



# MMbot

## Title

Towards Building an Open-Domain Dialogue System Incorporated with Internet Memes

建立一个结合互联网表情包的开放领域对话系统

## 思维导图

大概的流程总结：

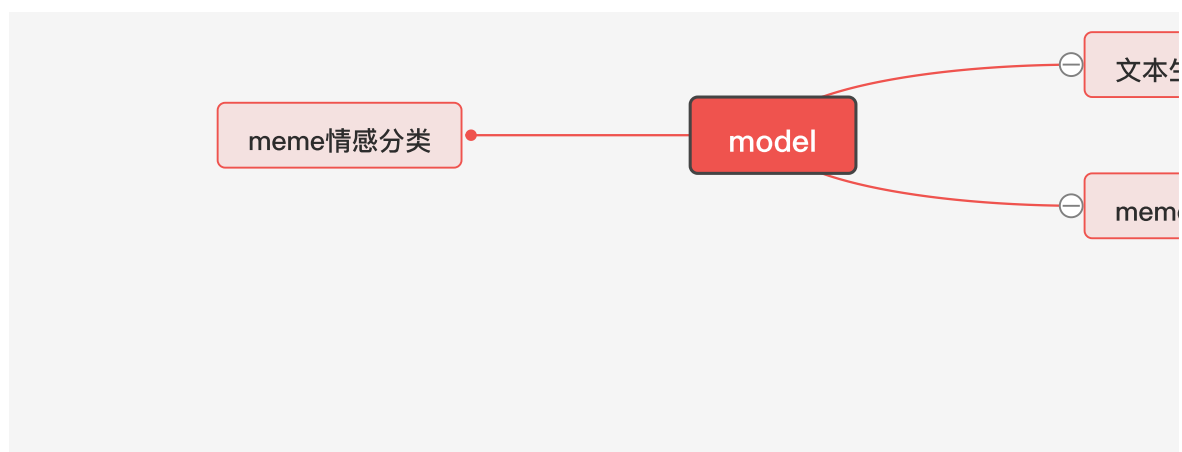
首先使用大规模预训练模型生成回复文本；

如果需要使用 meme 梗图进行回复，那么就进行 meme 梗图检索任务，把梗图都变成 title+ocr text 的格式，拼接到对话上下文中（分别拼接正例负例然后通过 cls 来计算分类）；

如果要完成 meme 情感分类任务，就把对话历史上下文，梗图的 title 和 ocr text 以及对话情感流拼接到一起，做分类任务。

100%

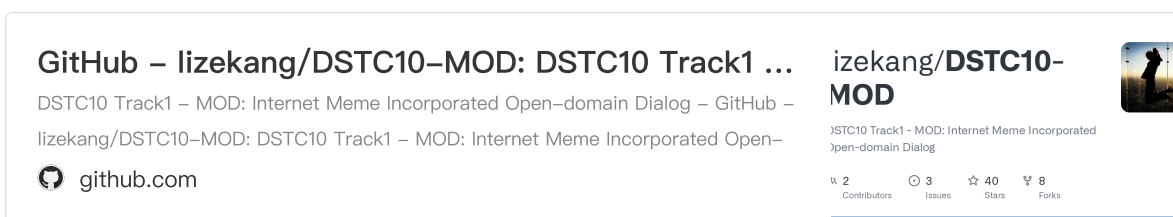




## Abstract

对比只有文字的聊天，使用互联网 meme 的聊天更加有表达力和吸引力。

本文介绍了我们对 DSTC10 的 meme incorporated open-domain dialogue 挑战的解决方案



其中涉及三个任务：

1. 利用大规模预训练对话模型生成连贯而丰富的反映
2. 基于互动的文本匹配，我们的方法可以检索到合适的 meme，并具有良好的泛化能力
3. 我们对对话的情感流进行建模，引入情感描述预测的辅助任务，提高 meme 分类的性能。

