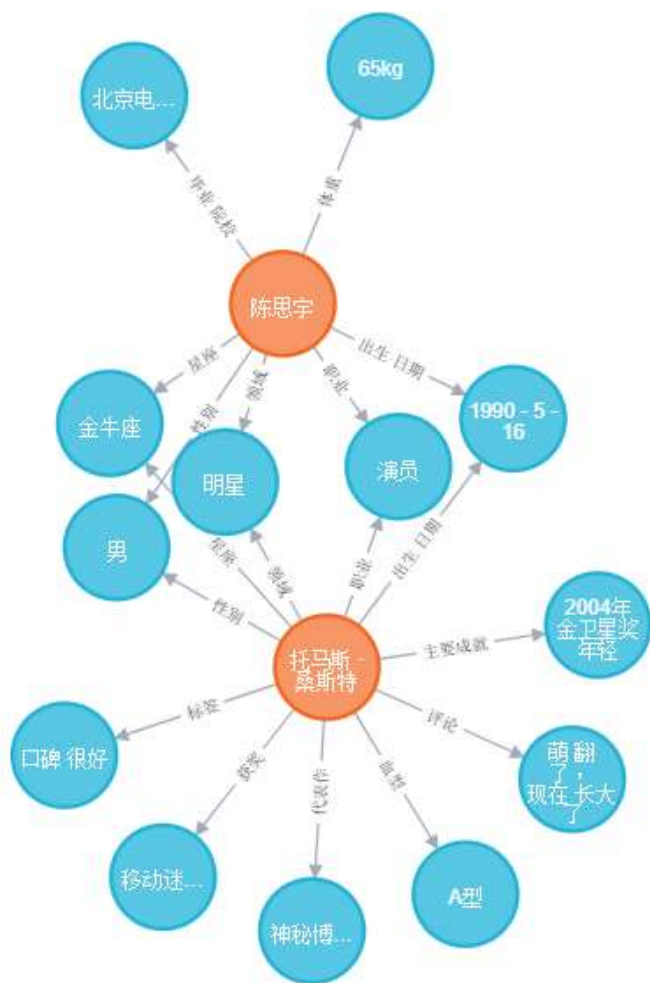


基于知识驱动的对话聊天 机器人

背景

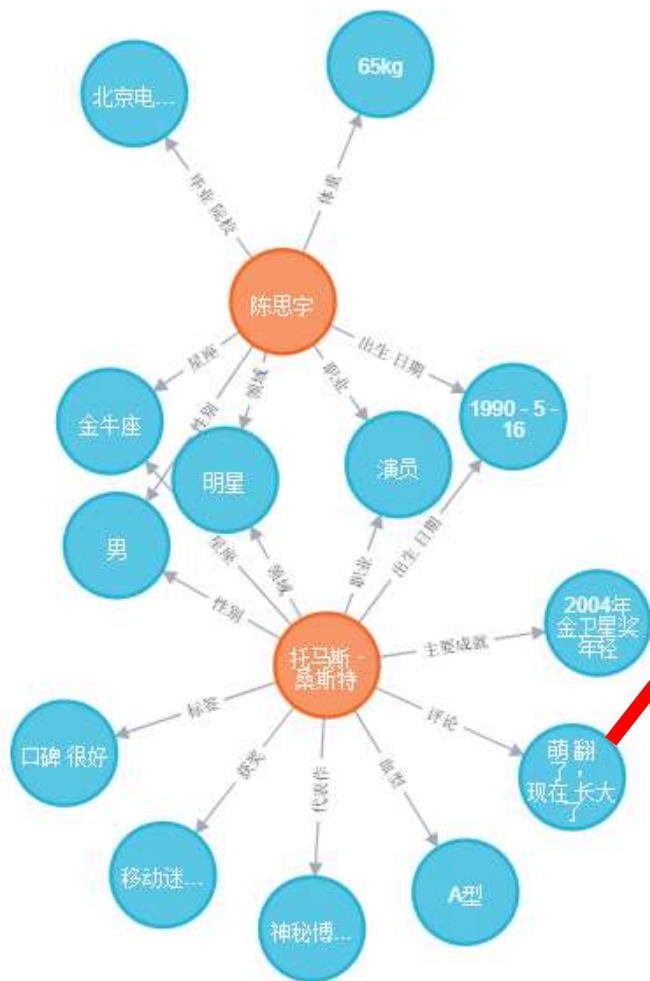
- 知识驱动对话---当前的聊天机器人聊天不够主动（更多仅仅是回复），且回复信息不够丰富
- 目标：对话系统为机器设定了一个对话目标，goal 为“START->TOPIC_A->TOPIC_B”，表示从冷启动状态主动聊到话题A，然后进一步聊到话题B；
- 提供的相关知识信息包括：话题A的知识信息，话题B的知识信息，话题A和话题B的关联信息。

