

自然语言处理基础



杨 猛

<https://cse.sysu.edu.cn/content/2970>

中山大学

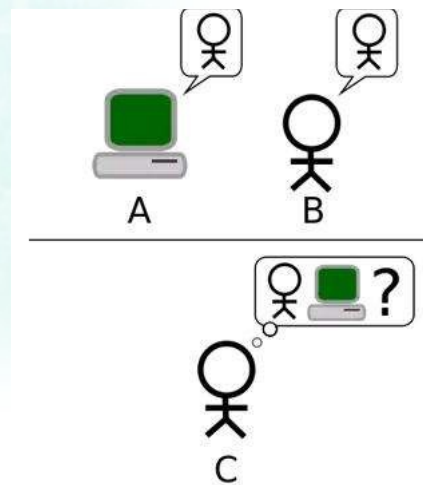
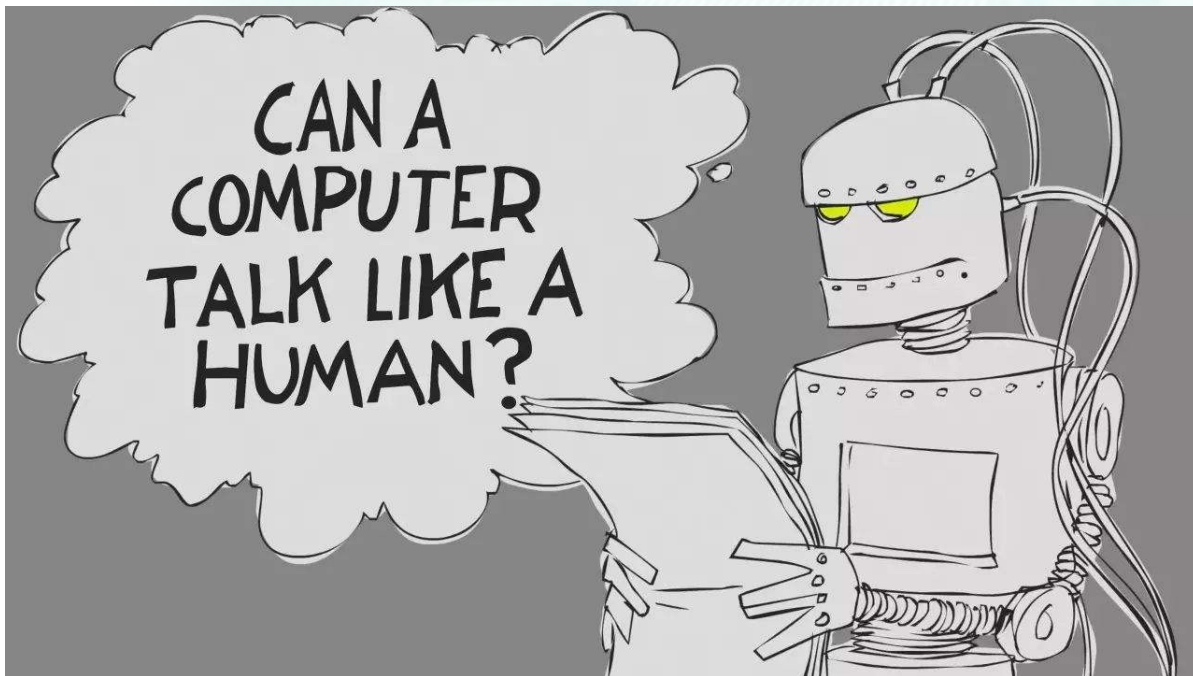
**机器智能与先进计算
教育部重点实验室**

声明：该PPT只供非商业使用，也不可视为任何出版物。由于历史原因，许多图片尚没有标注出处，如果你知道图片的出处，欢迎告诉我们 at wszheng@ieee.org.



图灵测试

- 如何知道一个系统是否具有智能？
- 1950年计算机科学家图灵提出了著名的“图灵测试”。
- 通过人和机器之间的自然语言对话来判断机器是否具有智能





目录

A NLP基础

B 对话系统

C ChatGPT

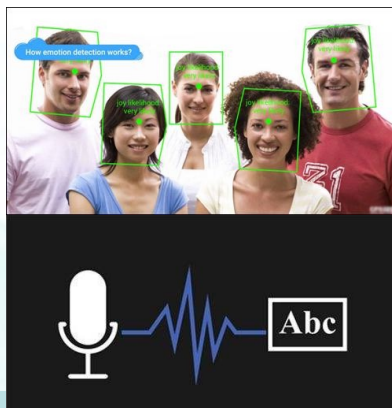


第一部分：NLP基础

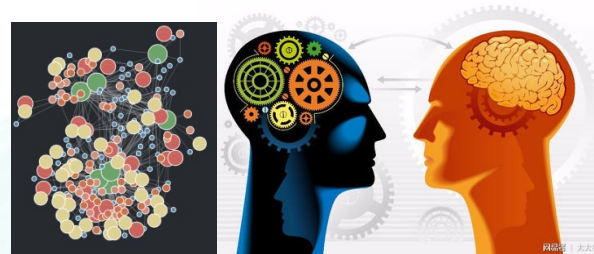
人工智能发展



运算智能



感知智能

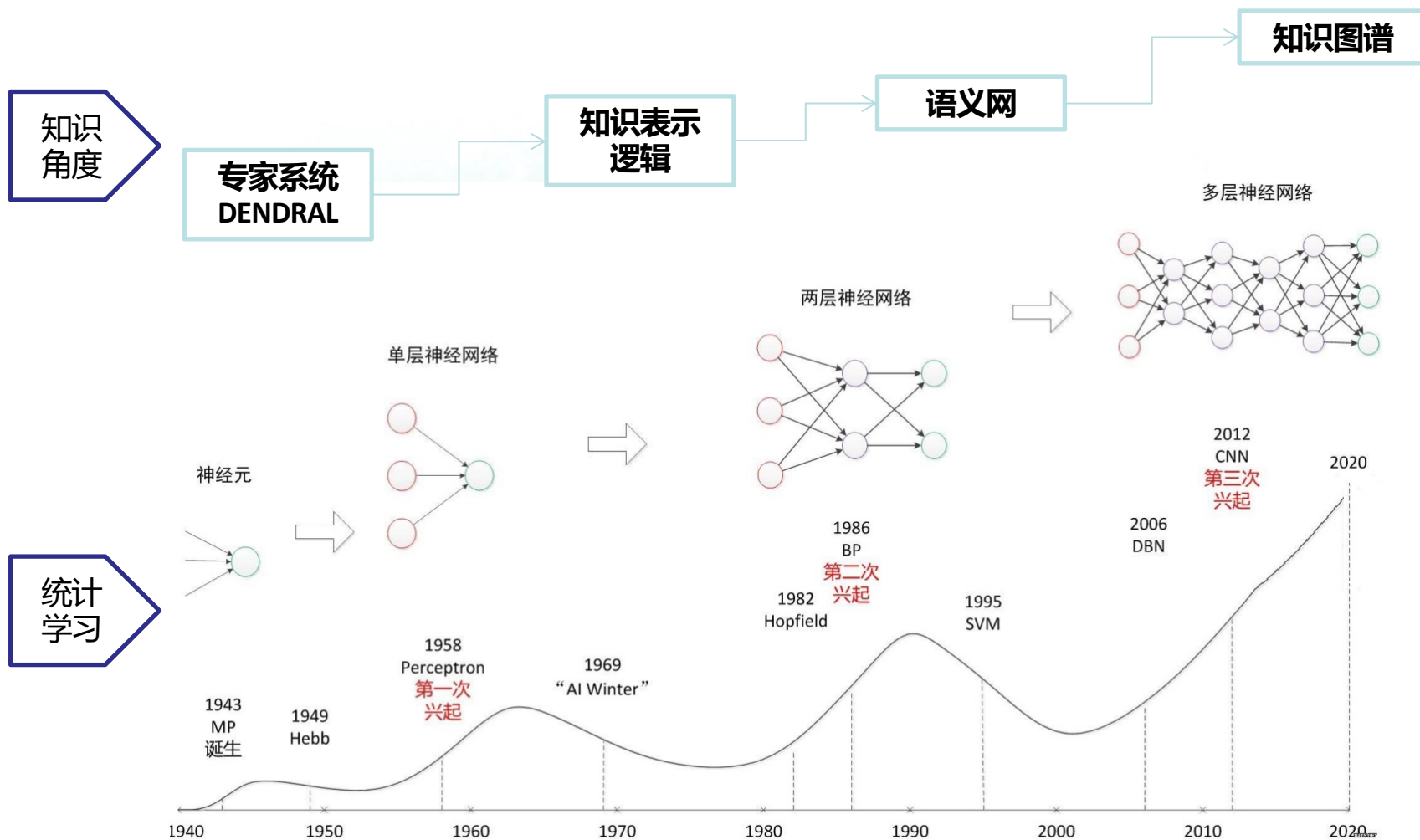


认知智能

掌握知识、进行推理

能理解会思考
知识引导+数据智能

人工智能发展



NLP是认知智能的核心



“深度学习的下一个大的进展应该是让神经网络真正理解文档的内容”

深度网络之父：Geoffrey Hinton



“如果给我10亿美金，我会用这10亿美金建造一个NASA级别的自然语言处理研究项目。”

机器学习专家、美国双院院士
Michael I. Jordan



“深度学习的下一个前沿课题是自然语言理解。”

Facebook人工智能负责人：Yann LeCun



“下一个十年，懂语言者得天下”

微软全球执行副总裁：沈向洋

刘挺，哈工大，中文信息处理前沿技术进展



什么是NLP

- **自然语言处理** (Natural Language Processing, NLP) 是指用计算机对语言信息进行处理的方法和技术。
- NLP是一门**交叉学科**。
 - 计算机科学
 - 人工智能
 - 逻辑学
- **目标**是让计算机能够处理和理解自然语言并实现一些有用的任务。
 - 智能助理 (Siri, Google Assistant, Facebook M, Cortana...)
 - 机器翻译
 - ...

