Python的智能问答之路

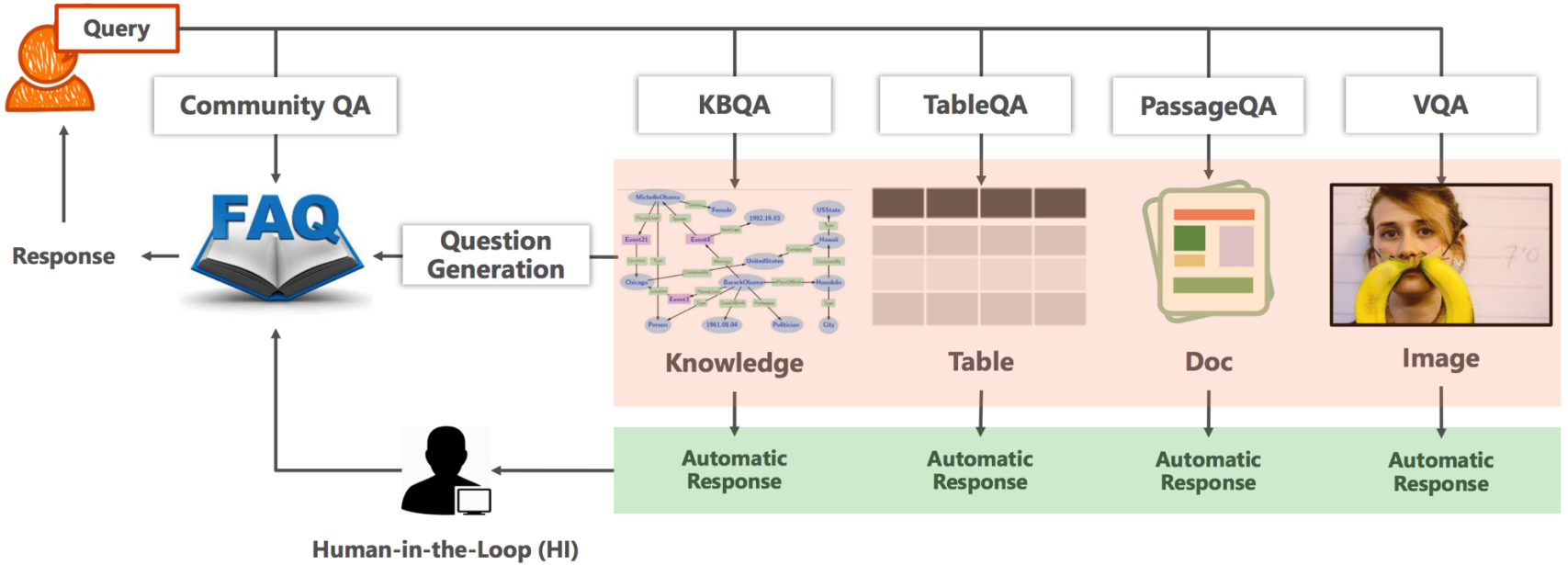
张晓庆

目录 CONTENTS

1 智能问答简介

智能问答领域分类、举例、应用场景

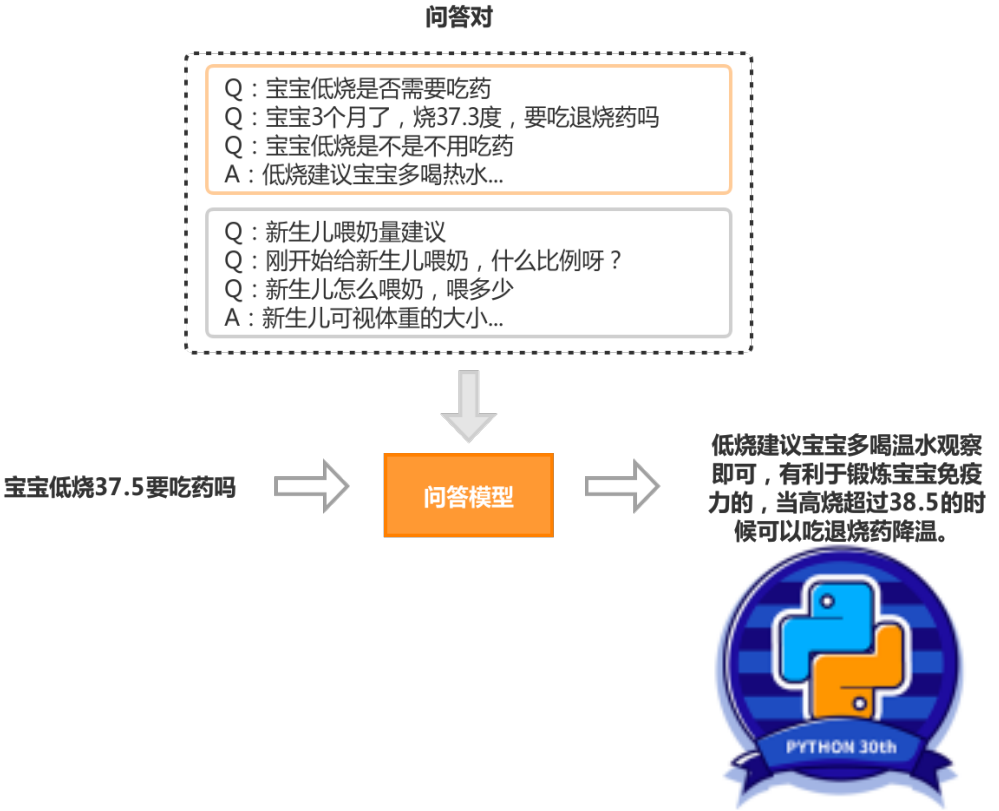
智能问答领域分类



[Duan 2017]

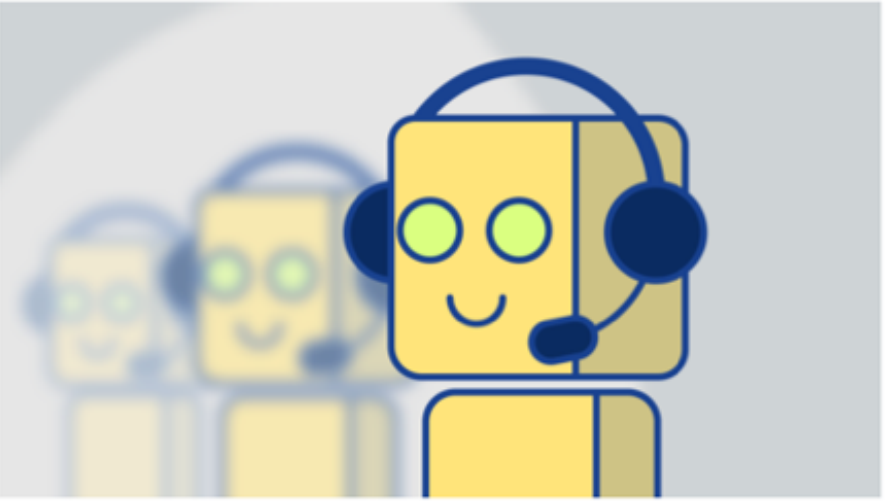
智能问答举例-Community QA

- **数据结构化**
  - 用问答对的方式进行知识表示
  - 知识点：由若干个问题（相似问）、以及能回答这些问题的答案组成
  - 知识库：由若干个知识点组成
- **模型**
  - 找到和用户query最匹配的问题，进而给出对应的答案
- **特点**
  - 易于维护
  - 符合实际业务场景
  - 为什么用这种形式？
    - ✓ 减轻人工维护答案的工作量
    - ✓ 同一知识点下的问题语义相同，是很好的训练数据



智能问答应用场景

- 辅助人工
  - 客服
  - 营销
  - 特定领域、重复性的对话
- GUI补充
  - 语音助手
  - 电话助手
- Voice-only Apps
  - 智能音箱
  - 车载设备
  - 可穿戴设备



2 QA快速实践

任务拆解、各个击破

任务拆解

- **业务**
  - 解决什么问题？
- **数据**
  - 标注数据
  - 训练数据
  - 测试数据
  - 评估数据
- **建模**
  - 输入输出？
  - 工作流？
- **语言工具**
  - C++
  - Python
  - Java
  - GO
- **模型**
  - 统计模型
  - 传统机器学习模型
  - 深度学习模型
  - 如何选择？是否组合？
- **评估**
  - 评估指标
  - 工具
- **迭代**
  - 策略？
- **服务化**
  - 服务框架
  - 性能
  - 稳定性