一个例子



- 话题A: 托马斯·桑斯特
- 话题B: 陈思宇
- 话题间的关联: 两者的出生日期都是1990-5-16

一个例子



机器人: 知道外国有个明星长得很萌吗?

用户: 这个还真不知道呢, 请问是谁啊?

机器人: 是托马斯·桑斯特, 颜值太高了。

用户: 哦, 没听说过呢, 你能给大体说说么?

机器人: 给你大体说说, 他口碑很好的, 也很有才华, 我们国家有个小哥哥跟他一样都是1990年5月16日出生的。

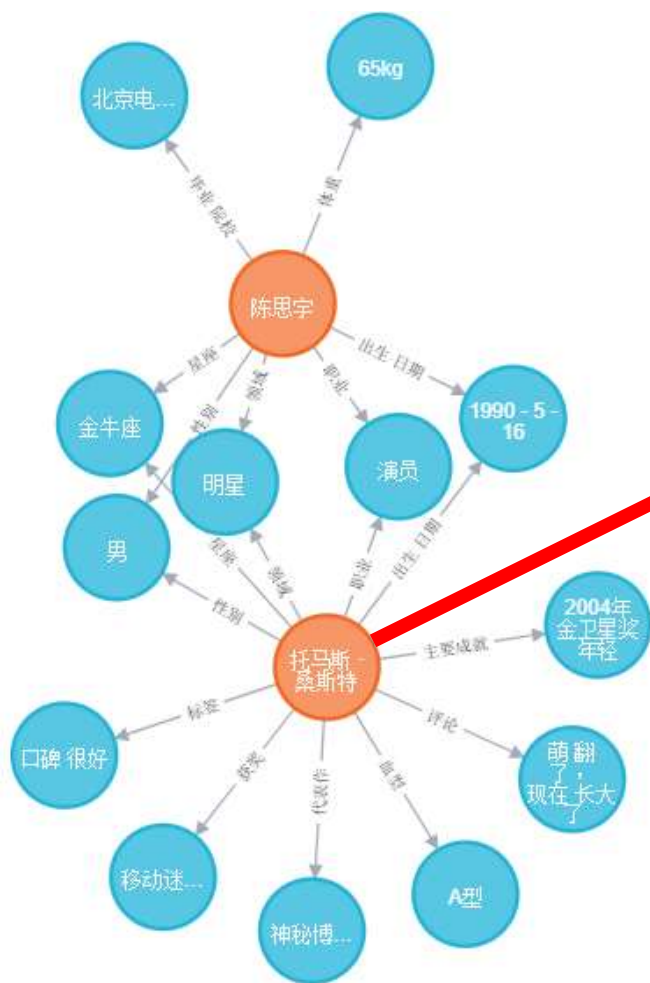
用户: 是谁啊?

机器人: 陈思宇, 金牛座的, 毕业于北京电影学院。

用户: 有时间了解一下。

Topic:
"托马斯·桑斯特" ->
"陈思宇"

一个例子



机器人: 知道外国有个明星长得很萌吗?

用户: 这个还真不知道呢, 请问是谁啊?

机器人: 是托马斯·桑斯特, 颜值太高了。

用户: 哦, 没听说过呢, 你能给大体说说么?

