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

EMNLP 2022 Figurative Language Processing Workshop
Euphemism Detection Shared Task

Sedrick Scott Keh[1*], Rohit K. Bharadwaj[2*], Emmy Liu[1**], Simone Tedeschi[3,4**], Varun Gangal[1], Roberto Navigli[3]

[1]Carnegie Mellon University, [2]Mohamed bin Zayed University of Artificial Intelligence, [3]Sapienza University of Rome, [4]Babelscape, Italy

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)?

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)

1. Data Cleaning

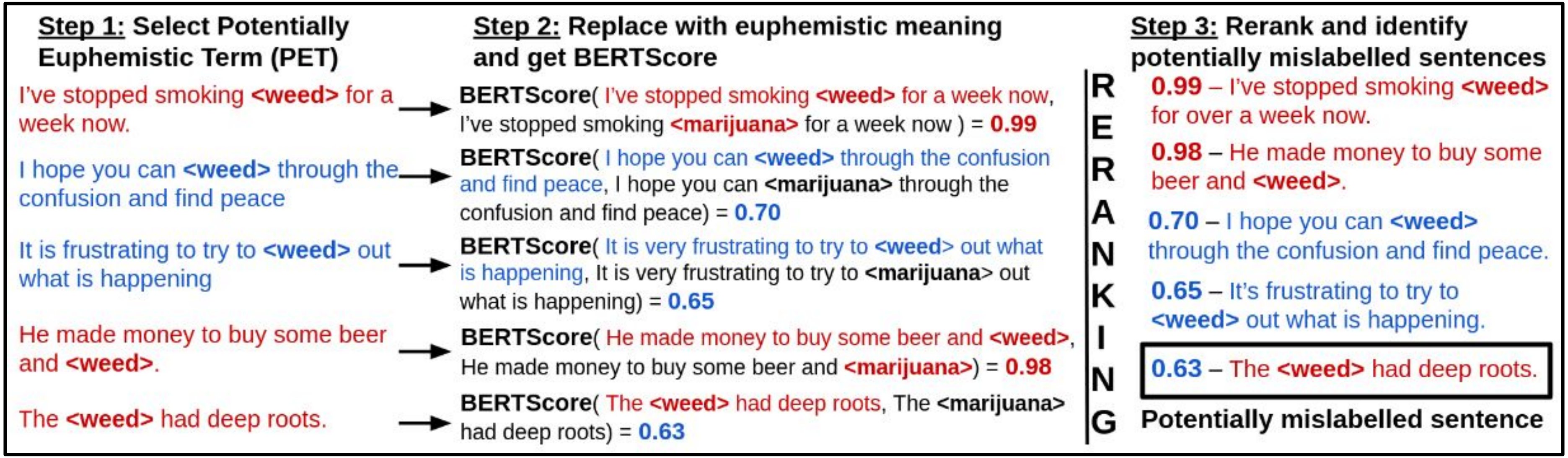
Sentence Containing PET	Sense (Euph.)	Sense (Non-Euph.)	Label (Original)	Label (Corrected)
Does your software collect any information about me, my listening or my surfing habits? Can it be <disabled>?	Handicapped	Switched off	1	0
Europe developed rapidly [...] Effective and <economical> movement of goods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

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.

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

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 $\{s_1, s_2, \dots s_k\}$ 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

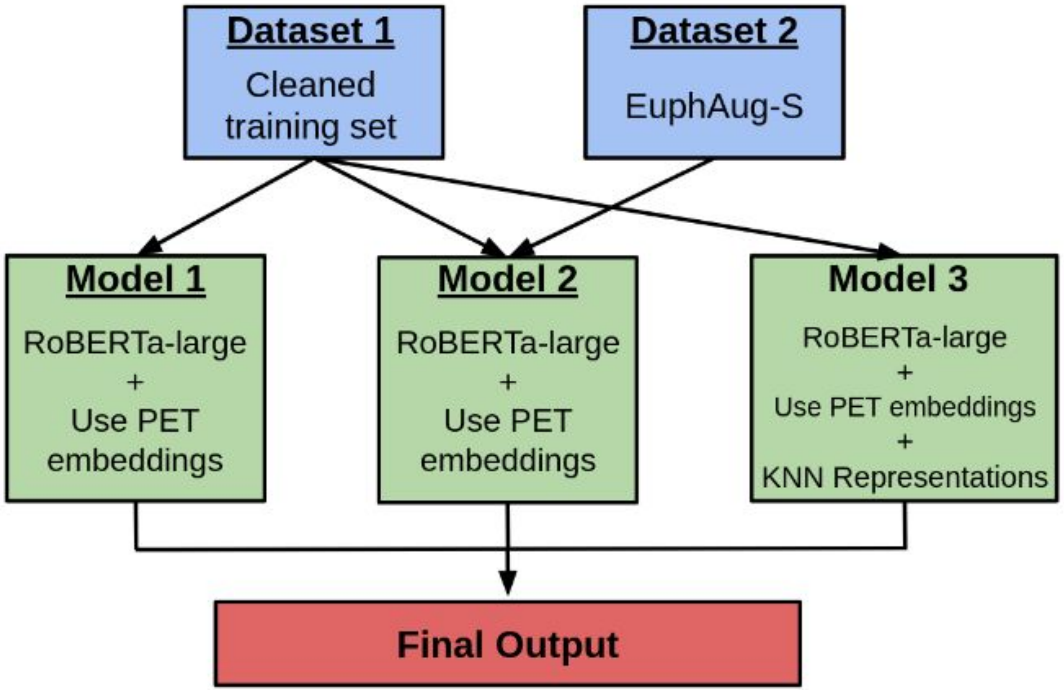


Figure 2: Models and datasets used in the ensemble.

Feature Tested	Model	Dataset	P	R	F1
-	RoBERTa-large	Original	0.8756	0.8168	0.8399
1) Data Cleaning	RoBERTa-large	Cleaned	0.8617	0.8300	0.8435
2) Data Augmentation	RoBERTa-large	Original+*EuphAug-R*	0.8529	0.8388	0.8452
	RoBERTa-large	Original+*EuphAug-S*	0.8728	0.8306	0.8481
3) PET Embedding	RoBERTa-large+PET	Original	0.8694	0.8408	0.8533
4) Additional Context	RoBERTa-large+KNN	Original	0.8769	0.8210	0.8411
	RoBERTa-large+DAN	Original	0.8481	0.7983	0.8181
Final Models	RoBERTa-large+PET	Cleaned	0.8728	0.8471	0.8582
	RoBERTa-large+PET	Cleaned+*EuphAug-S*	0.8692	0.8584	0.8633
	RoBERTa-large+PET+KNN	Cleaned	0.8792	0.8517	0.8635
Final Ensemble	Model 1 + Model 2 + Model 3	-	**0.8994**	**0.8788**	**0.8884**

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!

arXiv: https://arxiv.org/abs/2210.12846

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

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