各个击破 - 业务

- 想给小孩报名英文课，不清楚课程内容和价格怎么办？
- 课程看着不错，能直接帮忙预约一次体验课？
- 想给爸妈买点红酒，该怎么挑？怎么给爸妈讲解红酒的喝法？红酒要怎么保存？

- **营销场景机器人**
- **特性**：
  - ✓ 商务团队好帮手，多平台多渠道获客
  - ✓ 回答标准且及时，第一时间有效引导
  - ✓ 有效减少人力投入，提升线索收集数量

各个击破 - 业务

- 准备出国旅游，不会上网怎么办？需要提前准备什么？国外充电插座和国内是一样的吗？
- 30块的流量包是多少G？
- 办理海淀区高新技术企业需要准备哪些材料？

- **客服场景机器人**
- **特性**：
  - ✓ 永远积极向上，比传统客服更”善解人意”
  - ✓ 回答标准且及时，永不打烊
  - ✓ 支持多平台，支持语音、文字、图片等多种形式
  - ✓ 有效减少人力投入，有效提升应答准确率

各个击破 - 业务

- 公司需要打卡吗？公司的文化是什么？年假多少天？
- 打车发票要怎么报销？
- 物业一年物业费多少钱？能帮忙换水龙头？

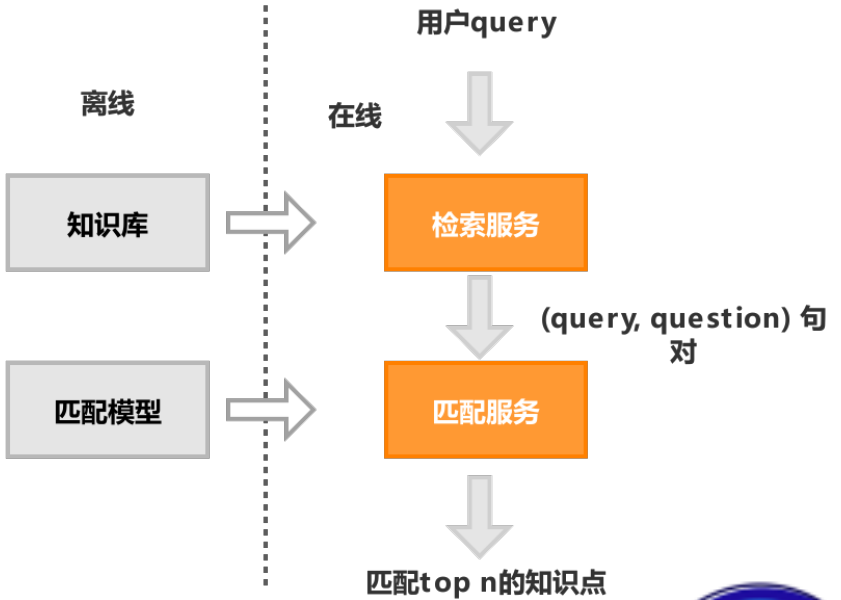
- **其它场景机器人**
- **特性**：
  - ✓ 有效渗透
  - ✓ 横向复制的可行性

各个击破 - 语言工具

- **开发成本**
  - C++：简洁紧凑，灵活方便，需要精细设计，开发成本较高
  - Python：语法简单，完全面向对象，容易入门和使用
  - Java：语法简单，面向对象，但框架较重，相对而言较适用业务程序开发
  - GO：语法简单，支持面向对象、函数、接口编程，开发速度媲美Python
- **平台迁移性**
  - C++：受环境和编译器影响较大
  - Python：安装简单，服务器ubuntu、centos等都默认兼容
  - Java：跨平台可用
  - GO：支持交叉编译，可在不同平台直接运行
- **运行速度**
  - C++：最快
  - Python：最慢，但是可以通过外调C/C++/Java分担慢速计算的压力
  - Java：较快
  - GO：C语言一样的执行速度
- **工具完备性**
  - C++：多为开发者开源，如切词、词性标注等基础工具
  - Python：海量的第三方开源工具库
  - Java：较多开源NLP工具，LingPipe、FudanNLP等
  - GO：较多开源NLP相关工具，gonlp,goml等

