

CHATBOT在智能电视行业的应用

暴风TV智能语音助手AIDA项目介绍

By 孙梦琪

原暴风TV服务端产品经理
原微软小冰商业化项目产品经理

大纲

- 产品简介
- NLU需求
- 解决方案
- 项目介绍
 - 项目结构
 - 技术简介
 - NLU流程

产品简介

- 一款智能电视的语音助手，功能以影视搜索为主，涉及多个 **domain**，如多媒体点播、指令控制、IOT控制、语音游戏等，使用规则、词法分析、短文本相似度、拼音相似度、语言模型等技术，应用于语音交互各场景，为用户提供语音搜索和控制服务

NLU需求

NLU需求

语音助手的人机对话以**任务型对话**为主。

电视端语音影视搜索特有的**3个特征**:

1. 意图识别:

- 用户意图明确，不需要多次澄清
- 主要任务是**槽位提取**

2. 以**影视搜索**和**指令控制domain**为主:

- 影视搜索**skill**: 主要任务是影视知识库的健全，包括实体、属性和关系
- 指令**skill**: 指令明确，类型较少，扩展性低

3. **ASR**错误率高，置信度低

解决方案

解决方案

采用单轮任务型对话，侧重于**影视搜索场景**，基础架构：**意图识别+skills**。

1. 健全影视实体库
2. 完善影视搜索和指令控制的规则
3. 建立并优化影视搜索模型
4. 通过拼音相似度转发解决**ASR**影视热词识别错误问题；根据语音准确率的置信度设置不同的回应方式。

解决方案——选择什么模型？

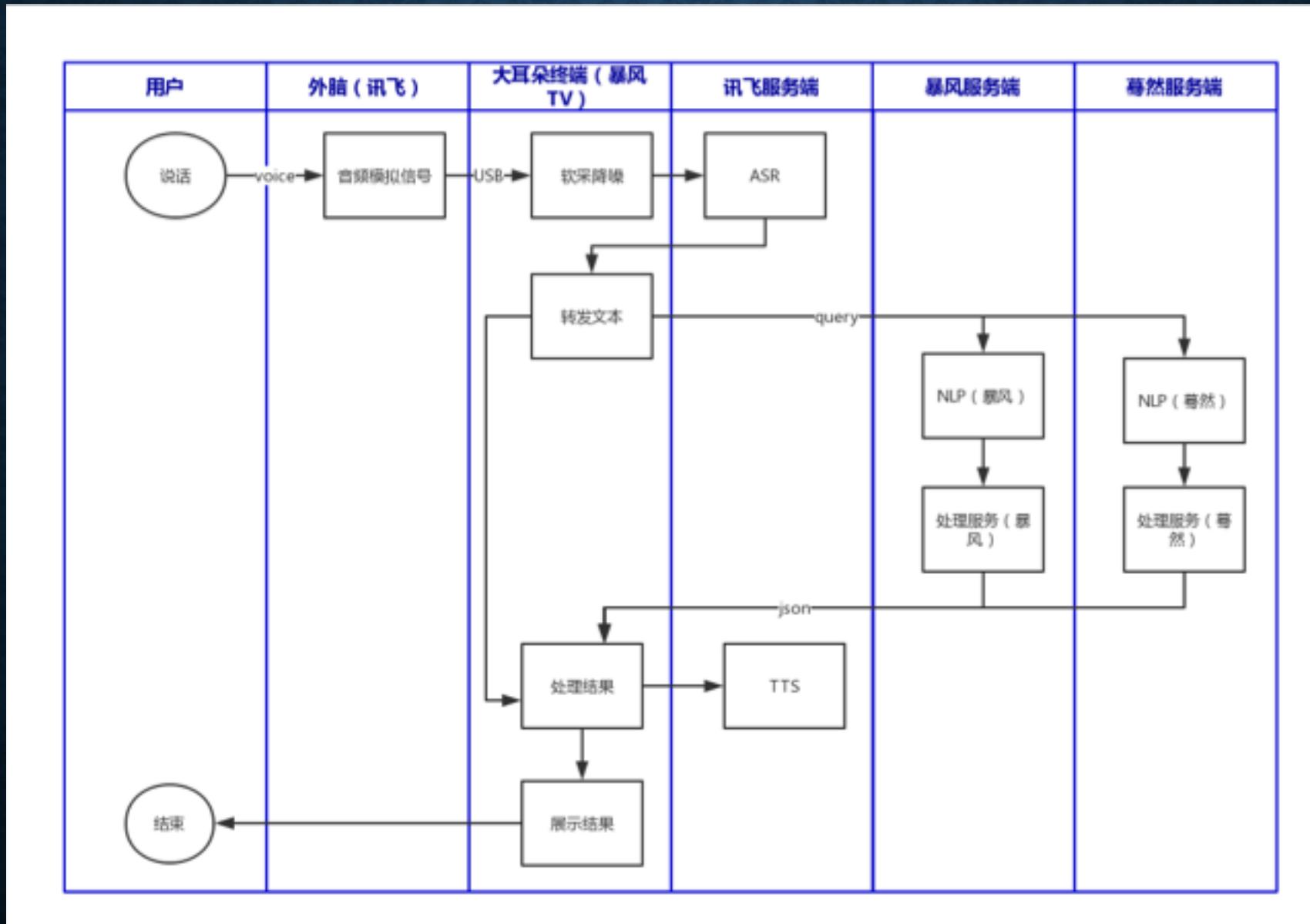
1. 电视影视搜索场景的特征：对于用户的意图判断可以不那么精确。
2. 尽可能覆盖更多句式，提高召回率

