Special Project HW3
Few-shot Emotion Classification

TA: 陳宣叡
r10942105@ntu.edu.tw

Task Introduction

Emotion classification

Classify emotion for the input text.

Input text	Emotion
I’m not even sure what it is, why do people hate it.	confusion
This caught me off guard for real. I’m actually off my bed laughing.	surprise, amusement
I tried to send this to a friend but [NAME] knocked it away.	disappointment

Task Introduction

Few-shot setup

Use GoEmotions as the source domain and DailyDialog as the target domain. Here are emotion tags in both dataset.

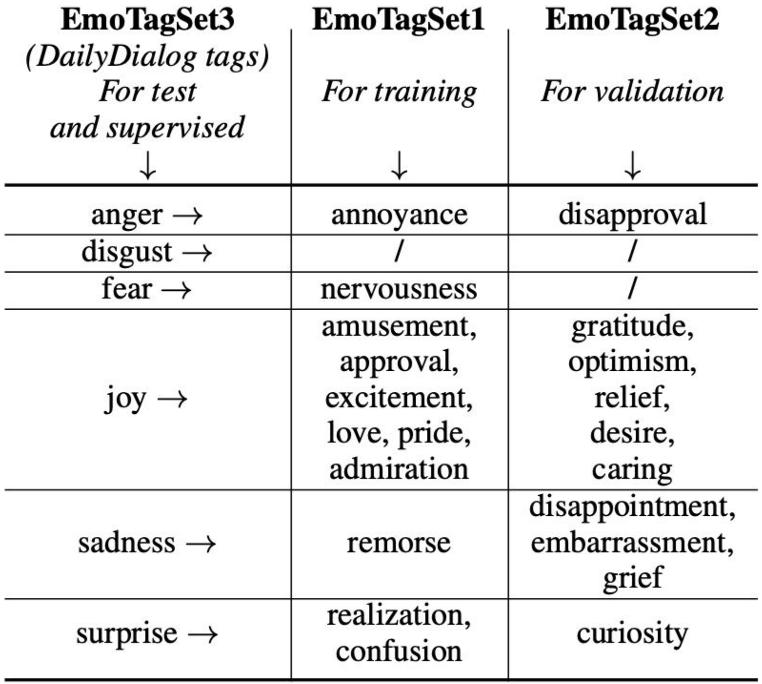
GoEmotions (source)	DailyDialog (target)
admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise	anger, disgust, fear, joy, sadness, surprise

Task Introduction

Few-shot setup

Emotion tags are split for training, validation, and testing.

The horizontal mapping is **only used in supervised scenarios** (for comparison with meta).



EmoTagSet3 *(DailyDialog tags)* *For test* *and supervised* ↓	**EmoTagSet1** *For training* ↓	**EmoTagSet2** *For validation* ↓
anger →	annoyance	disapproval
disgust →	/	/
fear →	nervousness	/
joy →	amusement, approval, excitement, love, pride, admiration	gratitude, optimism, relief, desire, caring
sadness →	remorse	disappointment, embarrassment, grief
surprise →	realization, confusion	curiosity

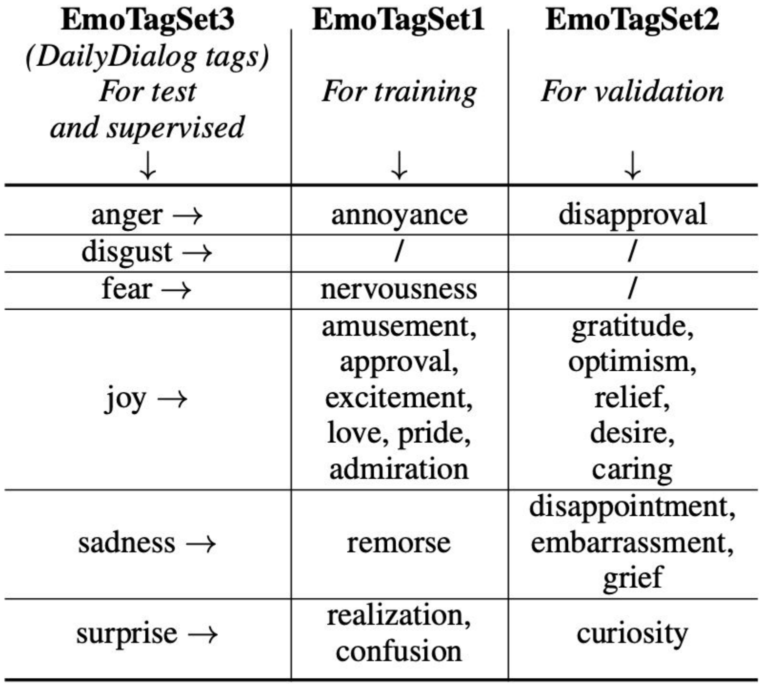
Task Introduction

Few-shot setup

An episode is defined as a 5-shot 6-way 30-query learning task.

