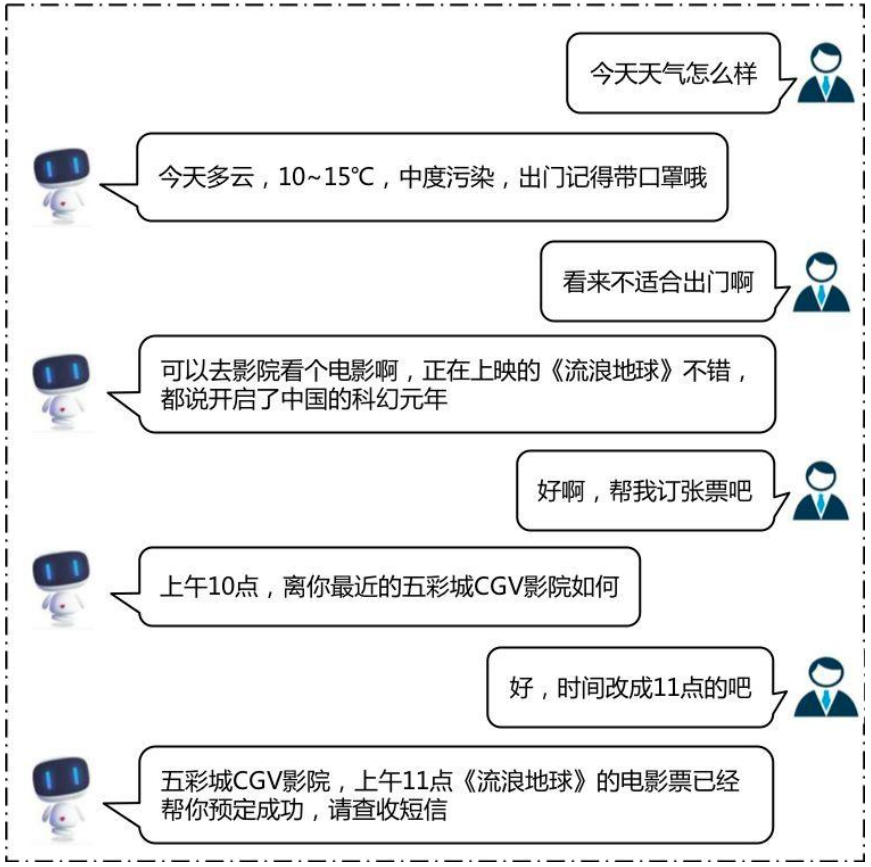
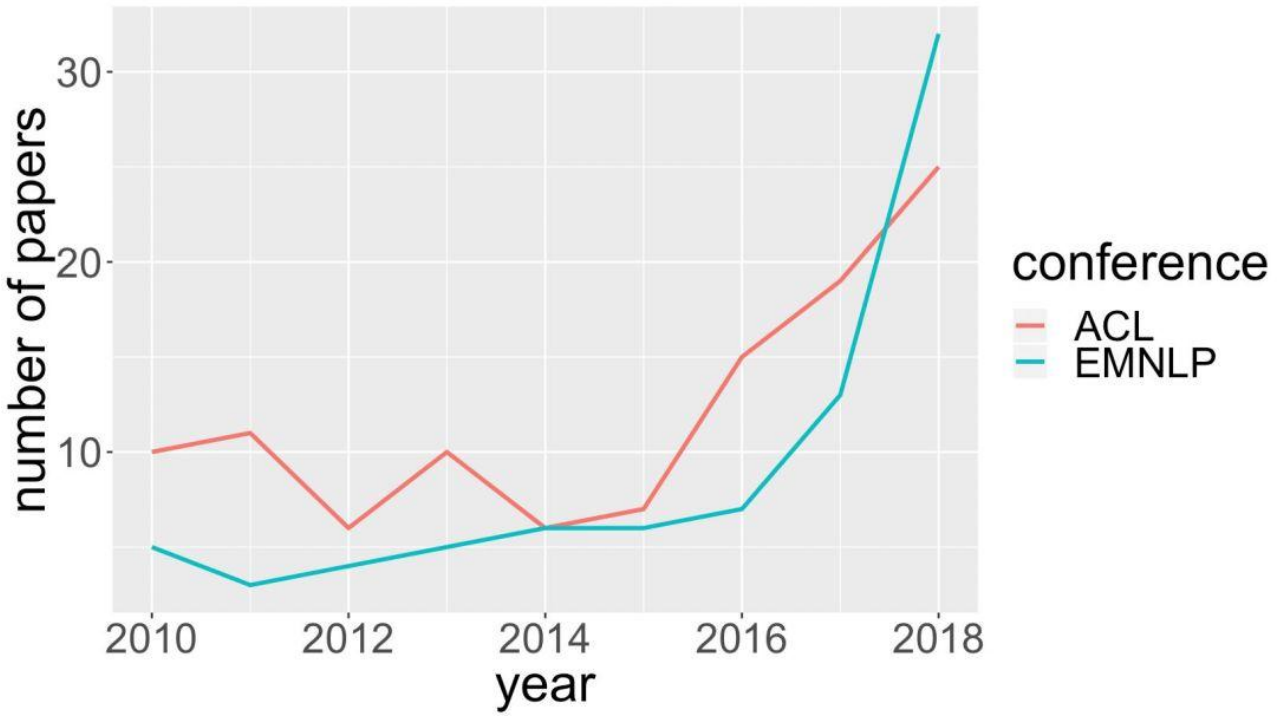
实验5