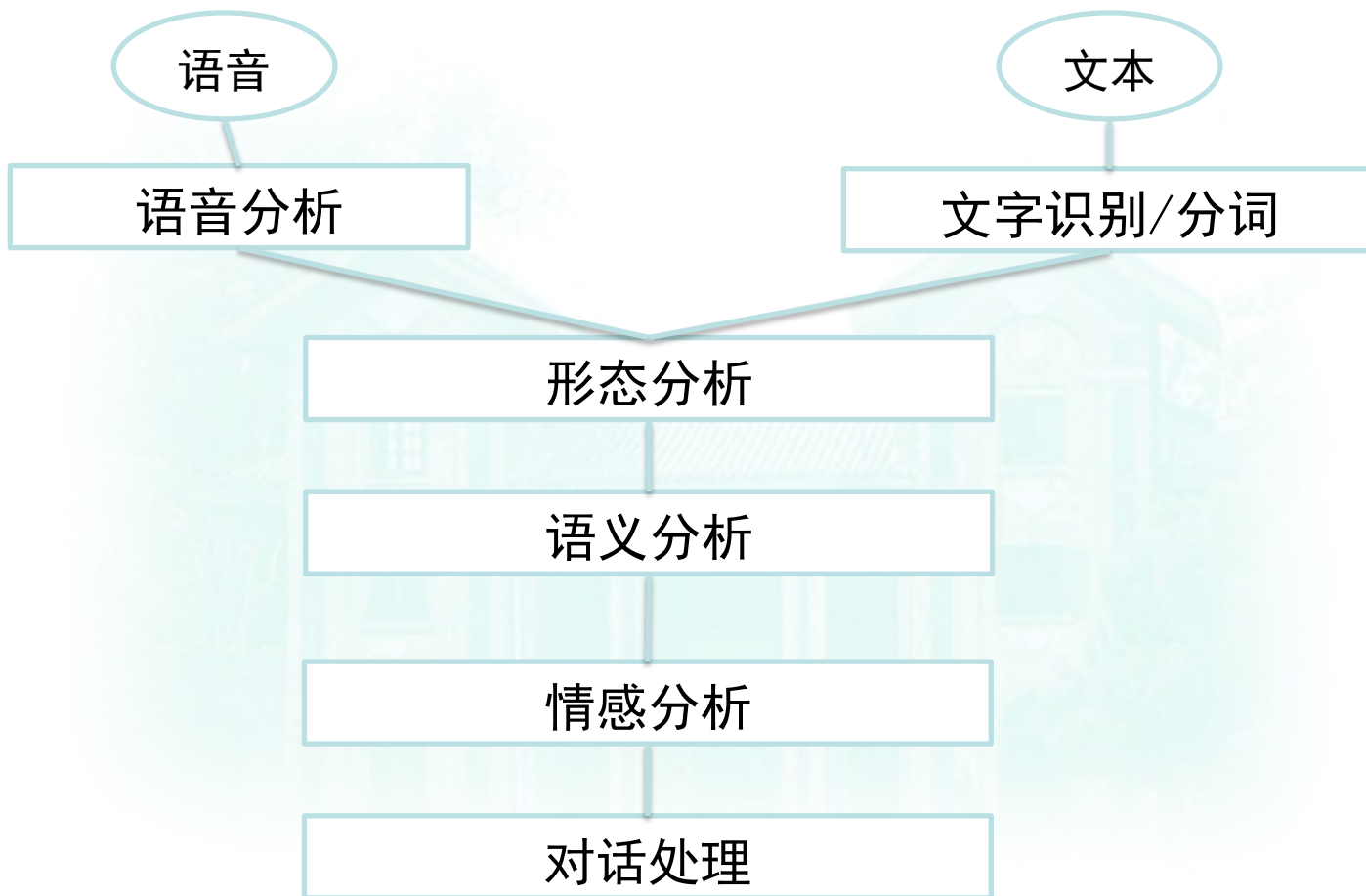


NLP发展





NLP的层次





NLP的层次

□ (文字识别) 百度OCR中文识别API:

<https://console.bce.baidu.com/ai/?fromai=1#/ai/ocr/overview/index>

9

(2) 借方与第三者发生诉讼、可能导致无力向贷方偿还贷款本息；

(3) 借方的资产总额不足抵偿其负债总额；

(4) 借方经营不善出现亏损或虚盈实亏；

(5) 借方发生重大变动，可能影响到贷款的回；

(6) 借方不按期支付利息；

(7) 借方的担保人违反或丧失担保书中规定的条件，或抵押财产发生毁损、灭失，危及贷款安全时；

(8) 其他贷方认为贷款本息收回存在风险的情况。

第十条：违约责任

1、借方不按用款计划用款，贷方有权就借方不按计划用款部份向借方收取2%的承担费。

2、借方不按合同规定用途用款，贷方有权停止发放贷款或收回贷款，并在原贷款利率基础上加收50%的罚息。

3、借方不按本合同第七条约定的每一还款计划还款，也未与贷方签订展期协议，或展期期限已到仍不能归还贷款时，贷方有权从贷款逾期之日起对逾期部分在原贷款利率基础上加收20%—50%的罚息，并有权限期或立即追回逾期贷款。

4、借方违反本合同第九条第1款的约定，贷方可要求借方支付贷款总额的10%作为违约金，造成损失的，借方还须赔偿损失。

第十一条：保险

借方应对使用本合同项下贷款购置的资产向贷方认可的保险公司投保，并将保险权益转让给贷方。

第十二条：生效

本合同经借贷双方盖章签字后生效，至本合同项下贷款本息及有关款项全部清偿时本合同自动失效。本合同一式两份，各方各执一份，具有同等法律效力。



○ (2) 借方与第三者发生诉讼、可能导致无力向贷方偿还贷款本息，(3) 借方的资产总额不足抵偿其负债总额；(4) 借方经营不善出现亏损或虚盈实亏；(5) 借方发生重大变动，可能影响到贷款的回；(6) 借方不按期支付利息；(7) 借方的担保人违反或丧失担保书中规定的条件，或抵押财产发生毁损、灭失，危及贷款安全时；(8) 其他贷方认为贷款本息收回存在风险的情况。

^第十条= 违约责任1、借方不按用款计划用款，贷方有权就借方不按计划用款部份向借方收取2%的承担费。2、借方不按合同规定用途用款，贷方有权停止发放贷款或收回贷款，并在原贷款利率基础上加收50%的罚息。3、借方不按本合同第七条约定的每六还款计划还款(拖塞与贷方签订展期协议，或展期期限已到仍不能归还贷款时，贷方有权从贷款逾期之日起对逾期部分在原贷款利率基础上加收20%—50%的罚息，并有权限期或立即追回逾期贷款。4、借方违反本合同第九条第1款的约定，贷主可要求借方支付贷款总额的10%作为违约金，造成损失的，借方还须赔偿损失。第十一条 保险 借方应对使用本合同项下贷款购置的资产向贷方认可的保险公司投保，并将保险权益转让给贷方。第十二条= 生效 本合同经借贷双方盖章签字后生效，至本合同项下贷款本息及有关款项全部清偿时本合同自动失效。本合同一式两份，各方各执一份，具有同等法律效力。 —3—



NLP的层次

- （语音识别）科大讯飞语音识别API：

<https://www.xfyun.cn/services/voicedictation>

- TTS (语音合成) <http://speech.diotek.com/en/text-to-speech-demonstration.php>

- （形态分析）NLTK工具包的词干提取：

<http://www.nltk.org/>

```
>>>from nltk.stem.porter import PorterStemmer
>>>stem = PorterStemmer()

>>>word = 'playing'
>>>stem.stem(word)

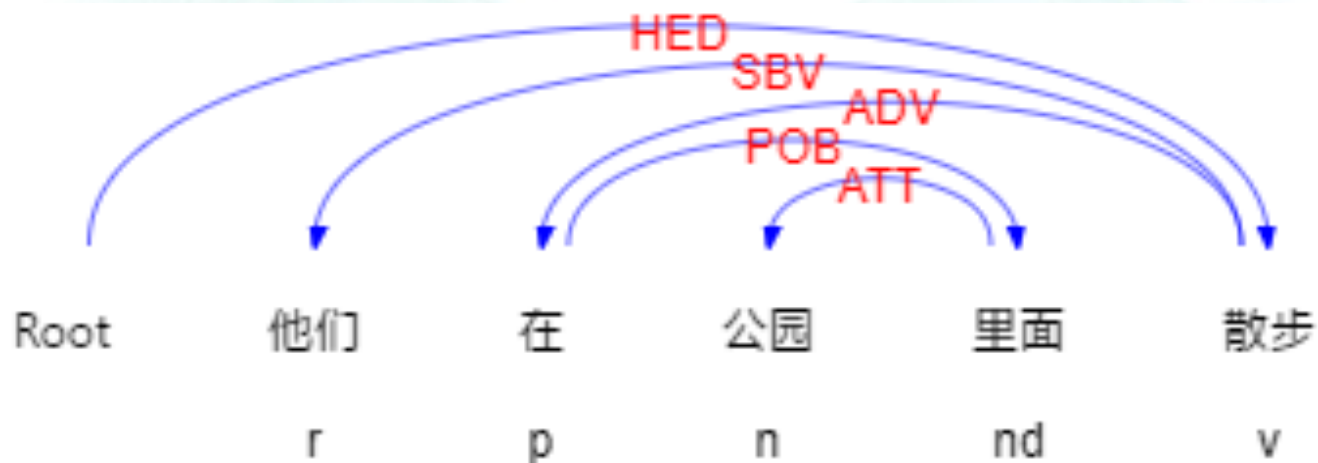
'play'
```



NLP的层次

- （分词、句法分析）哈工大NLP平台LTP:

<http://ltp.ai/>



NLP的层次

- （情感分析）讯飞的情感分析API：

<https://www.xfyun.cn/services/emotion-analysis>

我觉得这个电影不错

体验版最多100字

情感分析结果



褒义

今天天气不错，但是这家饭店服务很差

体验版最多100字

情感分析结果



贬义

NLP的层次

□ （对话处理）任务导向型（task-oriented）

对话系统（智能客服、助理）：



京东客服

没有收到货物怎么办

京东智能客服

下单前可以在产品页面查询到预计送达时效，下单后您可以点击我的订单查询物流详情哦。
❤️ 查询您账户无最近的订单记录，如已经下单成功，为了更好的给您解答，辛苦您登录下单账户后再咨询呢。

京东智能客服

您可在商品页面选择您的送货地址，查看商品预计送达时间哦~

预计送达时间查询方法

首页 品牌专区 商家 专题 促销 帮助 支付 隐私 反馈

帮助中心 > 退换货 > 物流 > 物流时效 > 预计送达时间查询方法



NLP技术的一些应用

- ❑ **拼写检查** (Word自动拼写检查)
- ❑ **机器翻译** (Google、Baidu翻译)
- ❑ **自动摘要**
 (Text rank抽取式方法、end to end的生成式方法)
- ❑ **文本分类和信息过滤**
 (特征提取的方法、深度学习的方法)
- ❑ **信息检索** (Google学术搜索)
- ❑ **信息抽取和文本挖掘**
 (机器阅读理解、LDA主题分析)
- ❑ **情感分析** (基于情感词典的方法、深度学习的方法)
- ❑ **问答系统** (各类智能客服、助理,)



NLP的主要任务

- **语言分析**：分析语言表达的结构和含义
 - 词法分析：形态还原、词性标注、命名实体识别、分词等
 - 句法分析：组块分析、结构分析、依存分析
 - 语义分析：词义、句意（逻辑关系）、上下文（指代、实体关系）
- **语言生成**：从某种内部表示生成语言表达
- **多语言处理**：语言之间的对应、转换（机器翻译、跨语言检索）
- **不同的应用对以上任务有不同需求**



NLP的主要实现方法

□ 基于规则的理性方法

- 基于规则的知识表示和推理（符号计算）
- 强调人对语言知识的理性整理（知识工程）
- 受计算语言学理论指导

□ 基于语料库的经验方法

- 以大规模语料库为语言知识基础
- 利用统计学习和基于神经网络的深度学习自动获取和运用隐含在语料库中的知识
- 学习到的知识体现为一系列模型参数

□ 混合方法

- 理性方法处理效率比较高，但是鲁棒性差、知识扩充困难
- 经验方法鲁棒性较好，但是缺乏对语言学知识的深入描述和应用、处理效率低
- 结合两种方法的优点



NLP的难点

- 歧义
 - 有限的词汇和规则表达复杂、多样的对象
- 语言知识的表示、获取和运用
- 成语和惯用语的处理
- 对语言的灵活性和动态性的处理
 - 同一个意图的不同表达，包括包含错误语法的惯用语
 - 语言在不断地变化，新词的出现
- 上下文和世界知识（常识）的利用和处理



词向量

- ❑ 为了使计算机能够处理自然语言符号，引入词向量的方法。
- ❑ **one-hot表示**：把每个词表示为一个长向量。这个向量的维度是词表大小，向量中只有一个维度的值为1，其余维度为0，这个维度就代表了当前的词。

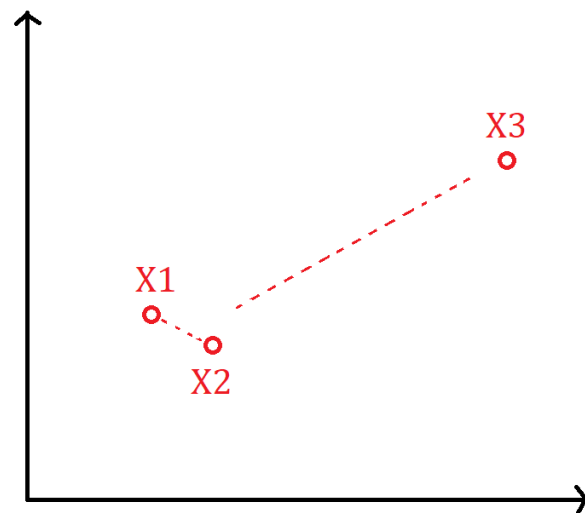
苹果	->	(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ...)
香蕉	->	(0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, ...)

