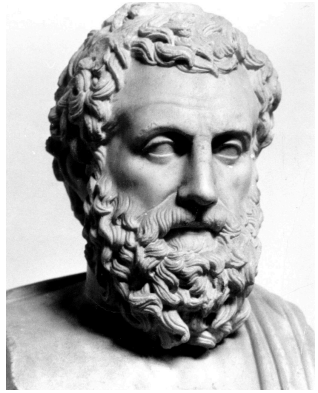
Artificial Intelligence for NLP Lesson-15 对话系统

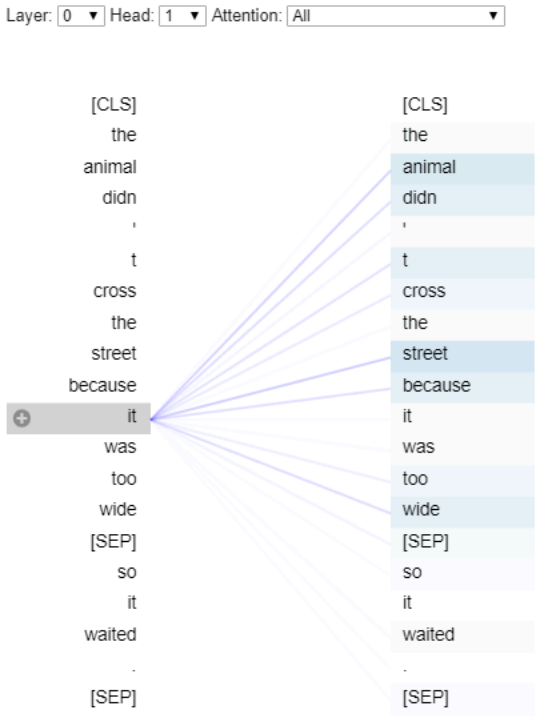


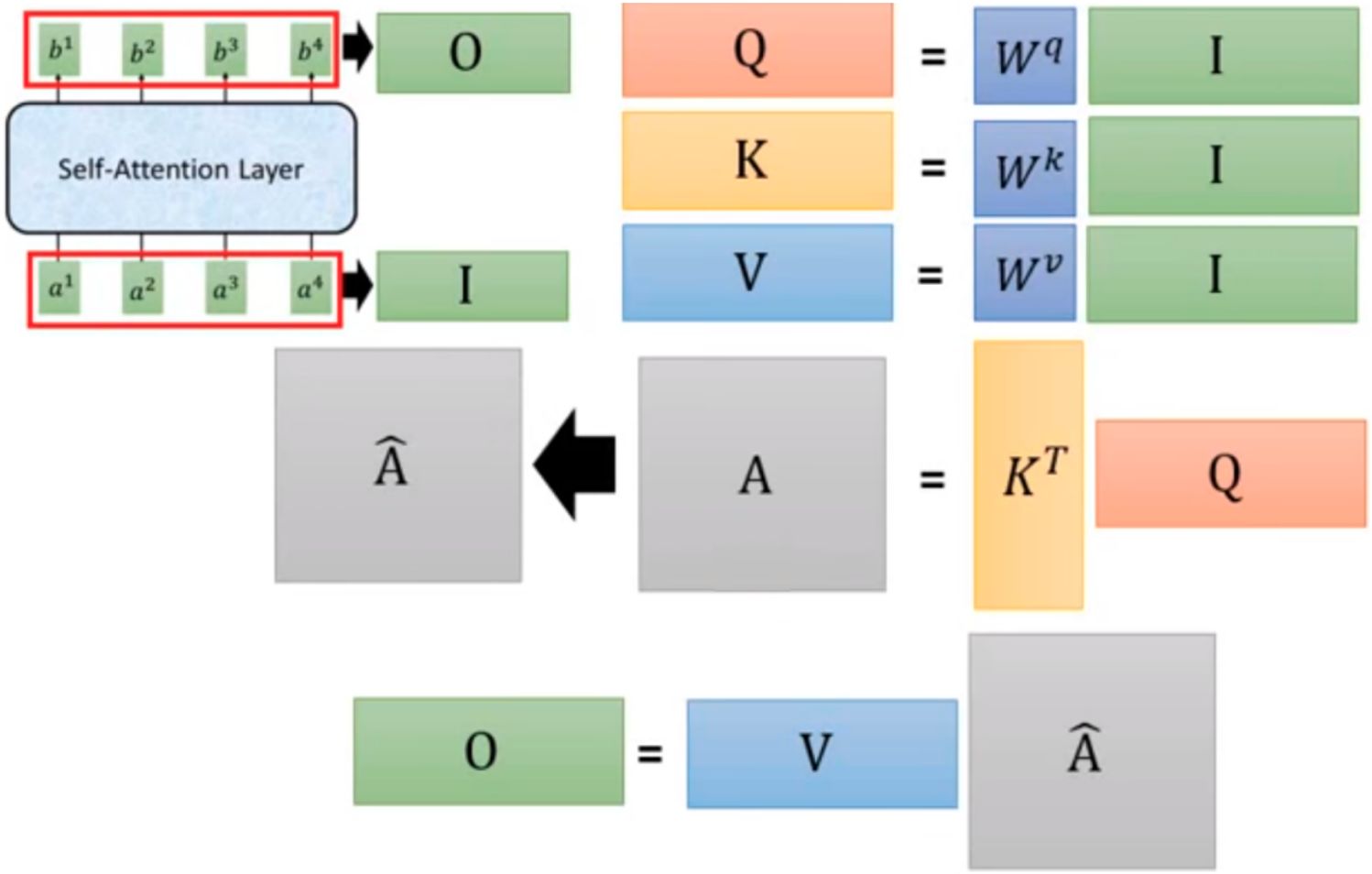
人工智能与自然语言处理课程组

2019. October. 19

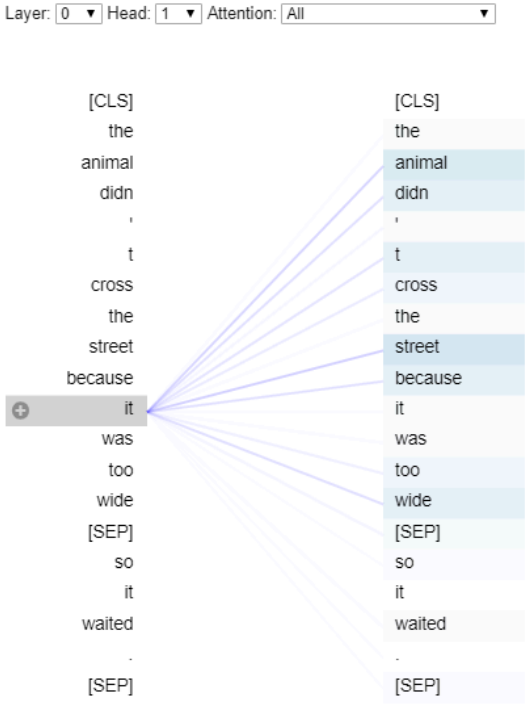
Last Course Remain

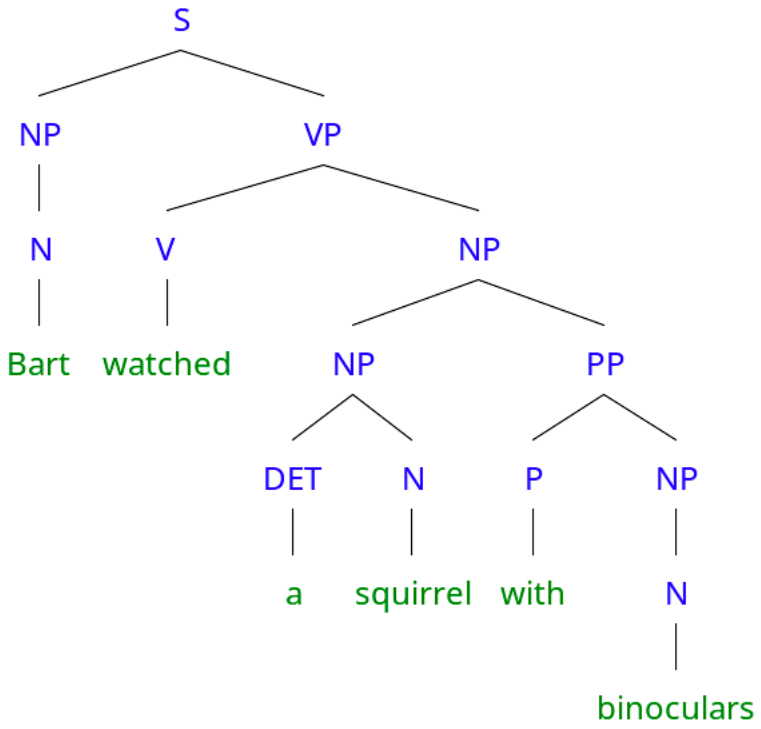
- Transformer

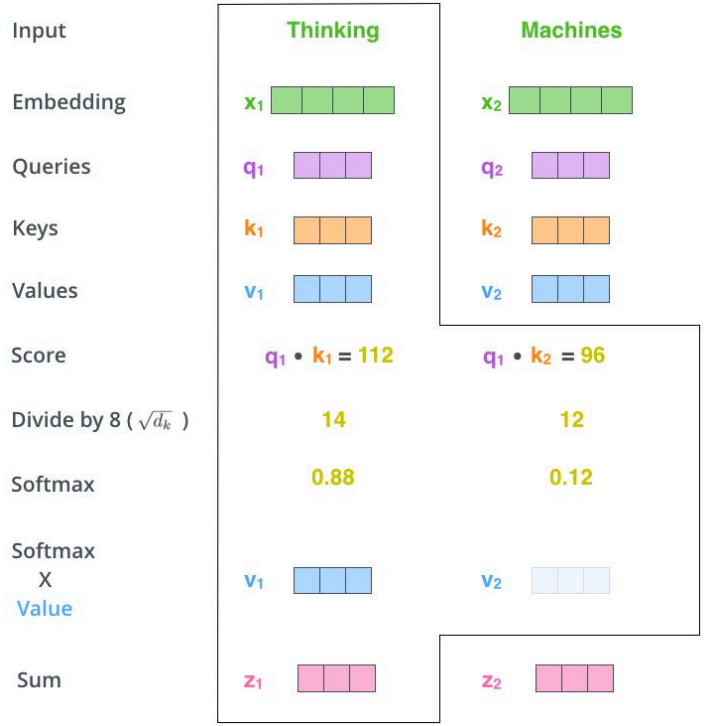




Transformer and BERT







BERT, which stands for Bidirectional Encoder Representations from Transformers

