对话情感识别与生成

毕冠群

2020.8.21

对话情感识别

- 特点

  识别对话中说话人的情绪，本质上是分类问题，即从预定义好的情绪类别中，为对话中的每一句表达确定其情绪的类别。

  对话存在三个特点：

  1、对话文本是短的、非正式文本；

  2、对话中的主题时常快速切换，因此上下文是动态的；

  3、对话者之间的交互会改变用户的情绪和状态。

  由于对话本身具有很多要素，话语的情绪识别并不简单等同于单个句子的情绪识别，而是需要综合考虑对话中的**背景**、**上下文**、**说话人**等信息，这些都是对话情感识别任务中独特的挑战。

  1. 话语本身及其上下文（由对话者在对话中的先前话语定义）以及意图和话题对话；
  2. 说话者的状态，包括诸如性格和论证逻辑之类的变量
  3. 前述言语表达的情感。

对话情感识别

- 挑战
  - 交互方面：说话人的差异；听众的反应；多方对话
  - 情感动态方面：动态上下文；情感惯性；人际影响
  - 话题方面：细粒度的情感；讽刺挖苦

Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

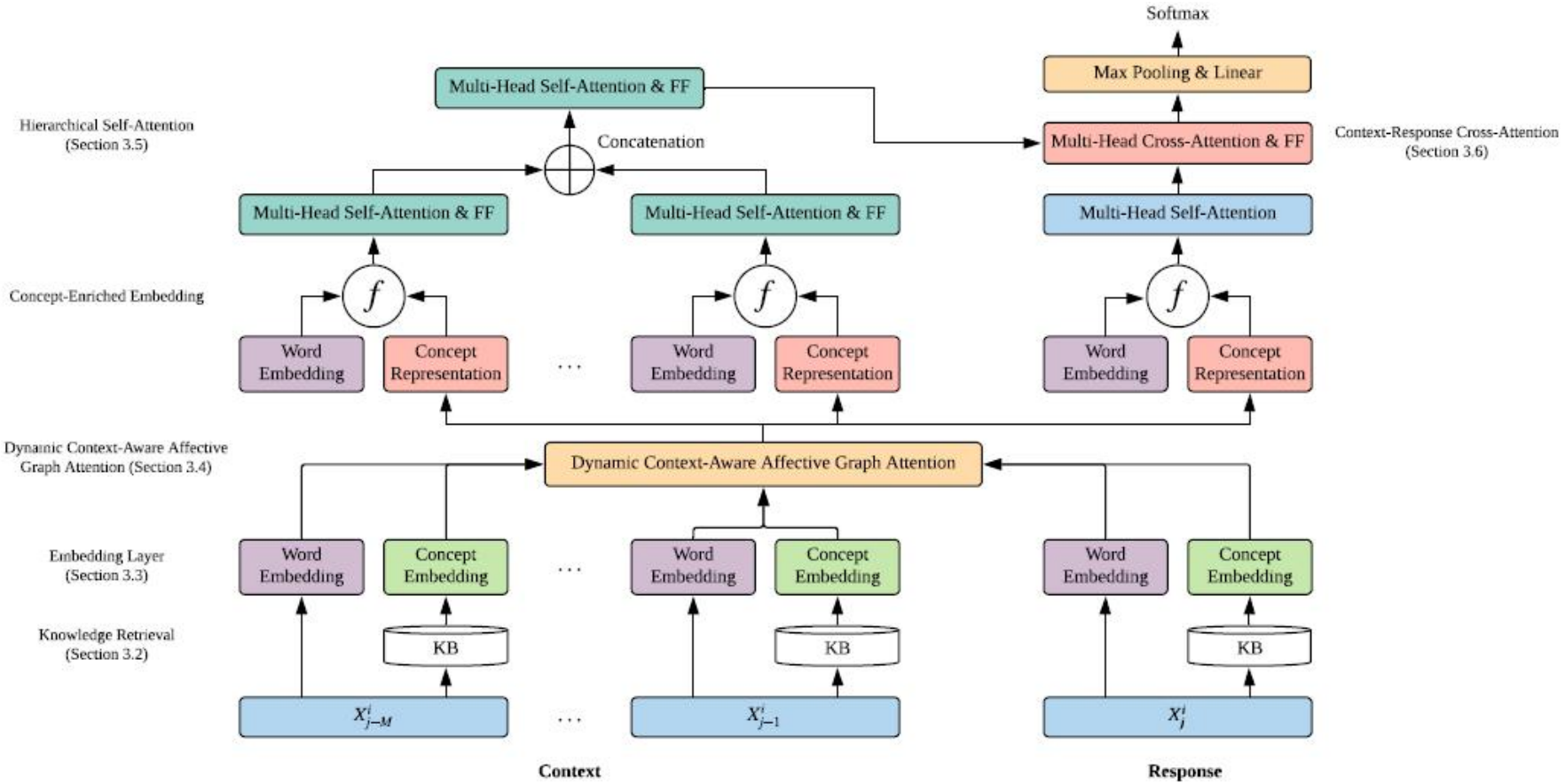


Figure 2: Overall architecture of our proposed KET model. The positional encoding, residual connection, and layer normalization are omitted in the illustration for brevity.

Knowledge—Enriched Transformer for Emotion Detection in Textual Conversations

- **任务定义**

  $\{X_j^i, Y_j^i\}, i = 1, ...N, j = 1, ...N_i$ 是一系列的{*utterance*, *label*}对，

  其中*N*表示conversation数量，N_i表示第*i*场会话里的utterances数量

  $$\Phi = \prod_{i=1}^{N} \prod_{j=1}^{N_i} p(Y_j^i | X_j^i, X_{j-1}^i, ..., X_1^i; \theta), \quad (1)$$

  目标：最大化

  where $X_{j-1}^i, ..., X_1^i$ denote contextual utterances and θ denotes the model parameters we want to optimize.

Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- **知识检索**
  - **常识知识：ConceptNet**

    一个大规模多语言语义图。一个triple<concept1，relation，concept2>作为一个assertion，每个assertion会有一个confidence score。

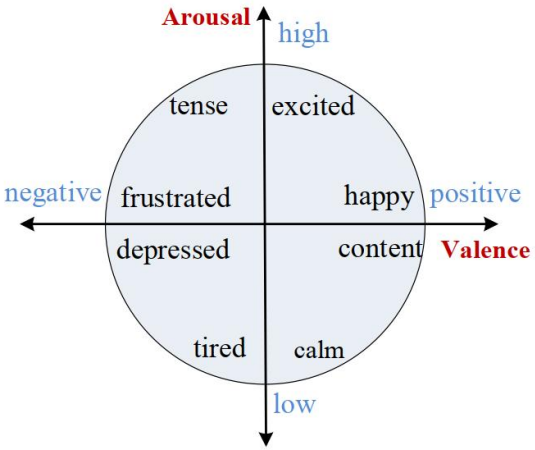
    e.g. An example assertion ***< friends, CausesDesire, socialize >***

    对应confidence score 3.46

  - **情感知识：NRC_VAD**

    一个英语单词和它的VAD分数的列表，VAD数值范围[0,1]，表示的是valence(negative-positive)、arousal(calm-excited)、dominance(submissive-dominant)。

    e.g. *socialize* {*V*: 0.907, *A*: 0.683, *D*: 0.726}



Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- **知识检索**
  1. 对于每个x_j^i中非stopwords的token t，从ConceptNet中检索得到一个包含所有直接邻居的知识图谱g(t)
  2. 对每个g(t)，移除属于以下三种情况的concept：
     - stopwords
     - 不在词典中
     - confidence scores $<$ 1
  3. 对每个concept，从NRC_VAD中获取VAD值
  4. token t的最终知识表示：一系列tuples：

$$(c_1, s_1, VAD(c_1)), (c_2, s_2, VAD(c_2)), \ldots, (c_{|g(t)|}, s_{|g(t)|}, VAD(c_{|g(t)|}))$$

$c_k \in g(t)$是第k个相连的concept，s_k是对应的confidence score，$VAD(c_k)$是c_k的VAD值

Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- **嵌入层**
  - word embedding layer：