- ❑ one-hot表示相当于给每个词分配一个ID，这就导致这种表示方式不能展示词与词之间的关系。并且有维度过高的问题。

词向量

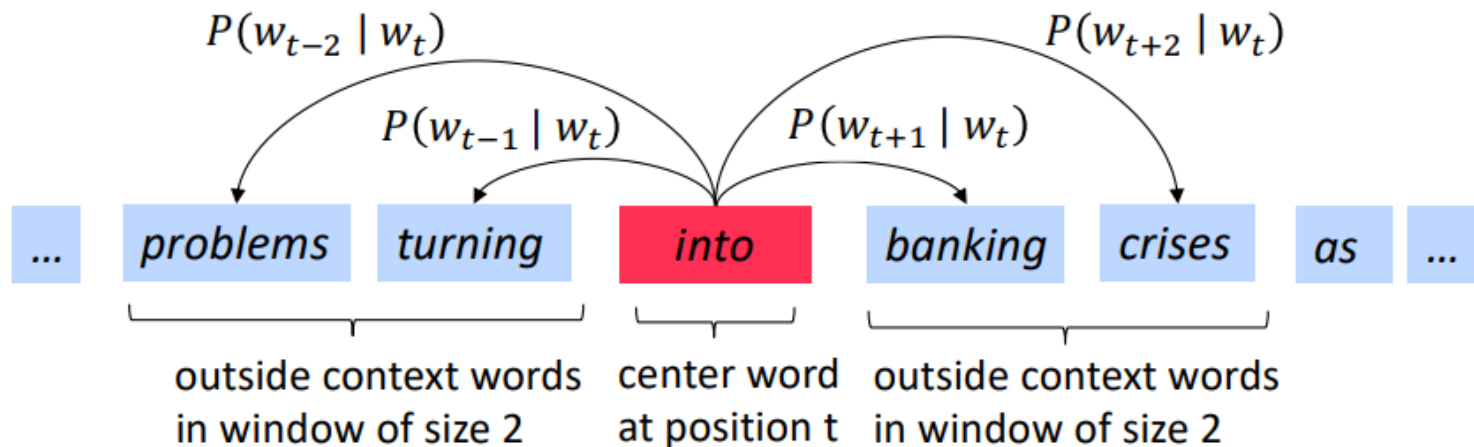
- ❑ **分布式表示**：将词表示成一个定长的连续的稠密向量。
- ❑ **Word2Vec**是流行的分布式表示方法，它的思想是利用中心词来预测上下文（skip-gram）或者利用上下文来预测中心词（CBOW）。
- ❑ 分布式表示很好地解决了维度过高的问题，并且可以一定程度上反映词语之间的关联。

苹果 $\rightarrow x_1$
香蕉 $\rightarrow x_2$
风扇 $\rightarrow x_3$



词向量

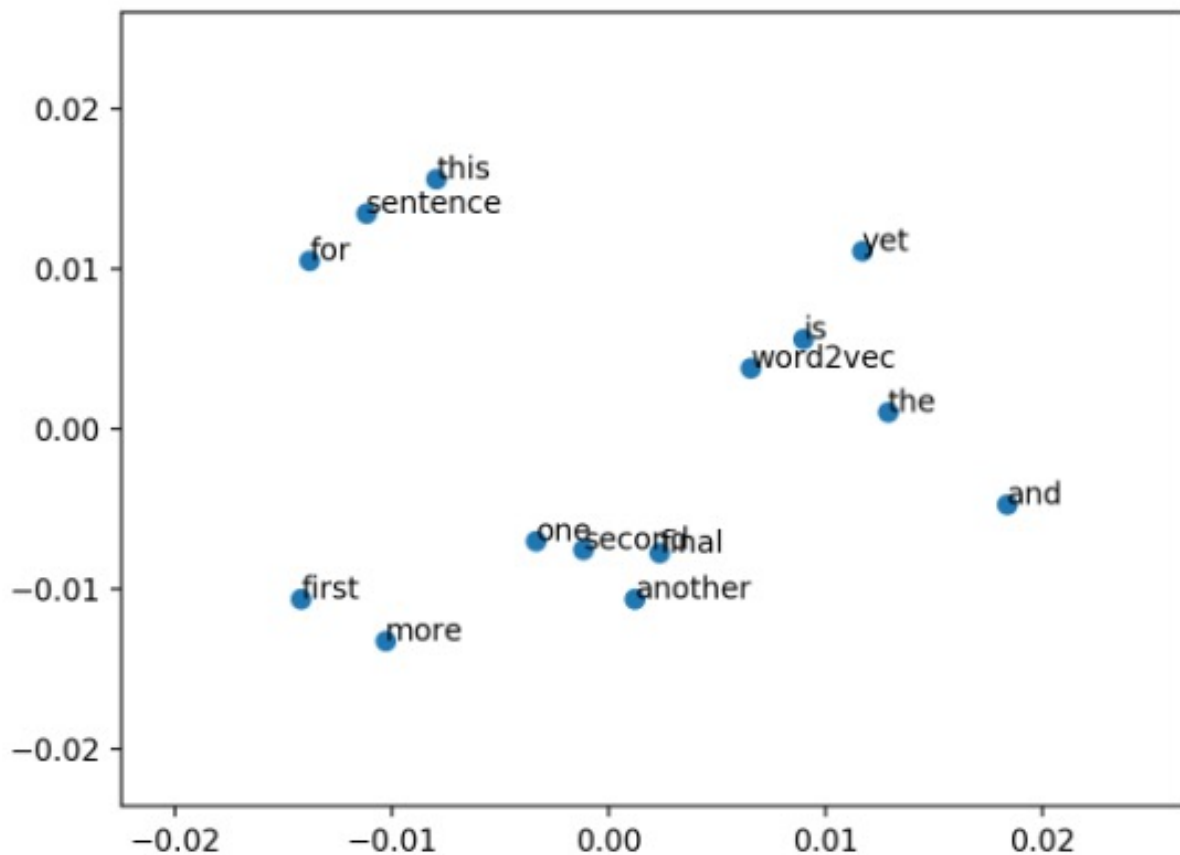
Word2Vec的skip-gram方法



- 使用滑动窗口遍历语料，用中心词来预测上下文的词（词表中每个词都计算出现概率），不断更新词向量从而使预料中出现的词概率最大化。

$$P(o|c) = \frac{\exp(u_o^T v_c)}{\sum_{w \in V} \exp(u_w^T v_c)}$$

词向量



□ PCA降维后的 Word2Vec 向量。

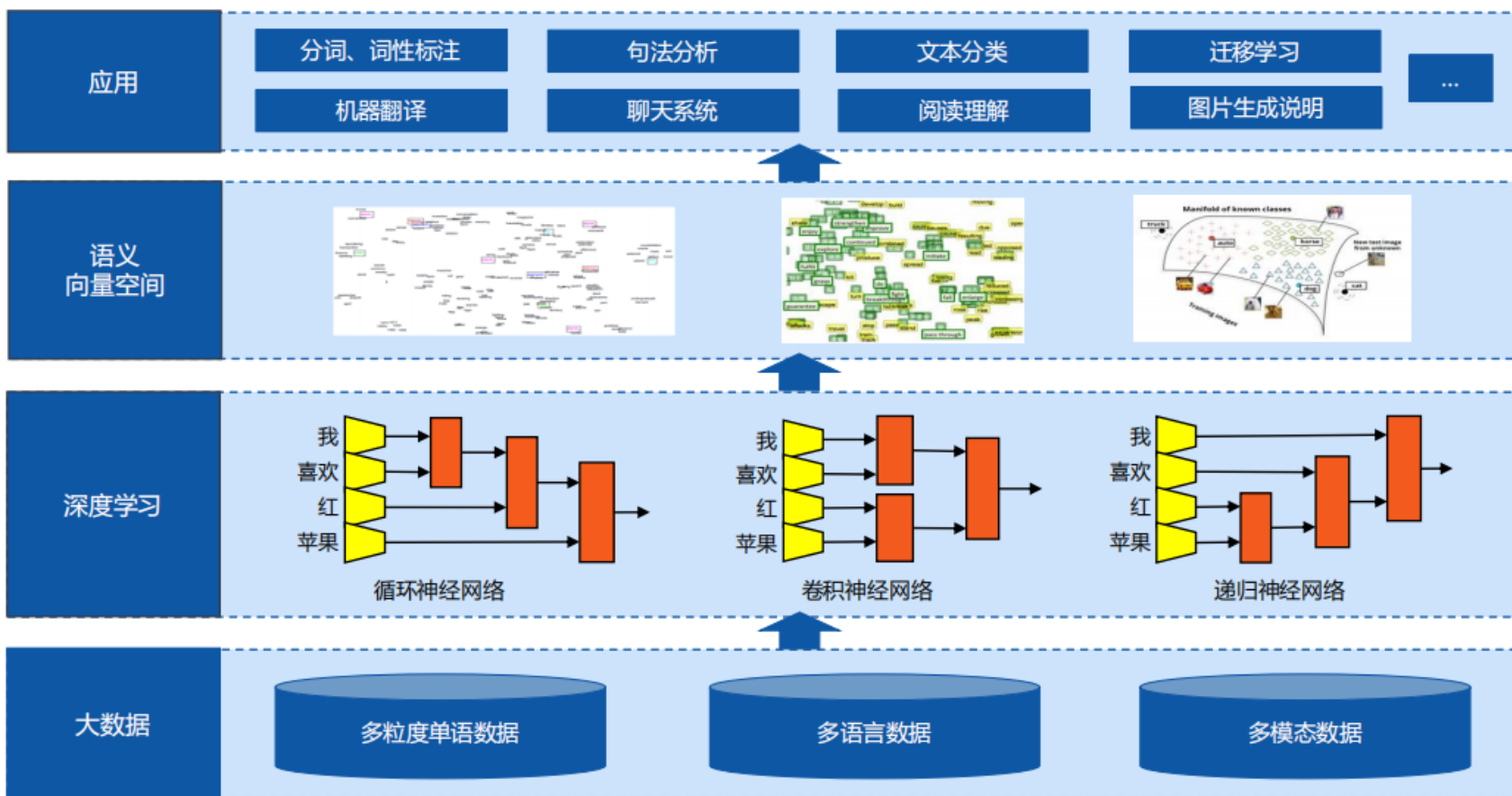
Word2Vec源码地址:

<https://code.google.com/archive/p/word2vec/source/default/source>

常用Python实现:

<https://github.com/RaRe-Technologies/gensim>

深度学习：目前NLP所采用的主要技术手段



刘挺，哈工大，中文信息处理前沿技术进展



第二部分：对话系统



QA问答系统

- 什么是问答系统？
 - 问答系统是对话系统的子集，主要指人需要从对话中获取知识的一类对话系统。划分不绝对，有时会与对话系统混用。



QA问答系统

□ 对话系统目标：

- 用自然语言给系统一句话，系统能够用自然语言给出合理的回应

□ 对话系统分类：

- 任务型对话系统

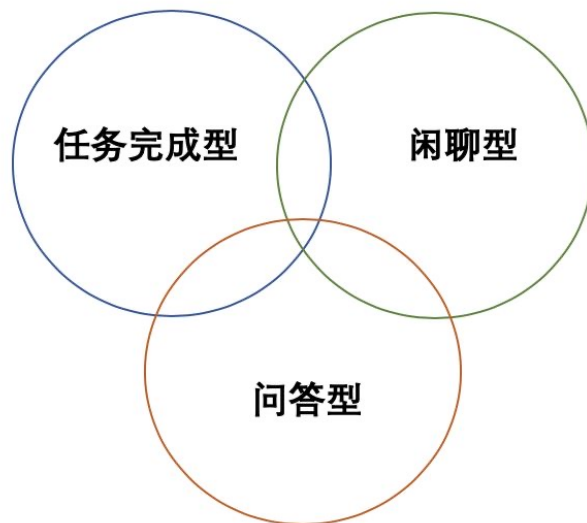
- ❖ 智能音箱

- ❖

- 问答系统

- 获取知识

- 闲聊型



QA问答系统

□ 任务型对话系统：

- 识别用户派遣的任务，然后完成相应的任务。需要对接并调用外部系统接口。
 - ❖ 语音助手/智能音箱：Siri、Cortana、亚马逊Echo、Google Home、
 - ❖ 智能客服





QA问答系统

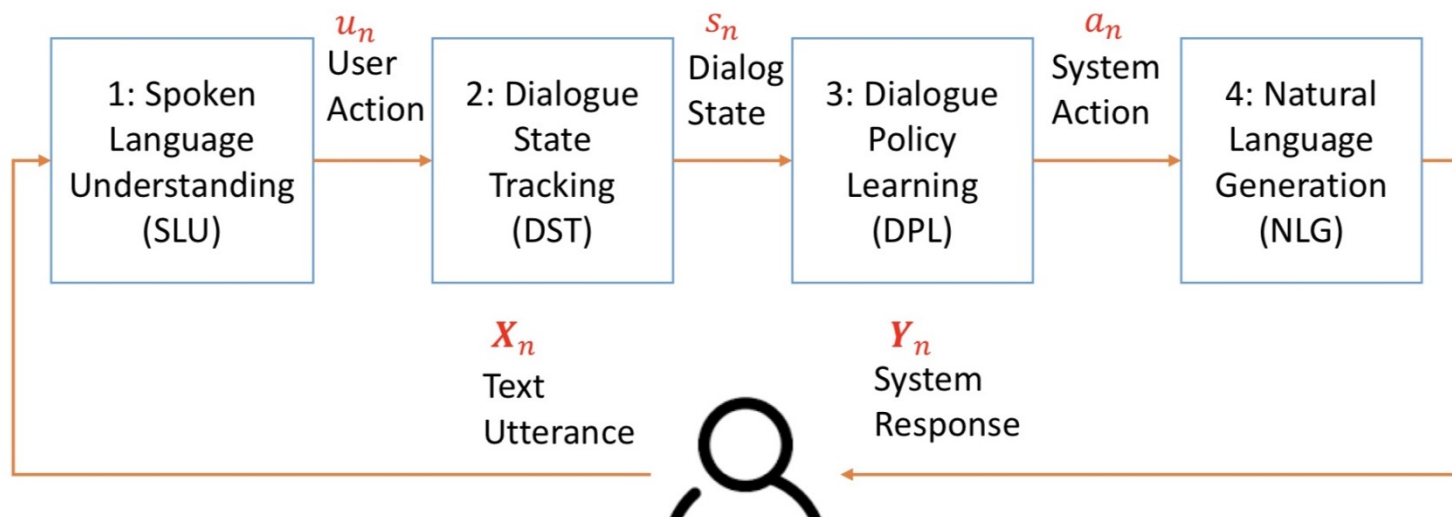
□ 任务型对话系统：

- 对用户向系统输入的自然语言进行目标性的理解，（有时还需要在用户给定信息不明确时对用户进行引导。）
- 实现技术：
 - ❖ 流水线 (pipeline) :
 - 将整个系统划分为多个独立的模块，单独设计各模块，模块之间协作完成任务型对话。
 - ❖ 端到端 (end-to-end) :
 - 由输入到输出整体构建和训练

QA问答系统

任务型对话系统——流水线方法 (pipeline):

- 主要包含四个部分：自然语言理解NLU，对话状态追踪DST，对话策略学习DPL，自然语言生成NLG。



- ❖ Natural Language Understanding: NLU自然语言理解。
- ❖ Dialogue State Tracking: DST对话状态追踪。
- ❖ Dialogue Policy Learning: DPL对话策略
- ❖ Natural Language Generation: NLG 自然语言生成。

QA问答系统

□ 任务型对话系统——流水线方法 (pipeline):

- 1. 自然语言理解: 将自然语言映射进行识别意图和信息抽取 (语义槽, semantic slots)。

Sentence	show	restaurant	at	New	York	tomorrow
Slots	O	O	O	B-desti	I-desti	B-date
Intent	Find Restaurant					
Domain	Order					

- ❖ 意图识别就是将用户的话分类到预定义好的类别中。使用深度学习、CNN进行意图识别和分类。类似的方法同样适用于类别和领域的分类。
- ❖ 槽填充 (slot filling) 是将用户的话中的词打上语义标签 (例如, 日期、地点等)。打标签方法有深度信念网络 (DBNs), RNN 等。
- ❖ NLU的结果 (intent和slot) 会进一步被对话管理器 (dialogue management component) 进行处理, 主要包括对话状态追踪和策略学习。



QA问答系统

□ 任务型对话系统——流水线方法 (pipeline) :

- 2. 对话状态追踪：在每一轮对话中估计用户的目标。常用的状态结构是槽填充 (slot filling) 或语义框架 (semantic frame)。
 - ❖ 传统方法是使用人工定义的规则来选择最有可能的结果。
 - ❖ 还有一些方法通过概论分布和条件随机场来做对话状态追踪
 - ❖ 在深度学习流行起来后有学者提出了基于深度学习、RNN的单领域或多领域的追踪模型；17年有学者提出了神经信念跟踪器 neural belief tracker (NBT) 来检测 slot-value pairs。
- 3. 对话策略学习：对话策略基于上一步对话状态的表示来生成系统的下一步动作 (action)。可以使用监督学习和强化学习。

2、3 构成对话控制部分



Dialog System Technology Challenge (DSTC)

- 对话系统技术挑战赛是一项对话系统的顶尖学术竞赛，由来自微软研究院、卡耐基梅隆大学、本田研究院的科学家于2013年发起。

DSTC7

Dialog System Technology Challenges

Honolulu, Hawaii, USA, January 27, 2019



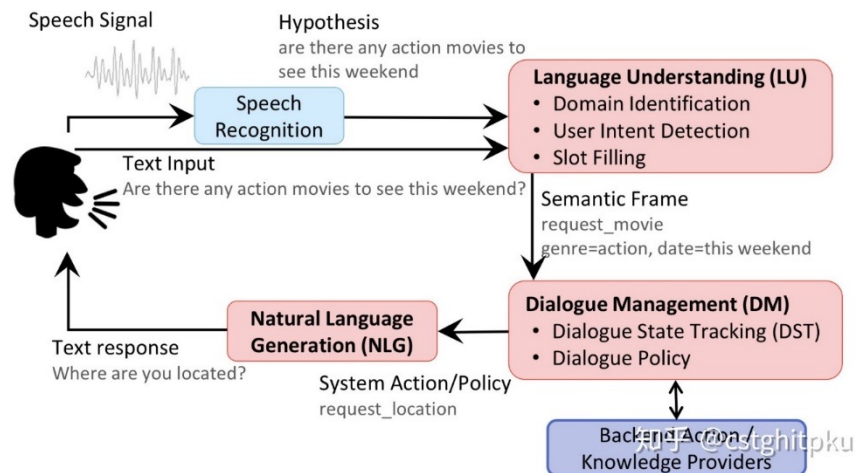


对话管理DM

- 闲聊型对话中的DM就是对上下文进行序列建模、对候选回复进行评分、排序和筛选等，以便于NLG阶段生成更好的回复；
- 知识问答型对话中的DM就是在问句的类型识别与分类的基础上，进行文本的检索以及知识库的匹配，以便于NLG阶段生成用户想要的文本片段或知识库实体
- 推荐型对话系统中的DM就是进行用户兴趣的匹配以及推荐内容评分、排序和筛选等，以便于NLG阶段生成更好的给用户推荐的内容

任务型对话机器人：对话管理DM

- 任务型对话中的DM就是在NLU(领域分类和意图识别、槽填充)的基础上，进行对话状态的追踪 (DST) 以及对话策略的学习 (DPL)，以便于DPL阶段策略的学习以及NLG阶段澄清需求、引导用户、询问、确认、对话结束语等。





DST状态追踪

- DST涉及到两方面内容：状态表示、状态追踪。
- 对话状态追踪DST：作用是根据领域 (domain) / 意图 (intention)、槽值对 (slot-value pairs)、之前的状态以及之前系统的Action等来追踪当前状态。它的输入是 U_n (n时刻的意图和槽值对，也叫用户Action)、 A_{n-1} (n-1时刻的系统Action) 和 S_{n-1} (n-1时刻的状态)，输出是 S_n (n时刻的状态)。
- $S = \{G_n, U_n, H_n\}$ ， G_n 是用户目标、 U_n 同上、 H_n 是聊天的历史， $H_n = \{U_0, A_0, U_1, A_1, \dots, U_{n-1}, A_{n-1}\}$ ， $S = f(S_{n-1}, A_{n-1}, U_n)$ 。