人机对话系统

前言

人机对话（Human-Machine Conversation）是指让机器理解和运用自然语言实现人机通信的技术。





前言

人机对话技术的研究最早可以追溯到上世纪六十年代，自阿兰·图灵提出通过图灵测试来检验机器是否具有人类智能的设想以来，研究人员就开始致力于人机对话系统的研究。

如何构建人机对话系统或者对话机器人呢？

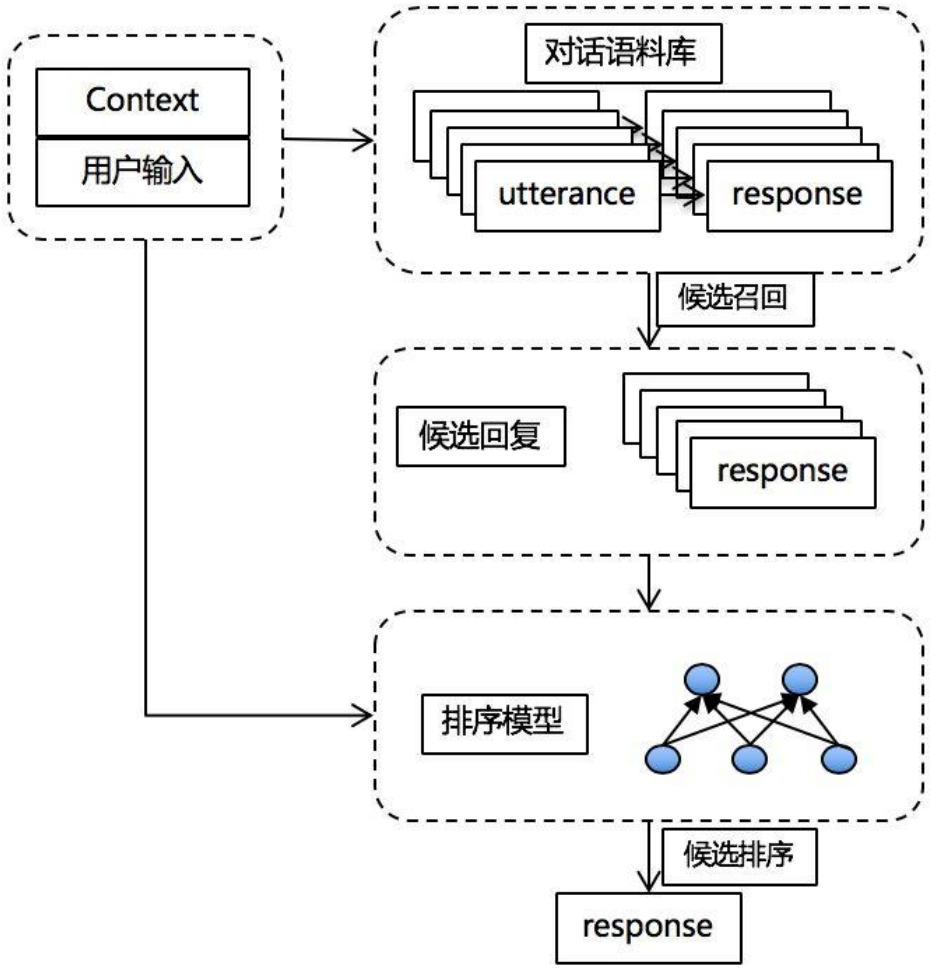
对话系统经过三代的演变：

- 规则对话系统：垂直领域可以利用模板匹配方法的匹配问句和相应的答案。优点是内部逻辑透明，易于分析调试，缺点是高度依赖专家干预， 缺少灵活性和可拓展性。
- 统计对话系统：基于部分可见马尔科夫决策过程的统计对话系统，先对问句进行贝叶斯推断，维护每轮对话状态，再跟进对话状态进行对话策略的选择， 从而生成自然语言回复。基本形成现代的对话系统框架，它避免了对专家的高度依赖，缺点是模型难以维护，可拓展性比较受限。
- **深度对话系统：基本延续了统计对话系统的框架，但各个模型采用深度网络模型。利用了深度模型强大的表征能力，语言分类和生成能力大幅提高， 缺点是需要大量标注数据才能有效训练模型。**