## 数据集分析

对话包含每句话的情感，是否使用表情包，表情包（若使用），文本，说话人 id

```
Python
1  {
2      "dialogue 43992": [
3          {
4              "speaker_id": "[speaker1]",
5              "emotion_id": 0,
```

```

6         "img_id": "195",
7         "txt": "\u53ef\u4ee5\u7684\u5ba2\u5b98\u62cd\u4e0b\u8bf4\u4e0d\u53ef\u4ee5",
8     },
9     {
10        "speaker_id": "[speaker2]",
11        "txt": "\u90a3\u6211\u4e70\u4e24\u4ef6\u4e24\u4e2a\u5730\u4e0d\u53ef\u4ee5",
12    },
13    {
14        "speaker_id": "[speaker1]",
15        "emotion_id": 1,
16        "img_id": "272",
17        "txt": "\u5ba2\u5b98\u8fd9\u4e2a\u662f\u5728\u540c\u4e00\u4e2a\u5730\u4e0d\u53ef\u4ee5",
18    },
19    {
20        "speaker_id": "[speaker2]",
21        "txt": "\u6211\u7684\u610f\u601d\u662f\u6211\u4e70\u516d\u4e24\u4e24\u4e2a\u5730\u4e0d\u53ef\u4ee5",
22    }
23 ]
24 }

```

## Introduction

在 DSTC10 中，将 meme 融入 open-domain dialogue (MOD) 的挑战。

它包含 3 个任务：

1. text response modeling:  
给定多模态上下文，任务时产生一个连贯的、信息丰富的**文本回复**
2. meme retrieval:  
给定一个多模态上下文和一个文本响应，检索一个适当的 meme
3. meme emotion classification  
给定一个多模态上下文和一个有 meme 的文本回复，预测 meme 的情感类型

MOD 数据集被划分为一个简单测试版本和一个困难测试版本。后者包含训练集中没有出现的 meme，被用作评估对话系统的泛化能力。

## Methodology

## Text Response Modeling

开放谈话用户自由的谈话任何话题，并且系统的回答应该在许多方面达到高标准，包括连贯性，一致性，信息量等。

$C_t = \{ \langle u_1, m_1 \rangle, \langle u_2, m_2 \rangle, \dots, \langle u_t, m_t \rangle \}$ ，其中  $u_i$  表示第  $i$  轮句子文本， $m_i$  指的是与它相关的 meme。如果第  $i$  个文本没使用 meme， $m_i$  表示为无。

文本回复模型的任务是生成反映  $r$ ，并给出多模态对于上下文  $C_t$ 。

本文使用 PLATO-2 作为基线模型。

## Meme Retrieval

meme retrieval 是至关重要的组成部分。任务目标是从 Internet meme set 中选择一个合适的 meme。

在这个工作中，我们用文本信息表示 meme 包括 meme 的 title 以及 ocr 文本（从 meme 中提取），尽管已经有了很多多模态预训练的工作，但是他们在 meme 特征提取上的效果比较差。实验结果表明 meme 的 title 以及 ocr 文本提取能够充分地表示 internet meme 的含义。

因此，我们把 meme 检索任务视作一个文本匹配问题并且使用较差编码器结构来计算相关性。

input: 对话文本上下文  $C_t$ ，文本回复  $u_{t+1}$ ，候选的 meme 梗图  $m_c$ 。

在整个训练过程中，一对 positive 和 negative 样本被送入网络中。

output 的[CLS]token 通过一个全连接层，并且经过一个 sigmoid 函数来获得相关概率  $p_{\text{matching}}(l_{mc} = 1 | C_t, u_{t+1}, m_c)$ ，其中  $l_{mc}$  代表在对话背景和相应的文本反应下，选择 meme  $mc$  与否的标签。

$$\mathcal{L}_{\text{Hinge}} = \max \left( 0, p_{\text{matching}}(l_{m_c^-} = 1 | C_t, u_{t+1}, m_c^-) - p_{\text{matching}}(l_{m_c^+} = 1 | C_t, u_{t+1}, m_c^+) + \alpha \right)$$