For training and validation, an epoch is consists of 100 random episodes.

For testing, randomly sample 1000 episodes.



EmoTagSet3 *(DailyDialog tags)* *For test* *and supervised* ↓	**EmoTagSet1** *For training* ↓	**EmoTagSet2** *For validation* ↓
anger →	annoyance	disapproval
disgust →	/	/
fear →	nervousness	/
joy →	amusement, approval, excitement, love, pride, admiration	gratitude, optimism, relief, desire, caring
sadness →	remorse	disappointment, embarrassment, grief
surprise →	realization, confusion	curiosity

Methodology

Sentence representation

- **From word representation (FastText)**
  - Average word representation
  - CNN for sequence representation
  - Transformer encoder layer
- **From a pre-trained BERT family**
  - BERT
  - roBERTa
  - ...

Methodology

Meta-learning

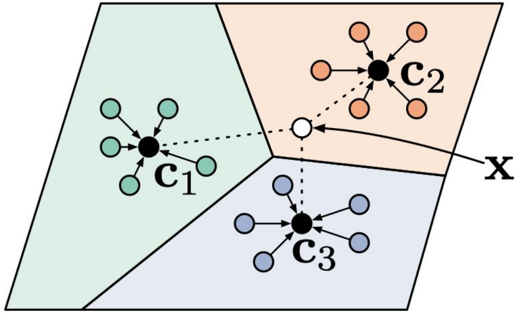
- **Optimization-based**
  - MAML
  - First-Order MAML
  - Reptile
- **Metric learning**
  - Prototypical Networks
  - Matching Networks
  - Relation Networks

The code provided implements prototypical networks, however, you are encouraged to try different meta-learning algorithms.

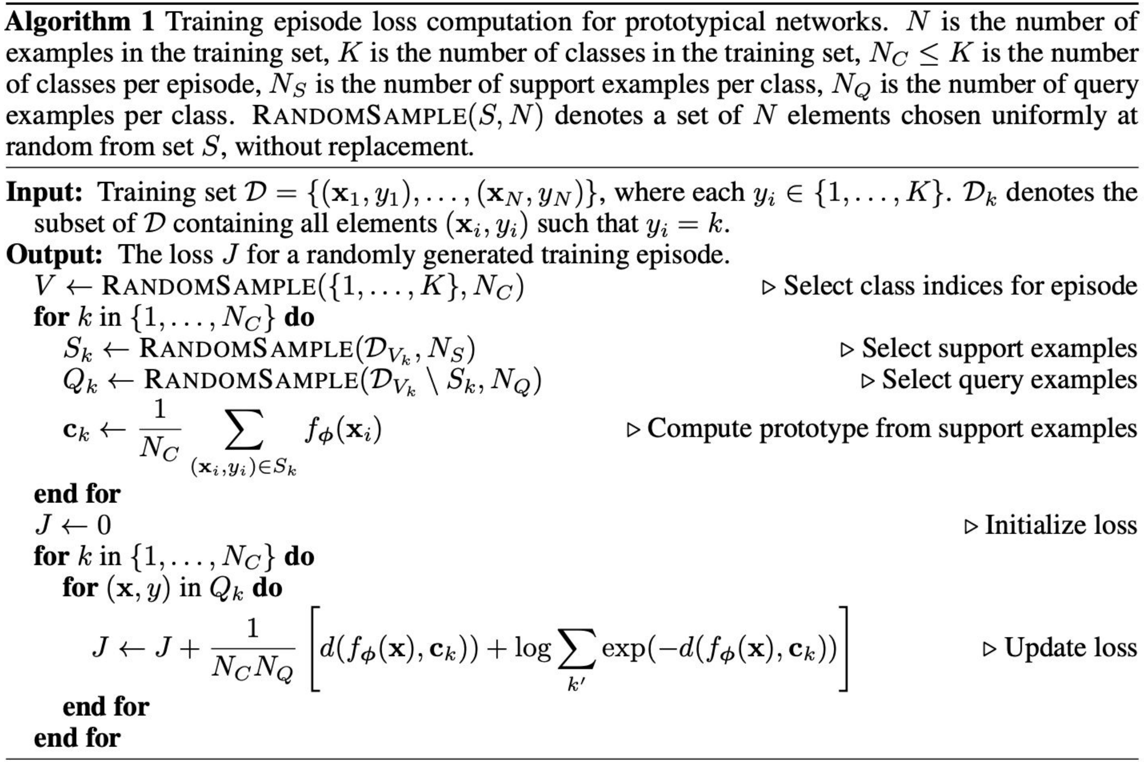
Methodology

Prototypical Networks

1. Learn a prototype network that can map an input to a prototype.
2. Compute the cluster mean from **the support set**.
3. Classify the sample from **the query set** by the closest cluster class.