各个击破 - 第一次建模

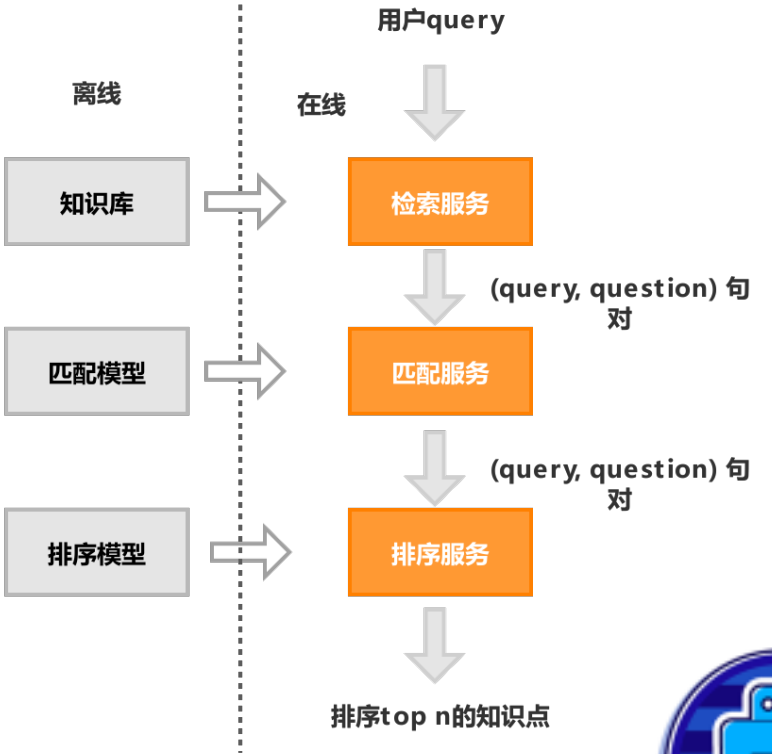
- **检索**（Retrieval）
  - 每个相似问question作为可被检索的文档
  - 输入为用户query
  - 输出为若干个（query，question）句对
  - 每个相似问都对应一个知识点
- **匹配**（Matching）
  - 计算（query，question）的语义相关性
  - 返回top n做排序
- 依赖工具
  - Python：数据预处理、搭建pipeline
  - Python扩展包：elasticsearch
  - Python扩展包：wmd