机器人: 给你大体说说, 他口碑很好的, 也很有才华, 我们国家有个小哥哥跟他一样都是1990年5月16日出生的。

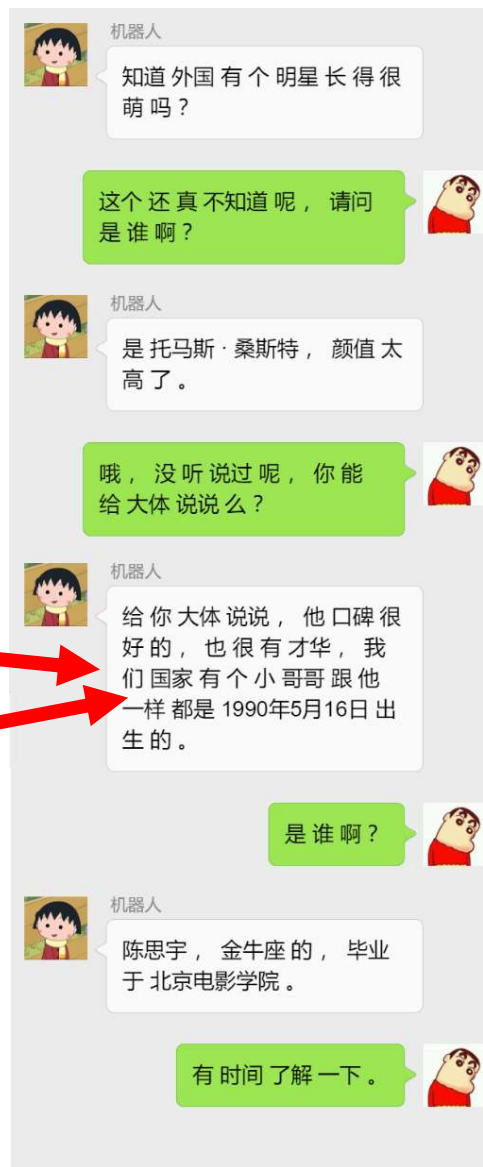
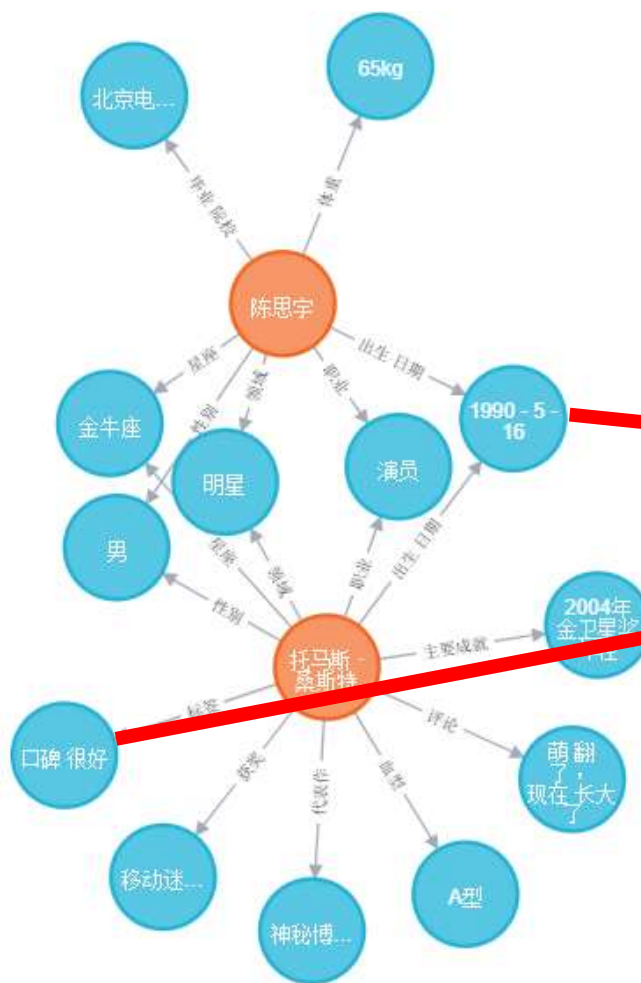
用户: 是谁啊?

