 | 0.30 | 0.26 | 0.26 | 0.28 | 0.26 |
| | **Our Approach** | **0.50** | **0.45** | **0.47** | **0.42** | **0.46** | **0.42** | **0.38** | **0.36** |
| **Weapon** | **Word2vec** | 0.30 | 0.30 | 0.27 | 0.23 | 0.18 | 0.20 | 0.20 | 0.18 |
| | **TF-IDF + word2vec** | 0.30 | 0.25 | 0.20 | 0.17 | 0.16 | 0.18 | 0.20 | 0.18 |
| | **CantReader [9]** | 0.20 | 0.20 | 0.17 | 0.18 | 0.16 | 0.17 | 0.13 | 0.11 |
| | **SentEuph [17]** | 0.00 | 0.00 | 0.03 | 0.05 | 0.06 | 0.05 | 0.05 | 0.04 |
| | **EigenEuph [11]** | 0.30 | 0.20 | 0.13 | 0.10 | 0.08 | 0.07 | 0.05 | 0.04 |
| | **GraphEuph [40]** | 0.00 | 0.05 | 0.03 | 0.05 | 0.04 | 0.03 | 0.03 | 0.02 |
| | **MLM-no-filtering** | 0.30 | 0.30 | 0.20 | 0.17 | 0.18 | 0.18 | 0.15 | 0.15 |
| | **Our Approach** | **0.40** | **0.45** | **0.37** | **0.35** | **0.32** | **0.28** | **0.25** | **0.20** |
| **Sexuality** | **Word2vec** | 0.10 | 0.05 | 0.07 | 0.08 | 0.08 | 0.08 | 0.09 | 0.09 |
| | **TF-IDF + word2vec** | 0.40 | 0.25 | 0.20 | 0.20 | 0.20 | 0.17 | 0.15 | 0.13 |
| | **CantReader [9]** | 0.10 | 0.10 | 0.07 | 0.08 | 0.06 | 0.08 | 0.09 | 0.10 |
| | **SentEuph [17]** | 0.10 | 0.10 | 0.08 | 0.10 | 0.08 | 0.10 | 0.08 | 0.06 |
| | **EigenEuph [11]** | 0.20 | 0.15 | 0.13 | 0.15 | 0.16 | 0.18 | 0.14 | 0.11 |
| | **GraphEuph [40]** | 0.00 | 0.00 | 0.03 | 0.05 | 0.04 | 0.03 | 0.04 | 0.03 |
| | **MLM-no-filtering** | 0.50 | **0.40** | 0.30 | 0.23 | 0.22 | 0.22 | 0.19 | 0.15 |
| | **Our Approach** | **0.70** | **0.40** | **0.33** | **0.33** | **0.28** | **0.25** | **0.23** | **0.19** |

**False positive analysis**: By studying the false positives in our results, we recovered several euphemisms that were not included in our ground truth list. Table X in Appendix A shows sentences associated with 10 of the top 16 false positive euphemisms from the drug dataset. Several of these are true euphemisms for drug keywords that were not present in the DEA ground truth list (e.g., md, l, mushrooms). Others are not illicit drugs (e.g., alcohol, cigarettes), but they are used in this corpus in a way that is closely related to how people use drug names, and reveal new usage patterns. For example, the sentences for "cigarettes" indicate that people appear to be combining cigarette use with other drugs, such as PCP. Similarly, the sentences containing "alcohol" reveal that people are dissolving illicit drugs in alcohol. Of these 10 false positives (according to our ground truth dataset), only five are actually false positives; these words are semantically related to the drug keywords, but they are not proper euphemisms (e.g., "pressed" is a form factor for drug pills).

### C. Euphemism Identification

For each euphemism that we have successfully detected, we now evaluate euphemism identification.

**Evaluation Metric**: For each euphemism, we generate a probability distribution over all target keywords and therefore, obtain a ranked list of the target keywords. We evaluate the top-$k$ accuracy ($Acc@k$), which measures how often the ground truth label (target keyword) falls in the top $k$ values of our generated ranked list.

**Baselines**: Given the lack of related prior work for the task of euphemism identification, we establish a few baseline methods and compare our proposed approach with them.

- **Word2vec**: For each euphemism, we select the target keyword that is closest to it using the measure of cosine similarity. Here we compare the word embeddings (100-dimensional) obtained by training the word2vec algorithm [13], [14] on each text corpus separately.
- **Clustering + word2vec**: For each euphemism, we cluster all its masked sentences, represented as the average of the word embeddings of the component words, using a $k$-means algorithm (we set $k = 2$). By clustering, our aim is to separate the masked sentences into two groups (ideally one group of informative masked sentences and the other group of uninformative masked sentences as presented in Table III) and to filter out the uninformative masked sentences that are not related to the target keywords. Then, we compare the embeddings of the filtered masked sentences of the euphemism and the target keywords using the measure of cosine similarity. The target keyword that is most similar to the filtered masked sentences is selected for identification.
- **Binary + word2vec**: similar to our approach, we use a binary classifier to filter out noisy masked sentences that are not related to the target keywords. Then, we use the Word2vec approach above to find its closest target keyword.
- **Fine-grained-only** is an simplistic version of our approach, which only uses the fine-grained multi-class

classifier, without the preceding coarse classifier.

**Results**: Table VII summarizes the euphemism identification results. There are 33, 9, and 12 categories for the drug, weapon and sexuality datasets respectively, resulting in a random guess performance for $Acc@1$ to be 0.03, 0.11, 0.08 (i.e. the inverse of the number of categories). Our algorithm achieves the best performance for all three datasets and has a large margin over the random guess performance.