双向 LSTM

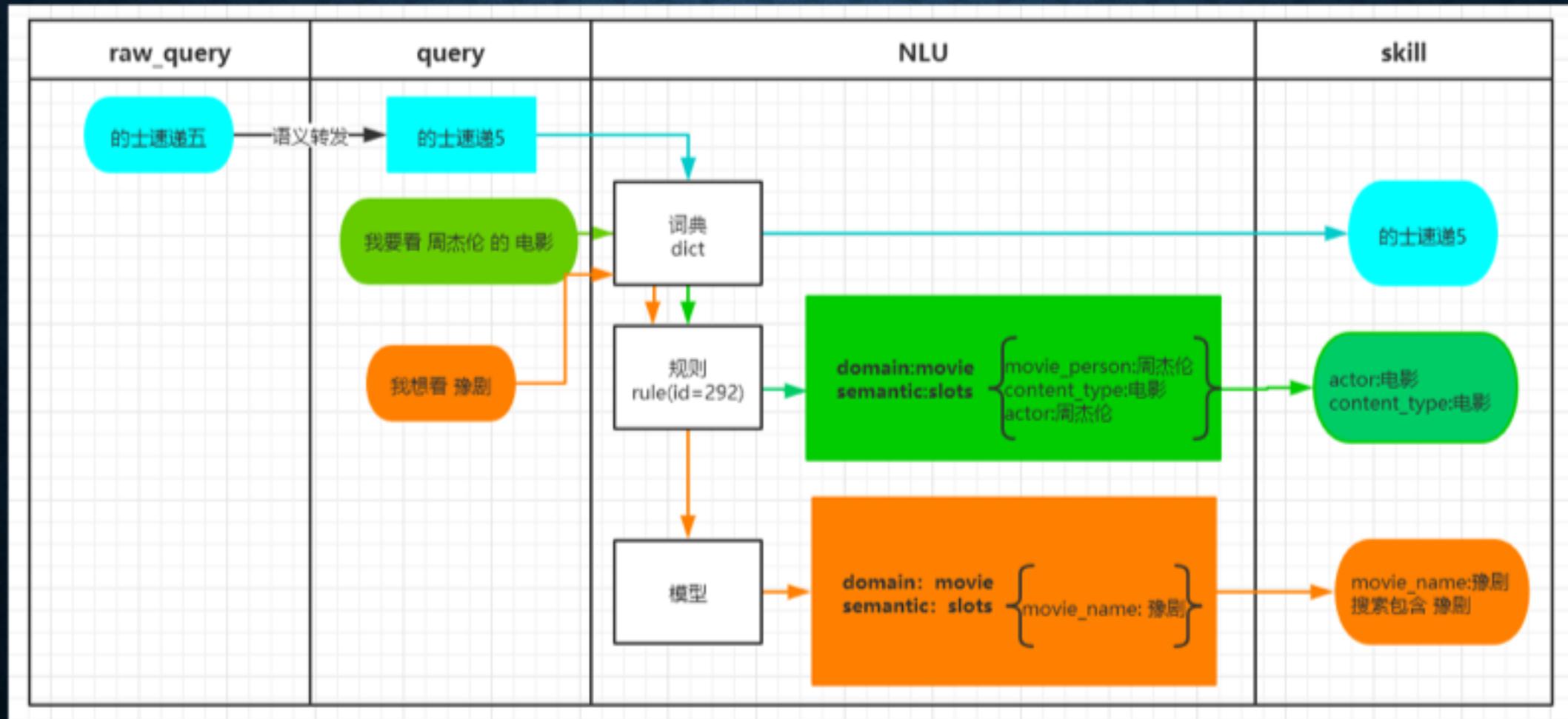
项目介绍

- 项目结构
- 技术简介
- **NLU**流程

总体技术架构



AIDA流程



AIDA使用的技术

词法分析：中文分词、词性标注、命名实体识别（NER）

词向量

词义相似度

短文本相似度（目前仅用于处理闲聊型问答）

拼音相似度（目前主要用于解决ASR识别错误问题）

语言模型（LM），RNN网络

正则

知识图谱（目前只包括影视domain）

AIDA使用的技术

- 槽位提取

1. 中文分词 **sentence cutter** : 结巴分词、海量分词

2. 序列标注: 模型 **BiLSTMCRF+** 规则

- 1) 基于模型: 采用预训练的**word2vec**模型, 运用**Bi-LSTM**结构+条件随机场(**CRF**)进行序列标注。

- 2) 基于规则: 通过规则和实体词典相结合进行实体识别。

例如: “我想看‘2012’年‘成龙’主演的‘动作片’”, 结果如下:

```
[  
  { "value": "2012", "entity": "year", "start": 4, "end": 6 },  
  { "value": "成龙", "entity": "person", "start": 8, "end": 9 },  
  { "value": "动作片", "entity": "movie_tag", "start": 13, "end": 15 }  
]
```

