Embodied Real-Time Emotion Recognition in Conversation

Peizhen Li

Faculty of Science and Engineering
Macquarie University

May 17, 2024

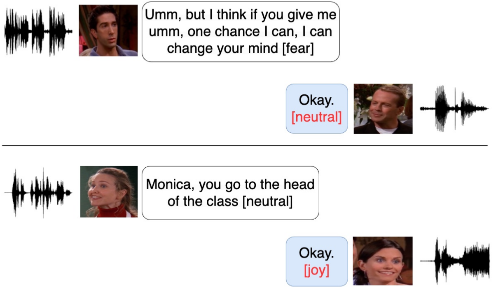
Outline

Task Formulation

Given:

- a collection of speakers $\mathcal{S}$,
- a set of emotion labels $\mathcal{E}$,
- a conversation $\mathcal{C}$, $[(s_1, u_1), (s_2, u_2), \cdots, (s_N, u_N)]$

Goal: identify the emotion label at each conversation turn



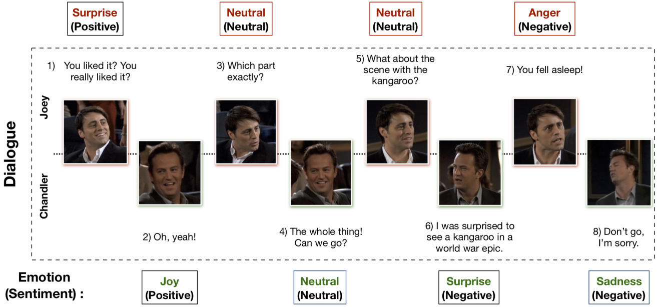
Datasets

Text-only:

- EmoryNLP

Multimodal:

- The Interactive Emotional Dyadic Motion Capture (IEMOCAP)
- Multimodal EmotionLines Dataset (MELD)



Context Modeling

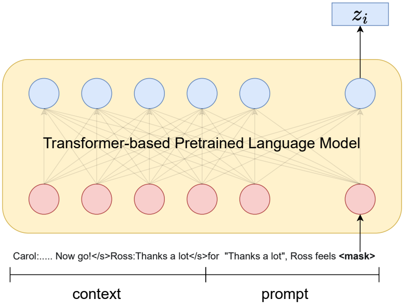
To get the text embedding of t-th turn in a dialogue:

- option 1: concatenate all contextual turns (not suitable in real-time setting)
- option 2: most recent k turns + prompt

$$C_t = [s_{t-k}, u_{t-k}, s_{t-k+1}, \cdots, s_t, u_t] \tag{1}$$

$$P_t = \text{for } u_t, < s_t > \text{ feels } < \text{mask} > \tag{2}$$

$$H_t = \text{TextEncoder}(C_t \oplus P_t) \tag{3}$$



Context Modeling

```
for _, dialogue in enumerate(dialogues):
    utterance_ids = []
    query = 'For utterance:'
    query_ids = tokenizer(query)['input_ids'][1:-1]
    for idx, turn_data in enumerate(dialogue):
        text_with_speaker = turn_data['speaker'] + ':' + turn_data['text']
        token_ids = tokenizer(text_with_speaker)['input_ids'][1:]
        utterance_ids.append(token_ids)
        if turn_data['label'] < 0:
            continue
        full_context = [CONFIG['CLS']]
        lidx = 0
        for lidx in range(idx):  # idx: curr utt_id in curr dialogue
            total_len = sum([len(item) for item in utterance_ids[lidx:]]) + 8
            if total_len + len(utterance_ids[idx]) <= CONFIG['max_len']:
                break
        lidx = max(lidx, idx - 8)        # max dis=8
        for item in utterance_ids[lidx:]:
            full_context.extend(item)
        query_idx = idx
        prompt = dialogue[query_idx]['speaker'] + ' feels <mask>'
        full_query = query_ids + utterance_ids[query_idx] + tokenizer(prompt)['input_ids'][1:]
        input_ids = full_context + full_query
        input_ids = pad_to_len(input_ids, CONFIG['max_len'], CONFIG['pad_value'])
        ret_utterances.append(input_ids)
        ret_labels.append(dialogue[query_idx]['label'])
        self.all_utt_idx_with_extra.append(all_utt_idx + idx)
```

Context Modeling

Prompt

Text Encoder

RoBERTa: A Robustly Optimized BERT Pretraining Approach

- How to use this model to get the features of a given text in PyTorch

```
from transformers import RobertaTokenizer, RobertaModel
tokenizer = RobertaTokenizer.from_pretrained('roberta-large')
model = RobertaModel.from_pretrained('roberta-large')
text = "Replace me by any text you'd like."
encoded_input = tokenizer(text, return_tensors='pt')
output = model(**encoded_input)

# encoded_input - "input_ids": torch.size([1, 12])
# tensor([[0, 9064, 6406, 162, 30, 143, 2788, 47, 1017, 101, 4, 2]])
# output - sequence output:
# torch.size([1, 12, 1024])
```