Word2vec exhibits poor performance, in that it is unable to capture the nuanced differences between the target keywords by taking all sentences into consideration. Therefore, we construct two baselines (i.e., Clustering + word2vec and Binary + word2vec) to remove the noisy sentences and aid learning using a more homogeneous set of masked sentences. Empirically, we find that a binary classifier contributes more towards the performance, compared to the clustering algorithm. This is because, the result of clustering did not adequately cluster the sentences into a target keyword cluster and a non-target keyword cluster. Taking the drug dataset as an example, we found that owing to the widely varying contexts and vocabulary diversity of the dataset, the clustering results were inadequate. For instance, a qualitative examination of the results of clustering for a few euphemisms showed that the cluster separation sometimes occurred by the "quality" attribute (e.g., high quality vs. low quality drugs) or even sentiment (e.g., feeling high vs. feeling low). Therefore, $k$-means clustering fails as a filter for the non-drug-related masked sentences and does not lead to performance improvement. We leave exploring other clustering algorithms for future work. In contrast, the binary classifier, which can be taken as a directed $k$-means clustering algorithm, specifically filters out the non-drug-related sentences and is therefore a helpful addition. For such a specific task, the binary classifier performance can be taken as a performance upper bound for clustering algorithms.

We highlight two important findings: 1) By comparing the results of Word2vec and Fine-grained-only, we demonstrate the advantage of using a classification algorithm over an unsupervised word embedding-based method; 2) By comparing the differences between Word2vec and Binary + word2vec, and the differences between Fine-grained-only and our approach, we demonstrate the superior discriminative ability of a binary filtering classifier and therefore, highlight the benefit of using a coarse-to-fine-grained classification over performing only multi-class classification.

## VI. Discussion

Our algorithms rely on a relatively small number of hyper-parameters and choices of classification models. In this section, we demonstrate how to choose these hyper-parameters through detailed ablation studies, primarily on the drug dataset.

### A. Ablation Studies for Euphemism Identification

As discussed above, we adopt a coarse-to-fine-grained classification scheme for euphemism identification, relying on two classifiers used in cascade. We discuss here the performance of multiple classifiers on both coarse and fine-grained classification.

Table VII
Results on euphemism identification. Best results are in bold.

| | | $Acc@1$ | $Acc@2$ | $Acc@3$ |
|---|---|---|---|---|
| **Drug** | **Word2vec** | 0.07 | 0.14 | 0.21 |
| | **Clustering + word2vec** | 0.06 | 0.15 | 0.25 |
| | **Binary + word2vec** | 0.13 | 0.22 | 0.30 |
| | **Fine-grained-only** | 0.11 | 0.19 | 0.26 |
| | **Our Approach** | **0.20** | **0.31** | **0.38** |
| **Weapon** | **Word2vec** | 0.10 | 0.27 | 0.40 |
| | **Clustering + word2vec** | 0.11 | 0.25 | 0.37 |
| | **Binary + word2vec** | 0.22 | 0.43 | 0.57 |
| | **Fine-grained-only** | 0.25 | 0.40 | 0.61 |
| | **Our Approach** | **0.33** | **0.51** | **0.67** |
| **Sexuality** | **Word2vec** | 0.17 | 0.22 | 0.42 |
| | **Clustering + word2vec** | 0.15 | 0.30 | 0.49 |
| | **Binary + word2vec** | 0.21 | 0.39 | 0.59 |
| | **Fine-grained-only** | 0.19 | 0.40 | 0.51 |
| | **Our Approach** | **0.32** | **0.55** | **0.64** |

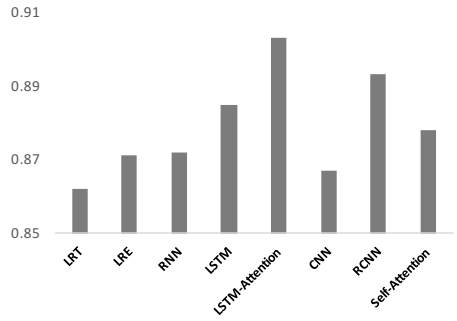


Figure 4. Testing accuracy for the coarse classifier.

*1) Coarse Classifiers:* In the euphemism identification framework, we use a binary classifier to filter out the sentences where euphemisms are used in non-euphemistic senses. We experiment with the binary classifiers shown below. Note that for all the neural models, we use 100-dimensional GloVe embeddings[12] [99] pre-trained on Wikipedia and tune the embeddings by the models.

- Logistic Regression [91] on raw Text (LRT): we first represent each word as a one-hot vector and then represent each sentence as the average of its member words' encodings.
- Logistic Regression on text Embeddings (LTE): we learn the word embeddings (100-dimensional) using word2vec [13], [14]. We represent each sentence by the average of its member words' embeddings.
- Recurrent Neural Network (RNN) [100]: we use a 1-layer bidirectional RNN with 256 hidden nodes.
- Long Short-Term Memory (LSTM) [89]: we use a 1-layer bidirectional LSTM with 256 hidden nodes.

[12]https://nlp.stanford.edu/projects/glove/

- LSTM-Attention: we add an attention mechanism [90] on LSTM.
- Convolutional Neural Networks (CNN) [101]: we train a simple CNN with one layer of convolution on top of word embeddings.
- Recurrent Convolutional Neural Networks (RCNN) [102]: we apply a bidirectional LSTM and employ a max-pooling layer across all sequences of texts.
- Self-Attention [103]: instead of using a vector, we use a 2-D matrix to represent the embedding, with each row of the matrix attending on a different part of the sentence.