机器人: 陈思宇, 金牛座的, 毕业于北京电影学院。

用户: 有时间了解一下。

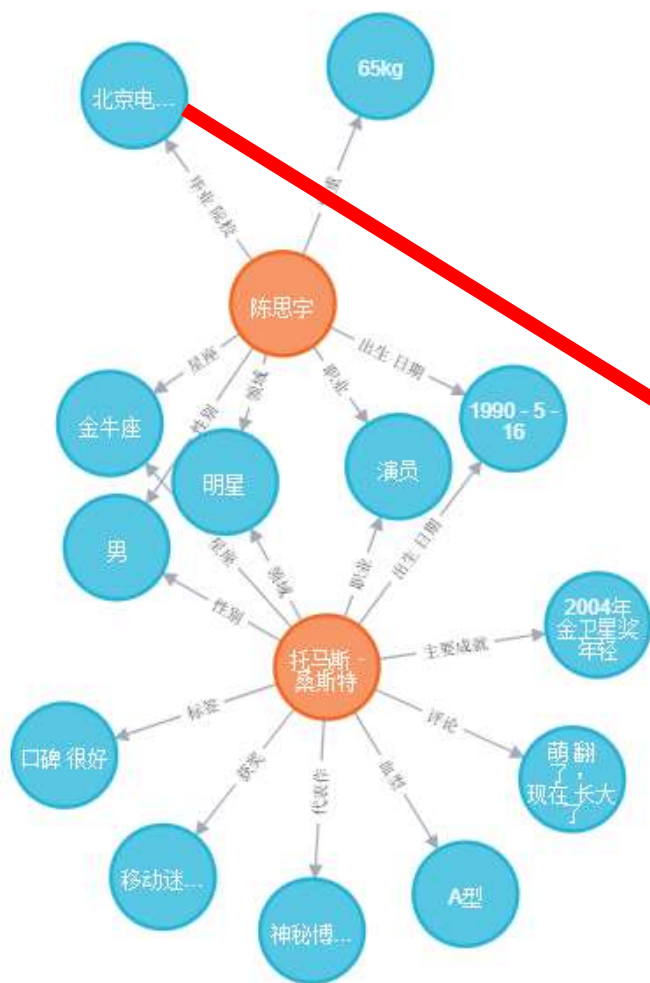
Topic:
"托马斯·桑斯特" ->
"陈思宇"

一个例子



Topic:
"托马斯·桑斯特" ->
"陈思宇"

一个例子



机器人: 知道外国有个明星长得很萌吗?

这个还真不知道呢, 请问是谁啊?

机器人: 是托马斯·桑斯特, 颜值太高了。

哦, 没听说过呢, 你能给大体说说么?

机器人: 给你大体说说, 他口碑很好的, 也很有才华, 我们国家有个小哥哥跟他一样都是1990年5月16日出生的。

是谁啊?

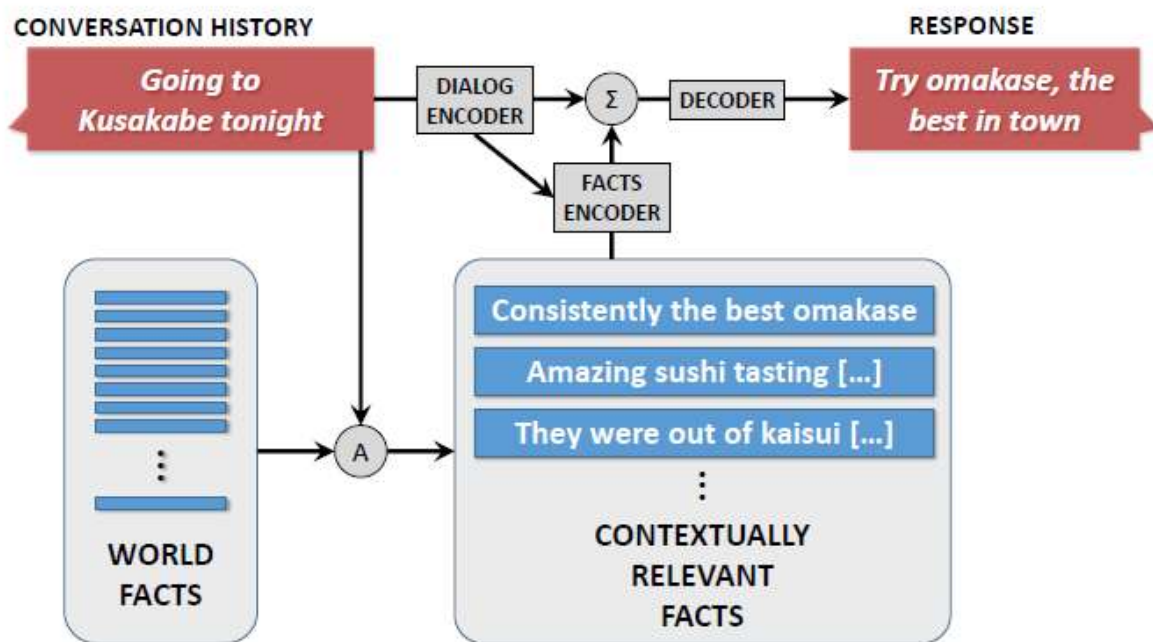
机器人: 陈思宇, 金牛座的, 毕业于北京电影学院。

有时间了解一下。

Topic:
"托马斯·桑斯特" ->
"陈思宇"