DST技术发展

CRF

Lee, Sigdial
2013
Kim et al. 2014

Transfer Learning

Mrksic ACL
2015
Willams 2013

Neural Belief tracker

Mrksic ACL
2017



NN - Based

Mrksic ACL 2015
Henderson et al.
2013

Multichannel tracker

Shi et al. 2016

End-to-end

单击此处添加文本
具体内容

DST-NN

- Zilka et al., 2015, 基于增量LSTM在DSTC2做对话状态追踪，具体思想如下：

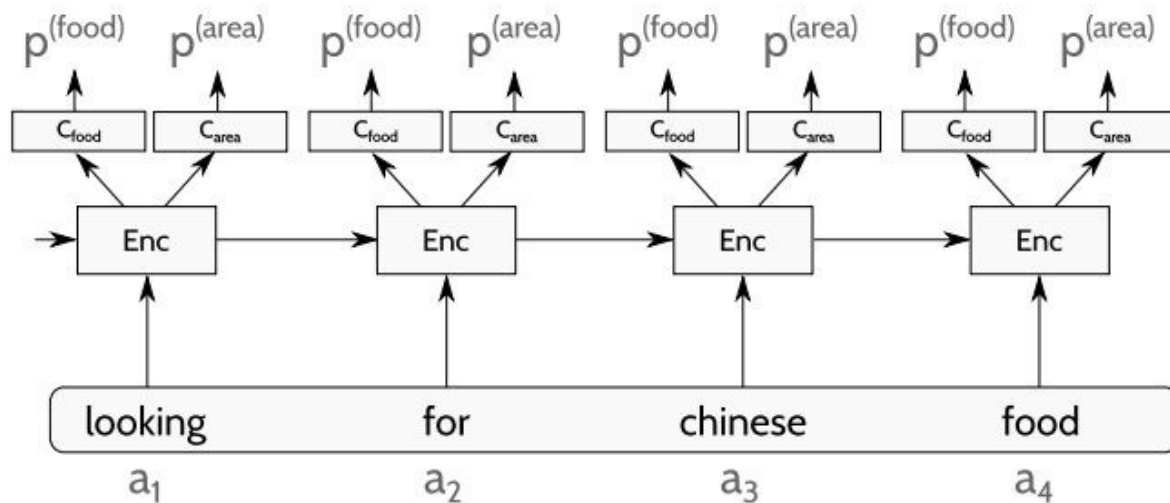
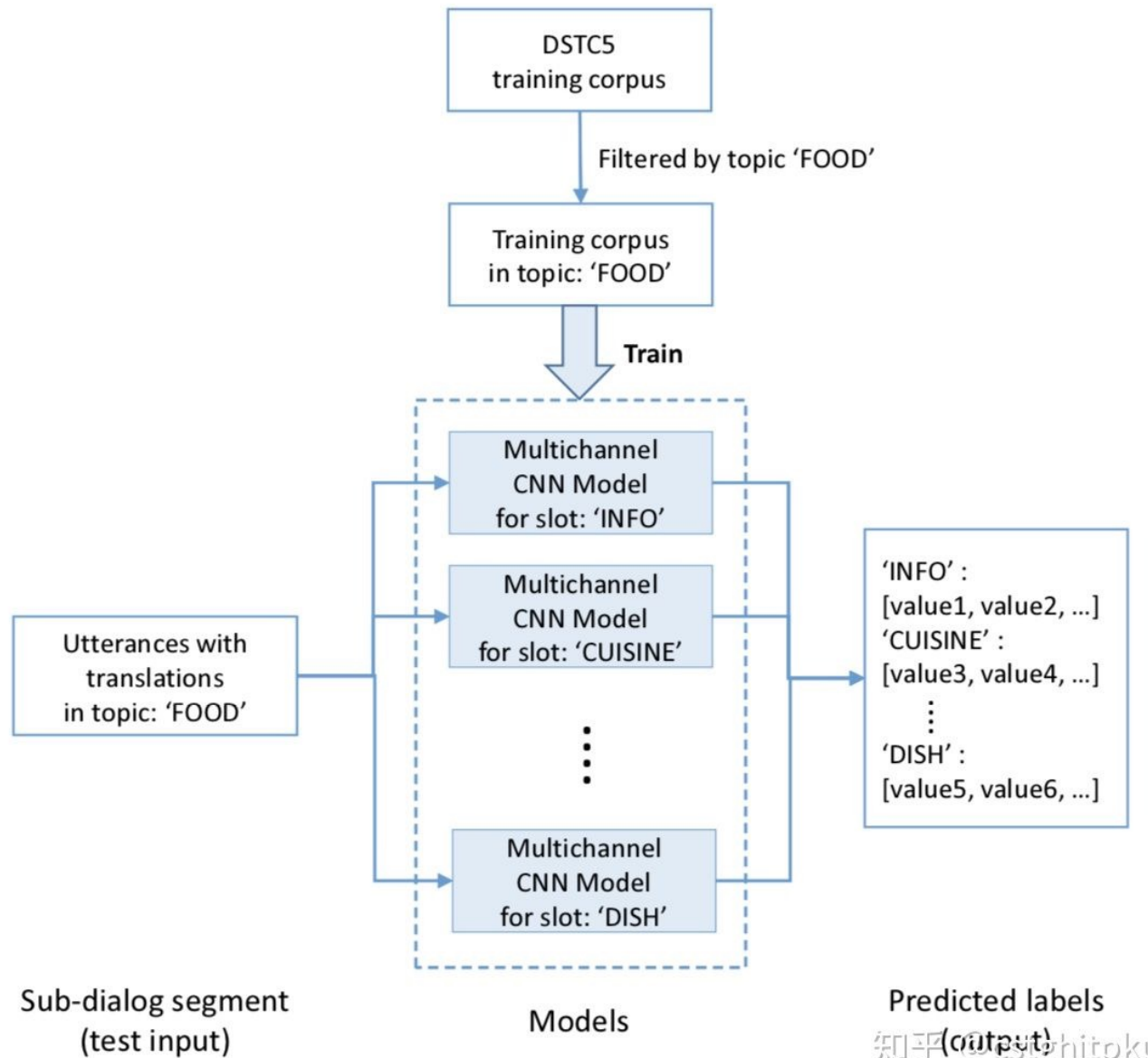


Fig. 1. A demonstration of the LecTrack LSTM dialog state tracker applied to a user utterance “looking for chinese food”. The encoding LSTM model Enc is sequentially applied to each input word and its hidden state is used to feed to the state component classifiers.

知乎 @cstghitpku

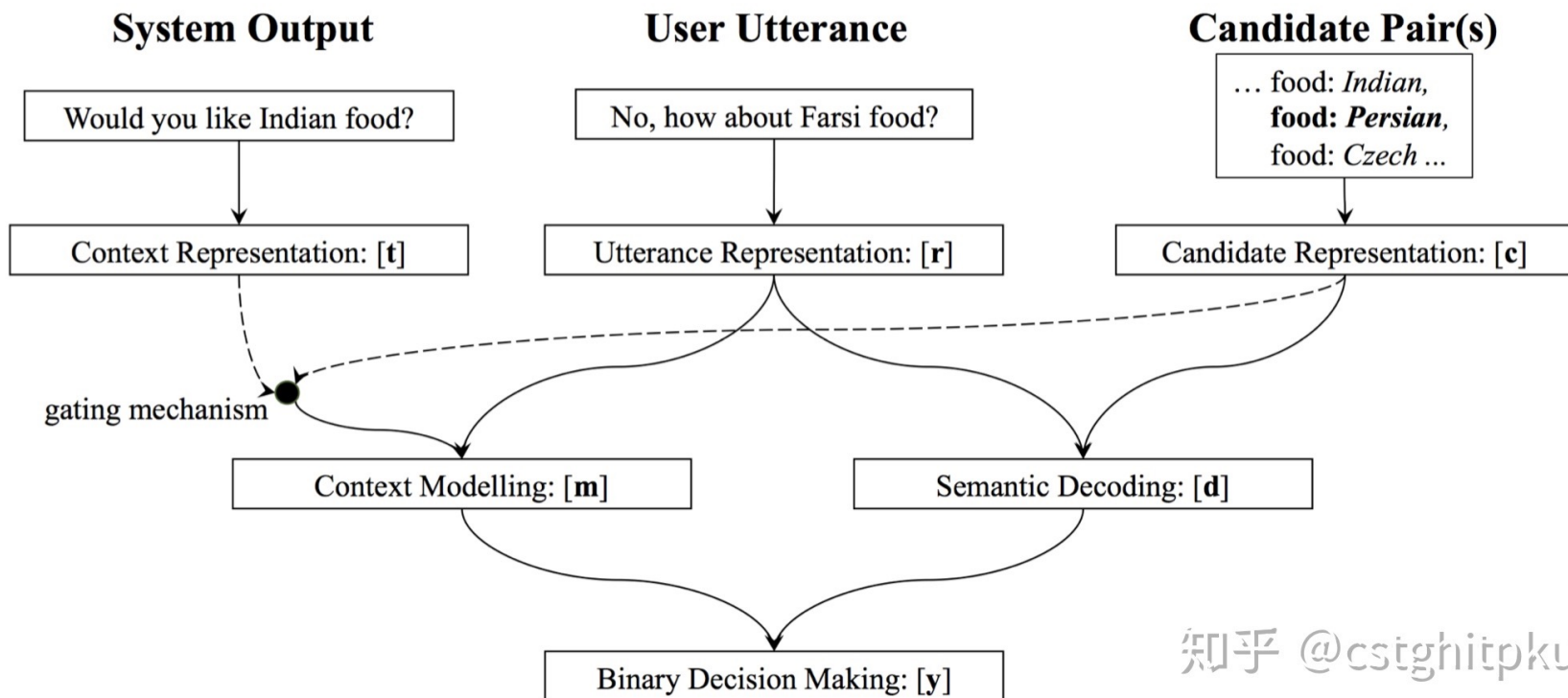
DST-Multichannel





Neural Belief Tracker

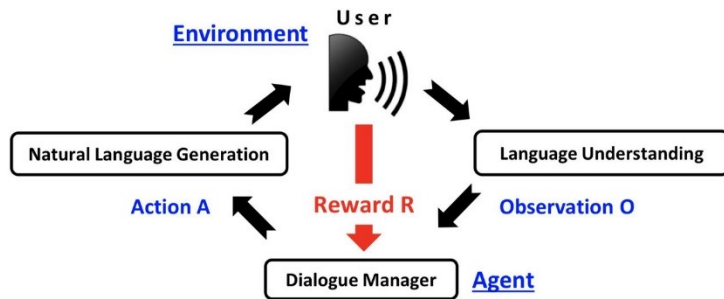
- 包含语义解码和上下文建模两部分：语义解码：判断槽值对是否出现在当前query；上下文建模：解析上一轮系统Act，系统询问 (tq) + 系统确认 (ts+tv)



DPL的输入和输出

Dialogue Policy Learning (DPL)

DPL decides which system action to take based on the dialogue state.



Optimized dialogue policy selects the best action that can maximize the future reward.
Correct rewards are a crucial factor in dialogue policy training.