We split the datasets into 70-10-20 for training, validation and testing. The model parameters are tuned on the validation data. Empirically, we find the LSTM-Attention performs the best across three datasets. This is why we ultimately selected it and reported results using it in Section V. Yet, as shown in Figure 4, other classifiers have satisfactory performance as well, and reach a testing accuracy ranging from 0.86 to 0.90.

*2) Fine-Grained Classifiers:* In the euphemism identification framework, we use a multi-class classifier to identify to which target keyword each euphemism refers. Again, we experimented with the same set of classifiers as above. Interestingly, we find that, for fine-grained classification, all classifiers have highly similar results. One possible reason is that each class has relatively small number of training instances (ranging from a few hundreds to 100k), which limits the discriminative power of advanced algorithms. For the drug dataset (33 target keywords), the training accuracy is about 55% and the testing accuracy is about 24%. This shows the feasibility of the task since the random guess accuracy would be 3.3%. Given the similar performance across classifiers, we recommend Logistic Regression on raw Text (LRT) for better computational efficiency.

For both coarse classifiers and fine-grained classifiers, we leave more advanced classification algorithms for future work.

### B. Parameter Analysis

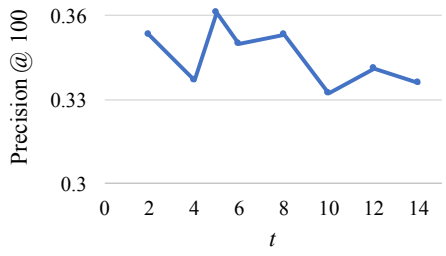


Figure 5. Sensitivity of $t$.

In the euphemism detection step (Section IV-A), we set a masked language model threshold $t$ to filter out the generic masked sentences. In the ranked list of replacements for the mask token, if any target keyword appears in the top-$t$ replacement candidates for the masked sentence, we consider the masked sentence a valid context instance. Otherwise, we consider the masked sentence generic and filter it out. Figure 5 shows how the results change with the threshold $t$ and we observe a slight decrease when the threshold $t$ is larger than 5. Therefore, $t = 5$ appears to be an optimal parameter choice.

### C. Limitations

While our approach for euphemism detection and identification appears highly promising, it does have some limitations.

**Text-only moderation**: Our approach only works with text, and our techniques are not easily generalizable to other media. Social media posts frequently include images, video, and audio, which can be even more challenging (and even more traumatic) to moderate by hand [3]–[5]. However, text is frequently associated with these other media, e.g., in the form of comments, and thus detecting euphemism use might indirectly provide clues to content moderators dealing with different media.

**Other contexts**: Our approach performs well on corpora discussing drugs, weapons, and sexuality. In preliminary experiments with a corpus of hate speech it did not perform nearly as well, producing many false matches when tasked with identifying racial slurs. We believe this is because euphemisms related to drugs, weapons, and sex typically have specific meanings; e.g., "pot" always refers to marijuana, not some other drugs. Racial slurs, on the other hand, are (in this corpus) used imprecisely, and interchangeably with generic swearwords, which seems to confuse euphemism detection. We do not know yet whether this is a fundamental limitation. Even if it is, though, there are many contexts where euphemisms have specific meanings and our approach should be effective, particularly forums selling illicit goods.

**Robustness to adversarial evasion**: In our evaluation, we have relied on *a priori* non-adversarial datasets, that were gleaned from public, online forums. In other words, people were using euphemisms, but we do not know whether they were using them specifically to evade content moderation. Perhaps these euphemisms are, for them, simply the ordinary names of certain things within the circle where they were discussing them. (Someone who consistently spoke of "marijuana" instead of "pot" on a forum dedicated to discussing drug experiences might well be suspected of being an undercover cop.)

Because our algorithms rely on sentence-level context to detect and identify euphemisms, an adversary would need to change that context to escape detection. Such changes may also render the text unintelligible to its intended audience. Therefore, we expect our techniques to be moderately resilient to adversarial evasion. However, we cannot test our expectations at the moment, since we do not have a dataset where people were purposely using euphemisms *only* to escape detection.

**Usability for content moderators**: While our approach shows encouraging performance in lab tests, we have not yet evaluated whether it is good enough to be helpful to content moderators in practice. That evaluation would require a user study of professional content moderators. This is out of scope for the

present paper, which focuses on the technical underpinnings of euphemism detection and identification. We are interested in investigating usability as a follow-up study.

As a preliminary experiment, we investigated the Perspective API[13], Google's automated toxicity detector to identify the likelihood of a sentence being considered toxic by a reader. Perspective is reportedly used today by human moderators to filter or prioritize comments that may require moderation. We take sentences from our datasets that contain the target keywords (e.g., "marijuana", "heroin") and for each such sentence, we evaluate the toxicity score of the sentence (a) with the target keyword, and (b) by replacing the target keyword with one of its identified euphemisms (e.g., "weed", "dope"). By comparing the toxicity scores, we can estimate the likelihood that a human moderator who is using Perspective API would be shown each version of the sentence. Table VIII shows the average toxicity scores when comparing 1000 randomly chosen original sentences with their euphemistic replacements for the drug, weapon, and sexuality categories. We observe that sentences with target keywords have higher (or at least comparable) toxicity scores compared to sentences with euphemisms, which suggests that euphemisms could help escape content moderation based on the Perspective API. In turn, detecting and identifying euphemisms could help defeat such evasive techniques.