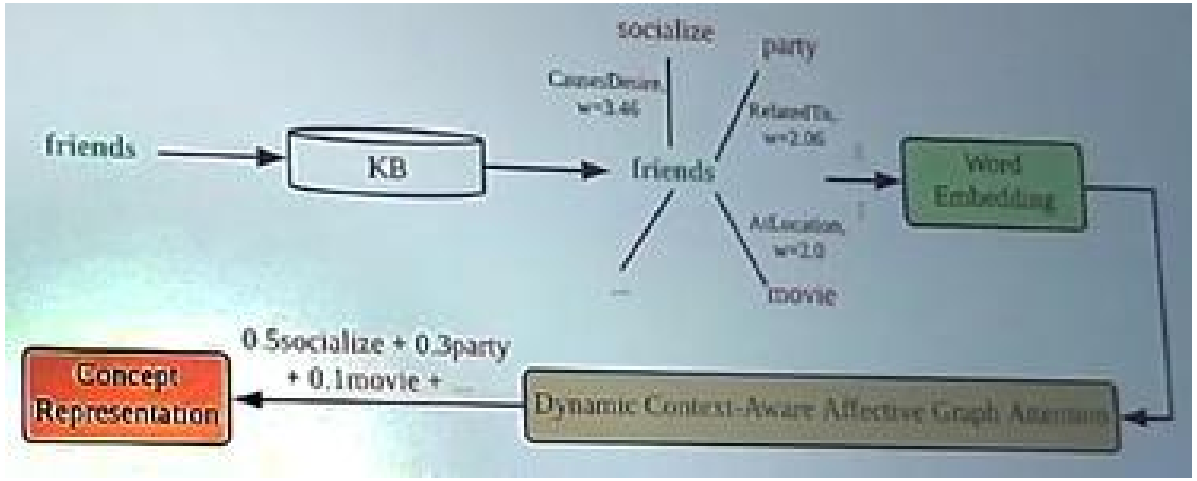
    将X^i中的每个token t转化成向量表示 $\mathbf{t} \in \mathbb{R}^d$

$$\mathbf{t} = Embed(t) + Pos(t)$$

  - concept embedding layer：

    将concept c转化成向量表示 $\mathbf{c} \in \mathbb{R}^d$

    没有位置编码。



Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- Dynamic Context-Aware Affective Graph Attention
  - 目的：用concept representation充实word embedding，即计算每个token融入知识后的上下文表示 concept representation $\mathbf{c}(t) \in \mathbb{R}^d$

$$\mathbf{c}(t) = \sum_{k=1}^{|g(t)|} \alpha_k * \mathbf{c_k},$$

$$\alpha_k = softmax(w_k),$$

  - w_k表示c_k的权重。
  - 标准图注意力机制计算w_k，但是在检测情绪时concept虽然相关但不一定是平等的。
  - 文章做了这样一个假设，即重要的concept除了和**对话上下文**相关，还与**情绪强度**相关。

Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- Dynamic Context-Aware Affective Graph Attention
  - 因此提出上下文感知情感图注意力机制，即在w_k时，考虑相关性relatedness和情感affectiveness两方面因素。
  - **Relatedness**