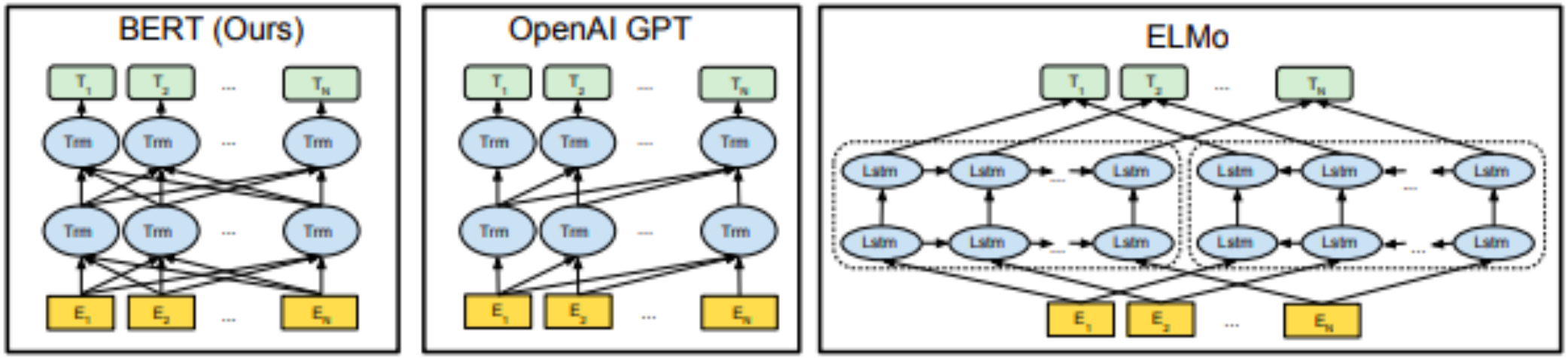
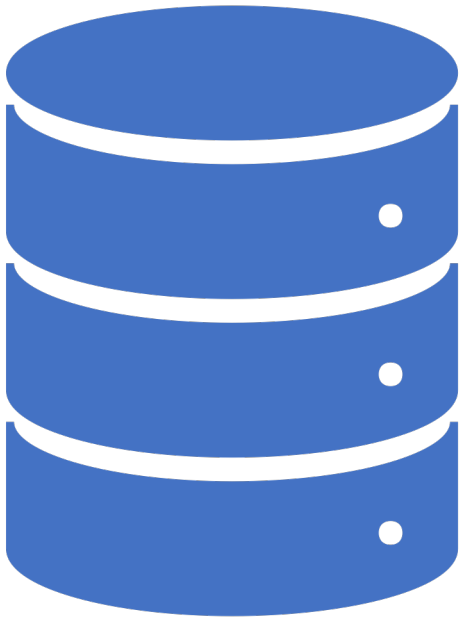


Figure 1: Differences in pre-training model architectures. BERT uses a bidirectional Transformer. OpenAI GPT uses a left-to-right Transformer. ELMo uses the concatenation of independently trained left-to-right and right-to-left LSTM to generate features for downstream tasks. Among three, only BERT representations are jointly conditioned on both left and right context in all layers.