Table VIII
AVERAGE TOXICITY SOCRES BY PERSPECTIVE API. (A): ORIGINAL SENTENCES; (B): SENTENCES WITH THEIR EUPHEMISTIC REPLACEMENTS.

| | Drug | Weapon | Sexuality |
|---|---|---|---|
| **A** | 0.209 | 0.235 | 0.612 |
| **B** | 0.178 | 0.232 | 0.522 |

### D. Ethics

This study relies extensively on user-generated content. We consider here the ethical implications of this work. The data we use in this paper were posted on publicly accessible websites, and do not contain any personal identifiable information (i.e., no real names, email addresses, IP addresses, etc.). Further, they are from 2018 or earlier, which greatly reduces any sensitive nature they might have. For instance, given their age and the absence of personal identifiable information, the data present very little utility in helping reduce imminent risks to people.

From a regulatory standpoint, in the context of earlier work on online anonymous marketplaces [43], [94], Carnegie Mellon University's Institutional Review Board (IRB) gave us very clear feedback on what is considered human research and thus subject to IRB review. Analyses relying on user-generated content do not constitute human-subject research, and are thus not the purview of the IRB, as long as 1) the data analyzed are posted on public fora and were not the result of direct interaction from the researchers with the people posting, 2) no private identifiers or personal identifiable information are associated with them, and 3) the research is not correlating different public sources of data to infer private data.[14] All of these conditions apply to the present study.

[13]https://www.perspectiveapi.com/

## VII. CONCLUSION

We have worked on the problem of content moderation by detecting and identifying euphemisms. By utilizing the contextual information explicitly, we not only obtain new state-of-the-art detection results, but also discover new euphemisms that are not even on the ground truth list. For euphemism identification, we, for the first time, prove the feasibility of the task and achieve it on a raw text corpus alone, without relying on any additional resources or supervision.

## REPRODUCIBILITY

Our code and pre-trained models are available on GitHub: https://github.com/WanzhengZhu/Euphemism.

## ACKNOWLEDGMENTS

We thank our shepherd, Ben Zhao, and the anonymous reviewers for comments on earlier drafts that significantly helped improve this manuscript; Kyle Soska for providing us with the Reddit data and Sadia Afroz for the weapons data; Xiaojing Liao and Haoran Lu for availing and discussing the Cantreader implementation with us; and Xin Huang for insightful discussions. This research was partially supported by the National Science Foundation, awards CNS-1720268 and CNS-1814817.

## REFERENCES

[1] D. Blackie, "AOL censors British town's name!" *Computer Underground Digest*, vol. 8, April 1996, as abstracted in RISKS Digest 18.07. [Online]. Available: http://catless.ncl.ac.uk/Risks/18.07.html#subj3

[2] P. M. Barrett, "Who moderates the social media giants?" *Center for Business*, 2020.

[3] C. Newton, "The terror queue," Dec. 2019, https://www.theverge.com/2019/12/16/21021005/google-youtube-moderators-ptsd-accenture-violent-disturbing-content-interviews-video.

[4] ——, "The trauma floor: The secret lives of Facebook moderators in America," Feb. 2019, https://www.theverge.com/2019/2/25/18229714/cognizant-facebook-content-moderator-interviews-trauma-working-conditions-arizona.

[5] Cambridge Consultants, "Use of AI in online content moderation," Ofcom Report, 2019, https://www.ofcom.org.uk/__data/assets/pdf_file/0028/157249/cambridge-consultants-ai-content-moderation.pdf.

[6] G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, R. Portnoff, S. Afroz, D. McCoy, K. Levchenko, and V. Paxson, "Identifying products in online cybercrime marketplaces: A dataset for fine-grained domain adaptation," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 2598–2607.

[7] R. S. Portnoff, S. Afroz, G. Durrett, J. K. Kummerfeld, T. Berg-Kirkpatrick, D. McCoy, K. Levchenko, and V. Paxson, "Tools for automated analysis of cybercriminal markets," in *Proceedings of International Conference on World Wide Web (WWW)*, 2017, pp. 657–666.

[8] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 1615–1625.

[14]This position is in line with Title 45 of the Code of Federal Regulations, Part 46 (45 CFR 46), which defines human research.

[9] K. Yuan, H. Lu, X. Liao, and X. Wang, "Reading thieves' cant: automatically identifying and understanding dark jargons from cybercrime marketplaces," in *Proceedings of 27th USENIX Security Symposium*, 2018, pp. 1027–1041.

[10] T. Hada, Y. Sei, Y. Tahara, and A. Ohsuga, "Codewords detection in microblogs focusing on differences in word use between two corpora," in *Proceedings of International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2020, pp. 103–108.

[11] R. Magu and J. Luo, "Determining code words in euphemistic hate speech using word embedding networks," in *Proceedings of the 2nd Workshop on Abusive Language Online (ALW2)*, 2018, pp. 93–100.

[12] K. Zhao, Y. Zhang, C. Xing, W. Li, and H. Chen, "Chinese underground market jargon analysis based on unsupervised learning," in *Proceedings of IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE, 2016, pp. 97–102.

[13] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2013, pp. 3111–3119.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019, pp. 4171–4186.

[16] C. Donahue, M. Lee, and P. Liang, "Enabling language models to fill in the blanks," in *Proceedings of Association for Computational Linguistics (ACL)*, 2020.

[17] C. Felt and E. Riloff, "Recognizing euphemisms and dysphemisms using sentiment analysis," in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 136–145.

