



知

写文章

...

领域应用 | 基于知识图谱的厨房领域问答系统构建

SUMMBA 索答科技 · 4 个月前

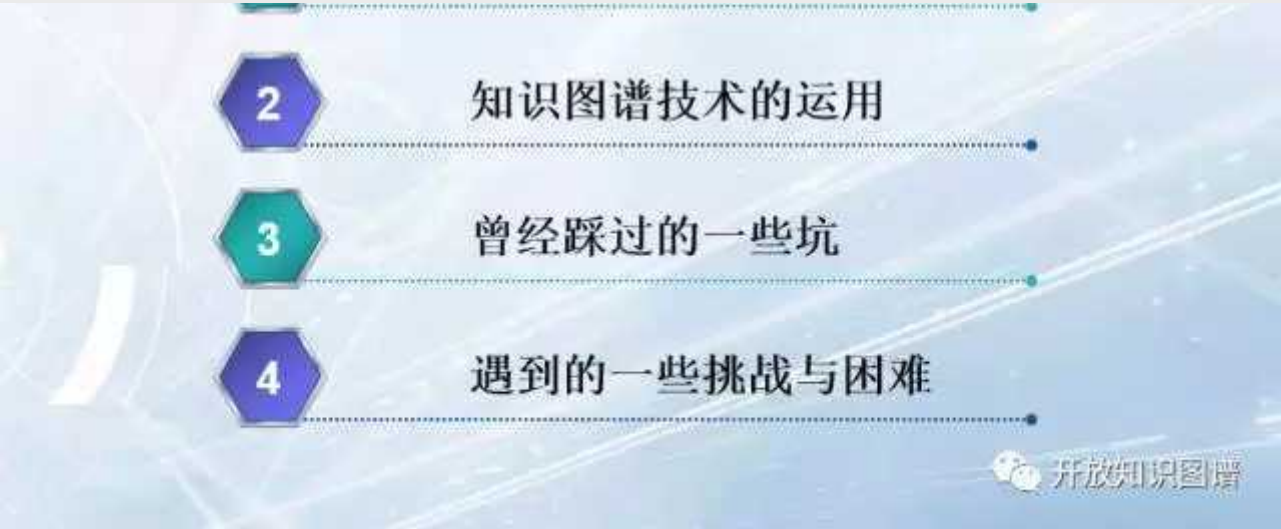
本文整理自广州索答信息科技有限公司 CTO 徐叶强在 4 月 29 日广州知识图谱与问答系统论坛上的演讲。

大家好，我是索答科技的徐叶强。首先感谢杜剑锋老师的邀请，还有我们的蒋院长给我们专门一个平台，有幸在这个地方，给大家分享一下我们索答科技目前做的一个产品里面所用到的一些技术，以及解决的一些问题，那么下面我会分成四个部分，跟大家一起分享。



知

写文章



第一个部分就是我们厨房这个领域的问答，到底我们有哪些问题，或者我们应该怎么去做问答；接下来就是在问答系统当中，我们有很多的方式都可以去解决一个问题，那最后我们选择目前看来比较科学的一种方式，就是知识图谱技术；之后会跟大家分享一下，我们在整个的过程当中踩过的一些坑，就是遇到的一些问题，最后也给大家探讨一下我们遇到的一些困难。

1 厨房领域的问答系统



知

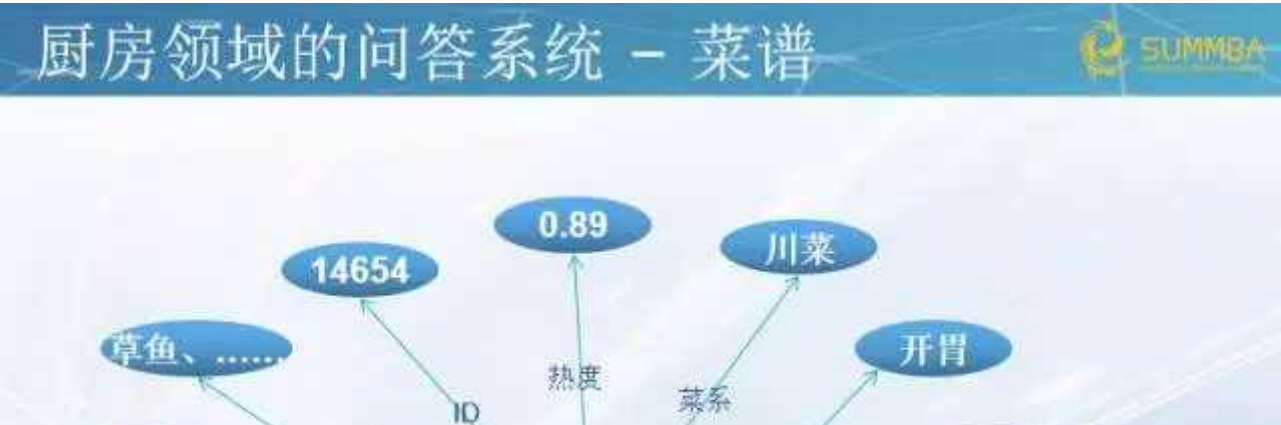
写文章

...



首先我们看一下，在我们这个厨房产品当中，我们有四个部分的回答。第一个部分就是菜谱，菜谱我们分成了很多的维度，通过问答系统，你可以知道哪一道菜，比如说红烧肉怎么做等等。第二个部分就是音乐，那在厨房里面这个枯燥的时间，我们可以说：“我想听一个轻松的音乐”等等。视频也是类似的，比如说我想看《人民的名义》第九集等等。最后一块是厨电的控制，那我们的清单是把厨房的厨电部分，做了一个中控系统，就是说我可以打开油烟机，比如说打开灶具，你可以做到这样的一个联动，再比如说我想炒鸡蛋，那可能你的灶具跟你的油烟机，要同时打开来这样的。

那么接下来的部分，我们主要是探讨菜谱这一块的内容，就是我们如何用知识图谱这样一种方式，去处理这样的一个菜谱的问答的问题。



知

写文章

...