其中， $\alpha$  是一个预先定义的边缘参数， $m_c^+$  是正例 meme， $m_c^-$  是负例 meme。在整个训练过程中，我们使用动态随机负抽样，意味着在模型训练过程，不同的 epoch 会对不同的负向样本集合进行动态采样。

在推理阶段，internet meme 使用如下公式进行选择：

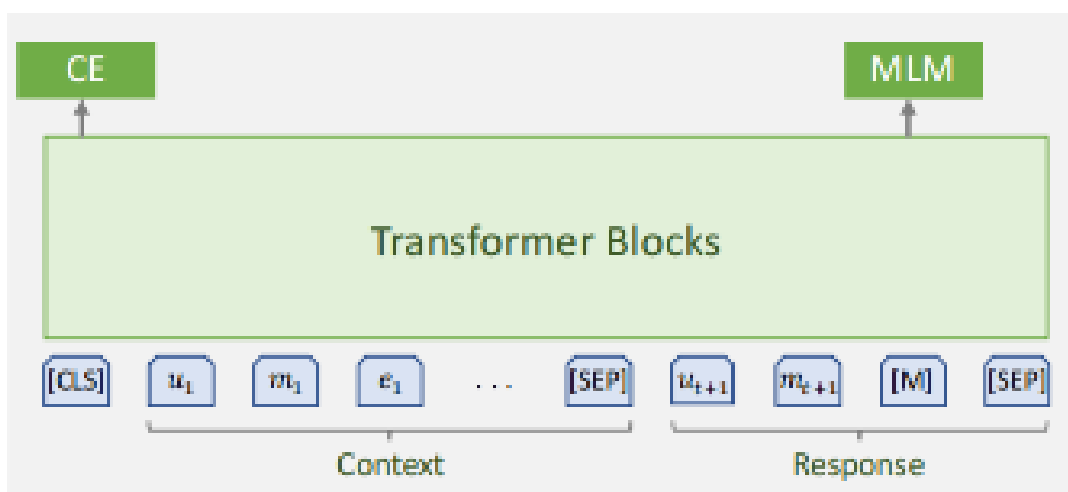
$$m^* = \operatorname{argmax}_{m_c \in M} p_{\text{matching}}(l_{m_c} = 1 \mid C_t, u_{t+1}, m_c)$$

## Meme Emotion classification

与其说是对互联网 meme 的情感进行分类，不如说这项工作的目的是预测 meme 的情感类型。

考虑到对话中对话者的情绪很少会突然变化，我们对多轮对话中的情绪流进行建模。情感的文本描述被整合进对话上下文。

特别的，第  $i$  轮的句子被以下元素组成  $\langle u_i, m_i, e_i \rangle$ 。meme 情感识别可以被视为一个经典的序列分类任务。损失函数是交叉熵。



(c) Meme Emotion Classification

此外，我们引入了一个额外的情感描述预测(EDP)任务去促进 meme 情感识别的性能。额外任务是去回复 masked token 通过 MLM 任务（mask 是情绪掩码）。在这个任务中，训练目标是让 meme 情感分类最小化。

## Experiments

meme 有 title 标识

	# dialogue	# utterance	# Internet meme	# meme emotion
Train	41,644	558,181	274	50
Valid	1,000	13,666	274	47
Easy Test	1,000	13,999	274	50
Hard Test	1,530	20,258	307	50

Table 1: Statistics of the MOD dataset.

## Settings

评价指标：

- ▼ task1:bleu-2/4, DIST-1/2

[点击创建内容](#)

- ▼ task2:Recall 10@1, Recall0@3, Recall10@5 以及 MAP

[点击创建内容](#)

- ▼ task3:accuracy@1, accuracy@3, accuracy@5

[点击创建内容](#)

实现细节：

text response model: plato-2

正例比负例：1:5

## Experimental Results