How to use BERT as your word embedding?

- 1. Github BERT Service: https://github.com/hanxiao/bert-as-service
- 2. Fine Tuning: https://www.kaggle.com/taindow/bert-a-fine-tuning-example

Outline

1. Outline: 概览

2. General: 闲聊型

3. QA: 问答型

4.Task Oriented: 任务型

5. Bot落地应用场景

1.概览

- 你中有我，我中有你的态势

闲聊型

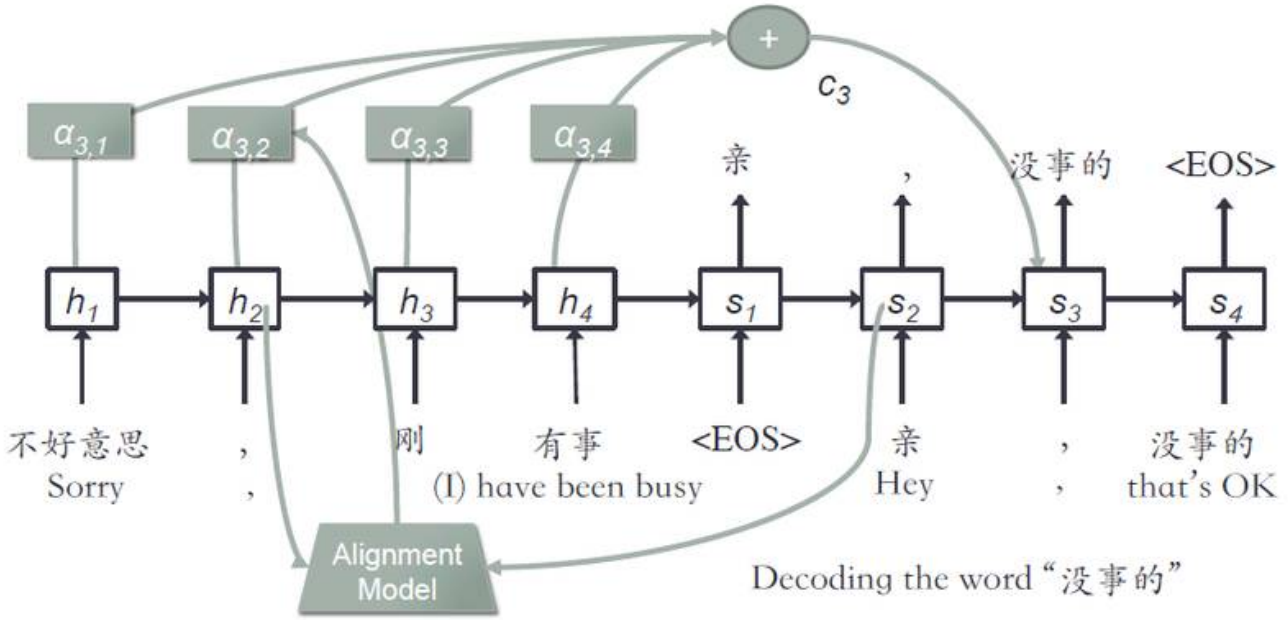
- Rule-Based System?
  - 通过事先定义的规则处理,譬如关键词、if-else,产生回复.
  - 类似于我们课程第一节: Pattern Match
  - 优点:简单、易理解,可以快速上线,适合处理小数据集
  - 缺点:规则定义需要人工参与,随着数据集的增大,规则越来越复杂,不易维护.同时在NLP层面也没有很好的语义理解.

闲聊型

- IR-Based System?
  - 通过使用传统的BOW方法(TF-IDF, BM25)来产生回复.这种方法在对话和响应之前存在一定的相关性.
  - 优点:有一定的语义表达能力,可以和Rule-Based System结合使用
  - 缺点:无法对语序结构、连贯性及深层语义进行很好的捕捉
- 问答型机器人的主流技术目前还是IR-Based System,在问答型里会详细阐述

Generation-Based System

闲聊型



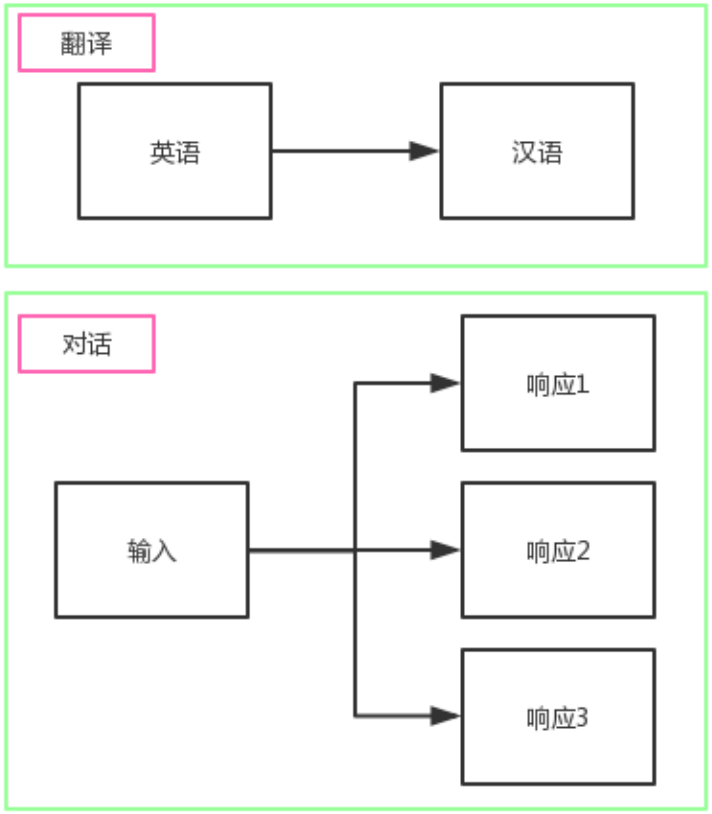
Generation-Based System?

可以近似看做一个Seq2Seq的问题,但这就万事大吉了吗?