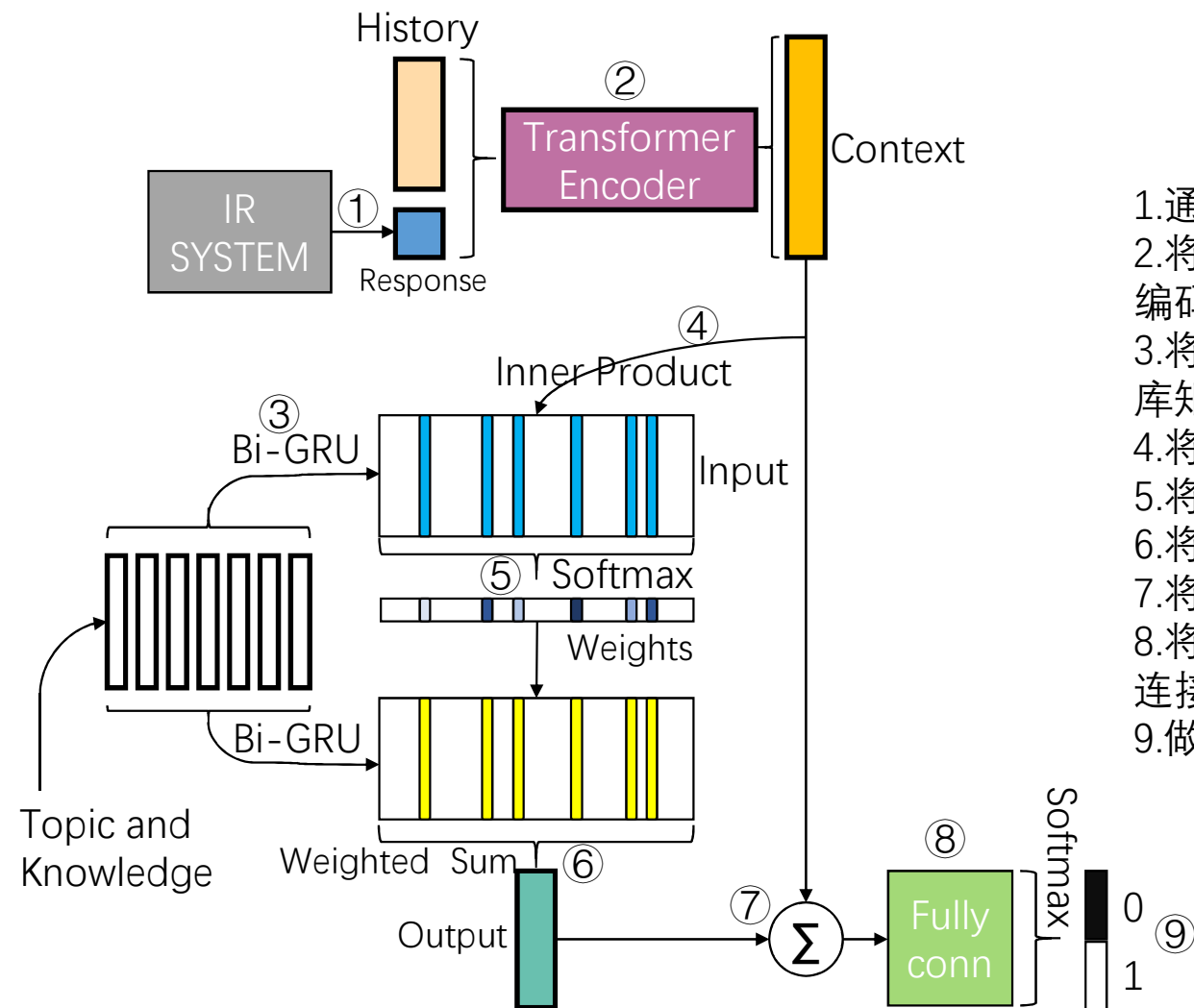
相关工作

- A Knowledge-Grounded Neural Conversation Model. (Marjan Ghazvininejad et al.2017)