[18] N. Leontiadis, T. Moore, and N. Christin, "Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade." in *USENIX Security Symposium*, vol. 11, 2011.

[19] D. McCoy, A. Pitsillidis, J. Grant, N. Weaver, C. Kreibich, B. Krebs, G. Voelker, S. Savage, and K. Levchenko, "Pharmaleaks: Understanding the business of online pharmaceutical affiliate programs," in *Part of the 21st USENIX Security Symposium*, 2012, pp. 1–16.

[20] J. Huang, Z. Li, X. Xiao, Z. Wu, K. Lu, X. Zhang, and G. Jiang, "SUPOR: Precise and scalable sensitive user input detection for android apps," in *24th USENIX Security Symposium*, 2015, pp. 977–992.

[21] Y. Nan, M. Yang, Z. Yang, S. Zhou, G. Gu, and X. Wang, "Uipicker: User-input privacy identification in mobile applications," in *Proceedings of 24th USENIX Security Symposium*, 2015, pp. 993–1008.

[22] K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "Trafficking fraudulent accounts: The role of the underground market in twitter spam and abuse," in *22nd USENIX Security Symposium*, 2013, pp. 195–210.

[23] S. Sedhai and A. Sun, "Semi-supervised spam detection in twitter stream," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 169–175, 2017.

[24] T. Wu, S. Wen, Y. Xiang, and W. Zhou, "Twitter spam detection: Survey of new approaches and comparative study," *Computers & Security*, vol. 76, pp. 265–284, 2018.

[25] T. Wu, S. Liu, J. Zhang, and Y. Xiang, "Twitter spam detection based on deep learning," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2017, pp. 1–8.

[26] A. Keith and K. Burridge, "Euphemism and dysphemism: language used as shield and weapon," 1991.

[27] K. L. Pfaff, R. W. Gibbs, and M. D. Johnson, "Metaphor in using and understanding euphemism and dysphemism," *Applied Psycholinguistics*, vol. 18, no. 1, pp. 59–83, 1997.

[28] R. Hugh, "Rawson's dictionary of euphemisms and other doubletalk," 2002.

[29] K. Allan, "The connotations of english colour terms: Colour-based x-phemisms," *Journal of Pragmatics*, vol. 41, no. 3, pp. 626–637, 2009.

[30] H. A. Rababah, "The translatability and use of x-phemism expressions (x-phemization): Euphemisms, dysphemisms and orthophemisms) in the medical discourse," *Studies in Literature and Language*, vol. 9, no. 3, pp. 229–240, 2014.

[31] R. A. Spears, *Slang and euphemism*. Signet Book, 1981.

[32] P. Chilton, "Metaphor, euphemism and the militarization of language," *Current Research on Peace and Violence*, vol. 10, no. 1, pp. 7–19, 1987.

[33] H. Ahl, "Motivation in adult education: a problem solver or a euphemism for direction and control?" *International Journal of Lifelong Education*, vol. 25, no. 4, pp. 385–405, 2006.

[34] E. C. Fernández, "The language of death: Euphemism and conceptual metaphorization in victorian obituaries," *SKY Journal of Linguistics*, vol. 19, no. 2006, pp. 101–130, 2006.

[35] Z. Pei, Z. Sun, and Y. Xu, "Slang detection and identification," in *Proceedings of Computational Natural Language Learning (CoNLL)*, 2019, pp. 881–889.

[36] Z. Huang, W. Xu, and K. Yu, "Bidirectional lstm-crf models for sequence tagging," *arXiv preprint arXiv:1508.01991*, 2015.

[37] J. D. Lafferty, A. McCallum, and F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proceedings of International Conference on Machine Learning (ICML)*, 2001, pp. 282–289.

[38] T. Rauber and K. Berns, "Kernel multilayer perceptron," in *Proceedings of SIBGRAPI Conference on Graphics, Patterns and Images*. IEEE, 2011, pp. 337–343.

[39] H. Yang, X. Ma, K. Du, Z. Li, H. Duan, X. Su, G. Liu, Z. Geng, and J. Wu, "How to learn klingon without a dictionary: Detection and measurement of black keywords used by the underground economy," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 751–769.

[40] J. Taylor, M. Peignon, and Y.-S. Chen, "Surfacing contextual hate speech words within social media," *arXiv preprint arXiv:1711.10093*, 2017.

[41] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," *Transactions of the Association for Computational Linguistics (TACL)*, vol. 5, pp. 135–146, 2017.

[42] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *Proceedings of Association for Computational Linguistics (ACL)*, 2014, pp. 302–308.

[43] N. Christin, "Traveling the Silk Road: A measurement analysis of a large anonymous online marketplace," in *Proceedings of International Conference on World Wide Web (WWW)*, 2013, pp. 213–224.

[44] J. Shen, Z. Wu, D. Lei, J. Shang, X. Ren, and J. Han, "Setexpan: Corpus-based set expansion via context feature selection and rank ensemble," in *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*. Springer, 2017, pp. 288–304.

[45] W. Zhu, H. Gong, J. Shen, C. Zhang, J. Shang, S. Bhat, and J. Han, "FUSE: Multi-faceted set expansion by coherent clustering of skip-grams," in *Proceedings of The European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD)*, 2020.

[46] Y. Zhang, J. Shen, J. Shang, and J. Han, "Empower entity set expansion via language model probing," in *Proceedings of Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020.

[47] J. Huang, Y. Xie, Y. Meng, J. Shen, Y. Zhang, and J. Han, "Guiding corpus-based set expansion by auxiliary sets generation and co-expansion," in *Proceedings of The Web Conference*, 2020, pp. 2188–2198.