Generation-Based System

闲聊型

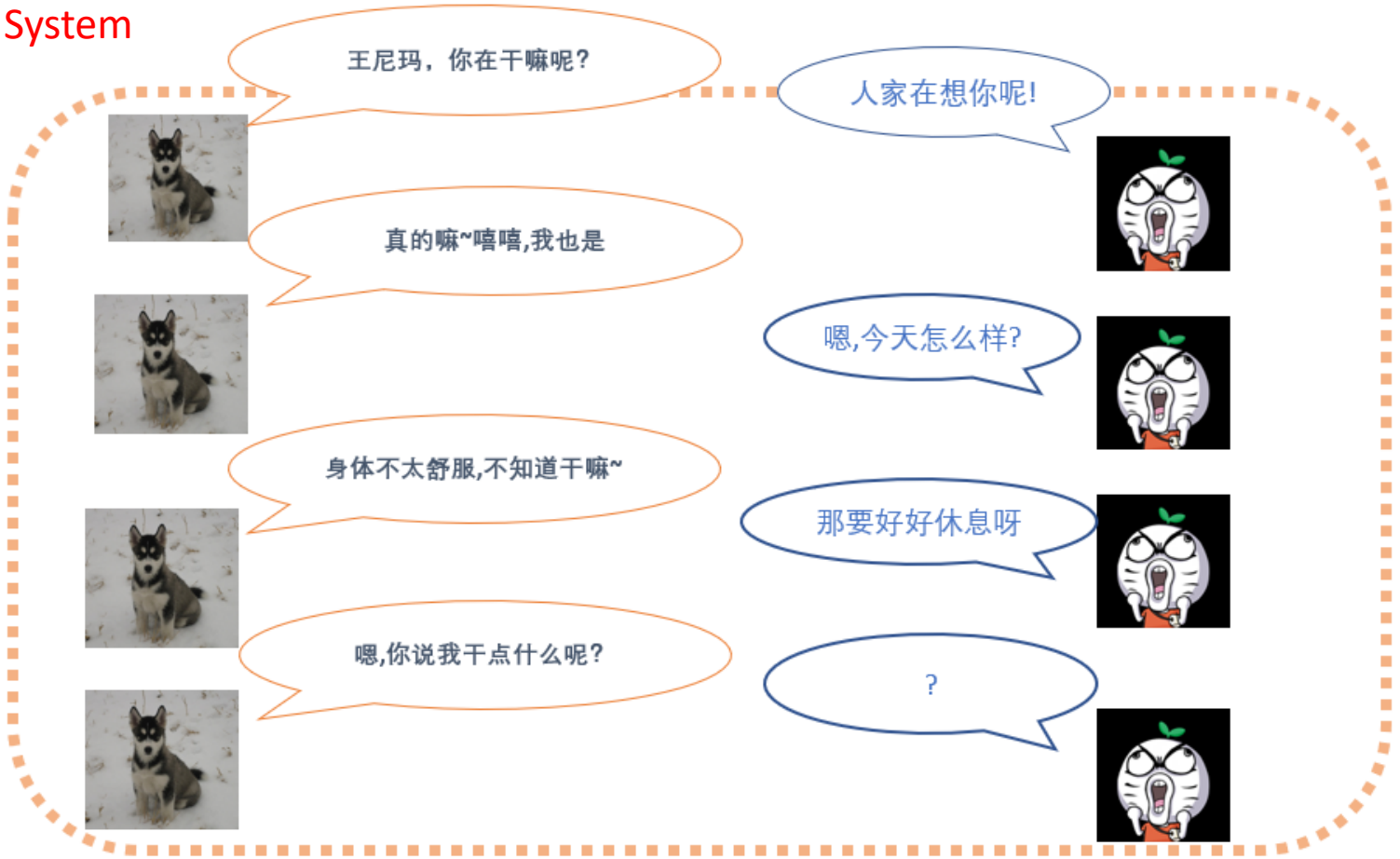
- seq2seq在翻译上的成功是因为从英语的输入，到汉语的输出是确定的,可评价的



- 但是在对话上,一个输入可能有多个响应.对于同一个响应而言,可能是汝之蜜糖,彼之砒霜.

Generation-Based System

闲聊型



在不考虑上下文信息的时候，王尼玛作为一个机器人可以回复”你可以 出去健身呀~”

但作为一个闲聊对话，可以不考虑上下文语境？生成式要怎么办？

问答型

IR-Based System

检索式对话系统可以处理问答型和闲聊型对话.

闲聊型对话一般为多轮对话,对场景上下文有依赖.

问答型对话一般为单轮场景,但从目前看也出现了多轮对话的场景

KG-Based System

以本体、实体、RDF三元组为基础构造的问答系统.

优点:具有一定的多轮、多意图识别和因果推理能力

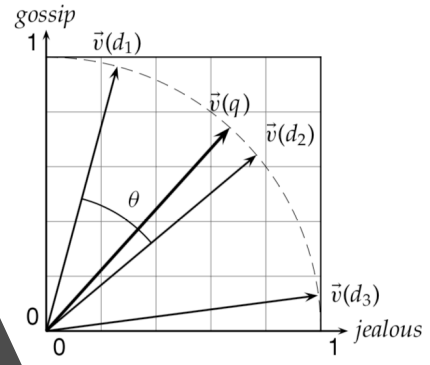
缺点:当前阶段的图谱知识系统仍旧属于专家系统,需要专家人工参与schema的构建.

(on-line coding using scikit learning)

TFIDF 以及向量相似性

Cosine similarity illustrated: $\text{sim}(d_1, d_2) = \cos\theta$.

$$\text{sim}(d_1, d_2) = \frac{\vec{V}(d_1) \cdot \vec{V}(d_2)}{|\vec{V}(d_1)||\vec{V}(d_2)|},$$



布尔搜索：Boolean Search

1. 快速检索大量文件信息

2. 灵活添加其他适配规则

3. 能够进行排序

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0
…						

.