AIDA使用的技术

- 词义相似度

主要用于影视搜索标签、简介的同义词映射

- 短文本相似度

主要使用 **levenshtein** 编辑距离，目前仅用于闲聊型问答。

例如：

Q: 你多大了/啦。

A: 我今年两岁了。

- 拼音相似度、声调相似度

用于**ASR**纠错，以提高影视意图识别的召回率、影视搜索**skill**的召回率。

例如，‘再创世纪’可能被**ASR**识别为：在创世纪，在创世记，再创实际、在创时机……

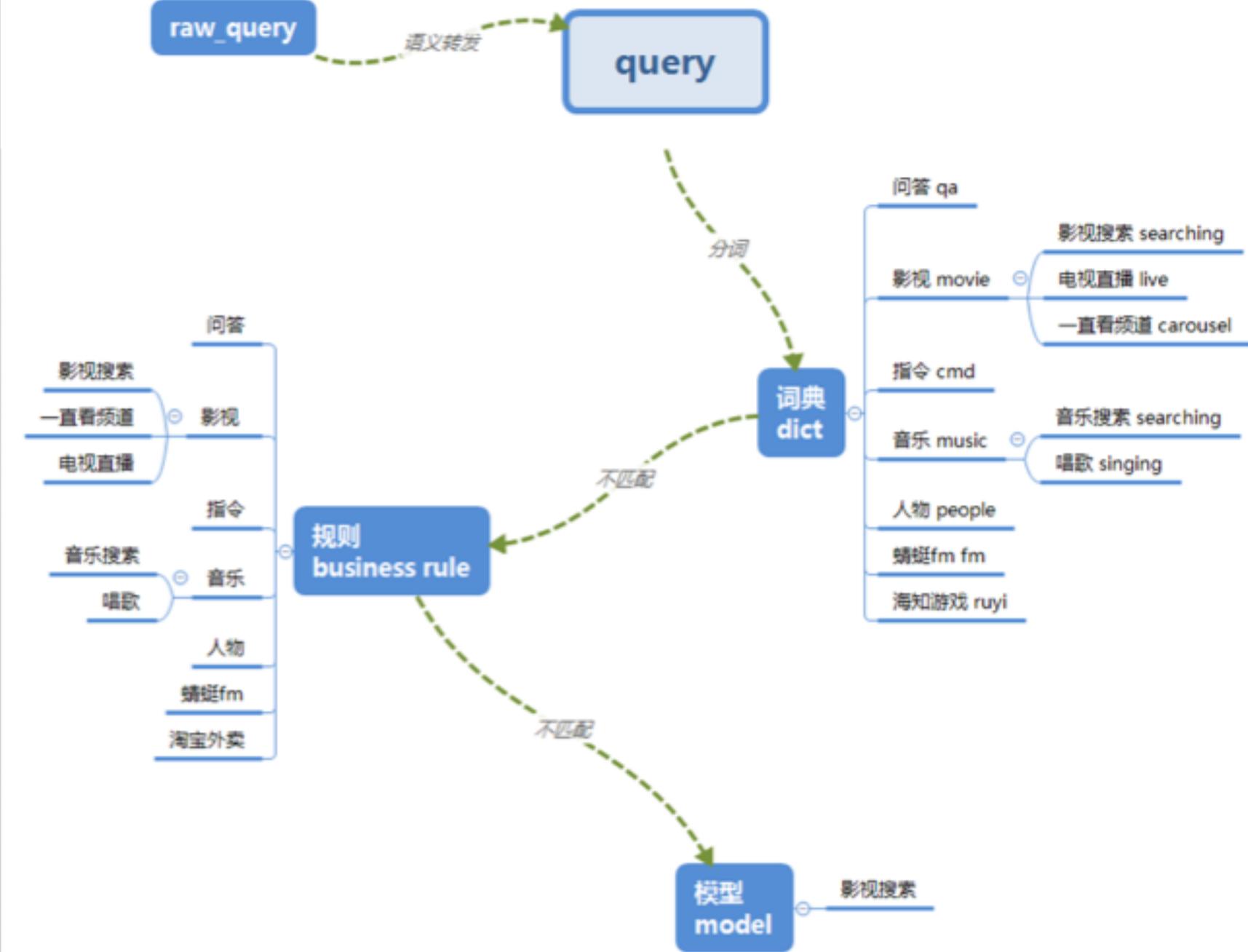
NLU流程

- 场景
- 意图
- 槽位
- **NLU流程**

槽位

影视搜索slot	参数说明
movie_name	电影名称, e.g. 捉妖记2、权力的游戏
sort_by	e.g. 评分高、高分；热门、好看；最新，最近
date_relative	最新、今年、昨天
num_g	评分, e.g. (评分) 9, 4 (分以上)
content_type	媒体类型 e.g. 电影、电视剧、综艺
movie_tag	内容标签：动作、悬疑、动作片、动作电影
year_pick	年份, e.g. 1888, 2010
age	年代, e.g. 80 (年代)、90 (年代)
language	语言, e.g. 英语、日语
country	地区, e.g. 美国、日本、中国
movie_person	包括演员actor、导演director
quality	清晰度, e.g. 高清、超清
pay	是否免费, e.g. 付费、收费、VIP、免费、会员
num_s	部, 季, 期, e.g. (权力的游戏第) 3 (部)、(第) 3(季)、(第)3(期)
order_num	集, e.g. 第3集

- # NLU流程
1. 人工干预
 2. 实体词典
 3. 规则
 4. 模型



NLU流程——人工干预

1. 语义转发

2. **query fix:** 指定**query**的意图或**query**处理方式，如分词和标注方式等。

e.g.

- 将“刺客伍六七第二季”标注为

```