$$rel_k = min\text{-}max(s_k) * abs(\cos(\mathbf{CR}(X^i), \mathbf{c_k}))$$

$CR(X^i)$ 表示第i组对话的上下文表示，因为一组对话中可能存在多个句子，所以就表示为所有句子的向量平均

$$\mathbf{CR}(X^i) = avg(\mathbf{SR}(X^i_{j-M}), ..., \mathbf{SR}(X^i_j))$$

  - **Affectiveness**

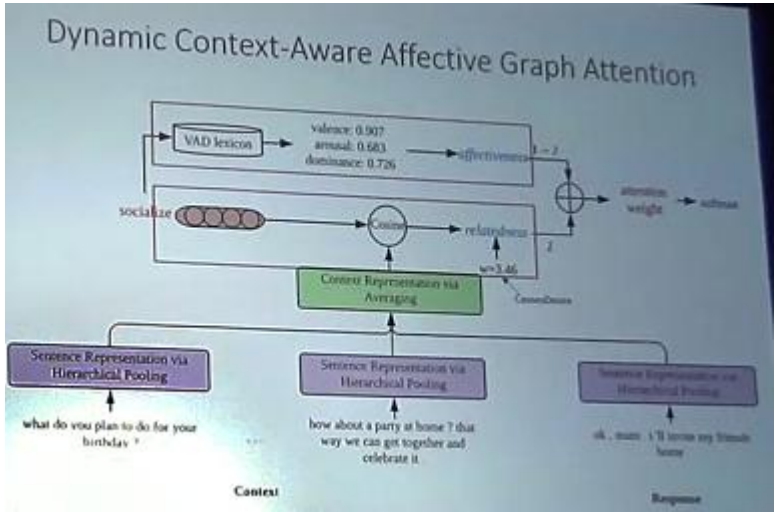
$$aff_k = min\text{-}max(||[V(c_k)-1/2, A(c_k)/2]||_2)$$

  - 结合二者计算w_k

$$w_k = \lambda_k * rel_k + (1-\lambda_k) * aff_k,$$

  - 最后知识扩充的词表示$\hat{t}$

$$\hat{\mathbf{t}} = \mathbf{W}[\mathbf{t}; \mathbf{c}(t)]$$



Knowledge-Enriched Transformer for Emotion Detection in Textual Conversations

- **多层自注意力**
  - 提出多层自注意力机制来获得对话的结构性表示，
    对上下文的表示 $X^i_{j-1}, ..., X^i_{j-M}$ 学习一个表示向量。
  - 多层注意力有两个步骤：
    1. 每个句子表示用一个句子级自注意力层来计算。
    2. 上下文表示由M个学习来的句子表示通过一个上下文自注意力层得到。
  - Step 1

$$\hat{\mathbf{X}}'^i_n = FF(L'(MH(L(\hat{\mathbf{X}}^i_n), L(\hat{\mathbf{X}}^i_n), L(\hat{\mathbf{X}}^i_n)))),$$

$$MH(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_s}})V,$$

$$FF(x) = max(0, xW_1 + b_1)W_2 + b_2,$$

  - Step 2

$$\mathbf{C}^i = FF(L'(MH(L(\hat{\mathbf{X}}^i), L(\hat{\mathbf{X}}^i), L(\hat{\mathbf{X}}^i))))$$

Knowledge—Enriched Transformer for Emotion Detection in Textual Conversations

- Context—Response Cross Attention
  - 目的：得到上下文感知的、知识扩充的response representation $\mathbf{R}^i \in \mathbb{R}^{m \times d}$

$$\mathbf{R}^i = FF(L^{'}(MH(L(\hat{\mathbf{X}}_j^{'i}), L(\mathbf{C}^i), L(\mathbf{C}^i)))), \quad (14)$$

$$\hat{\mathbf{X}}_j^{'i} = L^{'}(MH(L(\hat{\mathbf{X}}_j^i), L(\hat{\mathbf{X}}_j^i), L(\hat{\mathbf{X}}_j^i))), \quad (15)$$

  - 结果表示然后被输入到最大池化层，得到最终表示

$$\mathbf{O} = max\_pool(\mathbf{R}^i).$$

  - 计算输出情感标签概率

$$p = softmax(\mathbf{O}W_3 + b_3),$$

情感对话生成

- 对话情感生成是一个生成任务，旨在对话中生成蕴含情感、有针对性的回复。
- 对于待生成回复的情感，一般有两种观点：
  1. 认为待生成回复的情感需要明确指出。
     - 输入是对话上文和**目标情感**，输出是蕴含该情感的回复，
     - 优点是生成情感灵活可控，缺点是需要大规模情感标注的对话语料；
  2. 认为待生成回复的情感已经隐含在对话上文之中，不需要明确指出。
     - 只需要提供对话上文，
     - 优点是可利用已有的大规模对话语料，缺点是生成的情感不易控制。

MoEL：Mixture of Empathetic Listeners

- **任务定义**
  - 定义了speaker和listener两种角色
  - 给定：对话上下文 $C = \{U_1, S_1, U_2, S_2, \cdots, U_t\}$