DST:

输入:

$S_2 = \{G_2, U_2, H_2\}$, 其中:

$G_2 = \{\text{CoffeeType}=? , \text{Size}=? , \text{Temp}=?\}$

$U_2 = \{\text{Intention}=\text{Ask} , \{\text{CoffeeType}=?\}\}$

$H_2 = \{U_1, A_1\}$

$A_2 = \{\text{Action}=\text{Inform} , \{\text{CoffeeType}=\text{Espresso} , \text{Americano} , \text{Latte} , \text{Mocha}\}\}$

$U_3 = \{\text{Intention}=\text{Order} , \{\text{CoffeeType}=\text{Latte}\}\}$

输出:

$S_3 = \{G_3, U_3, H_3\}$, 其中:

$G_3 = \{\text{CoffeeType}=\text{Latte} , \text{Size}=? , \text{Temp}=?\}$

$U_3 = \{\text{Intention}=\text{Order} , \{\text{CoffeeType}=\text{Latte}\}\}$

$H_3 = \{U_1, A_1, U_2, A_2\}$

DPL:

输入是 S_3 , 输出 $A_3 = \{\text{Action}=\text{Ask} , \{\text{CoffeeType}=\text{Latte} , \text{Temp}=?\}\}$



基于强化学习的DPL的发展

Dialogue Policy Learning (DPL)

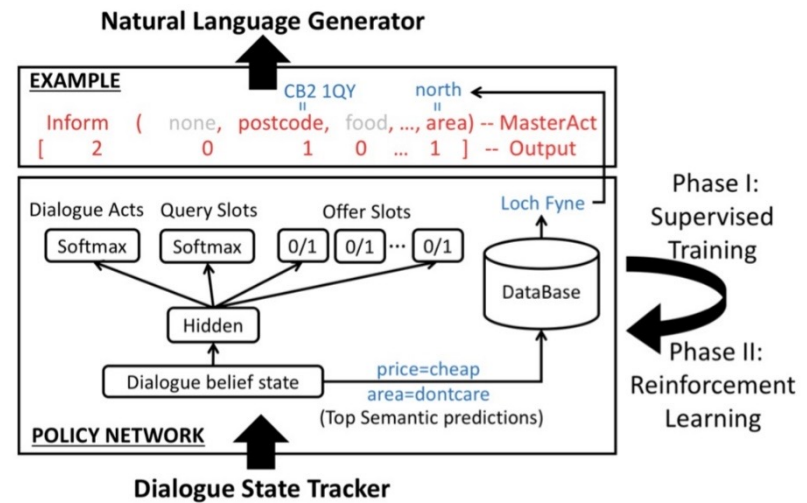
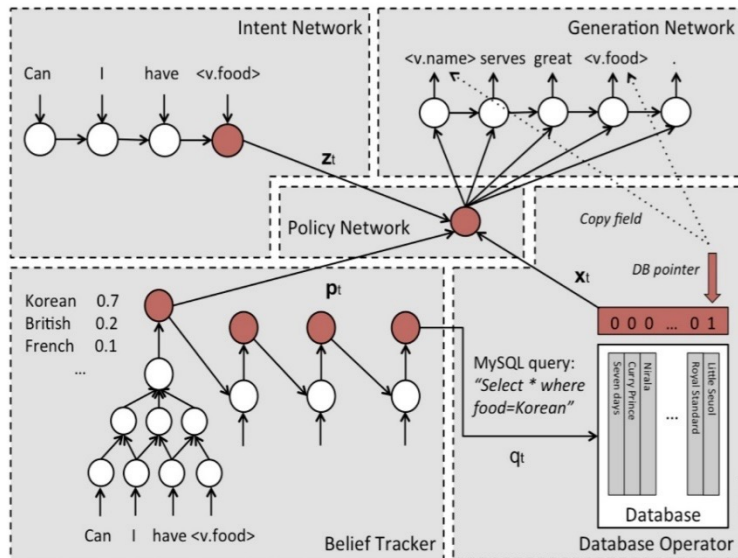
DPL decides which system action to take based on the dialogue state.



Policy based DPL

Wen et al. 2016b

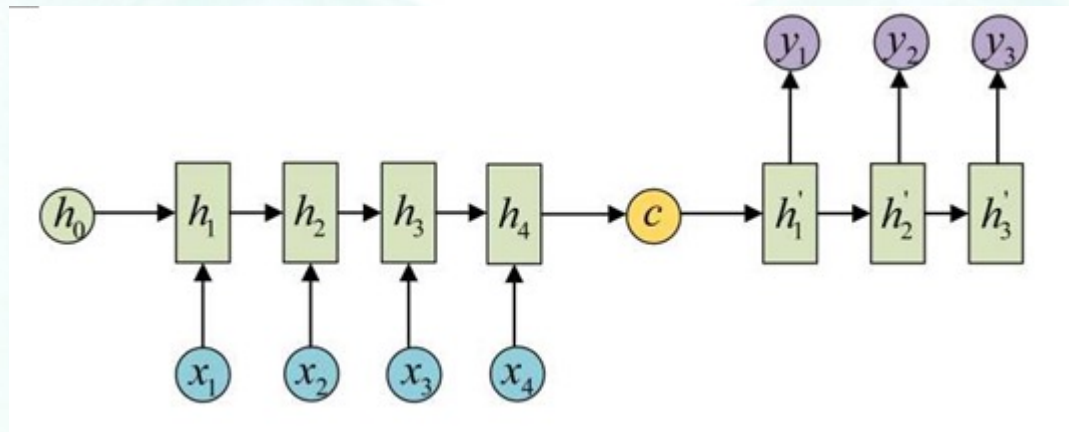
Su et al. 2016



QA问答系统

□ 任务型对话系统——流水线方法 (pipeline) :

- 4. 自然语言生成：根据上一步产生的动作生成自然语言。可以基于LSTM, seq2seq等模型进行自然语言生成。





QA问答系统

□ 任务型对话系统——流水线方法 (pipeline) :

○ 缺点:

- ❖ 基于流水线方法的系统，有很多在具体领域的人工设计，难以应用到其他领域，并且还有两点局限：一是用户的反馈难以传给模型，二是各个模块间相互依赖（一个模块的输出是另一个模块的输入），适应新环境时修改起来需要很多人力。



QA问答系统

□ 任务型对话系统——端到端方法 (end-to-end) :

- [network-based end-to-end trainable task-oriented dialogue system. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, pages 438 - 449, Valencia, Spain, April 2017. Association for Computational Linguistics.]

提出了基于神经网络的、端到端的、可训练的任务型对话系统，将对话系统的学习看作是一个从历史对话记录的匹配过程的学习，使用Encoder-Decoder模型来训练整个网络。缺点是监督学习需要大量数据，并且不够健壮。

- [T. Zhao and M. Eskenazi. Towards end-to-end learning for dialog state tracking and management using deep reinforcement learning. In Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 1 - 10, Los Angeles, September 2016. Association for Computational Linguistics.]

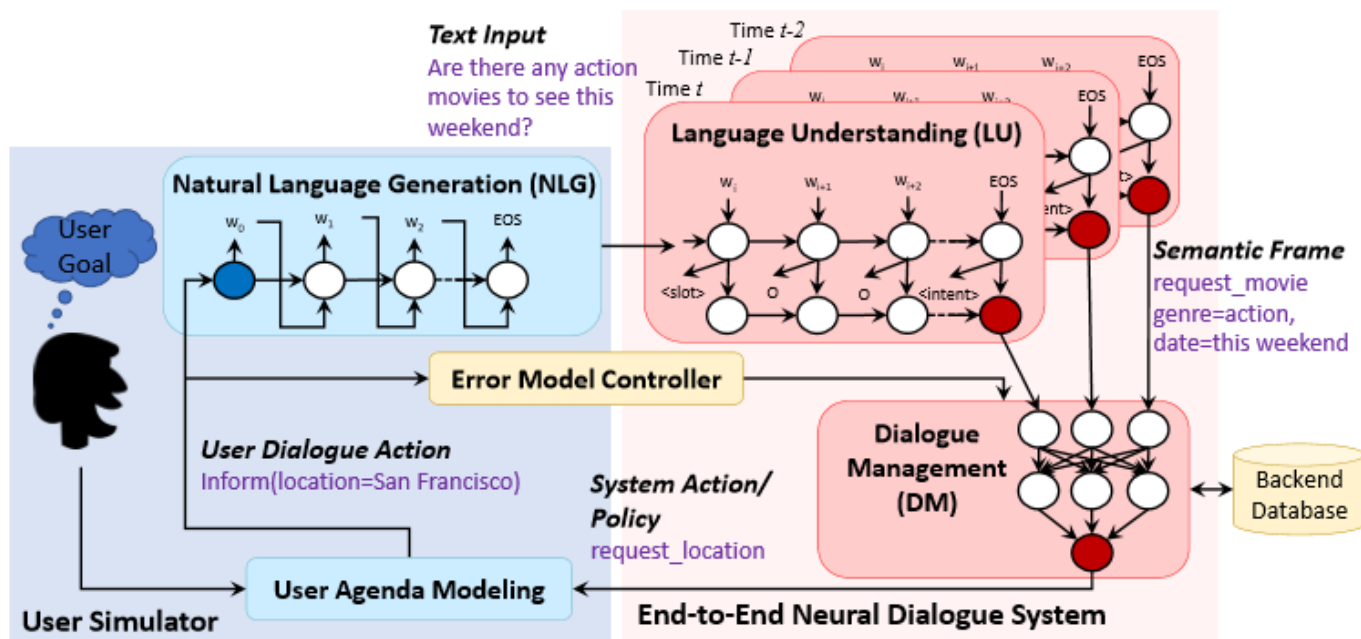
首先提出了端到端的强化学习方法来训练DM（包括对话状态追踪和策略学习），优化系统的鲁棒性（系统问用户一系列Yes/No问题来确定答案）。

- [X. Li, Y.-N. Chen, L. Li, and J. Gao. End-to-end task completion neural dialogue systems. arXiv preprint arXiv:1703.01008, 2017.]