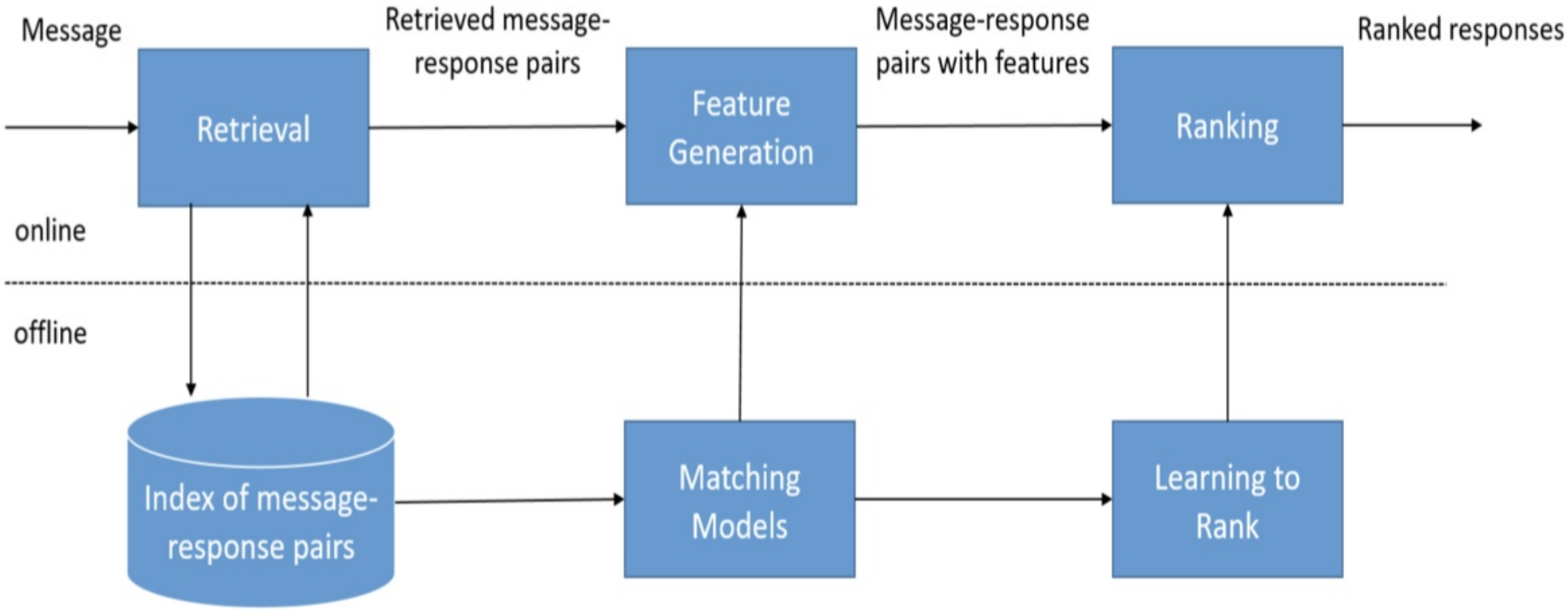
$$110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$$

Show Example

- (online code example)

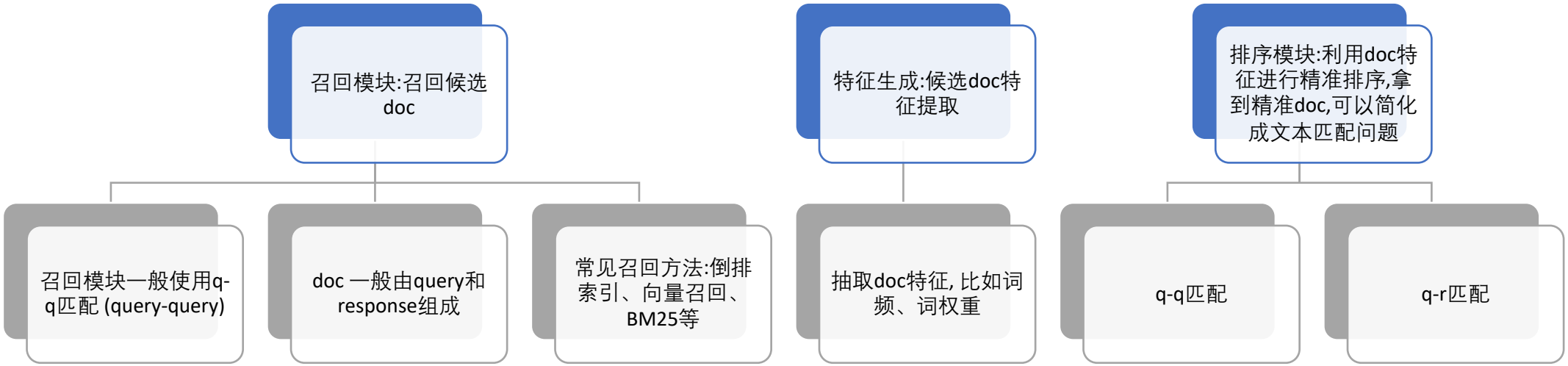
问答型

检索式系统框架图



问答型

检索式系统概述


召回模块:召回候选doc
召回模块一般使用q-q匹配 (query-query)
doc 一般由query和response组成
常见召回方法:倒排索引、向量召回、BM25等
特征生成:候选doc特征提取
抽取doc特征, 比如词频、词权重
排序模块:利用doc特征进行精准排序,拿到精准doc,可以简化成文本匹配问题
q-q匹配
q-r匹配

问答型

检索式下的多轮


王尼玛，我想问下白金卡的还款日一般是啥时候？
一般是每月15日
那紫金卡的呢？
一般是每个月20号
那黑金卡的呢？
一般是10号
好的,谢谢王尼玛
太客气了，我的粉红小可爱

问答型

从前述那个例子看, 多轮问答场景在检索式系统中该如何解决呢?

- 前述例子为卡片+还款时间的查询,如果FAQ的问答对中有query是”白金卡的还款时间是什么时候?”
- 可以对问答对进行扩写,新增”黑金卡的还款时间是什么时候?”和”紫金卡的还款时间是什么时候?”
- 问答服务需要将”那黑金的呢?”改写为”黑金卡的还款日一般是什么时候?”

问答型

检索式多轮

- FAQ问答对扩写
- 问答在线服务:上下文语义理解和指代消解
- 但是当卡片种类有上千种,类似于还款时间这样的属性有上千种呢?问答对扩写数量级就变成了百万级，这对问答对的管理是个灾难

图谱问答系统

- 上面的例子来看卡片对应着本体,白金卡、黑金卡和紫金卡对应着实体
- 还款时间、卡片额度对应着属性
- 图谱问答系统就是多轮问答场景下解决方案，天生支持多轮和推理.通过构建schema还可以解决FAQ对数据爆炸的问题