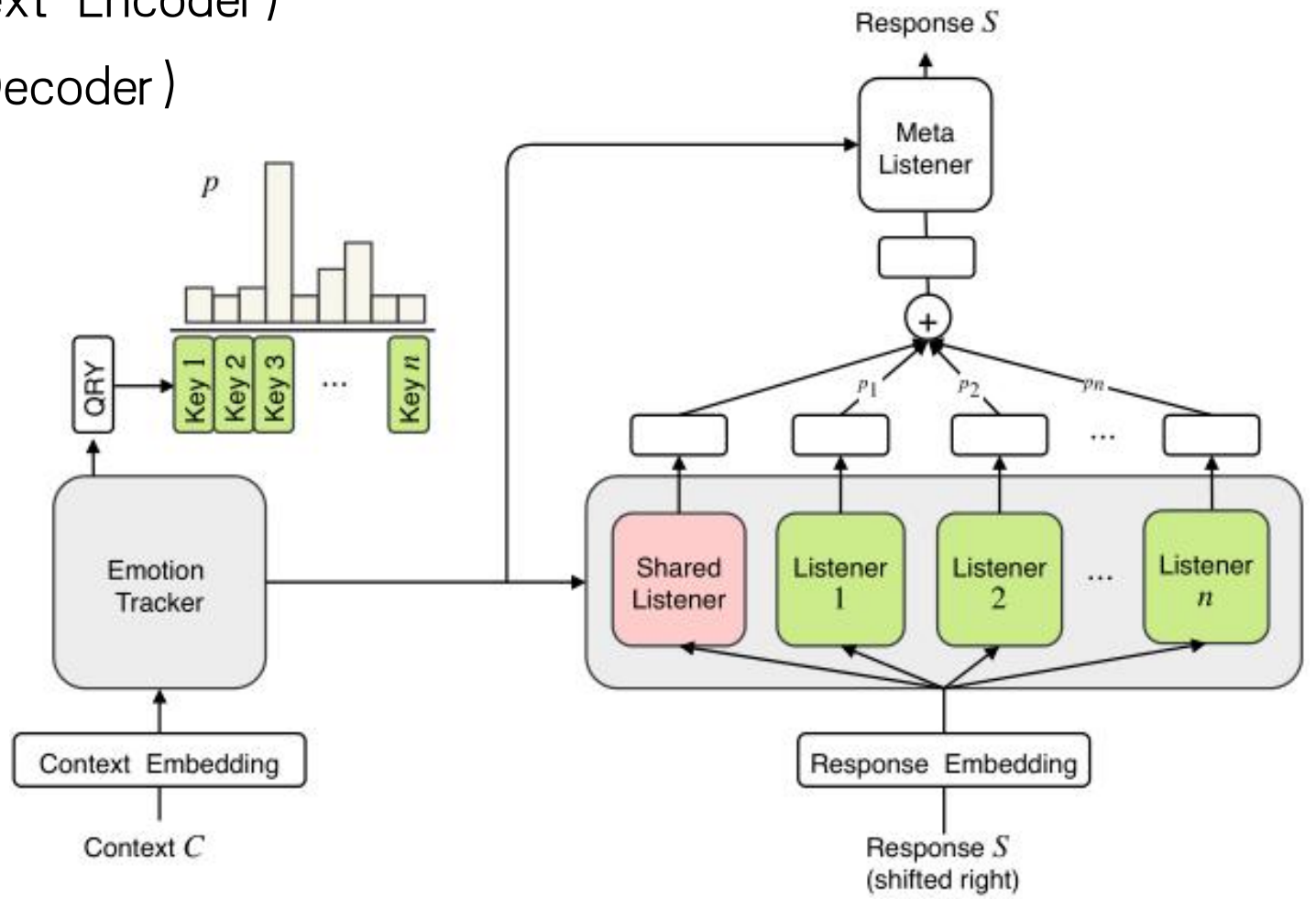
    speaker每句话对应的情绪 $Emo = \{e_1, e_2, \cdots, e_t\}$ where $\forall e_i \in \{1, \ldots, n\}$
  - 目标：根据上下文C，