将端到端系统训练为task completion neural dialogue，最终目的是完成一项任务，例如，订电影票

QA问答系统

- 任务型对话系统——端到端方法 (end-to-end) :
 - 端到端的方法中也存在一些方法单独设计模型的各个部件，不同部件解决Pipeline方法中的某个或多个模块。
 - ❖ E2E Task-Completion Bot (TC-Bot) (Li et al., 2017)





QA问答系统

□ 问答系统目标：

- 用自然语言给出一个问题，问答系统能够用自然语言给出合理的回答
- 获取知识

□ 分类：

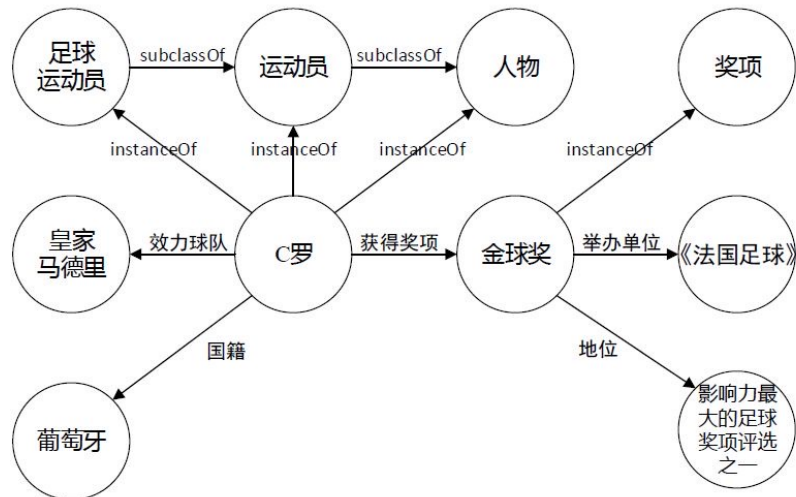
- 按用户问题数据域：限定领域、开放领域、FAQ
- 按答案知识来源：基于知识图谱、基于文本、基于问答对
- 按获取答案方式划分：检索式，生成式
- 按交互方式：单轮对话、多轮对话
-

QA问答系统

□ 基于知识图谱的问答系统：

○ 知识图谱

- ❖ 知识图谱旨在描述世界上各种实体（entity）或概念（concept）及其关系，可被看作是一张巨大的图，节点表示实体或概念，边表示属性或关系。本质上是一种语义网络。边代表了实体/概念之间的各种语义关系。



- 目标：给出问题，系统从知识图谱中推理得到答案

QA问答系统

基于知识图谱的问答系统：



C罗的国籍

网页 资讯 视频 图片 知道 文库 贴吧 采购

百度为您找到相关结果约3,740,000个



克里斯蒂亚诺·罗纳尔多国籍：

葡萄牙

克里斯蒂亚诺·罗纳尔多出生于葡萄牙，中锋、中锋



C罗的妈妈

网页 资讯 视频 图片 知道 文库 贴吧 采购

百度为您找到相关结果约578,000个



克里斯蒂亚诺·罗纳尔多母亲：

多洛雷斯·阿韦罗

C罗的母亲多洛雷斯·阿韦罗在葡萄牙的里斯本正记。作为全球最出色和有影响力的球员的母亲，传记自然是一经发布就受到了热烈的反响。穷...



QA问答系统

□ 基于知识图谱的问答系统：

- 知识图谱一般用RDF形式存储数据
- RDF：英语全称为Resource Description Framework，中文名称为资源描述框架。RDF是一种描述数据文件储存的数据模型，该数据模型通常描述由三个部分组成的事实，被称为三元组（triples）。三元组由主语（subject）、谓语（predicate）和宾语（object）组成，看上去很像一个简单的句子。比如：

subject	predicate	object
richard	homeTel	(229)276-5135
cindy	email	cindym@gmail.com

QA问答系统

□ 基于知识图谱的问答系统：

- 用RDF构建的知识图谱一般用SPARQL进行查询
- SPARQL的英文全称为SPARQL Protocol and RDF Query Language，是为RDF开发的一种查询语言和数据获取协议，它是为W3C所开发的RDF数据模型所定义，但是可以用于任何可以用RDF来表示的信息资源。





QA问答系统

□ 基于知识图谱的问答系统：

○ 传统方法：

❖ 基于语义解析

- 构建语义解析器，将自然语言转化为结构化形式，如逻辑形式。以此在知识图谱上查询和推理

❖ 基于信息检索和信息抽取

- 基于问题给出的信息（实体）在知识图谱中检索，得到以实体为中心的子图，子图中每个节点和边都可以作为答案。通过观察问题依据某些规则或模板进行信息抽取，得到问题特征向量，建立分类器通过输入问题特征向量对候选答案进行筛选，从而得出最终答案。



QA问答系统

□ 基于知识图谱的问答系统：

○ 传统方法：

❖ 向量建模 (Vector Modeling) :

- 该方法思想和信息抽取的思想比较接近，根据问题得出候选答案，把问题和候选答案都映射为分布式表达 (Distributed Embedding)，通过训练数据对该分布式表达进行训练，使得问题和正确答案的向量表达的得分 (通常以点乘为形式) 尽量高。





QA问答系统

□ 基于知识图谱的问答系统：

○ 深度学习加持：

- ❖ 使用卷积神经网络对向量建模方法进行提升
- ❖ 使用卷积神经网络对语义解析方法进行提升
- ❖ 使用长短期记忆网络（Long Short-Term Memory, LSTM），卷积神经网络（Convolutional Neural Networks, CNNs）进行实体关系分类
- ❖ 使用记忆网络（Memory Networks），注意力机制（Attention Mechanism）进行KB-QA：

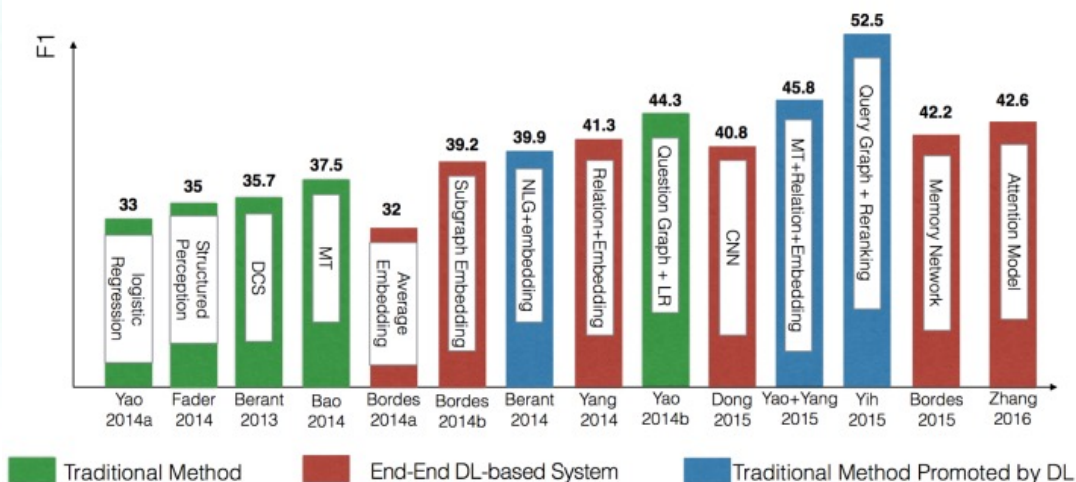
QA问答系统

基于知识图谱的问答系统：

深度学习加持：

- 自中科院刘康老师在知识图谱与问答系统前沿技术研讨会中的报告

Comparison in Benchmark of WebQuestion





QA问答系统

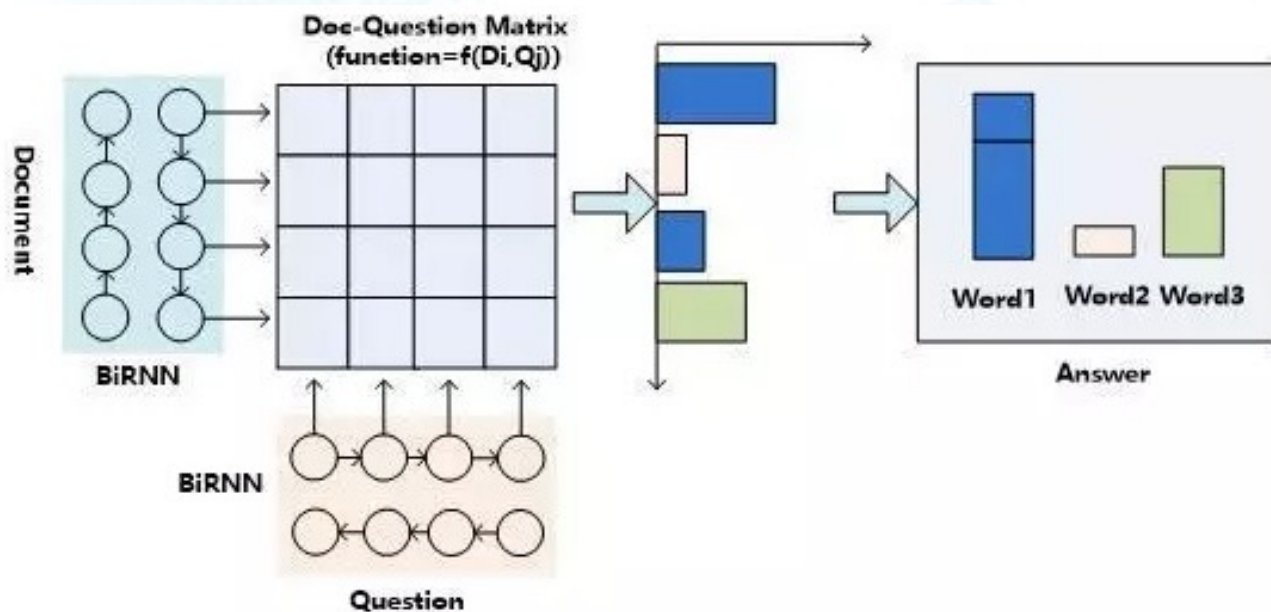
□ 基于文本的问答系统（阅读理解）：

- 对于每个问题，会给定几段文本作为参考，这些文本通常根据问题检索得到，每段文本中可能包含有答案，也可能只与问题描述相关，而不含有答案。我们需要从这些文本中抽取出一个词或几个词作为答案。
- 人工合成阅读理解任务示例

萨姆走进了厨房
萨姆捡起了一个苹果
萨姆走进了卧室
萨姆丢下了苹果
问：苹果在哪？
答：卧室

QA问答系统

- 基于文本的问答系统（阅读理解）：
 - 问答系统模型





QA问答系统

□ 基于文本的问答系统（阅读理解）：

○ 阅读理解中的推理

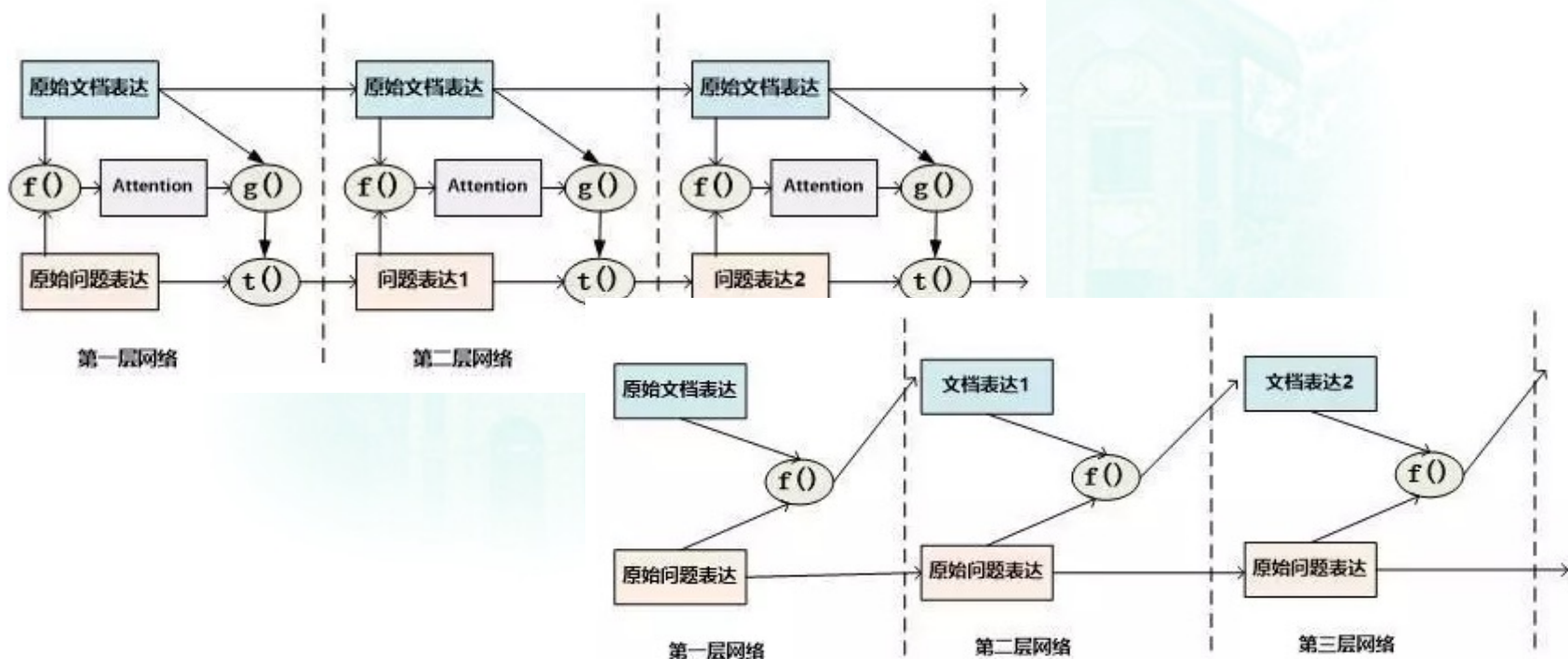
在理解阅读文章内容的时候，推理过程无处不在。没有推理几乎可以判断其是无法完全理解内容的。比起一阶逻辑，自然语言做推理是很困难的（模糊性）。并且用嵌入的方法表示后的自然语言更难以控制其推理过程。