问答型

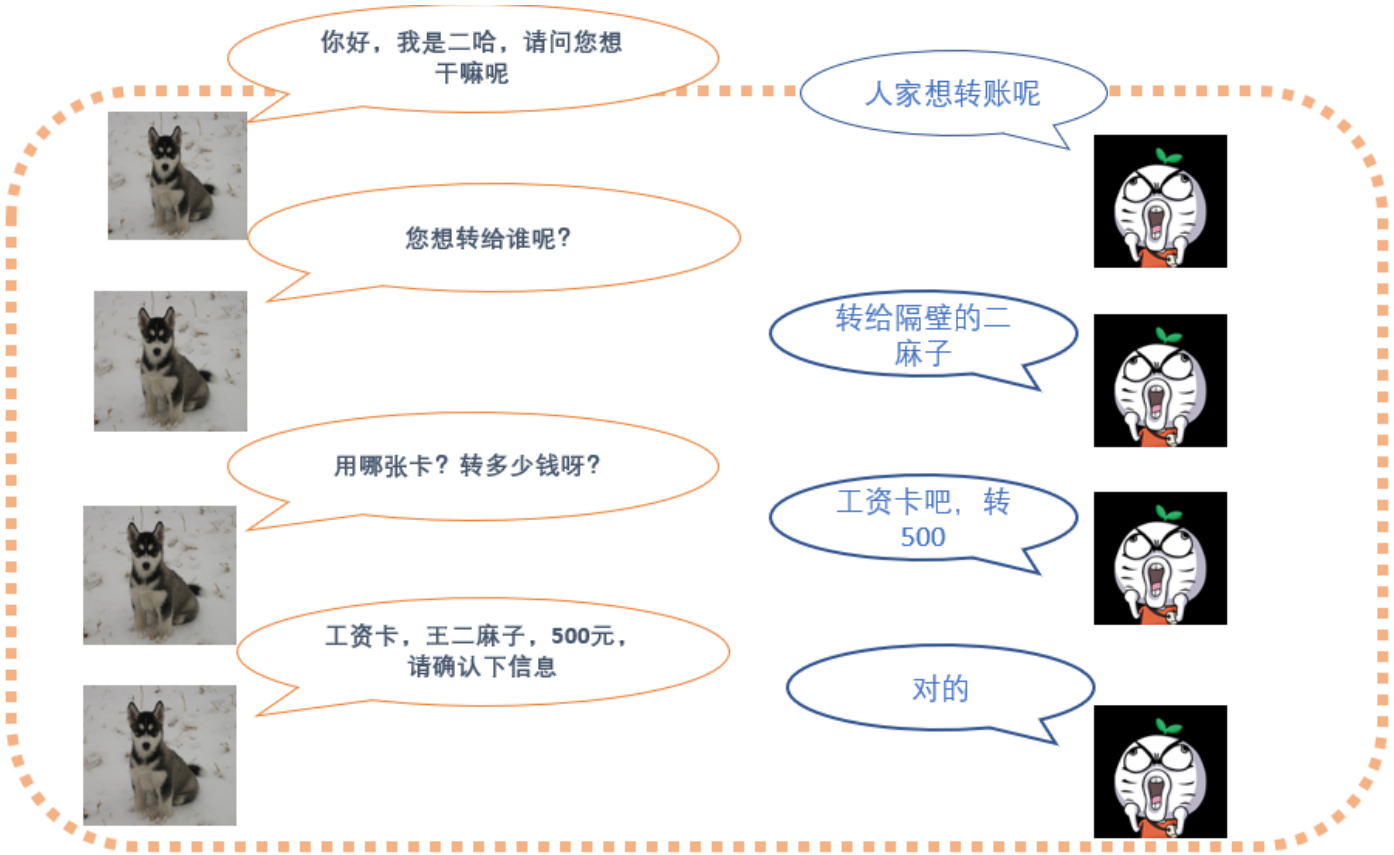
KB-QA

- Schema
- 本体、实体、属性、关系
- 详情参见知识图谱选修课程

任务型

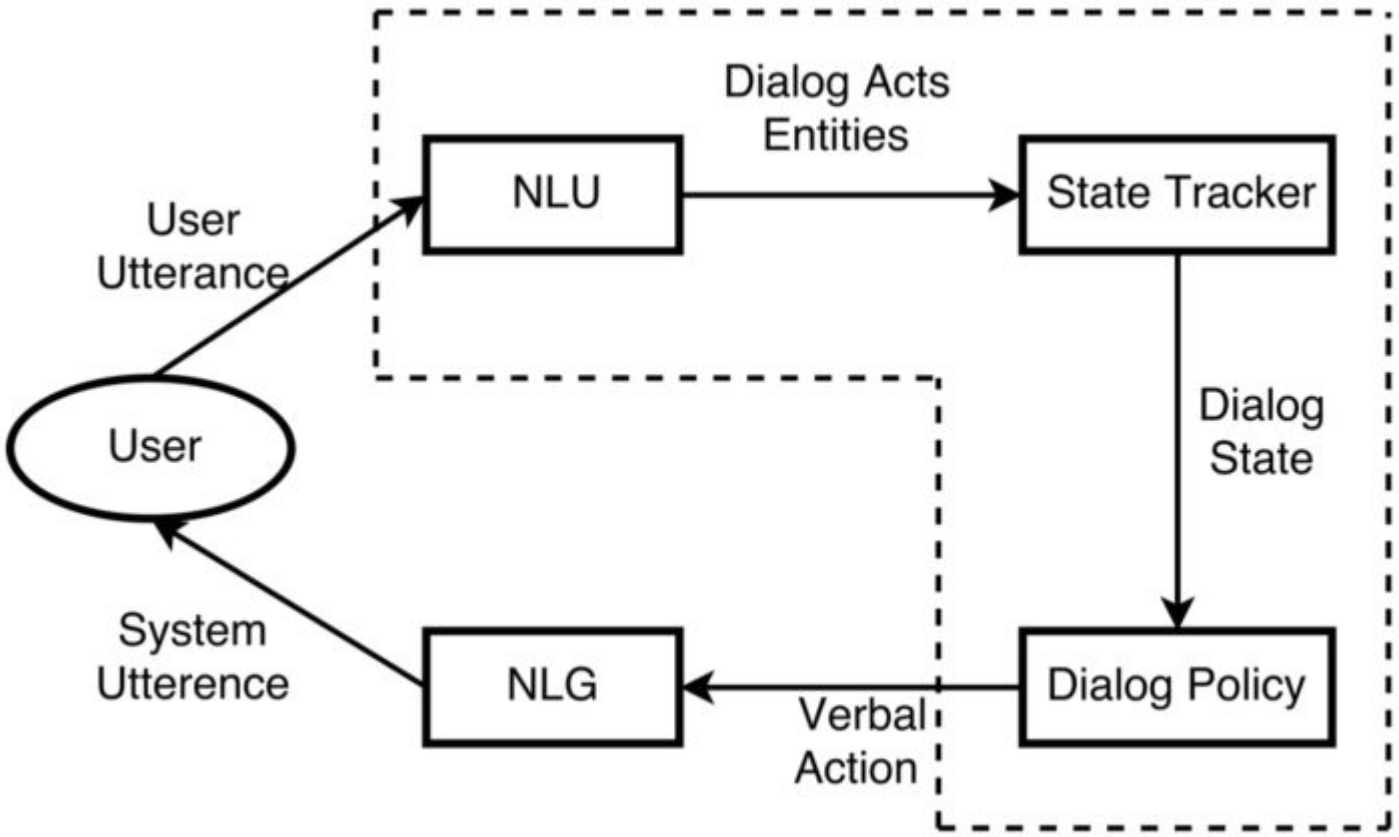
一个典型的任务型场景

意图:转账
词槽：{
人名:王二麻子
卡名:工资卡
金额:500元
}



任务型

任务型的整体框架



任务型

NLU模块

1.意图理解 (intent detection)
可以简化为文本分类任务

2.槽位填充 (slot fitting)
可以简化为序列标注任务或者多标签分类任务

当前的学术界的研究将意图理解和槽位填充联合训练
但是工业界仍旧是单独训练，且槽位填充是模型+规则混合使用.