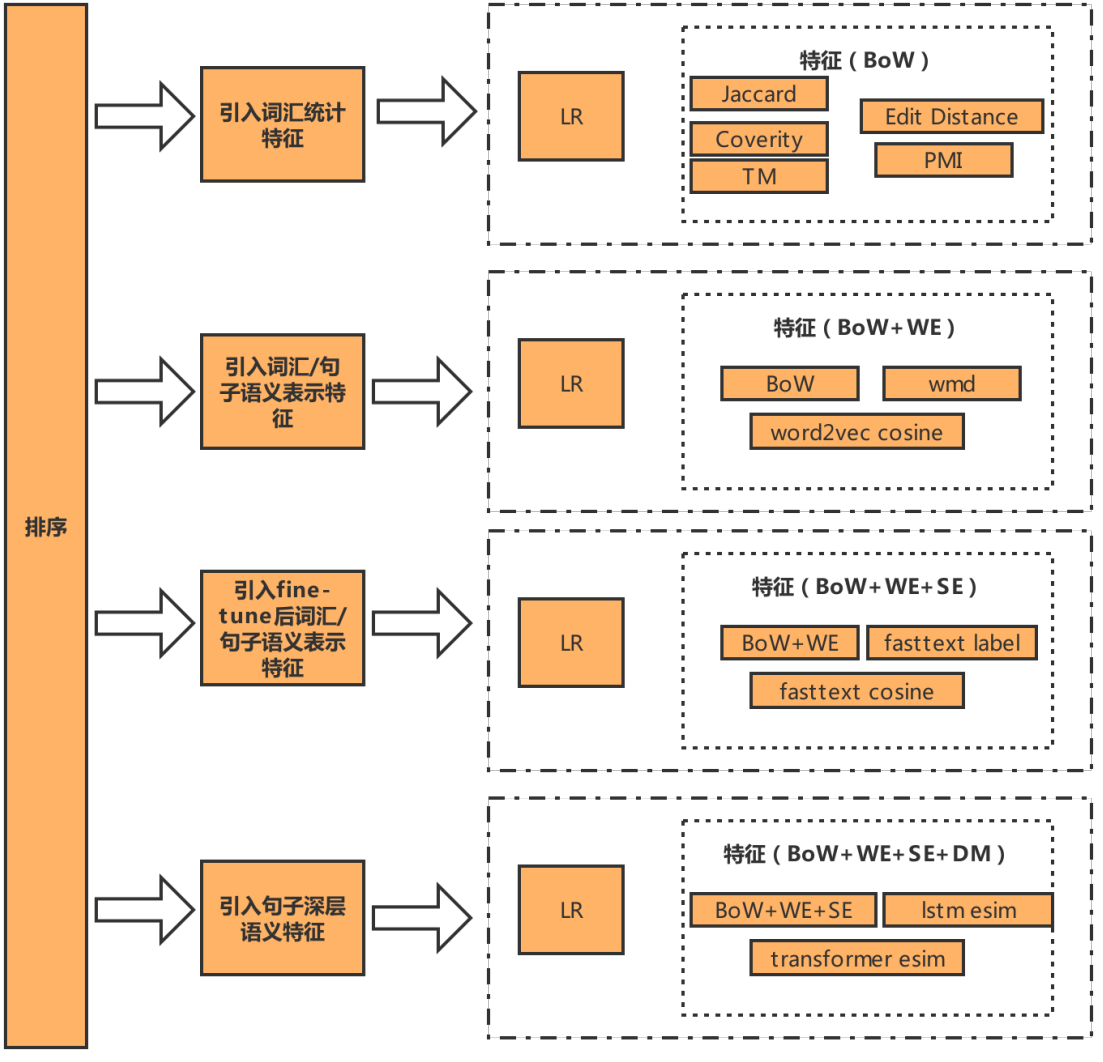
- **存在问题**
  - **相似意图区分能力弱**
  - **泛化能力差**

各个击破 - 第二次建模

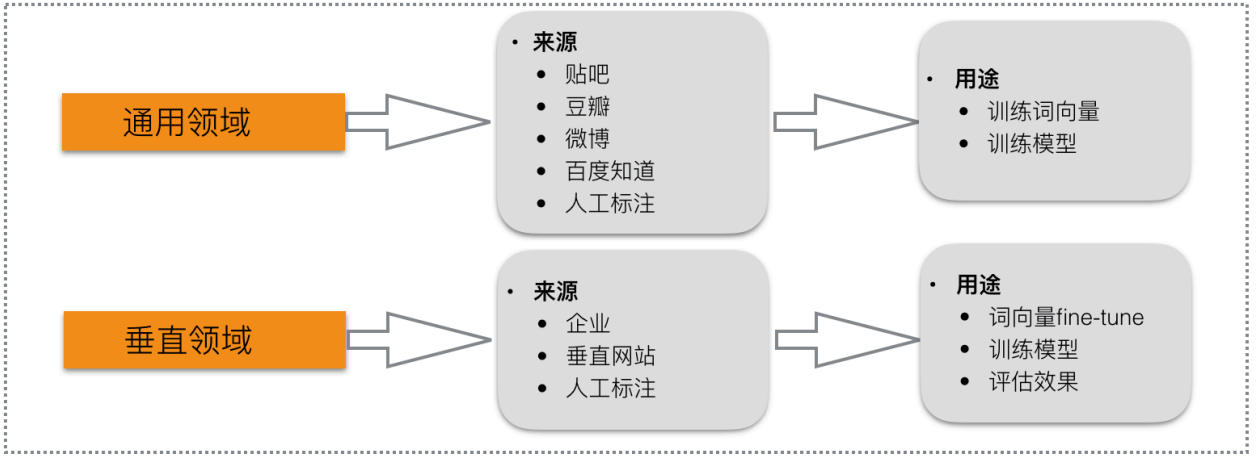
- **Baseline**：检索 + 匹配
- **排序**（Ranking）
  - 用知识库内的相似问，构造句对训练数据，训练有监督的模型
  - 基于通用领域的问答对，构造句对训练数据，训练通用领域内有监督的模型
  - 模型融合
  - 判断（query，question）相关性打分，返回top n作为最终命中知识点，给出对应知识点的答案回复用户
- 依赖工具
  - Python及第三方扩展包