[{"value": "刺客五六七", "entity": "movie_name", "start": 0, "end": 5}, {"value": "2", "entity": "num_s", "start": 6, "end": 8}]
```

- 将“那个”“这个”等，指定为“闲聊”意图（**others.default**）

NLU流程——实体词典

1. 分类

- 影视（影片名、标签等）
- 音乐（歌名、标签等）
- 人物（演员、歌手、导演）
- 有声书（节目、播客、内容分类等）
- 第三方应用
- 其他：地点（国家等）、年份、年代、集数、季数、语言等。

2. 应用场景

- 单独使用：词典匹配（精确匹配）
- 与实体正则表达式结合使用

NLU流程——规则

1. 正则表达式

- **指令类:** 大量、有规律、不需要扩展的句式
- **只需要意图识别，不需要做进一步处理的:** 只需要将 `query` 分到正确的分类，不需要提槽
- **影视、音乐等:** 缺乏有效的实体词典时，用正则表达式+权重调节

缺点：

- 识别准确率低，可能产生误召回的情况；
- 权重调节繁琐，不易维护，可以在开发初期使用，不适合长期使用。

2. 实体正则表达式

- 准确率高、维护方便，目前主要应用此方法。

如：影视搜索意图：地区(**country**)+视频类型(**content type**) 可以用此规则表示：

`^(?P< country>.[^的]{1,3})的?(?P< content_type>.{2,5})$`，其中 **country** 和 **content type** 需要调用相应的实体词典

- 缺点：对实体词典的要求高，召回率相对较低

NLU流程——语言模型

1. 数据准备（清洗、分析）

- 训练语料 **training corpus**

进行人工分词和实体标注后的句子，注意句式要尽可能丰富，如果句式过于单一会造成机器的“偏见”。如：我想看‘周杰伦’(actor)的‘恐怖片’(movie_tag)

可使用网络上的分词和标注工具进行初步标注，再进行人工标注

2. 制定指标、目标：精准度、召回率、准确率、F1

3. 整理测试集

4. 训练模型

5. 测试模型

6. 上线模型

7. 分析数据

8. 定期抽样，对比数据

9. 优化测试集、训练集

10. 不断迭代

谢谢！