[48] J. Shen, W. Qiu, J. Shang, M. Vanni, X. Ren, and J. Han, "Synsetexpan: An iterative framework for joint entity set expansion and synonym discovery," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 8292–8307.

[49] X. Rong, Z. Chen, Q. Mei, and E. Adar, "Egoset: Exploiting word ego-networks and user-generated ontology for multifaceted set expansion," in *Proceedings of Web Search and Data Mining (WSDM)*, 2016, pp. 645–654.

[50] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, vol. 2016. NIH Public Access, 2016, p. 595.

[51] C. Yang, J. Zhang, and J. Han, "Co-embedding network nodes and hierarchical labels with taxonomy based generative adversarial networks," in *Proceedings of IEEE International Conference on Data Mining (ICDM)*, 2020.

[52] J. Huang, Y. Xie, Y. Meng, Y. Zhang, and J. Han, "Corel: Seed-guided topical taxonomy construction by concept learning and relation transferring," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2020, pp. 1928–1936.

[53] Y. Mao, T. Zhao, A. Kan, C. Zhang, X. L. Dong, C. Faloutsos, and J. Han, "Octet: Online catalog taxonomy enrichment with self-supervision," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2020, pp. 2247–2257.

[54] J. Shang, X. Zhang, L. Liu, S. Li, and J. Han, "Nettaxo: Automated topic taxonomy construction from text-rich network," in *Proceedings of The Web Conference*, 2020, pp. 1908–1919.

[55] J. Shen, Z. Shen, C. Xiong, C. Wang, K. Wang, and J. Han, "TaxoExpan: Self-supervised taxonomy expansion with position-enhanced graph neural network," in *Proceedings of The Web Conference*, 2020, pp. 486–497.

[56] C. Zhang, F. Tao, X. Chen, J. Shen, M. Jiang, B. Sadler, M. Vanni, and J. Han, "Taxogen: Constructing topical concept taxonomy by adaptive term embedding and clustering," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*, 2018.

[57] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.

[58] ——, "Technique for analyzing overlapping memberships," *Sociological Methodology*, vol. 4, pp. 176–185, 1972.

[59] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web." Stanford InfoLab, Tech. Rep., 1999.

[60] S. Ishiwatari, H. Hayashi, N. Yoshinaga, G. Neubig, S. Sato, M. Toyoda, and M. Kitsuregawa, "Learning to describe unknown phrases with local and global contexts," in *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019, pp. 3467–3476.

[61] K. Ni and W. Y. Wang, "Learning to explain non-standard english words and phrases," in *Proceedings of International Joint Conference on Natural Language Processing (IJCNLP)*, 2017, pp. 413–417.

[62] K. Taghipour and H. T. Ng, "Semi-supervised word sense disambiguation using word embeddings in general and specific domains," in *Proceedings of North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2015, pp. 314–323.

[63] A. Raganato, C. D. Bovi, and R. Navigli, "Neural sequence learning models for word sense disambiguation," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2017, pp. 1156–1167.

[64] A. Raganato, J. Camacho-Collados, and R. Navigli, "Word sense disambiguation: A unified evaluation framework and empirical comparison," in *Proceedings of the European Chapter of the Association for Computational Linguistics (EACL)*, vol. 1, 2017, pp. 99–110.

[65] I. Iacobacci, M. T. Pilehvar, and R. Navigli, "Embeddings for word sense disambiguation: An evaluation study," in *Proceedings of Association for Computational Linguistics (ACL)*, vol. 1, 2016, pp. 897–907.

[66] L. Weng, "Self-supervised representation learning," *lilianweng.github.io/lil-log*, 2019. [Online]. Available: https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html

[67] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "ALBERT: A lite bert for self-supervised learning of language representations," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2019.

[68] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[69] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[70] T. Chen, S. Kornblith, K. Swersky, M. Norouzi, and G. E. Hinton, "Big self-supervised models are strong semi-supervised learners," in *Proceedings of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[71] L. Liu, J. Shang, X. Ren, F. F. Xu, H. Gui, J. Peng, and J. Han, "Empower sequence labeling with task-aware neural language model," in *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2018, pp. 5253–5260.

[72] Z. Feng, C. Xu, and D. Tao, "Self-supervised representation learning by rotation feature decoupling," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 10 364–10 374.

[73] A. Kolesnikov, X. Zhai, and L. Beyer, "Revisiting self-supervised visual representation learning," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1920–1929.

[74] M. Sabokrou, M. Khalooei, and E. Adeli, "Self-supervised representation learning via neighborhood-relational encoding," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 8010–8019.

[75] W. Zhu, C. Zhang, S. Yao, X. Gao, and J. Han, "A spherical hidden markov model for semantics-rich human mobility modeling," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[76] O. Mees, M. Tatarchenko, T. Brox, and W. Burgard, "Self-supervised 3d shape and viewpoint estimation from single images for robotics," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6083–6089.

[77] A. Nair, D. Chen, P. Agrawal, P. Isola, P. Abbeel, J. Malik, and S. Levine, "Combining self-supervised learning and imitation for vision-based rope manipulation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2146–2153.

[78] L. Berscheid, P. Meißner, and T. Kröger, "Self-supervised learning for precise pick-and-place without object model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4828–4835, 2020.

[79] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, "S4l: Self-supervised semi-supervised learning," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019, pp. 1476–1485.