各个击破 - 模型



各个击破 - 数据



- 开源数据抓取&清洗
- 依赖工具
  - ➢ requests抓取数据
  - ➢ retry重试

- 模型训练&特征生成
- 依赖工具
  - gensim：训练word2vec
  - jieba：切词，统计生成PMI/TM词典
  - difflib：调用SequenceMatcher生成edit-distance
  - fasttext：对词向量进行fine-tune，计算fasttext label
  - numpy：计算w2v cosine/fasttext cosine
  - wmd：计算wmd特征
  - esim：计算lstm-esim特征
  - tensorflow：计算transformer-esim特征
  - scikit learn：调用LR训练模型

各个击破 - 评估

婴儿咳嗽怎么食疗	新生儿黄疸吃什么药	没有快递取货码怎么办
宝宝流鼻水咳嗽可以喝什么么	新生儿黄疸可以服用的药物	取货吗被我不小心删了怎么办
宝宝咳嗽吃什么食疗好阿	什么药能治疗黄疸	提货码被我不小心删了
宝宝咳嗽吃食疗吃什么好响	孩子黄疸喝什么药	没有收到取货码
宝宝咳嗽有食疗吗？太小不想用药	新生儿黄疸应该吃点什么药	取货码短信不小心删了
治疗小孩咳嗽的偏方	正常吃什么药能缓解黄疸	取快递的取货码没收到
小孩咳嗽吃什么好的快	黄疸高可以吃什么药	没给我提货码
小儿止咳偏方最有效的	黄疸13.5高的话吃啥药	没收到取件短信
宝宝有点咳嗽怎么食疗	退黄疸用什么药	怎么能知道取货码

- 评估数据
  - 领域均衡：6个领域，每个领域50个知识点
  - 评估数据对标训练数据：每个知识点12个相似问用于训练，3个相似问用于评估
- 评估指标
  - 准确率/召回率/F1值

各个击破 - 迭代

方法	领域1	领域2	领域3	领域4	领域5	领域6	平均
BoW + LR	0.871	0.705	0.880	0.713	0.820	0.820	0.801
BoW + WE + LR	0.898	0.725	0.947	0.793	0.840	0.893	0.849
BoW + WE + SE + LR	0.871	0.933	0.953	0.887	0.880	0.947	0.912
BoW + WE + SE + DM + LR	0.918	0.933	0.960	0.893	0.933	0.953	0.932

- **badcase分析**
- **设计有效特征**
  - IDF加权
- **强化特征语义表示能力**
  - 词袋模型语义表示能力弱
  - 预训练词向量能提升模型的语义表示能力
  - 深度学习网络让句子产生交互，能进一步提升语义表示能力
  - 领域内数据fine-tune是有效的
- **拥抱业界新兴模型**
  - bert+MTL