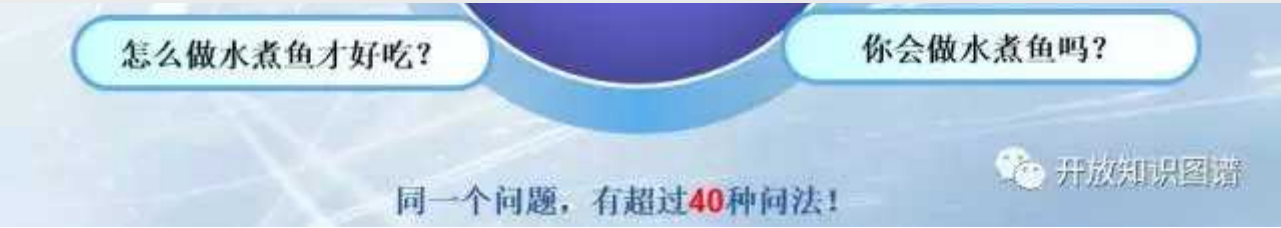


那菜谱首先是这样的，比如说我们以水煮鱼为例，它不仅仅是水煮鱼怎么做，就这样结束了对吧。它有很多很多的维度，就比如说你这个水煮鱼的食材到底是什么，你是用草鱼制作，还是用鲫鱼制作，然后你这个水煮鱼是什么样的味道？然后它有什么样的功效等等。那我们就把它形成了这样的一张图，那这个图中间的这个点就是我们菜谱的名称，然后后面有各种各样的属性，以及它的属性值，我们大体要做这样的一个原始的，一个最小化的节点就是这样的。



知

写文章



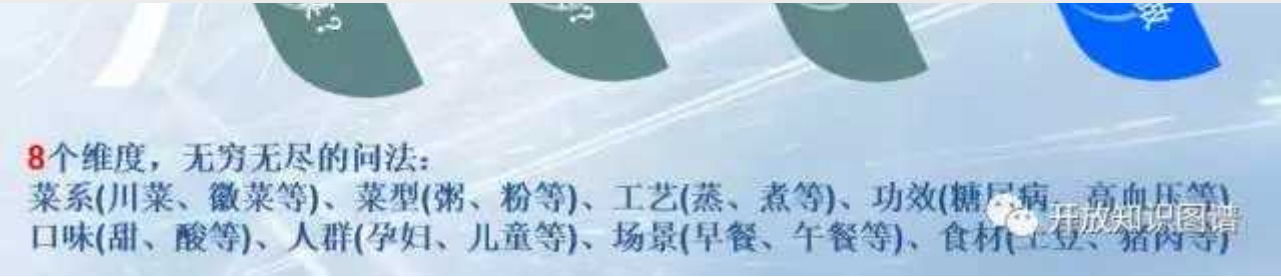
那下面我们有三种不同方式的可以去跟它沟通。第一个就是同意表达，就是说我就说了“水煮鱼的做法”就这样的一句话，那这个时候可能我们的用户在使用的时候，是各种各样的问法，因为你不清楚大家，你要怎么去问他。这个里面随便列了一些，比如说我针对厨房的一款产品，就跟他说水煮鱼，那你说它应该要出来一个什么样的结果，因为它这是有一个场景的限定的。另外比如说我想知道水煮鱼的菜谱，或者是查一下水煮鱼的做法，我应该怎么做水煮鱼等等。那么只要是大家问到这样的类似的，或者是同意的一种表达，那我们应该出来的结果都是同一个，就是我要教你做这个水煮鱼，怎么去做它，然后在同一个问题当中，我们现在是统计了一下，目前的语料当中显示，我们现在有超过40种的方法，都是同一个问题是这样的，我们要解决这样的一个问题。



知

写文章

...



第二个就是多个维度的问题。我们刚才说的这个话题，一个水煮鱼我们怎么去问它，下面就是在这个水煮鱼，我们从第一幅图看来，它有很多很多周边的一些维度，那么比如说就像食材，那食材我可以说洋葱能做什么菜，我也可以刚才说的那个草鱼能做什么菜，可能水煮鱼只是它的一个部分是这样的，然后还有功效，比如说减肥要吃什么菜对吧，尤其是现在健康饮食也是得到大家越来越关注的一个点。比如说一个肚子比较大的人，可能问他的时候，就问减肥要吃什么菜，还有菜系，比如说粤菜、川菜等等。另外还有一种组合的问法，比如说我现在冰箱里只有土豆和牛肉，那它能做什么菜，或者是能做什么不辣的菜，这样的一些非常复杂的一些问法，然后我们目前的这个维度一共有8个，如果把这8个维度，跟我们的食材本身，让它们交叉，相互的这样去组合的话，其实它的问法是无穷无尽的，你不清楚别人到底要怎么去问，所以这样的一个问题，你是没办法罗列，或者是用规则的方式去处理这样一个问题。



知

写文章

...



另外一个就是页面上的一些指令。这些指令跟我们本身的页面是有关系的，右下角的这张图，就是说宫保鸡丁怎么做，怎么做之后我们就会出来一个页面，这个页面上面有三个信息，一个就是有它的视频，右边是它的食材，再下面是它的步骤。那在这个页面的时候，我可能就不会再说我想看一下宫保鸡丁的视频，那在这个页面，我可能就会跟他说看一下视频，那这个就需要这样一个看信息，那在这个页面说看一下视频，那特指就是宫保鸡丁的视频，步骤也是一样的，然后除此之外，可能到步骤页面之后，我只会跟他说下一步，我不会说第三步或者是第四步等等。所以这一个菜谱，领域范围里面所有的这样的，我们说到三个部分的一些问法。



知

写文章

...



那么整个的厨房领域里面，我们处理流程是这样的，首先我们的这个产品，是用语音去跟它交互的，所以第一个部分我们要进行语音的识别。语音识别它不是一直都是监听的过程，就是说首先这个就像跟人说话这样的一个交流过程是一样的，那我可能首先打个招呼，然后再去跟你说话，那这个部分我们叫做语音唤醒。然后唤醒之后，我们就可以用语音去跟它说话，说完之后，它要把这个语音变换成文字。那这个环节叫语音听写，变成了文字之后，这个文字其实有很多的，也是有错误的信息的，那么前面这个部分，目前国内应该是科大讯飞做的是最好的，另外还有思必驰都会做这一块。那目前前面这个部分，我们也是调上科大讯飞的接口，那么调完了之后，我们就得到了文本。

那这个文本我们会发现，里面有很多很多不是我们想要的，举个例子说，就像鱼香肉丝怎么做？那最后我们调到检索之后，出来的文本是“雨”香肉丝，那么这个时候你再到实体里面去做匹配的时候，就会出错。所以我首先在做语义分析之前，第一步要进行文本的纠错，那文本纠错我们这个里面，主要用到两种方式，一种方式就是一个概率模型，就在我们的语料库里面，然后我们

