EUREKA: EUphemism Recognition Enhanced Through KNN-Based Methods and Augmentation

EMNLP 2022 Figurative Language Processing Workshop Euphemism Detection Shared Task

Sedrick Scott Keh¹*, Rohit K. Bharadwaj²*, Emmy Liu¹**, Simone Tedeschi³,⁴**, Varun Gangal¹, Roberto Navigli³

¹Carnegie Mellon University, ²Mohamed bin Zayed University of Artificial Intelligence, ³Sapienza University of Rome, ⁴Babelscape, Italy
We address the euphemism detection task along both the **data** side and the **modeling** side.

1. **Data Cleaning**
2. **Data Augmentation**
3. **Using PET Representations**
4. **KNN Augmentation and Deep Averaging Network (DAN)**

**Shared task performance:** F1 score **0.881** on public leaderboard (1st place) 🏆

How do we best incorporate the surrounding context of the Potentially Euphemistic Terms (PETs)?
1. Data Cleaning

We felt that some sentences in the dataset were incorrectly labeled. How to best detect mislabeled sentences?

<table>
<thead>
<tr>
<th>Sentence Containing PET</th>
<th>Sense (Euph.)</th>
<th>Sense (Non-Euph.)</th>
<th>Label (Original)</th>
<th>Label (Corrected)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does your software collect any information about me, my listening or my surfing habits? Can it be &lt;disabled&gt;?</td>
<td>Handicapped</td>
<td>Switched off</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Europe developed rapidly [...] Effective and &lt;economical&gt; movement of goods was no longer a maritime monopoly.</td>
<td>Prudent or frugal</td>
<td>Related to the economy</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>The Lancers continued to hang on to the &lt;slim&gt; one-point line as Golden West started a possession following [...]</td>
<td>Thin (physical appearance)</td>
<td>Thin (non-physical)</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Examples of incorrectly labelled sentences identified by our data cleaning pipeline. The label is 1 if the term is used euphemistically, 0 otherwise.
1. Data Cleaning

We manually curate a sense inventory (euphemistic vs. non-euphemistic senses) using context clues and BabelNet definitions.

We identify 203 potentially mislabelled sentences, then manually check through these and identify 25 incorrectly labeled instances.
2. Data Augmentation

The original corpus contains 1571 sentences.

We expand this corpus using by taking sentences from a larger corpus (i.e. WikiText), using two data augmentation strategies:

a) Representation-based augmentation (~4700 additional rows)
b) Sense-based augmentation (~950 additional rows)
2. Data Augmentation (**Representation-Based**)

Given PET \( p \), find new sentences \( \{s_1, s_2, \ldots s_k\} \) in WikiText containing \( p \)

Add \( s_i \) to our corpus if:

a) It’s “sufficiently similar” to all sentence containing \( p \) in our training corpus (add with same label)

or

b) It’s “sufficiently different” from all sentence containing \( p \) in our training corpus (add with opposite label)

To measure distance, use cosine distance of **sentence embeddings**
2. Data Augmentation (Sense-Based)

Instead of finding sentences \(\{s_1, s_2, \ldots s_k\}\) containing \(p\), we instead find sentences containing senses of \(p\).

E.g. Instead of searching for appearances of “disabled”:
- Search for appearances of “handicapped” → assign positive label
- Search for appearances of “switched off” → assign negative label

Use senses from previously defined sense inventory.
3. Using PET Representations

- Instead of passing the [CLS] token embeddings to the final classifier, we instead pass the token embeddings of the Potentially Euphemistic Terms (PETs)
- If there are multiple tokens in a PET, we add the token embeddings
4. kNN Augmentation and Deep Averaging Network (DAN)

The goal of these methods is to further make use of the surrounding context

a) kNN Augmentation
   - Use kNN store of the training set
   - Interpolate the classification probabilities of the base model and a kNN-based model

b) Deep Averaging Network (DAN)
   - Take the mean vector for the entire sequence and pass it through a linear layer
Data Ensembling

We take a majority vote of 3 of our top-performing models.

**Figure 2:** Models and datasets used in the ensemble.
### Results and Discussion

1. **Data augmentations** lead to slight increase in performance
2. Using **embeddings of the PET tokens** (instead of the [CLS] classifier token) significantly increases performance
3. **KNN models** lead to slight increase, while **DAN models** lead to significant decrease, in performance.
Thank you for listening!


GitHub: https://github.com/sedrickkeh/EUREKA

For emails and questions, please send to sedrickkeh@gmail.com