Methodology



Algorithm 1 Training episode loss computation for prototypical networks. N is the number of examples in the training set, K is the number of classes in the training set, $N_C \leq K$ is the number of classes per episode, N_S is the number of support examples per class, N_Q is the number of query examples per class. RANDOMSAMPLE(S, N) denotes a set of N elements chosen uniformly at random from set S, without replacement.

```
Input: Training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$, where each $y_i \in \{1, \ldots, K\}$. $\mathcal{D}_k$ denotes the
  subset of $\mathcal{D}$ containing all elements $(\mathbf{x}_i, y_i)$ such that $y_i = k$.
Output: The loss $J$ for a randomly generated training episode.
  $V \leftarrow$ RANDOMSAMPLE$(\{1, \ldots, K\}, N_C)$                              ▷ Select class indices for episode
  for $k$ in $\{1, \ldots, N_C\}$ do
    $S_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k}, N_S)$                           ▷ Select support examples
    $Q_k \leftarrow$ RANDOMSAMPLE$(\mathcal{D}_{V_k} \setminus S_k, N_Q)$                   ▷ Select query examples
    $\mathbf{c}_k \leftarrow \frac{1}{N_C} \sum_{(\mathbf{x}_i, y_i) \in S_k} f_\phi(\mathbf{x}_i)$                 ▷ Compute prototype from support examples
  end for
  $J \leftarrow 0$                                                             ▷ Initialize loss
  for $k$ in $\{1, \ldots, N_C\}$ do
    for $(\mathbf{x}, y)$ in $Q_k$ do
      $J \leftarrow J + \frac{1}{N_C N_Q} \left[ d(f_\phi(\mathbf{x}), \mathbf{c}_k)) + \log \sum_{k'} \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k)) \right]$   ▷ Update loss
    end for
  end for
```

Evaluation Metrics

- **Accuracy**
- **Weighted F1 score**

  Calculate the F1 score for each label, then weighted by the number of true instances for each label.

- **Matthews Correlation Coefficient (MCC)**

$$\mathrm{MCC} = \frac{\mathrm{TP} \cdot \mathrm{TN} - \mathrm{FP} \cdot \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP}) \cdot (\mathrm{TP} + \mathrm{FN}) \cdot (\mathrm{TN} + \mathrm{FP}) \cdot (\mathrm{TN} + \mathrm{FN})}}$$

https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-019-6413-7

Experiment

Sample Code

Colab Link

https://colab.research.google.com/drive/1ZJAajK5NmOmXiXGzDntVkQenGECoasGo?usp=sharing

The sample code is modified from the official code of the paper Meta-learning for Classifying Previously Unseen Data Source into Previously Unseen Emotional Categories.

Experiment

Choose **at least two** of the sentence representations for the following experiment.

- **From word representation (FastText)**

  ```
  python3 metalearning.py --encoder [transfo, cnn, avg]
  ```

- **From pre-trained BERT family**

  ```
  python3 metalearning.py --encoder bert --pretrained_bert [bert-base-uncased, ...]
  ```

Experiment (1)

First, run the **supervised method** as the baseline.

Report the three evaluation metrics on the test set of GoEmotions and DailyDialog.

- **Command to run**

```
python3 metalearning.py --task supervised_goemotions_on_dailydialog
```

Experiment (2)

Next, run the **supervised method on DailyDialog** as the topline (Due to the limited data in the target domain, the topline might be worse than the meta-learning method).

Report the three evaluation metrics on the test set of DailyDialog.

- **Command to run**

```
python3 metalearning.py --task supervised_dailydialog
```

Experiment (3)

Next, run the **meta-learning** method (Prototypical network is implemented in the sample code).

Report the three evaluation metrics on the test set of GoEmotions and DailyDialog.

- **Command to run**

```
python3 metalearning.py --task metalearning
```

Experiment (4)

Finally, in this part, you can choose to either

- (4-1) Implement another meta-learning method,
- (4-2) Do some case study.

Experiment (4-1)

You can choose **any meta-learning method** to implement.

Report the three evaluation metrics on the test set of GoEmotions and DailyDialog.

The code you need to change should be in `code/train/regular.py` and `code/train/fintune.py`.

Experiment (4-2)

For your best supervised and meta-learned model (tested on DailyDialog).

- Draw the confusion matrix (6 classes).
- Provide some mistaken examples.
- Any other thing you want to observe.

The code you need to change should be in the `test` and `test_one` function in `code/train/regular.py`.

References

- Original paper and code
    - https://aclanthology.org/2021.metanlp-1.9/
    - https://github.com/gguibon/metalearning-emotion-datasource
- Datasets
    - GoEmotions: https://arxiv.org/abs/2005.00547
    - DailyDialog: https://arxiv.org/abs/1710.03957
- Sentence representation
    - FastText: https://fasttext.cc
    - Hugging Face: https://huggingface.co/transformers/pretrained_models.html
- Meta-learning
    - Prototypical Networks: https://arxiv.org/abs/1703.05175
- Saving files to Google Drive (from hw1)
    - https://www.wongwonggoods.com/python/python-colab-mount-google-drive/