各个击破 - 服务化

微服务架构1

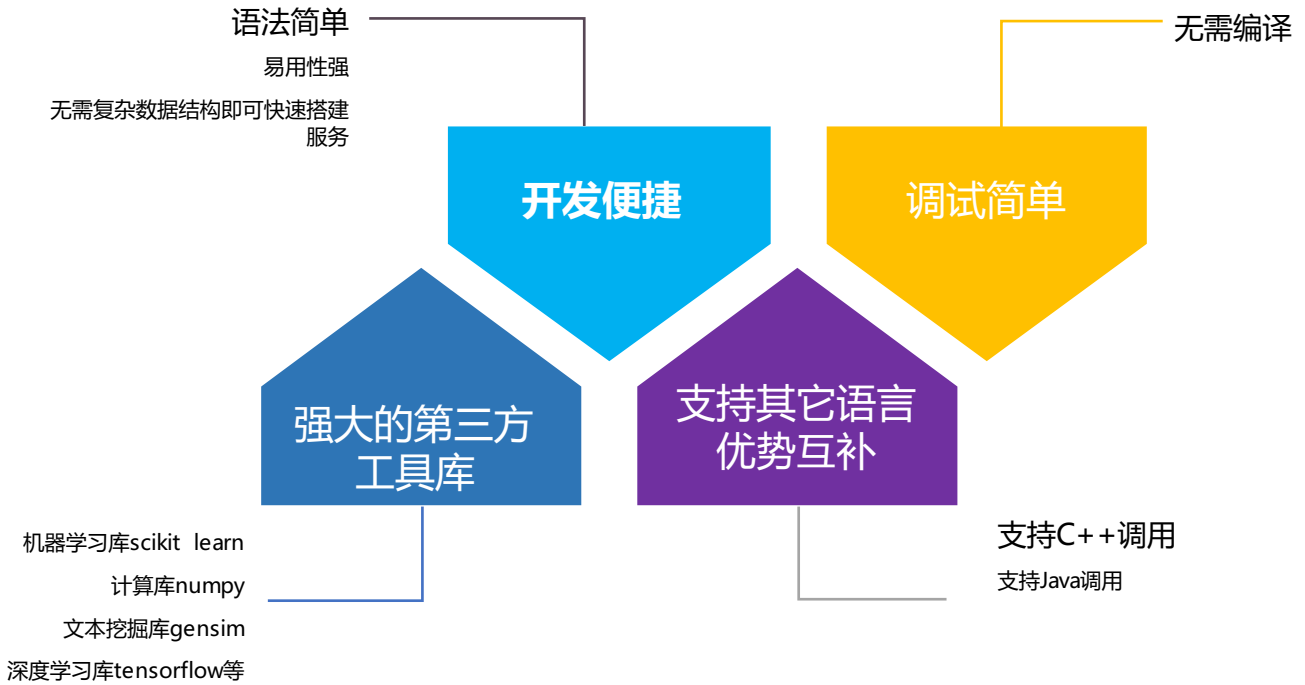
微服务架构2

- 微服务架构2
  - 降低资源占用
  - 方便灵活扩容
  - 资源充分利用
- 服务框架
  - http：短链接，简单，开发方便
  - grpc：长链接，安全性

3 Python开发的利与弊

优势总结、缺点举例

优势总结


语法简单
易用性强
无需复杂数据结构即可快速搭建服务
开发便捷
无需编译
调试简单
强大的第三方工具库
机器学习库scikit learn
计算库numpy
文本挖掘库gensim
深度学习库tensorflow等
支持其它语言优势互补
支持C++调用
支持Java调用

缺点举例1-内存占用高

Python：一切皆对象

对象	占用子节(sys.getsizeof)
int	28
float	24
string	50
list	64
dict	240

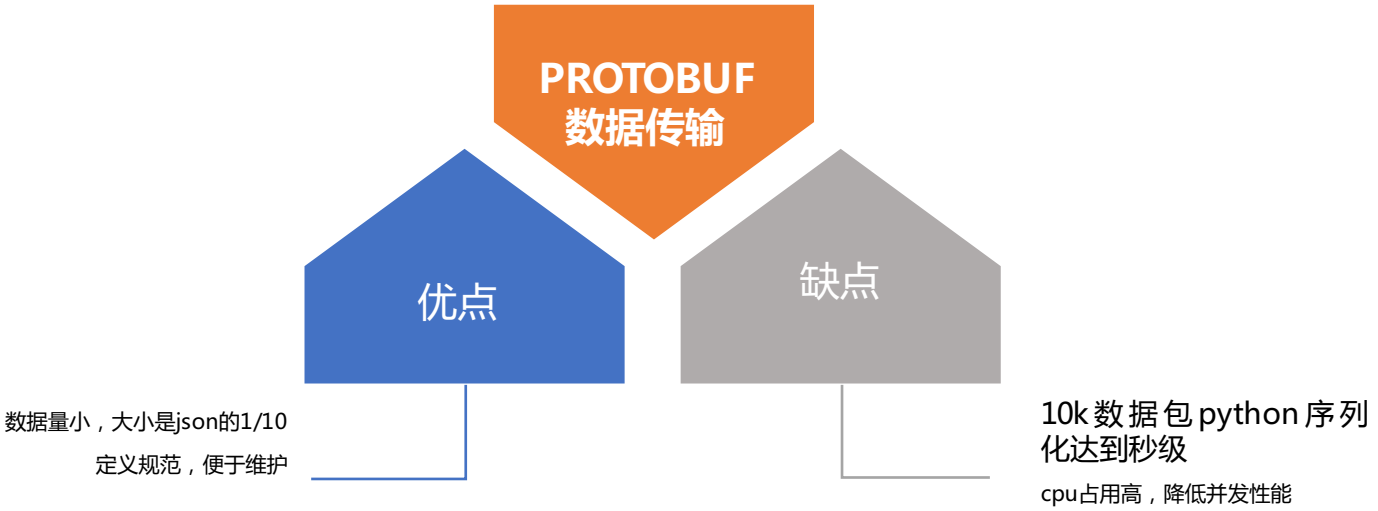
```
typedef struct {
    PyObject_HEAD
    long ob_ival;
} PyIntObject;



struct _longobject {
    long ob_refcnt;  // 引用计数
    PyTypeObject *ob_type; //变量类型
    size_t ob_size;  //实际占用内容大小
    long ob_digit[1];  //存储的实际python值
};
```