做了一个字跟词之间，序列这样的一个概率模型，那么一句话来了之后，我就会看，这个词它出

现下一个词的概率是多高，然后对它进行纠错。另外还有一些本身实体，本身的名称，我们就到实体铺里面，去给它进行纠错是这样的。那么纠完错之后，下一步我们是进行垃圾过滤，因为我们这个里面一共有四类，其实我们分类的结果，应该是五个类别，就是4+1，那这个里面的垃圾来自于我们是不是在跟机器说话，或者说我问的是不是跟它相关的问题，或者我是不是在跟别人说话，以及我们这个里面，还有一些其他的，就是跟它无关的这样一些垃圾，要把它过滤掉。那么过滤的这个准则，主要还是用检索这样的一种方式，就是我们有问题相关的实体库，然后你来这一句话，我会跟它进行比较是这样。第三步我们进行文本分类，这些地方的主要工作，是分本分类的技术，我们已跨过了两个台阶。首先我们做了一些问句的模板，比如说什么怎么做，或者是什么菜怎么吃，我们把所有的跟菜谱相关的动词都拿下来了，应该是 231 个，然后再

知

写文章

也就是说我所有的正文本的标注是足够的。那这样的话，任何一句话来了之后，我们在每一个里面做了一个二分类这样的问题，现在基本上全部转换成 Deep Learning 这样一种方式。文本分类之后，我们就知道它是哪一个类别的，那这个时候我就把它里面的实体以及属性把它抽取出来，比如说红烧肉怎么做，那我就把红烧肉拿出来，以及“做”这样的动词拿出来，那我就知道它们的组合方式，接下来我就会生成这个逻辑表达式，逻辑表达式我们现在生成的就是“与或非”之间的关系，比如说就像土豆、豆角它们两者其实是连着的，比如辣不辣是“非”的关系。然后最后我们要到知识图谱中查找的时候，我们会生成这个 SPARQL 查询语句，这个语句还是 Apache Jena 里面开源出来这样的一种方式，后面我们也会提到。

做好了语义分析的这一步之后，我们就可以得到我们想要的答案了，那这个答案我们是分在三个地方的，一个就是数据库，比如说我们有一些图片的 ID，或者是实体的 ID，这些就在数据库里面根据 ID 把它拿出来。另外一个部分，有一些实体，我们是放在 Elasticsearch 里面的，这样的话我们从里面检索出来就好了。但我们有一些语义跟实体之间的推理关系的时候，那我们就把它放在知识图谱里，然后从知识图谱里面出答案。在这个的这个流程当中，是我们目前的一个基本的流程。

厨房领域的问答系统 ES VS. KG

ES搜索技术采用基于关键字的搜索方式，在检索的时候只关注是否有该关键词，而不关注该关键词的语义信息。
知识图谱根据该关键词的上下位关系进行搜索，语义信息丰富，搜索结果更准确。

比如“萝卜可以做什么菜”，ES检索时只会根据“萝卜”的关键词进行检索，而不知道“萝卜”包含“白萝卜”、“红萝卜”等上下位关系，因此检索结果不够精确。

知

写文章

...



之前的话，我们也曾经尝试用 Elasticsearch 去做过这样一件事情。那也就是说 ES 其实它只是个检索的功能，就是在 Lucene 存储的基础之上然后做的一个分布式的一个框架。那在这个里面，它主要是基于关键字这样的一个搜索方式来做，是否包含着这个关键字。但是知识图谱，它会有关键词的上下位这样的一层关系，就是说它具有推理的功能。那下面举个例子，比如我们说萝卜能做什么菜，会发现有胡萝卜什么炒肉，或者是白萝卜什么炖汤，那这样的就找不出来了。就是说在一个仅仅是从词匹配的角度上来说，那只能做到这一步，那这个时候如果我们有上下位的关系，那我知道这个萝卜下面它是包含着红萝卜或者是白萝卜，并且它们都是食材的这样一个分类。大家如果是一直跟着 CCKS 这样的一个规律，就会发现这个图其实跟 CCKS 第二届的时候，漆桂林教授讲的一个知识图谱分类的那一张特别像，当时写的是疾病的那样的图，其实我们把它运用到这样的一个食材当中来了。那这个就是我们目前所做的这一个领域问答里面的相关问题。

2 知识图谱技术的运用



知

写文章

...



下一个话题我们来说一说我们的知识图谱，在我们刚刚说的这样的一些问题当中，是怎么得到解答的。首先我们看一看，目前这样的一个整体框架。

2.1 整体框架

2.1.1 数据存储

我们从下往上看，首先数据我们要把它存在什么地方。我们要把它的基础数据，目前是用这样的

的一个分布式框架，那么基础数据，我们会放在 Hbase 里面，比如说我们菜谱的图片数据，那

这样的数据它是很大的一个数据集，那就会放在 Hbase 里面。然后我们需要处理的一些数据，一

些原始数据，我们也会放在 Hbase 里面，然后我们通过 MapReduce 的程序去得到我们想要的一个结果，我们想要这个结果的时候，我们就会把它同步到 MongoDB 里面去。因为 Hbase 跟 MongoDB 它们各有优势，Hbase 写入的速度，如果跟 MongoDB 比起来的，它要快了更多，但是它读取的速度，是不如 MongoDB。那也就是说我最后呈现的数据，要从 MongoDB 里面呈现出来，那么 Hbase 主要是做基础数据的存储是这样的。然后还有一些关系性特别强的数据，我们会放在 Mysql 里面，但是这一块基本上已经非常弱化了。

知

 写文章

因为它们之间要个问的组百，那这个时候，我们采用的是网络本体语言就是 OWL 这样的一种方式，但是这个除此之外，前面应该还有 RDF 这样一种方式，其实我们是直接伴随了 RDF，因为现在有参考了很多的论文之后，OWL 其实是比 RDF 更先进的一种方式，那我们才开始采用的是这样一种方式。

然后对于大量的数据存储，那我们就必须要找一个持久化的工具，因为 OWL 它是一个文件格式的一种方式，Neo4j 我们做了，但是最后我们线上的系统并没有采用它，原因是什么呢？因为 Neo4j 首先它是个收费的，它那个节点的个数是有限定的。那我们现在采用的是 TDB 的方式，TDB 也是 Apache 基金会里面出来的 TDB，后来我们也会讲它。


另外我们有一些索引的数据，比如说我们有小的指令级的，就像我们刚刚说的上一页、下一页、翻页、换页这样的生成并不多，并且它匹配的准确率又非常高，就是它在一模一样的情况下，我才会让它去做，这个地方，可能就不需要做泛化，所以我就会把它放到 Trie 树里面去。然后有一些实体会放到 Elasticsearch 里面去查询，还有一些小的数据集也要索引，那我就会放到 Lucene 里面，然后之后的缓存数据，还是用的 Redis 这样的一个分布式的缓存处理。

