1. 知识图谱;
2. <https://www.bilibili.com/video/BV17s411n75M?from=search&seid=11298817144614489137>
3. P1, 知识图谱概论; 人的记忆偏重关联; web, 以链接为中心的系统; 语义网, 从链接文本到链接数据; 事物之间有多种类型的连接; 谷歌知识图谱; 搜索结果中体现关联; Facebook兴趣图谱; knowledge graph; KG的概念演化; 语义网络; 本体论; 从超文本链接到语义链接; 链接数据; 知识图谱; 知识表示与知识库; 框架系统、产生式规则、描述逻辑; KG辅助搜索; 搜索的理想是万物的链接, 搜索的理想是事物的搜索; freebase; wikidata; KG辅助问答; KG辅助决策; Palantir; kensho; KG辅助AI, 常识推理; KG本质; web视角, 数据之间的语义连接; NLP视角, 文本中抽取语义和结构化数据; KR视角, 利用计算机符号表示和处理知识; AI视角, 利用知识库辅助理解人的语言; DB视角, 用图的方式存储知识; 聪明的AI, 深度学习; 有学识的AI, 知识图谱; 典型知识库项目; 各种知识图谱项目; Cyc, 常识知识库; 术语terms+断言assertion; Wordnet; ConceptNet, 也是常识知识库, 三元组形式; Freebase; Wikidata, 构建全球最大的免费知识库; DBPedia; YAGO; Babelnet, 多语言词典知识库; NELL, 采用互联网挖掘的方法从web自动抽取三元组; 三元组就是主谓宾; 微软concept graph; Part 3, 知识图谱技术概览; 知识图谱的技术体系; 什么是知识表示; 语义网知识表示框架; RDF, triple-based assertion model; subject->predicate->object; RDF graph, directed labeled graph; 基本数据模型, 有向标记图; RDF的数据化格式; RDFS, simple vocabulary and schema; RDF的序列化格式; OWL, web ontology language; OWL extends RDF schema; OWL示例; 对称属性, 传递属性, 属性链; SPARQL简介, RDF查询语言; SPARQL示例; 知识图谱上的查询很大程度上属于子图匹配; JSON-LD, 数据交换格式; RDFa, HTML5 MircoData, 在网页中嵌入语义数据; 知识图谱的分布式表示, KG embedding; 张量分解; RESCAL, NTN, TransE; 知识抽取, NLP+KR; 知识工程, 正则表达式, 模板匹配, 规则约束; 基于本体的抽取, 知识挖掘; 基于模型的抽取; 模型, 训练; 知识存储; 知识问答, 以直接准确的方式回答用户自然语言的提问; KBQA例子; KBQA基本实现流程; 知识推理; 基于描述逻辑的推理, 本体推理; 基于统计规则挖掘的推理; 基于表示学习的推理--Representational Learning, 将实体和关系都表示为向量, 通过向量之间的计算代替图的遍历和搜索来预测三元组的存在; 知识融合; 典型工具Dedupe; 典型工具LIMES; 知识众包, Wikibase; Part 4, 典型案例简介; Open PHACTS; 中医药知识平台; 电商知识图谱; 让判断更智能; 企业知识图谱应用SAP; 链接金融数据; 大英博物馆, 知识图谱与本体设计; 大英博物馆, 语义搜索; 大英博物馆, 后端平台; BBC, 打通BBC全领域; 知识图谱在媒体出版行业应用; 知识图谱在伦敦奥运会的应用; BBC music-using linked data; (2020-9-18)
4. P2, 知识表示与知识建模; Part 1, 早期知识表示简介; 知识的特性, 相对正确性, 不确定性, 可表示性&可利用性; 知识分类, 常识性知识, 事实性知识, 确定性知识, 逻辑性知识; 一阶谓词逻辑; Horn逻辑, 一阶谓词逻辑的子集; Prolog语言; 描述逻辑, 一阶谓词逻辑的可判定子集; TBox语言, ABox语言; 产生式规则; advantage, 自然性, 模块性, 有效性, 清晰性; disadvantage, 效率不高, 不能表示具有结构性的知识; 框架; 槽名, 约束; 利用框架在知识库中匹配得到实例; advantage, 知识库质量高, 允许数值计算; disadvantage, 构建成本高, 形式不灵活; 语义网络; advantage, 结构性, 联想性, 自然性; disadvantage, 非严格性, 处理上的复杂性; Part 2, 基于语义网的知识表示框架; W3C推荐的语义网标准栈; 知识图谱所需使用的主要技术标准, SPARQL+RDF; 互联网发布信息规则(五星规则); linking open data; 语义网络-本体论-web-语义网-链接数据-知识图谱; RDF, 资源描述框架Resource Description Framework; 三元组, 主谓宾; RDF is also a graph model to link the descriptions of resources; vertex, edge, vertex; 使用namespace进行缩写; some objects are literals, not resources; RDF空白节点, 可以用于匿名和二跳; RDF是数据类型, 不是序列化格式; 