- 1.使用命名实体为key生成facts的倒排表
- 2.通过命名实体识别出History的中命名实体, 以此作为查询条件, 选出相关facts
- 3.将选出的facts和History进行attention操作, 得到每个facts的权重, 最后得到facts的 weighted sum
- 4.将History和处理之后的facts一起送进 decoder, 得到输出

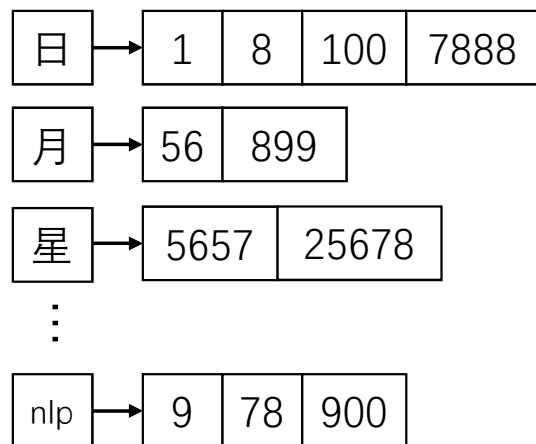
模型



- 1.通过IR SYSTEM得到可能的Response
- 2.将History和Response拼接后经过一个Transformer编码得到Context
- 3.将话题和知识库经过双向GRU编码后得到一个知识库矩阵Input
- 4.将Context和Input的每一列做内积运算
- 5.将内积运算经过一个softmax层，得到权重Weights
- 6.将Input和Weights做乘积后加和，得到Output
- 7.将Context和Output拼接
- 8.将7中的拼接结果经过一个全连接层(实验有多个全连接层)，输出为2个神经元-0或者1
- 9.做损失计算

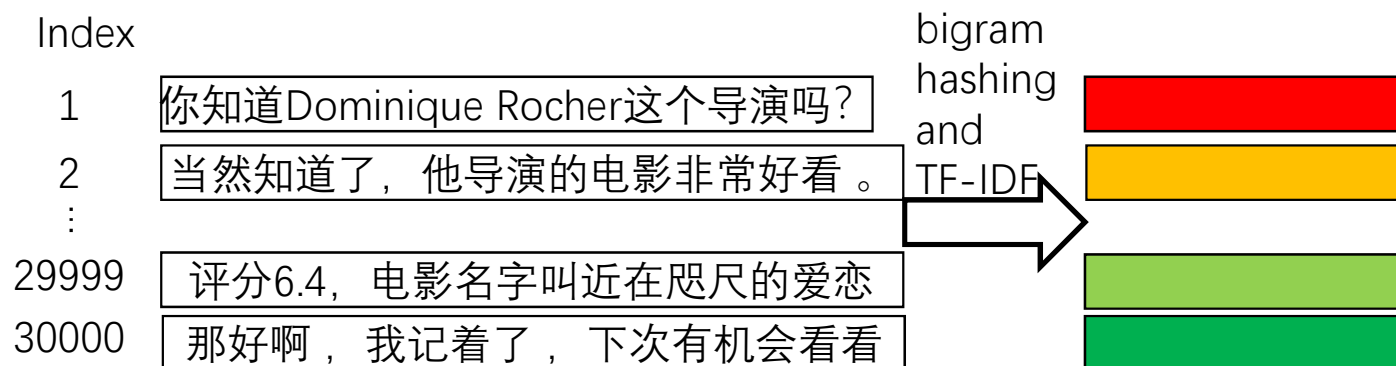
IR SYSTEM

◎以单个中文字或者英文字为key的倒排表



- 1.以Topic为key的倒排表没有画出
- 2.分别以Topic和对话历史倒数两句话为检索条件，各得到7个候选回答
- 3.第一步通过倒排表得到所有可能相似历史Candidate
- 4.计算最后一句话(如果是倒数第二句话则使用倒数第二句话)和Candidate的tf-idf相似度，取前7位

◎TD-IDF矩阵，任意相邻的两句话是一组对话



结果

	是否是最佳回答	precision	recall	F1-score
训练集	0	0.97	0.97	0.97
	1	0.05	0.05	0.05
验证集	0	0.94	1.00	0.97
	1	0.00	0.00	0.00