Vision Encoding

- Video level: Timesformer
- Frame level: ResNet

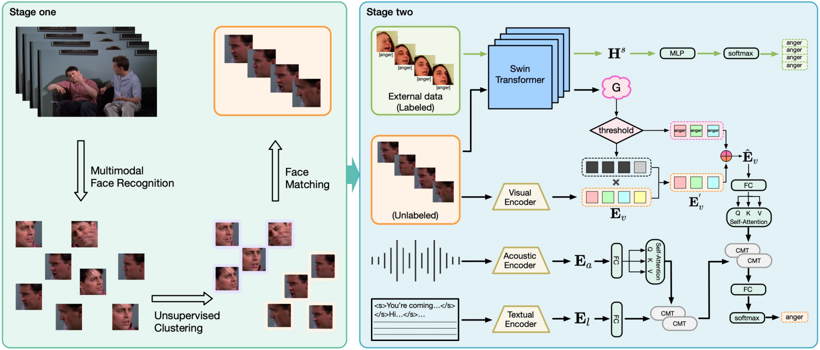
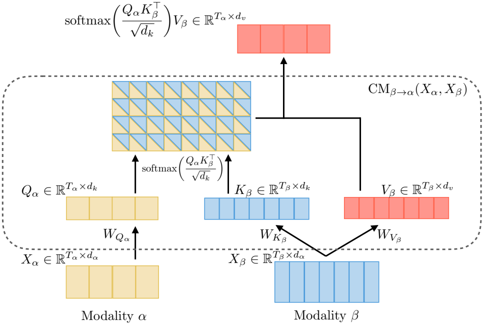


Figure credits to ACL-23: FacialMMT

Cross-Modal Attention

- Latent adaptation from β to α, $Y_\alpha = \text{CM}_{\beta\to\alpha}(X_\alpha, X_\beta)$:

$$\begin{aligned} Y_\alpha &= \text{softmax}\left(\frac{Q_\alpha K_\beta^T}{\sqrt{d_k}}\right)V_\beta \\ &= \text{softmax}\left(\frac{X_\alpha W_{Q_\alpha} W_{K_\beta}^T X_\beta^T}{\sqrt{d_k}}\right)X_\beta W_{V_\beta} \end{aligned} \tag{4}$$



ACL-19: Multimodal Transformer for Unaligned Multimodal Language Sequences

Cross-Modal Transformer

Given unimodal embeddings: $\mathbf{E}_l$, $\mathbf{E}_a$, $\mathbf{E}_v$

- intra-modal interactions: $\mathbf{H}_a = \text{Transformer}(\mathbf{E}_a)$, $\mathbf{H}_v = \text{Transformer}(\mathbf{E}_v)$
- inter-modal interactions:

$$\begin{aligned} \mathbf{H}_{l-a} &= \text{CM-Transformer}(\mathbf{E}_l, \mathbf{H}_a), \\ \mathbf{H}_{l-a-v} &= \text{CM-Transformer}(\mathbf{H}_{l-a}, \mathbf{H}_v) \end{aligned} \tag{5}$$

- emotion classification layer:

$$q(y) = \text{softmax}(\mathbf{W}^T \mathbf{H}_{l-a-v} + \mathbf{b}) \tag{6}$$

- pseudo code:

```
audio_emb=audio_transformer(audio_linear(audio_inputs),audio_mask)
vis_emb=vis_transformer(vis_linear(vision_inputs),vision_mask)
ta_feat=cm_ta_transformer(text_feat, audio_emb, audio_emb)
at_feat=cm_ta_transformer(audio_emb, text_feat, text_feat)
tat_feat=torch.cat((ta_feat, at_feat)) # concatenate
vta_feat=cm_tat_transformer(vis_emb, tat_feat, tat_feat)
tav_feat=cm_tat_transformer(tat_feat, vis_emb, vis_emb)
final_feat=torch.cat((vta_feat, tav_feat))
```

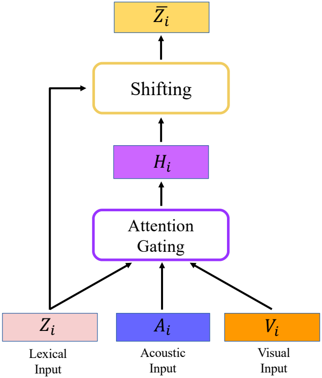
Refer to official implementation

Multimodal Adaptation Gate (MAG)

- Shifting by a displacement vector: $\bar{Z}_i = Z_i + \alpha H_i$

$$H_i = g_i^a \cdot (W_a A_i) + g_i^v \cdot (W_v V_i) + b_H \tag{7}$$

$$g_i^a = R(W_{ga}[Z_i; A_i] + b_a), \quad g_i^v = R(W_{gv}[Z_i; V_i] + b_v) \tag{8}$$