开放世界假设, 与经典系统的封闭世界假设相对应; 不知道并不代表不存在; RDF允许分布式地定义知识; 分布式定义的知识可以自动合并; 带标注RDF; (s, p, o): lambda; RDF和RDFS; RDF schema; eg, <rdfs:subClassOf>; RDFS推理示例; OWL和OWL2; RDF(S)表达能力上的缺陷; OWL词汇; 等价性声明, 声明属性的传递性, 声明两个属性互反, 声明属性的函数性, 声明属性的对称性; OWL词汇扩展, OWL中的其他词汇; OWL2语言家族树; OWL2 QL, 表达能力, 可以优化到多项式对数复杂度; OWL2 EL, 概念层复杂, 实例层很小; OWL2 RL, 函数性, 互反性, 对称性; SPARQL简介; SPARQL查询结构; SPARQL查询基本构成; RDF中的资源, 以?或者$指示; SPARQL查询语言; prefix是命名空间的声明, 为了写得更加简洁; 关键字optional, 关键字filter(加一些数值上的约束), 关键字UNION, 关键字FROM引入其他本体或者可访问的知识库; SPARQL查询图; 更多SPARQL查询示例; 本体可以填充知识与查询之间的语义间隙; RDF提供了知识建模地灵活性, 但查询可能成为噩梦; SPARQL用于新药发现的案例; 跨知识库SPARQL查询示例; JSON-LD, 通过引入规范的术语表示, 比如统一化表示name homepage image的URI, 使数据能够被机器理解; 围绕某类知识提供共享的术语; RDFa, 扩展了XHTML的几个属性; RDFa工作原理, 通过引入名字空间的方法在已有的标签中加入RDFa相应的属性; HTML5 Microdata, 在网页标记语言嵌入机器可读的属性数据; 浏览器可以很方便地从网页上提取微数据实体、属性以及属性值; RDF+SPARQL对比ER(entity relationship)+SQL; RDF语义模型--关系显示定义; 关系模型-关系隐式声明; 当数据发生了变更, 语义模型如何应对(changes are easy to make); 当数据发生了变更, 关系模型如何应对(changes should be avoided at all costs); 数据的智能性如何体现, smart data + uniform inference engine; Part 3, 典型知识库项目的知识表示; 各种知识图谱项目; DBpedia抽取知识示例; YAGO2 knowledge base, 涉及实体消歧; Freebase, 数据已经全部导入wikidata; 目前为止的知识图谱描述的都是事实性知识; 多元关系, compound value types; wikidata知识表示元素及示例; 多元关系-wikidata; ConceptNet基本表达要素; 多元关系, ConceptNet5; NELL抽取结果示例; 知识表示, 多元关系; 实用的知识表示, 接近人的自然语言, 够用的表达能力, 易于扩展; Part 4, 基于Protege的知识建模实践; Protege使用, 知识图谱示例; Protege已经是相对较好的了; (2020-9-23)
5. P3, 知识抽取与挖掘1; a是rdf:type的缩写, 代表is a; Part 1, 知识抽取任务定义和相关比赛; 背景, 大数据时代; 知识图谱数据来源; 示例应用, 音乐领域; 数据获取, 数据连接, 查询访问, 各种应用; 知识抽取, 从不同来源、不同结构的数据中进行知识提取, 形成知识存入到知识图谱; 结构化数据, 半结构化数据, 纯文本数据; 知识获取关键技术和难点; 从结构化数据库中获取知识, D2R; 从链接数据中获取知识, 图映射; 从半结构化数据中获取知识, 使用包装器; 从文本中获取知识, 信息抽取; 什么是知识抽取; 子任务; 命名实体识别; 术语抽取; 关系抽取; 事件抽取; 共指消解; 相关竞赛与数据集; message understanding conference; 命名实体识别(named entity recognition); 共指消解(co-reference resolution); automatic content extraction; 实体检测和识别(entity detection and recognition); 数值检测与识别(value detection and recognition); 时间表达检测与识别(time detection and recognition); 关系检测与识别(relation detection and recognition); 事件检测与识别(event detection and recognition); TAC knowledge base population; 实体发现与链接(entity discovery and linking); 槽填充(slot filling); 事件抽取(event); 信念和情感(belief and sentiment); 端到端冷启动知识构建; semantic evaluation; 应用领域; 实体抽取定义; 抽取文本中的原子信息元素(人名、组织/机构名、地理位置); 实体抽取举例; 序列标注方法; 人工特征(词本身的特征, 前后缀特征, 字本身的特征); 实体识别的序列标注实例; HMM, 隐马尔可夫模型; CRF, 条件随机场; LSTM+CRF; 方法效果比较; 实体抽取--参考文献; 实体识别与链接; wikipedia miner; DBpedia spotlight, disambiguation