2.1.2 数据采集

那么下面一层就是数据采集，那么我们要想达到这样的一些数据把它存储起来，所以我们要采集 8 万个数据。首先我们做菜厨，所以一些垂直网站，就比如说掌厨，下厨房，还有豆果网一些数据，另外我们人工编辑了很多的数据，比如说问句。因为这个问句人工也是不可能一步把它收集完了，但这个里面我们也有一个算法去做这种大量的问句，我们说了刚才的问句，同一个问题的

问句是非常非常多的，那这个时候往往大致过程是这样的，首先把这个问句进行分析，首先找种子问句，种子问句拿到了之后，我们会对它进行分词，然后把每一个词到 word2vec 里面去训练，就是找它相关的词出来，相关的词把它们的位置序列记好，然后做笛卡尔积。这样的做完了，我就会生成大规模这样问句的数据，当然里面有一些是不正确的句子，那这个时候我们也会用我们文本纠错的马尔可夫链的方式，做它的概率模型去纠正它，那这样的话，我会生成大量的这样一些问句，然后人工也会筛选一些是这样的。其实我们也申请了一个专利，就是我们怎么进行问句生成的。然后动词的收集，就是跟领域相关的这些动词，我们觉得它是有穷尽的，就是它是能够拿的到，虽然我们的问句是无穷无尽的，那我们收集了200多个跟菜谱相关的动词，以及同义词的收集。比如说番茄炒蛋怎么做你能够出来，然后西红柿炒鸡蛋你就出不来了，这样是不对

知

 写文章 ...

之后还有一些开放的知识库，比如说同义词词林出来的，刚才漆教授给大家分析的这个 OpenKG，OpenKG 其实有很多的这种开放的数据源，而且是免费的。我们接下来在做音乐的领域的时候，清华大学就在 OpenKG 里面贡献了音乐的数据，全部是三元组的方式，也希望大家能够去打开 OpenKG，去看一下里面有很多免费的数据源是非常不错的，然后还有 WordNet 一个开放的数据。另外我还抓了很多的百科数据是这样。

2.1.3 知识库构建

之后我们在知识库构建的时候，首先就是分成两个步骤去做它，一个就是我们知识的融合。首先我们要做时序融合，就是你之前做的实体，它的实体链接是不是根据时间的推理，而它换掉了它本身的这样一个含义，进而做本体的扩充。还有一个就是多源融合，我们会看到我们的菜谱，不止来自于一家网站，这样就会形成同一道菜，但是它的菜名不同，那我们要做这样的多源的融合，把它做一个实体的匹配和概念的对齐。做后了之后，我们就要做知识计算，就是把这一些知识融合起来之后，我们哪一些是它的属性，哪一些是它的实体，以及它们两者之间的关系到底是什么？

2.1.4 数据访问

之后我们到数据的访问，我们用 SPARQL 的访问，还有自然语言查询。自然语言查询就是说红烧

肉怎么做，那我就给你出来结果，你也可以自己能够写到SPARQL的时候，那我的系统也是支持你能够出结果的，但是我们还有SDK的方式，放在你的系统里面可以用。当然我现在也把逻辑表达式的那个查询放出来了，如果你用过逻辑表达式，我也会自己把它转化成SPARQL语句然后给你出结果是这样的。

2.1.5 知识运用

知

写文章

...

问答当中。



下面我们就把前面那一幅图抽象成这样的一个图谱，然后它旁边有很多很多的属性，之后我们一

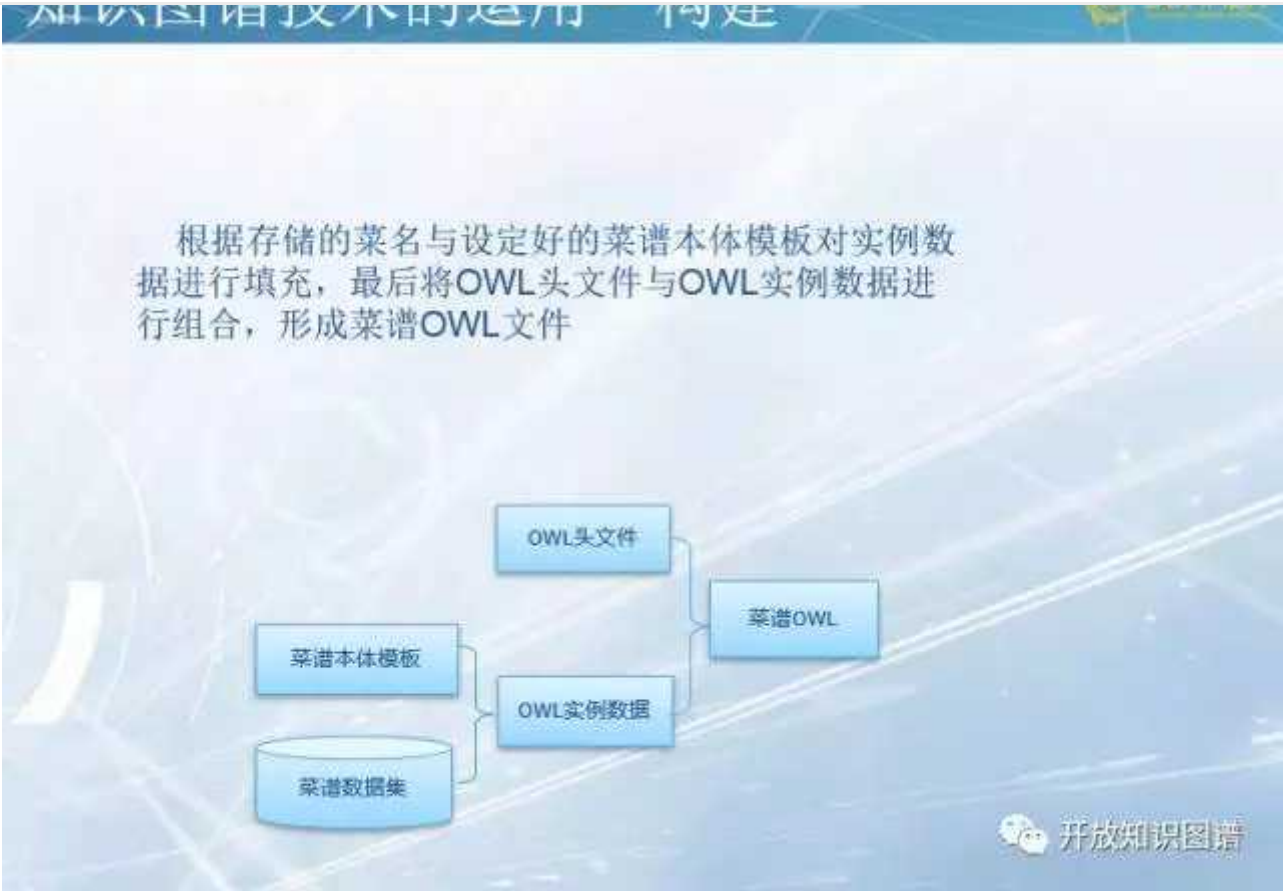
共抽出来 19 个菜谱的属性，那这个里面所有的跟它相关的东西，我们都可以，应该能问的出来，比如说就像人群，还有功效、菜系等等。

2.2 知识图谱的构建

知

写文章

...