    追踪speaker的情绪状态e_t，

    并生成具有同理心的反应S_t

MoEL: Mixture of Empathetic Listeners

- Emotion Tracker*1; (Context Encoder)
- Empathetic Listeners*n; (Decoder)
- Shared Listener*1;
- Meta Listener*1;



MoEL: Mixture of Empathetic Listeners

- **Embedding**
  - Context embedding $E^C \in \mathbb{R}^{|V| \times d_{emb}}$
  - Response embedding $E^R \in \mathbb{R}^{|V| \times d_{emb}}$

$$E^C(C) = E^W(C) + E^P(C) + E^D(C)$$

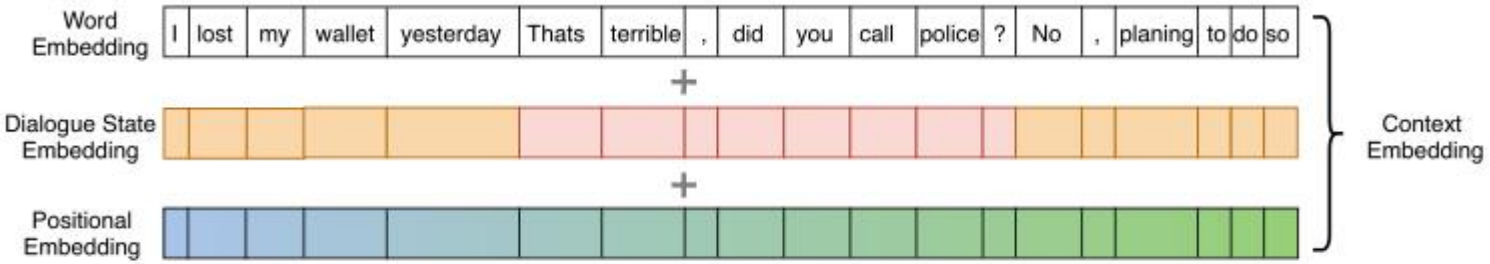


Figure 2: Context embedding is computed by summing up the word embedding, dialogue state embedding and positional embedding for each token.