前言

对话系统分为三类：

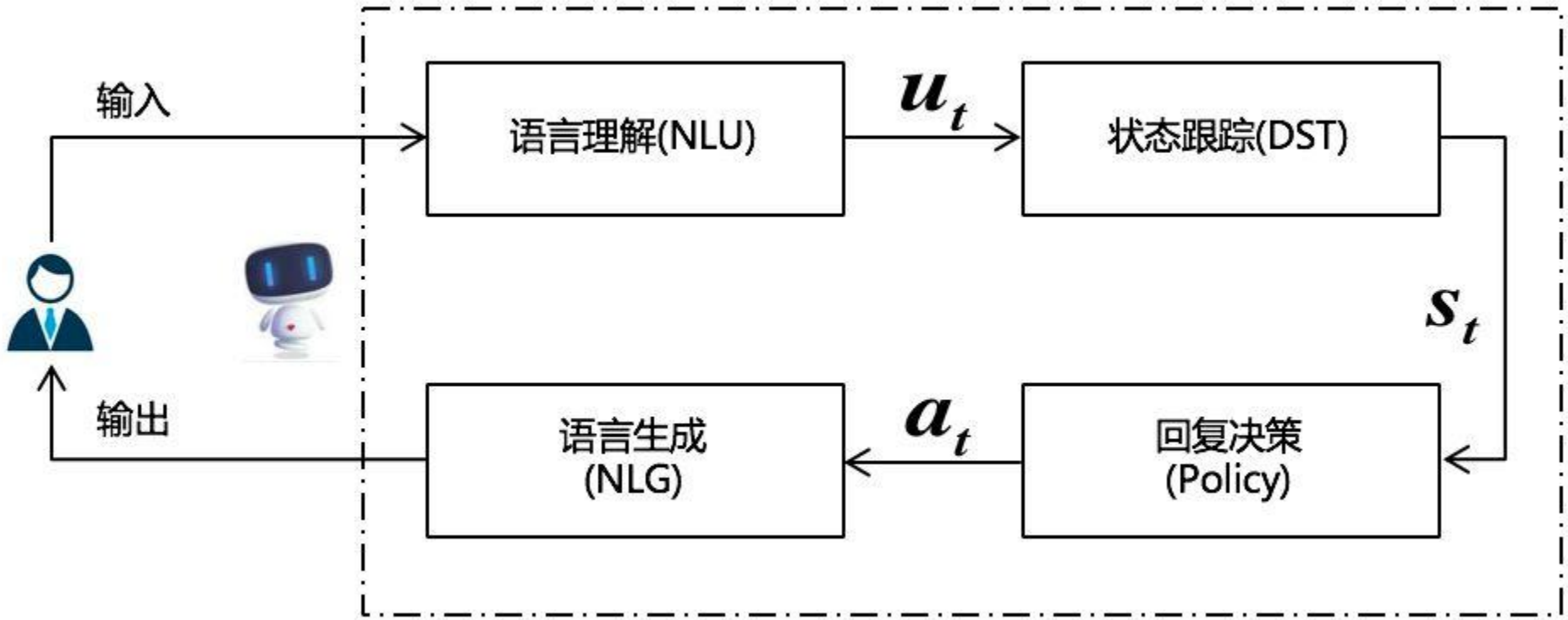
- 问答型对话：多是一问一答，用户提问，系统通过对问题解析和查找知识库返回正确答案，如搜索。