接下来我们说一说知识图谱的整个构建过程，就是我们刚刚看到了，我们整个的一个图，抽象出来是这个样子的，那我们怎么把它构建出来呢？首先我们应该有菜谱的本体的模板，那这个模板里面，我会定义好，我有哪一些菜，有哪一些属性是这样的，然后下面我们有一个菜谱的数据集，那这个数据集我们是放在 MongoDB 里面，就是我们已有这样的一个数据把它放好。之后把这两者结合，我们就成了一个本体语言这样的实例数据，再加上我们本体语言的一个头文件，之

后我们就变成了菜谱的OWL文件。那这个文件到底展开什么样子的，我们下面看一下，这四个部

知

写文章 000

COOKINGTIME_CLASS = "<rdf:type rdfs:resource>cookingtime"/>"//烹饪时间	581	葱油拌饭	葱油拌饭	14215	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
HARDWARE_CLASS = "<rdf:type rdfs:resource>hardware"/>"//硬件设备	590	凉拌黄瓜	凉拌黄瓜	13982	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
RECIPEID_CLASS = "<rdf:type rdfs:resource>recipeid"/>"//菜谱ID	599	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
COOKINGMETHOD_CLASS = "<rdf:type rdfs:resource>cookingmethod"/>"//烹饪方法	602	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
FUNCTION_CLASS = "<rdf:type rdfs:resource>function"/>"//功能	605	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
SCENE_CLASS = "<rdf:type rdfs:resource>scene"/>"//场景	606	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
PROPLRANGE_CLASS = "<rdf:type rdfs:resource>proplrange"/>"//属性范围	607	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
PLAYGROUND_CLASS = "<rdf:type rdfs:resource>playground"/>"//游戏区域	608	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
AGREEMENTPOINT_CLASS = "<rdf:type rdfs:resource>agreementpoint"/>"//协议点	609	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018
COREDISHNAME_CLASS = "<rdf:type rdfs:resource>coredishname"/>"//核心菜名	610	凉拌金针菇	凉拌金针菇	13983	http://owl.loveowl.cn/	http://owl.loveowl.cn/	http://owl.loveowl.cn/	2018

<https://zhuanlan.zhihu.com/p/29177019>



知

写文章



从本体的构建，那这个本体是我们本身有的这样的一个菜谱数据的本身，然后我有两条线。我们首先不看那个黄色的，看下面的这样的一个部分，通常的做法是这样的，我们把它放到这个本体文件当中，我们刚刚看到 OWL 到底是什么东西，其实你可以放到 RDF 里面这个没所谓的，之后我们会有一个 URI 的一个映射文件，ttl这样的一种映射方式，然后来同时解析，然后在 TDB 里面，我们会根据 Jena 的 TDB 写一个 TDB 的增删改查，通常的这样一条线，这样它就能走完。

另外我们公司就想做一个红色的一条线，中间我是不是可以不生成一个 OWL 文件，我只要把我的文件，原始数据能够放到 TDB 里面这种格式就好了，因为它支撑这种查询就好了。这一部分研究我们应该也就结束了，就是我们可以直接把一个数据库里面的菜谱文件，然后直接把它放到 Jena TDB 这样的一个存储系统当中，然后这样的话，我们文件把它放上去了，然后我们怎么把它查出来呢？查的时候也有两种方式，一种方式根据我们自己的 CRUD 读出来，读到我们的数据集段中，然后通过 URI 把它映射成我们的一个本体模型，这样通过 SPARQL 就可以把它读出来。

另外 Apache 还提供了一个 Fuseki 的 SPARQL 服务器，我们直接 Load 进来，Load 进来之后我们通过 HTTP 往外读是这样的。但下面是一个推理规则的时候，我们也是这样，我们要有 rules

文件跟推理器，那整个的流程就是这样去做的。



知

写文章

...



2.3 搜索流程

后面我们看一下搜索的流程，那么整个搜索，我们有一个语义逻辑的表达式，我们把它放进来了，举了个例子就是马铃薯能做什么不辣的菜？那我们现在就可以把它搜索出来，怎么搜索？第一步这个语义分析结束之后，我就知道这个马铃薯它是一个实体，并且我要找的不辣的。那么下面我们就开始分析，马铃薯是 MUST，就是它一定要也，这个辣就是 MUST NOT，就是一定没有这样的一层关系。再下来我就要做同义表达，同义处理，就马铃薯它也是土豆，所有土豆也是 MUST。之后我在做这种实体属性，它们的映射关系，然后来生成这个 SPARQL 语句。跟我们的数据库查询有一点像，但是也不完全一样的这样的一种方式。那最后我们查询的结果是菜谱里面

的ID值是这样的，那有了ID值的话，我就可以从数据库，通过ID的方式把它拿出来，这样的话就

会非常快。

下面我们举几个例子来说一说我们语义搜索的一些结果，现在针对于这种单属性值这样的一种表现方式，我们是怎么去做的呢？你比如说土豆和豆角可以做什么菜？这两者之间是“和”的关系，就是它们两个都要有，就是土豆跟豆角，那这样的话，我们查询出它的结果就会是这个样子。

2.4 语义搜索

知

写文章

知识图谱技术的运用- 语义搜索

针对单属性、单属性值或多属性值的逻辑表达式查询：菜名查询、食材查询、其他属性查询。

例如“土豆和豆角可以做什么菜”，相应的SPARQL查询语句如下：

```
SELECT ?heat ?id ?dish WHERE { ?dish fa:hasNameIngredient fa:土豆 . ?dish fa:hasNameIngredient fa:豆角 . ?dish fa:hasId ?id . ?dish fa:hasHeat ?heat } ORDER BY DESC(?heat) LIMIT 100
```