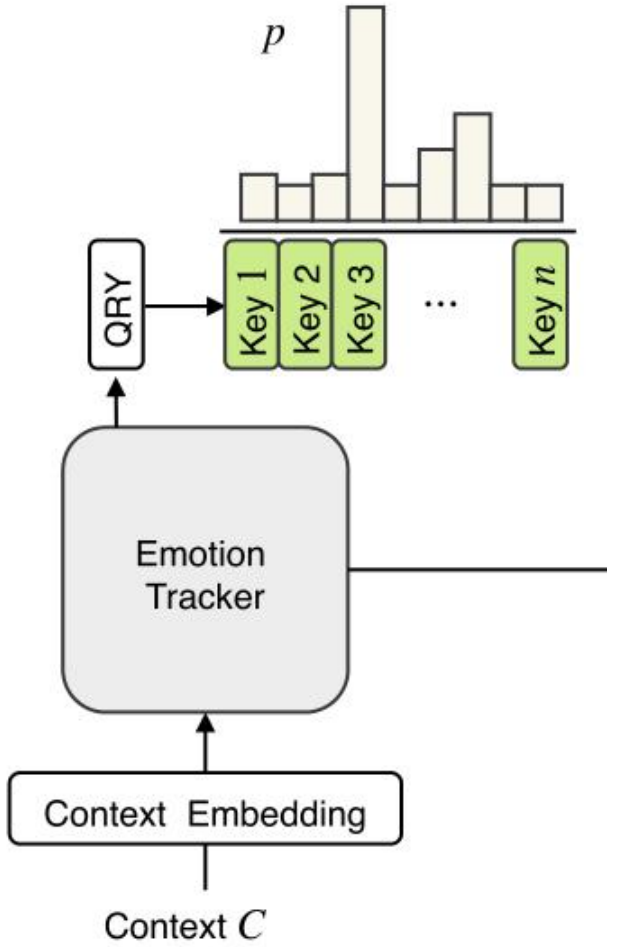
MoEL：Mixture of Empathetic Listeners

- **Emotion Tracker**
  - 标准Transformer Encoder

    QRY：在输入序列的起始处加的token，参考BERT

$$H = TRS_{Enc}(E^C([QRY; C]))$$

  - *QRY*的最终表示 $q = H_0$ ，可以用于生成情绪



MoEL：Mixture of Empathetic Listeners

- **Emotion Aware Listeners**
  - 计算listeners的情感回复表示V_i

$$V_i = TRS^i_{Dec}(H, E^R(r_{0:t-1}))$$

  - 确定不同listener的权重：
    - q为Emotion Tracker的输出query，k为预测情感分布，v为每个listener的输出

$$p_i = \frac{e^{q^\top k_i}}{\sum_{j=1}^{n} e^{q^\top k_j}}$$

    - p_i用作对应V_i的权重，通过交叉熵损失函数学习

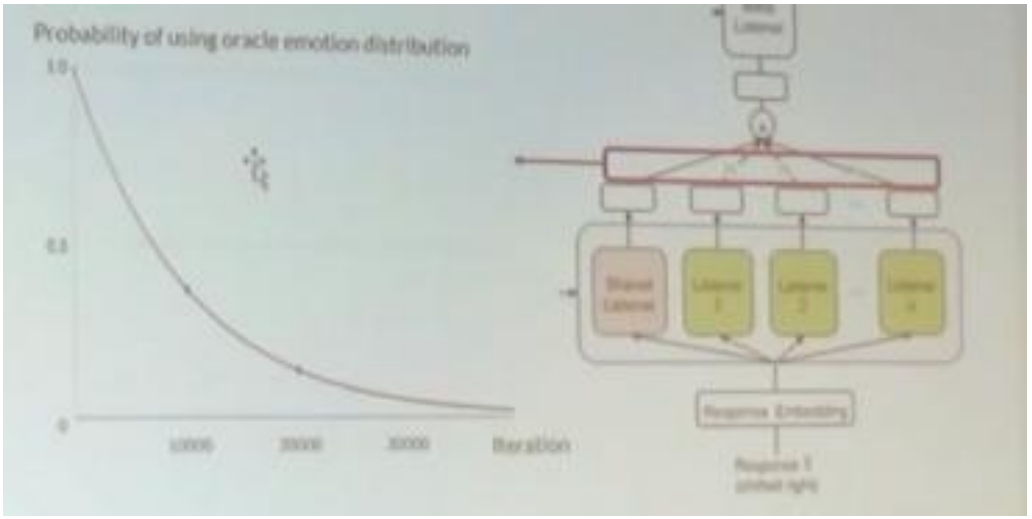
$$\mathcal{L}_1 = -\log p_{e_t}$$

    - 最终输出结合了不同情绪listeners V_i 的加权和 和 shared listener输出V_0

$$V_M = V_0 + \sum_{i=1}^{n} p_i V_i$$

MoEL：Mixture of Empathetic Listeners

- **Emotion Aware Listeners**
  - 在训练早期，emotion tracker随机初始化，分配给listeners权重的权重也是随机的，
  - 为了使训练更稳定，可以在训练早期直接使用emotion label作为emotion distribution
  - 以概率ϵ_{oracle}，用oracle emotion e_t替代distribution p，并在训练过程中逐渐减少该比例



$$\epsilon_{oracle} = \gamma + (1-\gamma)e^{-\frac{t}{t_{thd}}}$$

MoEL：Mixture of Empathetic Listeners

- **Meta Listener**
  - 每个listener专精于一种特定的情绪，Meta Listener综合所有listeners的信息，生成最终的response

$$O = TRS_{Dec}^{Meta}(H, V_M)$$

$$p(r_{1:t}|C, r_{0:t-1}) = \text{softmax}(O^\top W)$$

  - 损失函数：用标准MLE优化response的预测

$$\mathcal{L}_2 = -\log p\left(S_t|C\right)$$

  - 最终所有参数端到端联合训练，同时优化listener selection和response generation，最小化两个losses的加权和

$$\mathcal{L} = \alpha\mathcal{L}_1 + \beta\mathcal{L}_2$$

THANKS