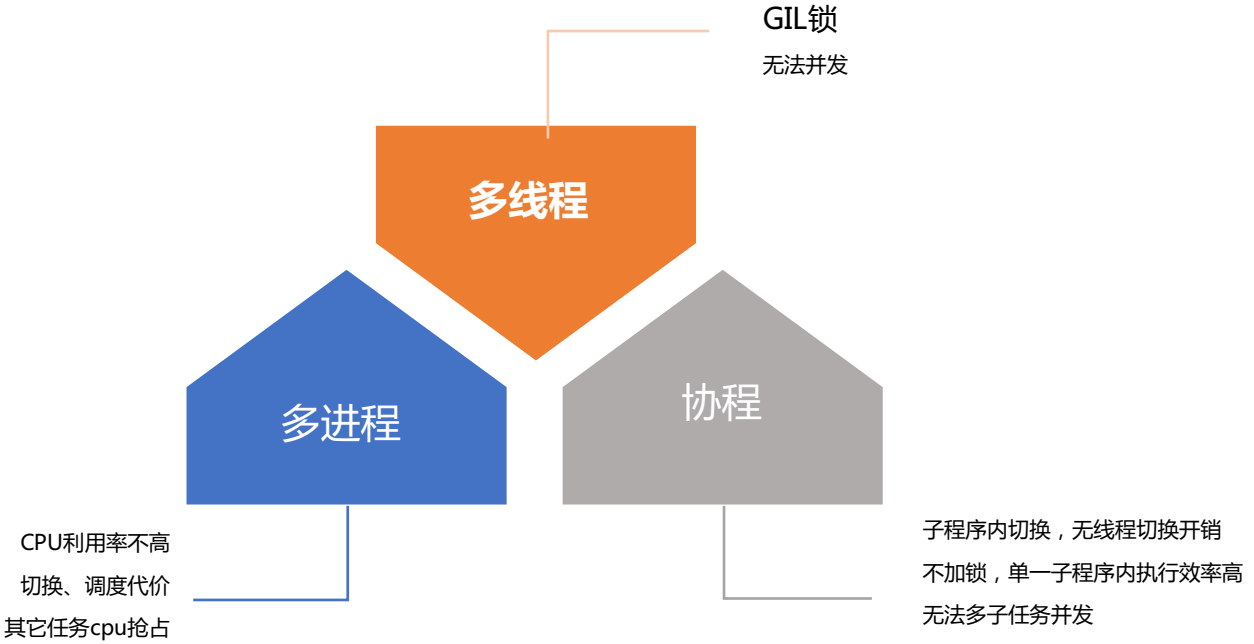
- python执行由解析器解析为C语言对应的结构
- python对象取值仅对应C结构的一个属性
- 附加字段、引用指针均消耗内存
- 解决方案：c++封装kv存储，编译成so供python调用，内部采用unordered_map实现
- 提升：1g文件，python load dict占用5~6g，kvdict存储占用1~1.5g

缺点举例2-pb序列化慢



解决办法：引入c++ implementation，解析速度提升10倍+

缺点举例3-并发机制



解决办法：CPU密集型并发任务独立为微服务，多线程/协程调用

4 总结展望

总结、期望

总结


理解业务，对问题进行合理建模
选择合适的语言完成任务
由浅入深稳步迭代：建立数据、模型、FEEDBACK的闭环
持续性学习以及拥抱技术


PYTHON 30th

期望

基础组件性能优化：存储、访问

语言本身支持更多的协程、高并发、多核

THANK YOU