3.领域分类(domain Identification)
这部分的存在是因为当对话系统涉及N多意图时，每新增一个意图就需要重新训练模型，会导致意图模型训练困难,系统维护困难。

Tips:国内看下百度对话团队工作

任务型

对话管理模块

1.对话状态追踪(Dialogue State Tracking)

这里的主要信息依赖于意图和槽位的填充值

主要方法是规则+统计, 其中规则适合冷启动和简单场景.

当然现在也有强化学习在这方面的探究,可以搜下Jianfeng Gao的相关工作

2.对话策略(Dialogue Policy)

根据当前的用户意图和对话状态来决定下一步动作是澄清意图,还是生成回复还是结束.

比方说:用户意图:转账 词槽:卡名+人名+金额都填充, 就引导客户输入密码完成转账动作.

目前RL在这方面的研究比较多

任务型

NLG模块

1.该模块通常使用基于模板、规则和模型的方法

2.模型的方法也可以归类为seq2seq问题

词向量 + 语法树
生成句子

N-Gram Model Formulas

- Word sequences

$$w_1^n = w_1 \ldots w_n$$

- Chain rule of probability

$$P(w_1^n) = P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)\ldots P(w_n \mid w_1^{n-1}) = \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

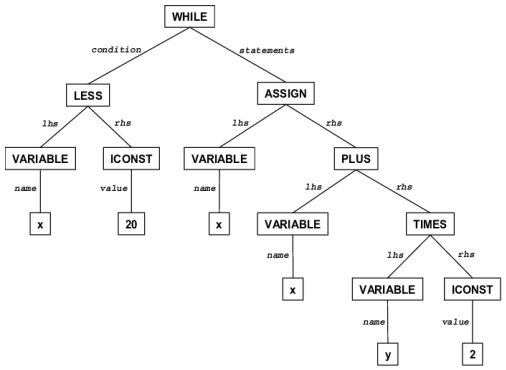
- Bigram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-1})$$