土豆豆角



土豆豆角焖面

推荐 7分钟 辣



豆角土豆烩五花肉

推荐 3.5分钟 咸



土豆炖豆角

一般 5.5分钟 甜



土豆豆角烧腊肉

一般 7分钟 辣



土豆南瓜炖豆角

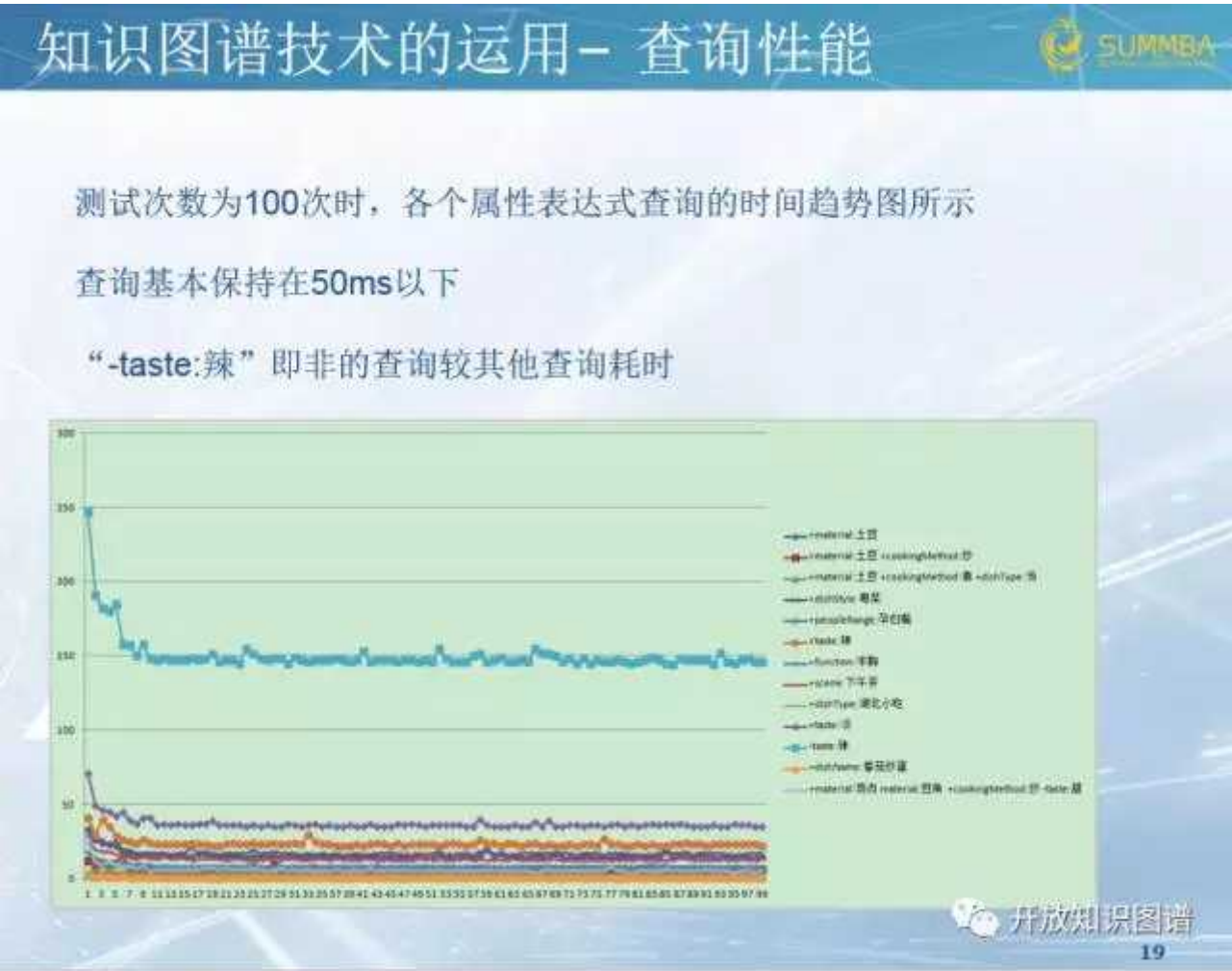
推荐 62分钟 辣

开放知识图谱

谱，这个首先我们是从属性基础角度上面去分析这一句话，就是说这个里面土豆，这个里面可以加豆角也可以不加，西红柿可以加豆角也可以不加，是这样的一种方式，是这样的一种或。然后最后它们出的菜，都应该是不辣的一个菜，所以我出来的结果是比如说西红柿炒土豆片，就是它们两个有吗，西红柿土豆炖牛肉是这样的。然后后面就是土豆跟豆角，它们两者之间的出现，因为它们是或，不是限定的，那这样的它们就出来了，那这个语句其实还是比较复杂的一个语言现象的处理方式。

知

写文章



后面我们看一下，我们知识图谱的一个性能问题。其实我们搭建知识图谱本身的时候，就刚刚开

始用OWL这样的一种方式把它放起来，然后发现这个文件越来越大，然后它的效率会越来越低是这样的。但后来的话我们采用了TDB的方式去处理了这个事情，这边是我们的一个测试，那么我们每一个测试是100次，然后各个属性，这个属性有一点小，比如说第一个是土豆，或者是土豆怎么烧，然后不辣等等，基本上我们所有的查询，都应该是在50毫秒之内能够完成它的结果。然后有一个特别的不正常，就是那个不辣，比如说我想做一个土豆不辣的菜，然后我们会发现这个查询，它的耗时超过了150，就是跨了好几倍这样的一种方式，那么这个原因是什么？就是说我们上面的标签没有不辣，我们要么就是甜，要么就是酸，要么就是辣，你如果要查它不辣怎么去做呢？那我就要查这个属性它是甜的，等于要把所有其他的属性都查了一遍，所以这个时候它就特别慢，那我们现在后来的这种处理方式是怎么去处理它呢？就是离线去处理，我们整个库当中的

知

 写文章 ...

3 曾经踩过的一些坑

后面一个部分我们看一下，我们在这个过程中遇到的一些问题，以及我们的解决方法。





知

写文章

...



首先就是相关性查询的问题，那么我们会遇到的第一个问题，比如说土豆可以做什么菜，然后我们就出现了一个结果叫紫苏炒三丁。因为我们的这个结果是从哪里找的，我们是从食材里面去找的，只要这道菜当中，包含着这个食材当中有土豆，那我们就会把它拿出来，那么就会发现它本身的这个菜的名称当中不包含，就查出来好像是错一样，就这种体验非常不好。后来我们就是处理它，怎么处理呢？我们首先离线去处理所有菜谱的名称，把菜谱名称当中含有食材的部分，单独拿一个列表，列出来，把这个字放进去，然后我们再查询的时候，就首先到这个里面去查，是不是跟它有相同的，然后再出这样一个结果，如果没有的话，那你只能从食材当中去查，然后我查完了之后跟百度比了一下，就是跟到百度的属性去找的时候，我们会发现有很多我们出来的结果跟它是重复的。