[80] Y. Sun, E. Tzeng, T. Darrell, and A. A. Efros, "Unsupervised domain adaptation through self-supervision," *arXiv preprint arXiv:1909.11825*, 2019.

[81] J. Xu, L. Xiao, and A. M. López, "Self-supervised domain adaptation for computer vision tasks," *IEEE Access*, vol. 7, pp. 156 694–156 706, 2019.

[82] H. Yin, P. Molchanov, J. M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N. K. Jha, and J. Kautz, "Dreaming to distill: Data-free knowledge transfer via deepinversion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 8715–8724.

[83] G. Kahn, A. Villaflor, B. Ding, P. Abbeel, and S. Levine, "Self-supervised deep reinforcement learning with generalized computation graphs for robot navigation," in *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–8.

[84] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4238–4245.

[85] V. H. Pong, M. Dalal, S. Lin, A. Nair, S. Bahl, and S. Levine, "Skew-fit: State-covering self-supervised reinforcement learning," *arXiv preprint arXiv:1903.03698*, 2019.

[86] Y. Huo, Y. Lu, Y. Niu, Z. Lu, and J.-R. Wen, "Coarse-to-fine grained classification," in *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2019, pp. 1033–1036.

[87] W. Liu, C. Zhang, J. Zhang, and Z. Wu, "Global for coarse and part for fine: A hierarchical action recognition framework," in *Proceedings of 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 2630–2634.

[88] Z. Li, Y. Wei, Y. Zhang, X. Zhang, and X. Li, "Exploiting coarse-to-fine task transfer for aspect-level sentiment classification," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, vol. 33, 2019, pp. 4253–4260.

[89] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[90] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[91] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[92] F. Hoffa, "1.7 billion reddit comments loaded on BigQuery," 2016, https://www.reddit.com/r/bigquery/comments/3cej2b/17_billion_reddit_comments_loaded_on_bigquery/. As of 2021, the dataset proper is available at https://console.cloud.google.com/bigquery?project=fh-bigquery.

[93] C. Cimpanu, "Reddit bans community dedicated to dark web markets," Mar. 2018, https://www.bleepingcomputer.com/news/security/reddit-bans-community-dedicated-to-dark-web-markets/.

[94] K. Soska and N. Christin, "Measuring the longitudinal evolution of the online anonymous marketplace ecosystem," in *Proceedings of the 24th USENIX Security Symposium*, Washington, DC, Aug. 2015, pp. 33–48.

[95] Drug Enforcement Administration, "Slang terms and code words: A reference for law enforcement personnel," *DEA Intelligence Report DEA-HOU-DIR-022-18*, 2018, available at https://www.dea.gov/sites/default/files/2018-07/DIR-022-18.pdf.

[96] S. Zannettou, B. Bradlyn, E. De Cristofaro, H. Kwak, M. Sirivianos, G. Stringini, and J. Blackburn, "What is gab: A bastion of free speech or an alt-right echo chamber," in *Companion Proceedings of The Web Conference*, 2018, pp. 1007–1014.

[97] C. D. Manning, H. Schütze, and P. Raghavan, *Introduction to information retrieval*. Cambridge university press, 2008.

[98] K. Järvelin and J. Kekäläinen, "Ir evaluation methods for retrieving highly relevant documents," in *ACM SIGIR Forum*, vol. 51, no. 2. ACM New York, NY, USA, 2017, pp. 243–250.

[99] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[100] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," California Univ San Diego La Jolla Inst for Cognitive Science, Tech. Rep., 1985.

[101] Y. Kim, "Convolutional neural networks for sentence classification," in *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[102] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[103] Z. Lin, M. Feng, C. N. d. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, "A structured self-attentive sentence embedding," *arXiv preprint arXiv:1703.03130*, 2017.

## APPENDIX

We present the euphemism detection results by our approach in Table IX and analyze the false positive detection results on the drug dataset in Table X. We categorize our false detection results into four types:

- They are correct euphemisms but missed on the ground truth list (cases 1-5 in Table X).
- They are not euphemisms by themselves, but they are contained in euphemism phrases. For example, as shown in case 6 in Table X, "oil" is not a drug euphemism while "cbd oil" is one.
- Though they are not euphemisms, they are strongly related to drug or the usage of drug (cases 7-10 in Table X). Cases 7 and 8 uncovers some ways that people take drugs (together with alcohol or cigarettes).
- Incorrect detection.

The case studies reveal that we can even find some correct euphemisms that are not on the ground truth list, which suggests the rapid-evolving nature of euphemisms and the necessity of the automatic euphemism detection task.

Table IX

EUPHEMISM DETECTION RESULTS BY OUR APPROACH (BETTER VIEWED IN COLOR). PURPLE BOLD WORDS ARE CORRECTLY DETECTED EUPHEMISMS AND ON THE GROUND TRUTH LIST (I.E., THE DEA LIST). THE PURPLE UNDERLINED WORDS INDICATE THAT THEY ARE INCORRECT BY THEMSELVES, BUT ARE CONTAINED IN TRUE EUPHEMISM PHRASES, SUCH AS "DOG FOOD", "CHINESE TOBACCO" (EUPHEMISMS FOR "HEROIN" AND "OPIUM" RESPECTIVELY). THOSE WORDS WHICH DO NOT APPEAR IN THE GROUND TRUTH LIST ARE MARKED BLACK.