Attention-based Modality Shifting Fusion

- Fusion by the displacement vector based on non-verbal information

$$Z_k = F_{T_k} + \lambda \cdot H_k \tag{9}$$

where $H_k = g_{AV}^k \cdot (W_2 \cdot F_{\text{attn}}^k + b_2)$, $g_{AV}^k = R(W_1 \cdot [F_{T_k}; F_{\text{attn}}^k] + b_1)$

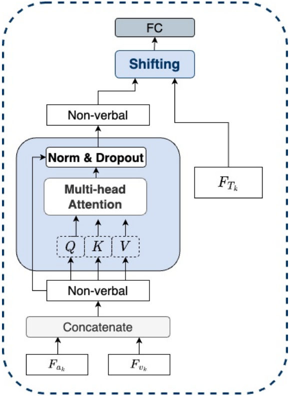
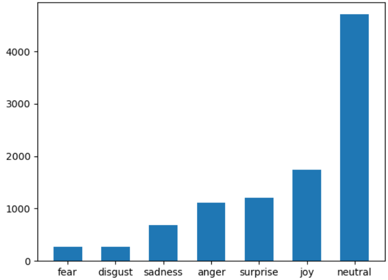


Figure credits to: TelME

Class Imbalance

- Emotion distribution on the training set of MELD dataset



- Evaluation metric: weighted-F1 score

$$
\begin{aligned}
\text{weighted-F1} &= \sum_{i=1}^{|\mathcal{E}|} w_i \times \text{F1}_i \\
\text{F1} &= 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\end{aligned}
\tag{10}
$$

Supervised Contrastive Learning

- Self-supervised contrastive loss

$$\mathcal{L}^{\text{self}} = \sum_{i \in I} \mathcal{L}_i^{\text{self}} = -\sum_{i \in I} \log \frac{\exp(z_i \cdot z_{j(i)}/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \tag{11}$$

- Supervised contrastive losses

$$\mathcal{L}_{\text{out}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{out},i}^{\text{sup}} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \tag{12}$$

$$\mathcal{L}_{\text{in}}^{\text{sup}} = \sum_{i \in I} \mathcal{L}_{\text{in},i}^{\text{sup}} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a \in A(i)} \exp(z_i \cdot z_a/\tau)} \right\} \tag{13}$$

where

$$i \in I \equiv \{1 \cdots 2N\},\ z_l = Proj(Enc(\tilde{x}_l)),\ A(i) \equiv I \backslash \{i\},\ P(i) \equiv \{p \in A(i) : \tilde{y}_p = \tilde{y}_i\}$$

given

$$\{x_k, y_k\}_{k=1 \cdots N},\ \{\tilde{x}_l, \tilde{y}_l\}_{l=1 \cdots 2N},\ \tilde{y}_{2k-1} = \tilde{y}_{2k} = y_k$$

NeurIPS-20: Supervised Contrastive Learning

Supervised Prototypical Contrastive Learning

Issue: limited batch size + class imbalance

- representation queue for each category: $Q_c = [z_1^c, z_2^c, \cdots, z_M^c]$
- support set by random selection: $S_K = \text{RANDOMSELECT}(Q_c, K)$
- prototype vector for each category: $\mathbf{T}_c = \frac{1}{K} \sum_{z_j^c \in S_K} z_j^c$
- supervised prototypical loss:

$$\mathcal{L}_i^{\text{spcl}} = -\log \left\{ \frac{1}{|P(i)| + 1} \cdot \frac{\sum_{p \in P(i)} \mathcal{F}(z_i, z_p) + \mathcal{F}(z_i, \mathbf{T}_{y_i})}{\sum_{a \in A(i)} \mathcal{F}(z_i, z_a) + \sum_{c \in \mathcal{E} \setminus \{y_i\}} \mathcal{F}(z_i, \mathbf{T}_c)} \right\} \quad (14)$$

where

$$\mathcal{F}(z_i, z_j) = \exp(\mathcal{G}(z_i, z_j)/\tau)$$

Challenges

- Embody the multimodal emotion recognition model
  - complementing it with sensor data from a robot agent
- End-to-end training
  - train on sensor data directly
  - discern good features from noisy inputs
- Real-time inference
  - reference speed: minimum of 1-3 HZ
  - cannot run large models directly on the robot
  - backend server/cloud service: round-trip delay

Refer to: PaLM-E and RT-2

Progress and Future Work

Progress:

- illustration of our framework
- preliminary results

Future work:

- deploy on Ameca
- collect more data and co-fine-tune

References

[1] **arXiv 2024** - TelME: Teacher-learning Multimodal Fusion Network for Emotion Reconition in Conversation

[2] **ACL 2023** - A Facial Expression-Aware Multimodal Multi-task Learning Framework for Emotion Recognition in Multi-party Conversations

[3] **EMNLP 2022** - Supervised Prototypical Contrastive Learning for Emotion Recognition in Conversation

[4] **NeurIPS 2020** - Supervised Contrastive Learning

[5] **ACL 2020** - Integrating Multimodal Information in Large Pretrained Transformers

[6] **ACL 2019** - Multimodal Transformer for Unaligned Multimodal Language Sequences

Thank you very much!
Q&A