曾经踩过的一些坑

SUMMIT

知识图谱的同步问题

问题描述：知识图谱的构建依赖了知识库和采集的数据，在实际的项目中，发现知识库和菜谱数据会不断的更新，中间生成OWL文件时，数据得不到及时更新。

解决方法：生成两份OWL文件OWL1-ready, OWL2,读取含有ready的OWL1-ready文件提供API服务，采用定时任务更新没有ready的OWL2，生成OWL2-ready，并删掉OWL2，再重新读取OWL2-ready

知

写文章



第二个就是知识图谱的同步的问题。我们生成了这样中间的 OWL 文件，我们的实体库每天都在不断地更新，不断地更新，那你线上的 OWL 文件，因为你是从数据库里读出来，读成这个三元组的这种方式，那么我们怎么才能够把我们最新的，最实时的做好的这个数据，能够在我们的线下系统去使用？后来我们生成了两份文件，就是 OWL1—ready，还有一个是 OWL2 是这样的，然后每次来读取的时候，我就搁一个小时一次，去看一看这个文件夹里面，到底谁是 Ready，就把它读走，然后读走了之后，这一份就把它删掉是这样的。然后它在更新的时候，更新完了之后，它会变成 Ready，这样两份的交替使用，那线上它就会做这样一个无缝的切换。

曾经踩过的一些坑

检索结果排序

问题描述：在使用知识图谱检索过程中，发现食材、功效、菜系等检索结果并没有进行排序。

解决方法：在图谱中增加“heat”类及“hasHeat”属性，根据检索结果进行排序，同时增加推荐知识库作为检索结果的补充。

heat	id	dish
------	----	------

知

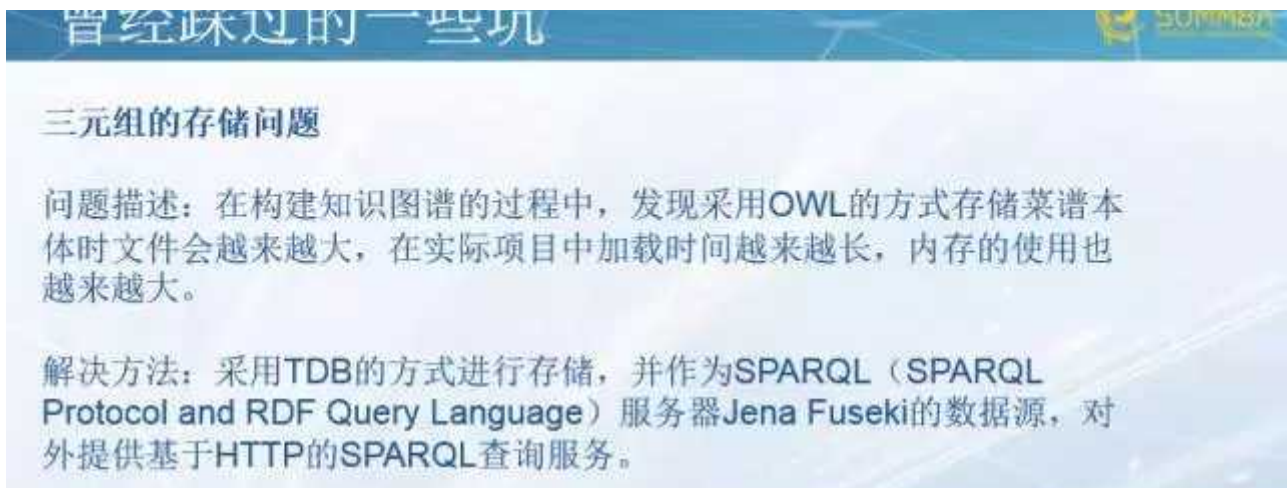
写文章

4	fa:0.7871752910626082	fa:zhangchuj2040	fa:豆角土豆煨五花肉
5	fa:0.7034956909518111	fa:zhangchuj2790	fa:土豆炖豆角
6	fa:0.5875672312488012	fa:zhangchuj3270	fa:土豆西红柿汤
7	fa:0.57856794046107	fa:zhangchuj10189	fa:土豆豆角辣蹄肉
8	fa:0.6759086074468732	fa:zhangchuj11252	fa:南瓜西红柿土豆汤
9	fa:0.5164840131078767	fa:zhangchuj9061	

开放知识图谱

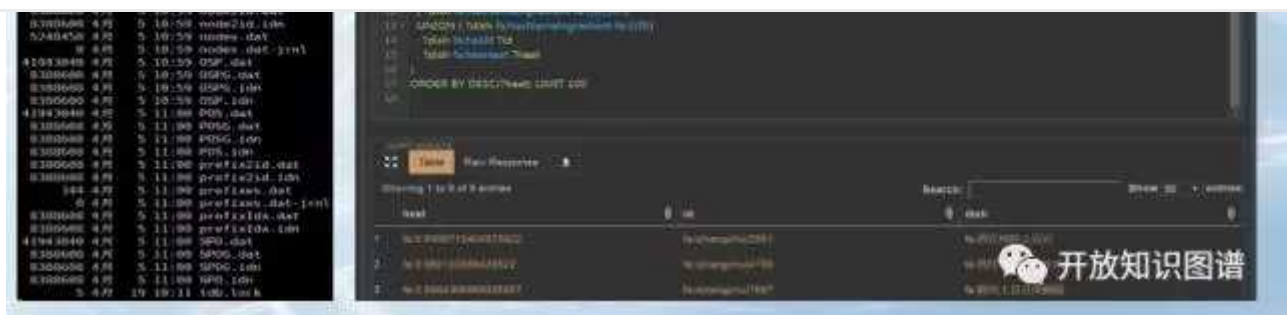
23

知识图谱里面的搜索是有一个问题的，就像我们经常用 ElasticSearch 的人就会发现，检索里面的排序其实是非常容易去做的，本身底层就写了一个排序打分的 TF-IDF，这样的一个模型。然后你会发现你用知识图谱的时候，你非常不习惯，因为它附近的这些节点的权重都是一样的。比如说我们土豆能做什么菜，那你所有出来的这些菜的话，在我们知识图谱里面，映射的本身其实它是扁平的，就是这个土豆它是一个节点，能够到哪一些菜谱上面，都可能拿出来。那这个时候你就会发现，如果是在产品当中的使用，你就会发现你所列出来的那些菜大家都不认识，我都没有见过，这个体验也是很不好的。后来我们又在知识图谱的这个属性当中，去加了一个热度的一个值，这个热度到底是多大，这个热度我们主要是通过点击次数去计算，就是哪一道菜，就是把它们查出来之后，它们哪一道菜的点击量会更大，它们更热，然后就会把它调制的参数把它往前排是这样的。