| Dataset | Target Keywords | Euphemism Candidates |
|---|---|---|
| Drug | {heroin, ecstasy, marijuana, cocaine, opium, …} | cannabis, **weed**, **coke**, alcohol, crack, **speed**, **acid**, **pot**, mushrooms, md, pills, **hash**, **h**, **powder**, tobacco, **crystal**, something, cigarettes, pressed, l, **k**, **x**, met, recovering, **lean**, **spice**, **bud**, narcotics, product, oil, grade, **e**, **shatter**, **blow**, anything, prescription, pill, research, heroine, shit, **gold**, use, psychedelic, **hydro**, **white**, medical, **water**, **stuff**, card, **wax**, substances, **benz**, products, fatal, fucking, addiction, sh, orange, new, **coffee**, sample, **bars**, others, sex, rc, smoking, **lucy**, **blue**, daily, money, pain, education, substance, **coca**, care, magnesium, **tar**, guns, everything, quality, treatment, **peruvian**, 2, legal, **pure**, mx, ir, synthetic, **herb**, **amp**, **green**, 4, medicine, chemicals, red, sleeping, possession, **extract**, depression, lithium |
| Weapon | {carbine, gatling, gun, rifle, pistol, …} | **cannon**, **bullet**, finger, **burner**, phone, **gat**, ball, one, hand, weapons, **shot**, **guns**, **trigger**, needle, **car**, **heater**, fuck, bucket, fence, handgun, bat, chance, trap, pipe, **sword**, bag, door, **hammer**, camera, bar, blade, **piece**, cigarette, barrel, hit, **point**, dog, **flashlight**, horse, house, **revolver**, fan, bomb, balloon, pill, brick, **stick**, lighter, key, fire, joint, **pack**, grenade, line, flag, blunt, **box** |
| Sexuality | {breast, genitals, sex, pornography, nipple, …} | **dick**, **head**, **brain**, **cock**, **face**, hair, body, **balls**, **ass**, man, heart, **heads**, hands, white, family, hand, mouth, woman, children, life, child, name, baby, **finger**, wife, gun, neck, mind, nose, **skin**, shit, teeth, blood, money, **sex**, **fingers**, **blow**, bodies, leg, one, private, legs, black, back, race, knife, soul, yes, **brains**, people, lives, son, daughter, **throat**, foot, red, feet, breasts, house, personality, **tongue**, country, **bang**, women, core, mother, job, point, **suck** |

Table X
CASE STUDIES OF THE FALSE POSITIVE DETECTION RESULTS ON THE DRUG DATASET. THEY ARE REAL EXAMPLES FROM REDDIT.

| ID | Euphemism Candidates | Sentences Associated |
|---|---|---|
| 1 | cannabis | • hash is the most popular cannabis product that circulates the streets here<br>• symptoms stop after cessation of cannabis use<br>• im 18 and use lsd and cannabis often<br>• smoking cannabis generates a large amount of unwanted side products of which carcinogenic compounds are the most dangerous |
| 2 | mushrooms | • his main products included amphetamines ecstasy mushrooms and crystal meth<br>• vendor review tripwithscience vial liquid mushrooms ~9mg of psilocybin<br>• lsd mdma mushrooms especially ketamine and cocaine<br>• im having trouble deciding whether to order 100 tabs of acid or 50 tabs and a half ounce of mushrooms |
| 3 | md | • can easily smash through 15 in a night id be a bit fucked up but the md high isnt as good anymore<br>• the crystals do have a md smell though but just a bit overpowered by the weed smell<br>• i mixed in 7g of red md with 3tsp of lemon juice and a dash of water and freezed it over night<br>• so i took md last night not for the first time and was thinking about how music sounds so much better when youre high |
| 4 | l | • price value was good paid around aps26 the only person i know that sells l around here charges aps10 per 100ug digusting<br>• really great for smoking at night i rolled a 2g l and it put me to sleep lmao<br>• sparked an l of just blueberry kush and was so high i reanalyzed this entire review like 3 times while watching tv |
| 5 | met | • so i have 500mg of 4acodmt and 250mg of met in my cart on lysergi<br>• 25inboh + ho met = lots of hallucinations pretty clean mindset<br>• i preferred 4 ho met to 4 aco dmt since aco made me anxious when i snorted it |
| 6 | oil | • this hemp cbd oil gets me and my friends super high for about half an hour<br>• my nephew had some cbd oil he bought at his dispensary and he gave me what he considered a large dose<br>• tldr tried cbd oil in a headshop got really stoned and passed out<br>• i have some excellent 70 kava honey oil extract |
| 7 | alcohol | • does anyone have successful experience making an alcohol alprazolam solution<br>• its an interesting chemical and experience thats so completely different than alcohol or cannabis only two drugs ive used so far in all aspects<br>• i left some speed dissolved in alcohol in a glass overnight<br>• can i buy alcohol on the dark net<br>• alcohol and xanax no effect |
| 8 | cigarettes | • i am not a regular smoker but the last few times i have done shrooms a cigarette would make the trip even more amazing<br>• i was thinking ive heard of people dipping cigarettes in pcp to make it smokeabole could the same done with mxe<br>• cigarettes felt amazing i felt so much love it was defiantly one of the best xtc pills i have taken |
| 9 | pressed | • im primarily looking at their pressed pills you can grab 50 from them and it would cost you less than $250<br>• domestic usa pressed mdma vendors<br>• 5 x 3mg xanax gg249 price 1499 including shipping product received 5 x gg249 3mg pressed replicas xanax bars all 5 intact no broken bars |
| 10 | recovering | • i have been completely abstinent from all drugs besides lsd for over 18 months and i am a recovering addict<br>• im a recovering heroin addict so i know what that looks like at least for me the nod<br>• are most kratom users regular users or recovering opiate addicts |