- N-gram approximation

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k \mid w_{k-N-1}^{k-1})$$

8

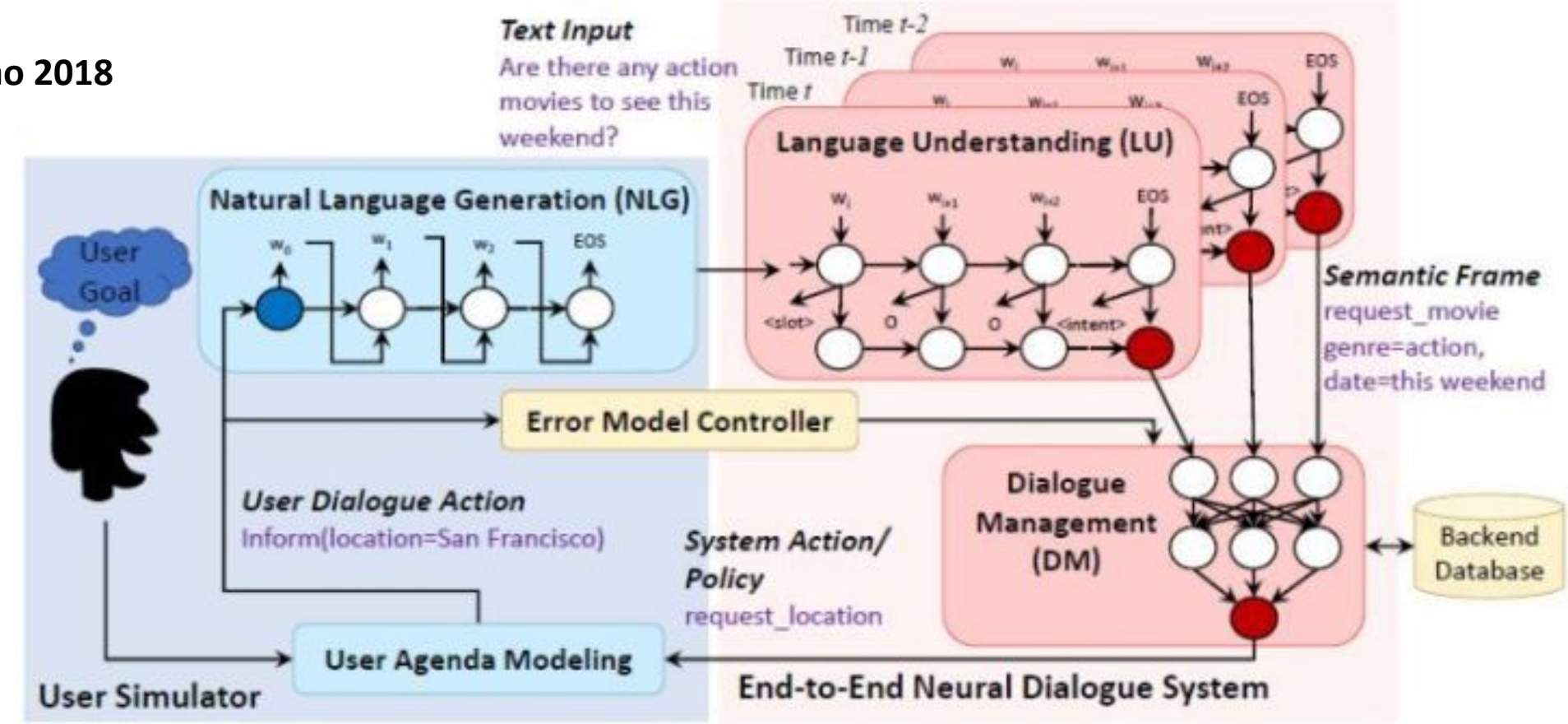


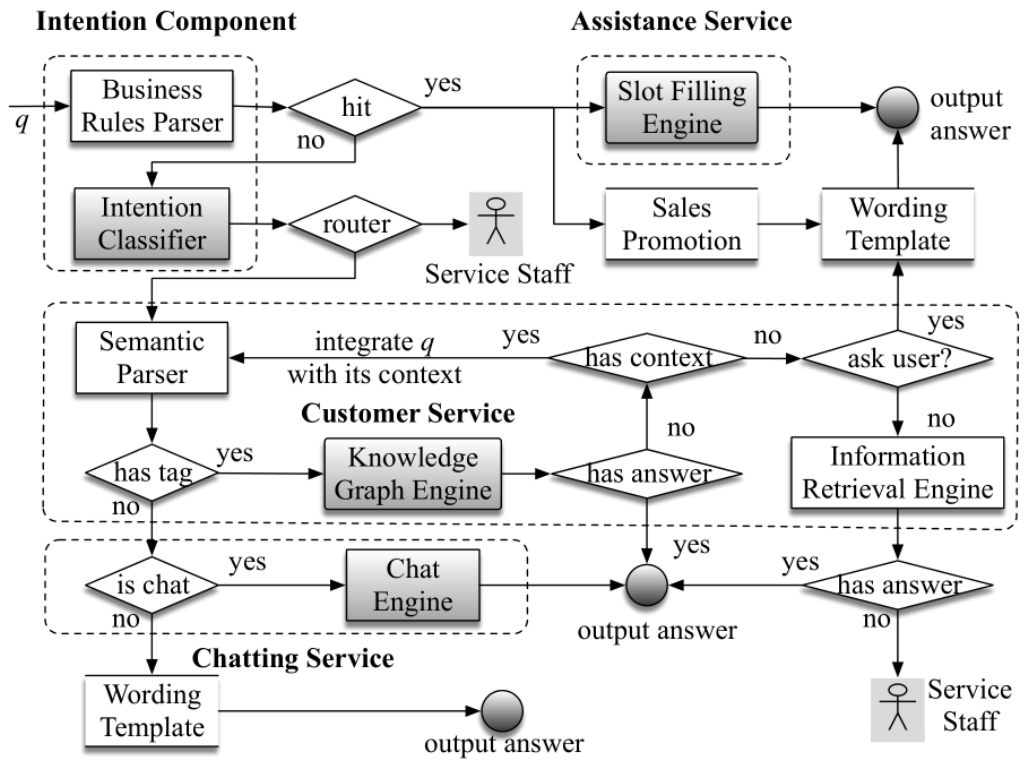

```
zh_bank_grammer = """
start => hello | ask
hello => hello_w , | None
ask => ask_w | None
ov => ov_seg
sen => start ov tail
hello_w => 你好| 您好| 不好意思 | 请问 | 给我解释 | 介绍一下 | 想知道
ask_w => 怎么 | 如何
tail => , tail_w | None | None | None
tail_w => 谢谢 | 感谢
"""
```

任务型

End2End的研究

微软 **Jianfeng Gao 2018**





Bot落地应用场景

阿里小蜜架构

- 从小蜜架构来看, 是规则和模型的混合体

- 从Bot的融合类型看，是Chat、KG和Task的结合体

- FAQ、Chat、Task单独很难形成竞争力,只有通过组合的方式才能发挥更大作用

Bot落地应用场景

阿里小蜜架构

1.小冰整体能力分为core chat 和 skills

2.core chat就是开防域聊天

3.skills就是各类技能,可以理解为Task

4. 截止18年5月有6.6亿个q-r对

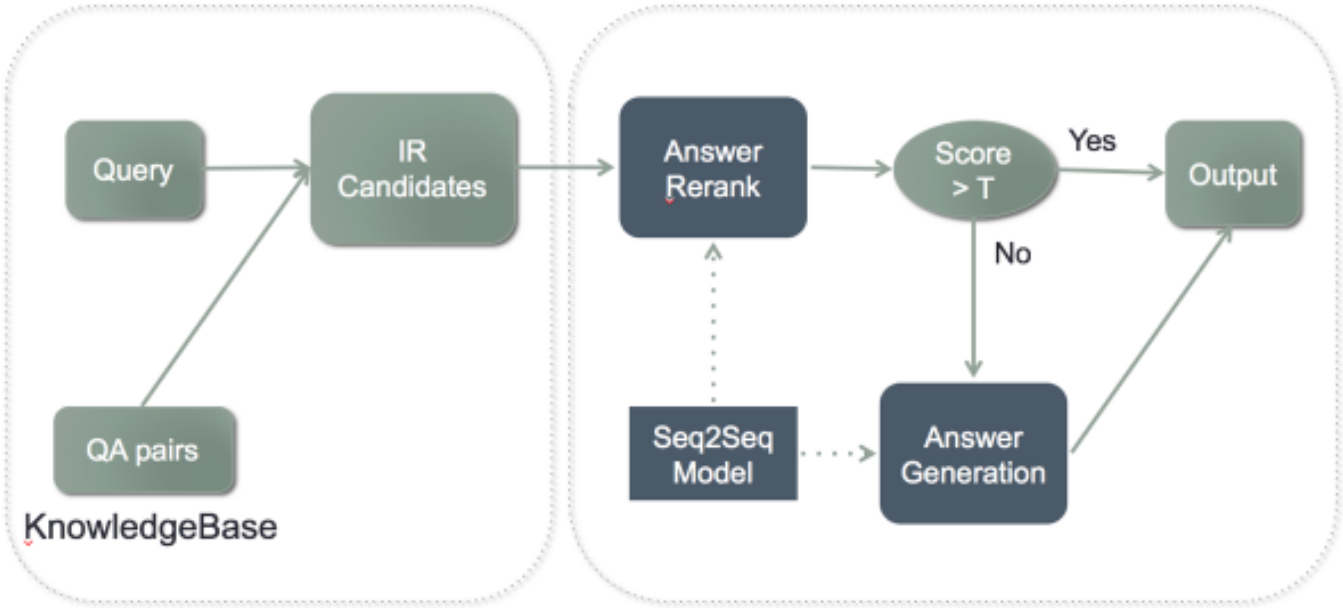
5.下图是最长使用记录

Full Duplex (voice)	Message-based Conversations		
China	China	Japan	USA
6 hours 3 minutes 8 domains 53 topics, 16 tasks	29 hours 33 minutes 7151 turns	17 hours 7 minutes 2418 turns	23 hours 43 minutes 2791 turns

Bot落地应用场景

微软小冰

检索和生成式的结合



Bot落地应用场景

阿里小蜜—Chat改进

1.百度AnyQ (检索式) https://github.com/baidu/AnyQ

2.Rasa(任务型) https://rasa.com/

Bot落地应用场景

Thanks