知

☰ 写文章 ○○○



最后一个看一下三元组的一个存储问题，刚才也提到了，就是我们实际在使用，全部用 OWL 这样的一种方式存储的话，那这个文件会越来越大，越来越大。其实它的查询是以加载到内存当中的这样一种方式，那这种太大的时候，就会非常影响你的速度，并且你的整个机器都会被拖垮。如果你是分析实验的话，你可以这样去做，但真正的你的这个数据量开始越来越大，那现在我们的菜谱的节点，已经超过了 80 多万，所以三元组的个数会更大，这样的一种方式。然后我们采用的就是 TDB 的这样的存储方式，刚刚也说过了。下面是 TDB 的一个页面，左边这一幅图是 TDB 存储完了之后，它里面的目录文件，右边是它提供的一个页面，我们可以看的到，用 STDB 的方式我们直接可以读到它。

4 遇到的一些挑战与困难





知

写文章



最后说一下我们遇到的一些问题，跟大家探讨一下。我们现在是在做菜谱的整个的这样一个方式，那在这个里面，我们还是花了很多的时间，花了很多的人力去做这样一件事情。然后我们刚刚 PPT 的开头，跟大家说到我们做厨房领域，是有四个方面的这样一个问答，接下来我们一个月视频怎么办，那我们同样也要用这样的一种方式去做，我们在菜谱里面下了更多的工夫，我们做了这么多的实体，三元组它的查询方式，那我们现在做音乐，我们怎么能够把它移植过去？就是我们一个跨领域的问题，那你会发现，音乐的时候其实它的那个属性是不一样的，它比如说有劲爆的，有小清新的等等，你在做这个，除了那个基础工作是一样的，就是这一套理论，这一套分布式，或者查询这样的一种方式相同的，但是这种属性和属性值，好像我们要重新来过一次。


第二个问题，我们看到我们语义理解过程当中，其实我们主要还是在做文本的分类，说明分完类之后，我们去做它的属性抽取，然后生成一个逻辑表达式，我们就知道这一句话到底在说什么，或者说它想查询什么，其实在于这样的一件事情。其实知识图谱的运用，还没有把它运用到自然语言理解这样的一个事情当中来。我们用多少数据的一个知识图谱的时候，我们就不需要再做这样的一个事情，那我们讲的一句话，就把它直接应用到我的知识图谱的应用里面去，如果它在我的每个节点上面都有体现，并且这个节点之间是有关系，那我就知道你就在说什么，我就不需要再做前面大量的这些工作，包括这种分类，是不是也就不需要了。因为我都知道你在说什么了，

那我一定知道你的类别是什么，但是这一块可能要慢慢去积累，去做这样的一件事情。总之目前

这样的一种方式，应该是还不能够达到语义理解的这样一个层面。

后面一个话题是这样的，比如是我们现在做的一些事情，其实我们用了大量的数据量，去训练一个模型。其实这个模型怎么形成的，就是一个规律。在这个过程当中，其实表达的意思是这样，就像我们用一个二元一次方程，如果我们用深度学习的方式怎么去解二元一次方程，那我本身这样的一个表达式，可能就是 $ax^2 + bx + c = 0$ ，这样的一个式子，最后我解出 x_1 跟 x_2 等于多少就行了。如果我们用这方面的方式去做，我会怎么做？我会把 a ，比如说从 1 到多大的一个数

知

 写文章

我就一定能够把 x 就解出来。那这样的话，我们用了这么大量的数据，其实我们解决的一个问题是什么？我们其实就是找了求根公式是不是就可以了。如果知道这样的一个公式就可以了，所以我们大量的工作需要放在，找这种大量的数据集，并且去做标注，还是我们应该去找的求根公式，当然这个找大量的数据集，肯定是找求根公式的一种方法，这个是毋庸置疑的，但是我们是不是还有一种方式，直接有另外的一种方式，能够找到这样的求根公式，能够去处理我们这样的问题。那这个就是我们也在尝试着去做一些事情，尤其是这个部分，我们想去做一些中文汉字和词，它们本身这样的一层关系，就比如说我们的汉字常见的有 3000 多个，你把那个大的词典拿出来，可能是 7000 多个，然后我们的汉语当中所有的词，大词典当中的词，现在收录的是 58000 多个，这样的加起来的话应该不到 7 万。那如果把它们之间的关系，以及它们之间的含义内容表示清楚，用某种方式表示清楚，那我们是不是就可以能够映射本身的一个图谱，或者是帮助我们的语义理解，那这个也是我们研究的一个话题。

5 总结

知

 写文章

最后总结一下，首先我们觉得这个问答系统，一定是未来的一个发展趋势，并且它在处理这样的具体的领域问题的时候，我觉得它还是能够带来一些实际的价值。因为未来这样的一个，我们所有使用的这样一个工具，它是有很多的历史变迁，比如说我们以前用电脑的时候，用鼠标用键盘这样的输入方式，后来我们再用手机的时候，跟你的键盘、鼠标的输入方式已经完全不同了，对不对，你不可能拿着手机在手里，拿着一个鼠标去点它，或者是键盘去敲它，你都是手去触碰它。然后未来的这种发展，它应该是两极分化的一种方式，要么就是没有屏幕，比如说一些音响的出现，它是没有屏幕的，那么你怎么去跟它交流，你不可能用鼠标，手滑这种趋势是不可能的，你一定是用语音的方式去跟它交互。

另外一种就是屏幕会越来越大，比如说这个屏幕大到这个样子的，或者说整面墙都是这样子的，那你可能去鼠标或者手滑，你一定是语音这样的一种方式，那我们觉得语音的问答这样的一种方式，是未来的一个趋势。第二个在问答系统当中，知识图谱在目前看应该还是比较科学的一种方法。第三个其实知识图谱的它的使用价值，应该是很多很多的，但是终将能够把它转换成运用的，这个你们应该还有很多很多的挑战，那我们觉得也是一个值得深入和广泛研究的一个领域，也希望大家能够更多的研究和探讨，谢谢大家。

- end -

Tip：索答科技已经将 50w 菜谱本体信息在 OpenKG 上开放出来，每个菜谱包含菜名，食材，味道，烹饪时间等属性。链接

知识图谱 语义分析 人工智能算法

☆ 收藏 ↗ 分享 ⚠ 举报

知 📄 写文章 ...



7 条评论



写下你的评论...



不留
PPT 做的真心好
4 个月前

SUMBA

索答科技（作者） 回复 不留
谢谢
4 个月前

🗨 查看对话



樱落清璃
下午好！看了您的分享，感觉非常有收获。
3 个月前

SUMBA

索答科技（作者） 回复 樱落清璃
谢谢
3 个月前

🗨 查看对话



具的历吉

2 个月前



赵二铮

你们的程序中会大量用到OWL API么

2 个月前

SUMMER

索答科技（作者） 回复 赵二铮

查看对话

写文章

知