QA问答系统

基于文本的问答系统（阅读理解）：

阅读理解中的推理

❖ 加深网络层数，对单层处理堆叠以期其在不同层做推理不同阶段。





QA问答系统

□ 基于问答对的问答系统：

- 构建常用问答对集合(可以即时构建)，在集合范围内寻找给定问题的答案。
- 问题与答案之间字面意义关联性可能很弱(比如在某些专业性较强领域)，不能或很难利用问题答案的关联性。

问题：如何开启Win10笔记本电脑热点

答案：首先XXXX其次XXX……

- ❖ 系统在集合中寻找与该问题语义最相近的问句，然后返回该问句对应的答案
- ❖ 这种情况下可以用反向索引/倒排索引(inverted index)保存问题的语义向量，做初步过滤，减少相似度计算量。
- 问题和答案关联性较强式还可以利用问题和答案的关联性
- 通常与检索式问答系统关联



QA问答系统

□ 参考

- https://blog.csdn.net/sinat_33231573/article/details/83473741
- <https://wydsb.xin/other/chatbot.html>
- <https://zhuanlan.zhihu.com/p/25735572>
- <https://zhuanlan.zhihu.com/p/83825070>
- 其他论文及博客
-



第三部分：CHAT-GPT



Instruction finetuning

- ❑ **Collect examples** of (instruction, output) pairs across many tasks and finetune an LM
- ❑ Evaluate on **unseen tasks**

Instruction finetuning

Instruction finetuning

Please answer the following question.
What is the boiling point of Nitrogen?

-320.4F

Chain-of-thought finetuning

Answer the following question by reasoning step-by-step.
The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?

The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$.

Language model

Inference: generalization to unseen tasks

Q: Can Geoffrey Hinton have a conversation with George Washington?
Give the rationale before answering.

Geoffrey Hinton is a British-Canadian computer scientist born in 1947. George Washington died in 1799. Thus, they could not have had a conversation together. So the answer is "no".

Reinforcement Learning from Human Feedback (RLHF)



- Let's say we were training a language model on some task (e.g. summarization).
- For each LM sample s , imagine we had a way to obtain a *human reward* of that summary: $R(s) \in \mathbb{R}$, higher is better.

SAN FRANCISCO,
California (CNN) --
A magnitude 4.2
earthquake shook the
San Francisco

...
overturn unstable
objects.

An earthquake hit
San Francisco.
There was minor
property damage,
but no injuries.

$$s_1 \\ R(s_1) = 8.0$$

The Bay Area has
good weather but is
prone to
earthquakes and
wildfires.

$$s_2 \\ R(s_2) = 1.2$$

- Now we want to maximize the expected reward of samples from our LM:

$$\mathbb{E}_{\hat{s} \sim p_{\theta}(s)}[R(\hat{s})]$$

Note: for mathematical simplicity we're assuming only one "prompt"

How do we model human preferences?

- Awesome: now for any **arbitrary, non-differentiable reward function** $R(s)$, we can train our language model to maximize expected reward.
- Not so fast! (Why not?)
- **Problem 1:** human-in-the-loop is expensive!
 - **Solution:** instead of directly asking humans for preferences, **model their preferences** as a separate (NLP) problem! [[Knox and Stone, 2009](#)]

An earthquake hit San Francisco. There was minor property damage, but no injuries.

The Bay Area has good weather but is prone to earthquakes and wildfires.

Train an LM $RM_\phi(s)$ to predict human preferences from an annotated dataset, then optimize for RM_ϕ instead.

$$R(s_1) = 8.0$$



$$R(s_2) = 1.2$$



How do we model human preferences?

- **Problem 2:** human judgments are noisy and miscalibrated!
- **Solution:** instead of asking for direct ratings, ask for **pairwise comparisons**, which can be more reliable [[Phelps et al., 2015](#); [Clark et al., 2018](#)]

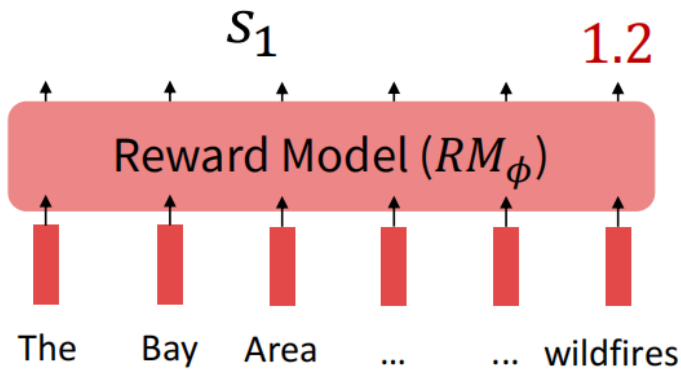
An earthquake hit San Francisco. There was minor property damage, but no injuries.

>

A 4.2 magnitude earthquake hit San Francisco, resulting in massive damage.

>

The Bay Area has good weather but is prone to earthquakes and wildfires.



Bradley-Terry [1952] paired comparison model

$$J_{RM}(\phi) = -\mathbb{E}_{(s^w, s^l) \sim D} [\log \sigma(RM_\phi(s^w) - RM_\phi(s^l))]$$

“winning” sample “losing” sample

s^w should score higher than s^l



RLHF: Putting it all together

[Christiano et al., 2017; Stiennon et al., 2020]

- Finally, we have everything we need:
 - A pretrained (possibly instruction-finetuned) LM $p^{PT}(s)$
 - A reward model $RM_\phi(s)$ that produces scalar rewards for LM outputs, trained on a dataset of human comparisons
 - A method for optimizing LM parameters towards an arbitrary reward function.
- Now to do RLHF:
 - Initialize a copy of the model $p_\theta^{RL}(s)$, with parameters θ we would like to optimize
 - Optimize the following reward with RL:

$$R(s) = RM_\phi(s) - \beta \log \left(\frac{p_\theta^{RL}(s)}{p^{PT}(s)} \right) \quad \text{Pay a price when } p_\theta^{RL}(s) > p^{PT}(s)$$

This is a penalty which prevents us from diverging too far from the pretrained model. In expectation, it is known as the **Kullback-Leibler (KL) divergence** between $p_\theta^{RL}(s)$ and $p^{PT}(s)$.

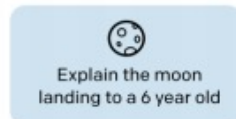
InstructGPT

Scaling up RLHF to tens of thousands of tasks

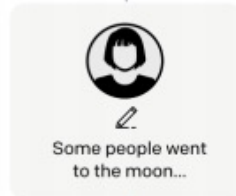
Step 1

Collect demonstration data, and train a supervised policy.

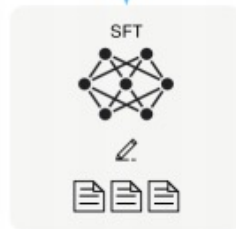
A prompt is sampled from our prompt dataset.



A labeler demonstrates the desired output behavior.



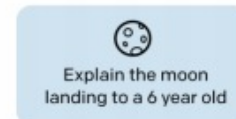
This data is used to fine-tune GPT-3 with supervised learning.



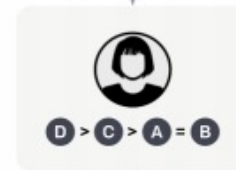
Step 2

Collect comparison data, and train a reward model.

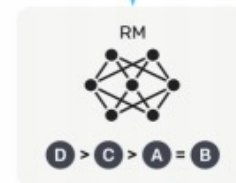
A prompt and several model outputs are sampled.



A labeler ranks the outputs from best to worst.



This data is used to train our reward model.



Step 3

Optimize a policy against the reward model using reinforcement learning.

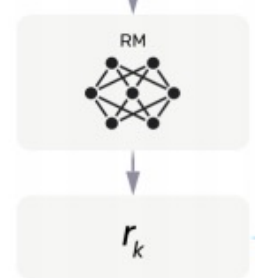
A new prompt is sampled from the dataset.



The policy generates an output.



The reward model calculates a reward for the output.



The reward is used to update the policy using PPO.



ChatGPT

Instruction Finetuning + RLHF for dialog agents

Methods

We trained this model using Reinforcement Learning from Human Feedback (RLHF), using the same methods as InstructGPT, but with slight differences in the data collection setup. We trained an initial model using supervised fine-tuning: human AI trainers provided conversations in which they played both sides—the user and an AI assistant. We gave the trainers access to model-written suggestions to help them compose their responses. We mixed this new dialogue dataset with the InstructGPT dataset, which we transformed into a dialogue format.

(Instruction finetuning!)



ChatGPT

Instruction Finetuning + RLHF for dialog agents

To create a reward model for reinforcement learning, we needed to collect comparison data, which consisted of two or more model responses ranked by quality. To collect this data, we took conversations that AI trainers had with the chatbot. We randomly selected a model-written message, sampled several alternative completions, and had AI trainers rank them. Using these reward models, we can fine-tune the model using Proximal Policy Optimization. We performed several iterations of this process.

(RLHF!)

ChatGPT

Instruction Finetuning + RLHF for dialog agents

MU

Explain in a few sentences what methods were used in training ChatGPT to the audience of CS224n, Stanford's Deep Learning for Natural Language processing course. A lot of the students are very Gen Z, so be sure to use a ton of emojis to seem hip and cool.



Hey CS224n peeps! 😎 ChatGPT here to spill the tea on how I was trained to be a savvy language model! 🧐

So basically, I was pre-trained using a combination of unsupervised learning techniques, specifically a transformer-based language model architecture, that allowed me to learn patterns and relationships within large amounts of text data. 🧠📖

The training data came from various sources such as books, articles, and websites, and I was fine-tuned on a variety of downstream NLP tasks, which involved supervised learning using labeled datasets to further improve my performance. 🏆