- 错误分析
 - 正负样本比例差距太大，训练集正负样本比例为30:1

引入Focal Loss

- $$L_{fl} = \begin{cases} -\alpha(1-y')^\gamma \log y' & , \quad y = 1 \\ -(1-\alpha)y'^\gamma \log(1-y'), & y = 0 \end{cases}$$
- 设置 $\alpha=0.96$, $\gamma=4$

	是否是最佳回答	precision	recall	F1-score
训练集	0	0.98	0.58	0.73
	1	0.05	0.64	0.09
验证集	0	0.96	0.33	0.49
	1	0.07	0.79	0.12

重采样

- 增加正样本的数量，使其比例为1:1, $\alpha=0.5$, $\gamma=2$

	是否是最佳回答	precision	recall	F1-score
训练集	0	0.82	0.67	0.74
	1	0.72	0.85	0.78
验证集	0	0.84	0.37	0.51
	1	0.60	0.93	0.73

- 错误分析
 - 增加正样本错误分类的惩罚导致大量负样本被错误分类
 - 提高正样本比例，准确率和召回率都有一定地提升，但是还是有大量的负样本被错误分类
 - 考虑在该维度空间正负样本不可分

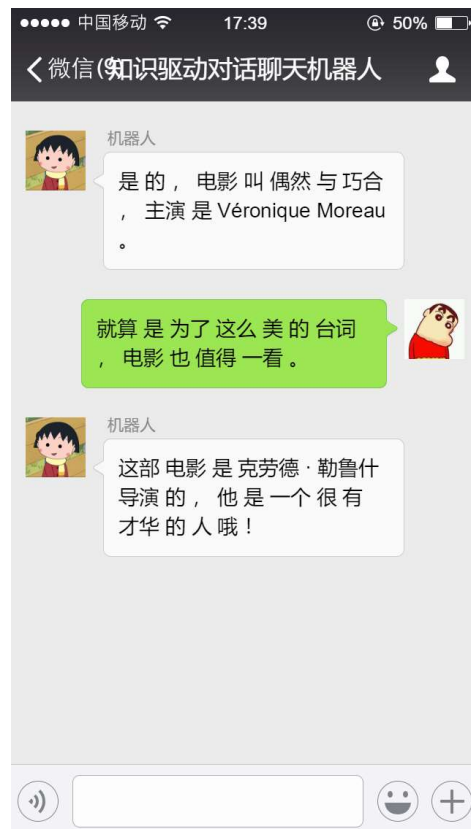
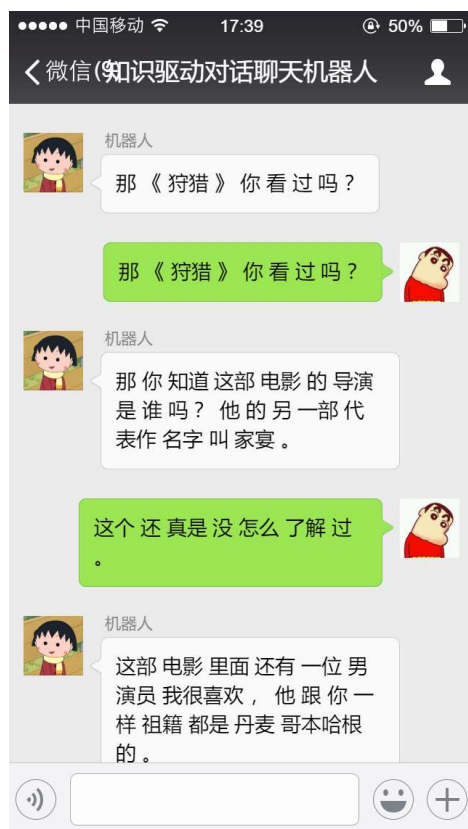
映射到高位空间

- 使用RBF神经网络—随机选取RBF中心

	是否是最佳回答	precision	recall	F1-score
训练集	0	0.97	0.98	0.97
	1	0.05	0.05	0.05

- 错误分析
 - 经过高维映射后反而效果更差，原因可能是：
 - I. 因为是随机选取RBF中心，选取的中心不合适

测试集效果



话题: "偶然与巧合", "克劳德·勒鲁什"

错误的回答, 且尝试引导到其他方向上去