前言

对话系统分为三类：

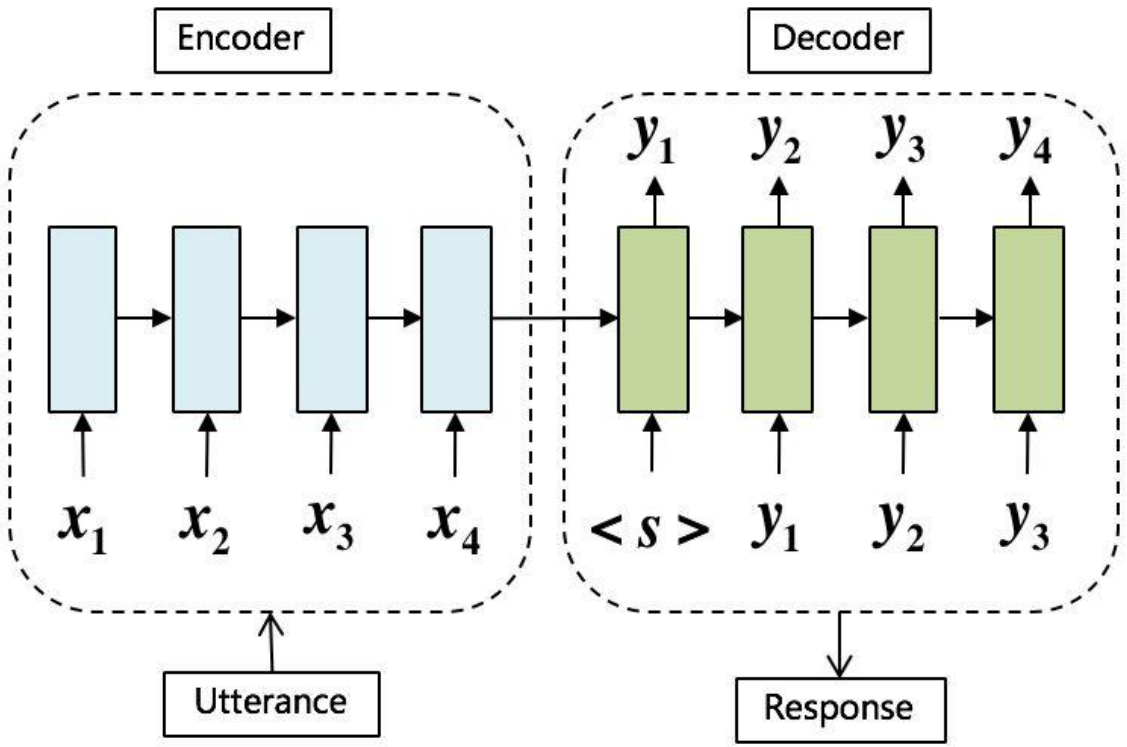
- 任务型对话（Task Oriented Dialogue Bot）用于完成用户的特定任务需求，比如电影票预订、机票预定、音乐播放等，以任务完成的成功率作为评价标准。这类对话的特点是用户需求明确，往往需要通过多轮方式解决，主流的解决方案是2013年Steve Young提出的POMDP框架，如图所示，涉及语言理解、对话状态跟踪、回复决策、语言生成等技术。



前言

对话系统分为三类：

- 聊天型对话（Generative Dialogue Bot）
  GPT2 Model
  Sequence To Sequence Model(seq2seq)
  Taobao dataset



- **实验数据**

中文闲聊语料	数据集地址	语料描述
常见中文闲聊	https://github.com/codemayq/chinese-chatbot-corpus	包含小黄鸡语料、豆瓣语料、电视剧对白语料、贴吧论坛回帖语料、微博语料、PTT八卦语料、青云语料等

中文闲聊语料的内容样例如下:

谢谢你所做的一切
你开心就好
开心
嗯因为你的心里只有学习
某某某，还有你
这个某某某用的好

你们宿舍都是这么厉害的人吗
眼睛特别搞笑这土也不好捏但就是觉得挺可爱
特别可爱啊

- 实验

问答型对话
本地检索问答： examples/ searchbot_local_demo. py
网络检索问答： examples/ searchbot_internet_demo.py

```
2024-06-28 20:49:57.292 | DEBUG    | dialogbot.search.searchbot:local_answer:65 - init_query=明天晚上能发出来吗?, filter_query=明天晚上能发出来吗
2024-06-28 20:49:57.292 | DEBUG    | dialogbot.search.searchbot:local_answer:67 - search_model=tfidf, qa_search_sim_doc=明天晚上能发出来吗, score=1.0000
2024-06-28 20:49:57.295 | DEBUG    | dialogbot.search.searchbot:local_answer:64 - --------------------
2024-06-28 20:49:57.295 | DEBUG    | dialogbot.search.searchbot:local_answer:65 - init_query=明天晚上能发出来吗?, filter_query=明天晚上能发出来吗
2024-06-28 20:49:57.296 | DEBUG    | dialogbot.search.searchbot:local_answer:67 - search_model=onehot, qa_search_sim_doc=明天晚上能发出来吗, score=1.0000
2024-06-28 20:49:57.299 | DEBUG    | dialogbot.search.searchbot:local_answer:64 - --------------------
2024-06-28 20:49:57.299 | DEBUG    | dialogbot.search.searchbot:local_answer:65 - init_query=明天晚上能发出来吗?, filter_query=明天晚上能发出来吗
2024-06-28 20:49:57.299 | DEBUG    | dialogbot.search.searchbot:local_answer:67 - search_model=bm25, qa_search_sim_doc=明天晚上能发出来吗, score=27.1936
2024-06-28 20:49:57.301 | DEBUG    | dialogbot.search.searchbot:local_answer:64 - --------------------
2024-06-28 20:49:57.301 | DEBUG    | dialogbot.search.searchbot:local_answer:65 - init_query=有5元的东西吗? 哪种口味好吃, filter_query=有元的东西吗哪种口味好吃
2024-06-28 20:49:57.301 | DEBUG    | dialogbot.search.searchbot:local_answer:67 - search_model=tfidf, qa_search_sim_doc=有4元的东西吗哪种口味好吃, score=0.9154
```

- 实验

问答型对话（Search Bot）
examples/bot_demo.py

```
from dialogbot import Bot

bot = Bot()
response = bot.answer('姚明多高呀？')
print(response)
```

```
2024-06-28 20:32:05.050 | DEBUG    | dialogbot.search.internet.search_engine:search_baidu:166 - 百度百科找到答案
2024-06-28 20:32:07.584 | DEBUG    | dialogbot.search.internet.search_engine:search_bing:218 - Bing找不到答案
2024-06-28 20:32:07.586 | DEBUG    | dialogbot.search.searchbot:local_answer:64 - ------------------
2024-06-28 20:32:07.586 | DEBUG    | dialogbot.search.searchbot:local_answer:65 - init_query=姚明多高？, filter_query=
2024-06-28 20:32:07.586 | DEBUG    | dialogbot.search.searchbot:local_answer:67 - search_model=bm25, qa_search_sim_doc=好的谢谢哦, score=0.0000
2024-06-28 20:32:07.586 | DEBUG    | dialogbot.search.searchbot:local_answer:75 - search_response=亲爱哒，还有什么小妹可以帮您呢~
{'search_response': '亲爱哒，还有什么小妹可以帮您呢~', 'gen_response': '，斤。'}
```

- 实验

聊天型对话（Generative Bot）
GPT2模型使用
基于GPT2生成模型训练的聊天型对话模型。
模型已经 release 到huggingface models：[shibing624/gpt2-dialogbot-base-chinese]

examples/genbot_demo.py

- **资源下载**

  下载代码，不要求重新训练模型，能运行examples文件下demo即可（解释一种demo 6-7分，每种对话系统增加1分）

- 完善代码　GPT2模型fine-tune（2分）

数据预处理

根目录下创建data文件夹，将原始训练语料命名为train.txt放在该目录下，train.txt每段闲聊之间间隔一行。
运行preprocess.py，对data/train.txt对话语料进行tokenize，然后进行序列化保存到data/train.pkl。train.pkl中序列化的对象的类型为List[List],记录对话列表中,每个对话包含的token。

```
cd dialogbot/gpt/
python preprocess.py --train_path data/train.txt --save_path data/train.pkl
```

训练模型

运行train.py,使用预处理后的数据，对模型进行自回归训练，模型保存在根目录下的model文件夹中。
在训练时，可以通过指定patience参数进行early stop。当patience=n时，若连续n个epoch，模型在验证集上的loss均没有下降，则进行early stop，停止训练。当patience=0时，不进行early stop。
代码中默认关闭了early stop，因为在实践中，early stop得到的模型的生成效果不一定会更好。

```
python train.py --epochs 40 --batch_size 8 --device 0,1 --train_path data/train.pkl
```

预测模型（人机交互）

运行interact.py，使用训练好的模型，进行人机交互，输入q结束对话，聊天记录将保存到sample.txt中。

```
python interact.py --no_cuda --model_dir path_to_your_model
```

执行interact.py时，可以尝试通过调整topk、topp、repetition_penalty、max_history_len等参数，调整生成的效果。更多的参数介绍，可直接看interact.py的set_args()函数中的参数说明 如果要使用GPU进行生成，则不要调用--no_cuda参数，并且通过--device gpu_id来指定使用哪块GPU。

谢谢

FOR YOUR LISTENING

陈雅妮.