in local context; OpenCalais; 实体链接; 属性消歧; 上下文消歧; 开源系统; 关系抽取简介; enterprise knowledge graph; 方法分类, 基于模板, 监督学习方法, 弱监督学习方法; 基于模板的方法--基于触发词的pattern; 基于依存句法分析的pattern; 基于模板的方法的优缺点; advantage, 在小规模数据集上容易实现, 构建简单; disadvantage, 特定领域的模板需要专家构建, 难以维护, 可移植性差, 规则集合小的时候召回率很低; 监督学习; 确定实体对的情况下, 根据句子上下文对实体关系进行预测; 监督学习-特征, 机器学习方法特征设计, 深度学习方法特征设计; 监督学习--深度学习方法; pipeline方法, 识别实体和关系分类完全分离; joint model, 实体识别和关系分类共同优化; pipeline方法, CR-CNN模型; Att-CNN模型; Att-BLSTM模型; 监督学习--效果对比; 监督学习--深度学习方法; Joint Model -- LSTM-RNN模型; 监督学习优劣; advantage, 准确率高, 标注数据越多越准确; disadvantage, 标准数据成本太高, 不能扩展新的关系; 半监督学习; 远程监督; 远程监督--优劣; advantage, 可以利用丰富的知识库信息, 减少一定的人工标注; disadvantage, 假设过于肯定, 引入大量噪声, 存在语义漂移, 很难发现新的关系; 半监督学习--bootstrapping; bootstrapping--优劣; advantage, 构建成本低, 适合大规模构建, 可以发现隐含的新的关系; disadvantage, 对初始给定的种子集敏感, 存在语义漂移问题, 结果准确率较低, 缺乏对每一个结果的置信度计算; 关系抽取--参考文献; 事件抽取; 事件抽取的定义; 事件描述, 事件触发, 事件元素, 元素角色; 更多事件抽取示例; 事件嵌套; 事件抽取任务; 识别事件触发词及事件类型, 抽取事件元素同时判断其角色, 抽出描述事件的词组或句子, 事件属性标注, 事件共指消解; 事件抽取的pipeline方法; trigger classifier, argument classifier, role classifier, attribute classifier, reportable-event classifier; 典型的分类特征; 事件抽取的联合抽取方法, 性能衰减严重; 一般的解决方法, joint inference; joint modeling; joint modeling with structured prediction; 几种方法的trigger和argument抽取效果; 基于深度学习的事件抽取方法, 好处是端到端; 基于动态多池化卷积神经网络的事件抽取方法; 扩充语料的方法; 从网络获取事件信息; 参考文献; 面向结构化数据的知识抽取; 垂直领域的知识往往来源于支撑企业业务系统的关系数据库; R2RML; 标准与工具; direct mapping idea; direct mapping, encoding principles; R2RML概述; triples maps; 示例; 数据库表映射; R2RML views; linking two logical tables; ontop, 把关系数据库映射成虚拟RDF; 面向半结构化数据的知识抽取; 百科类知识抽取; generic infobox extraction, 同一属性名不做映射, 保持wikipedia中原有内容; mapping-based infobox extraction, 定义depedia ontology, 将属性做好二次对齐; web网页数据抽取, 包装器生成; 网页中的信息; 如何获取网页信息--手工方法; XPath是一种用来确定XML文档中某部分位置的语言; 通过CSS元素实现对网页中元素的定位; 包装器简介及描述方式, 包装器是一个能够将数据从HTML网页中抽取出来, 并且将它们还原为结构化数据的软件程序; 包装器归纳; 网页输入-网页清洗-网页标注-wrapper space生成-wrapper评估-输出包装器; 准确率和召回率; 一个简单例子; 为什么要有自动抽取, 手工标注工作量大; 自动抽取, 将诸多网页通过聚类分成若干个组; 两个相似网页的比较过程; web table抽取简介; 什么是表格的实体链接, 将表格中各单元格的字符串映射到给定知识库的实体上; 表格实体链接的步骤; 候选生成-实体消歧(位于相同行或者列的字符串可能相关); 生成三元组; 抽取框架; 在线百科知识抽取技术应用--佛学知识图谱构建; knowledge collection(Buddhist figures); knowledge fusion(Buddhist figures); 主语融合, 同名不同实体, 同实体不同名; 谓语融合, inforbox属性; 宾语融合, 单值属性, 多值属性; knowledge completion(Buddhist figures); application; 实践任务1, 属性-值抽取, 不同的目标属性有不同的模板; 实践任务2, 实体清理; (2020-9-27)
6. P4, 知识抽取与挖掘2; Part 1, 面向文本的知识抽取; 关系抽取分类; deepdive关系抽取实战; KBC系统, 输入半结构化数据, 输出结构化数据库; deepdive; KBC流程; 特征抽取, 专业知识融合, 监督学习, 迭代优化; 框架实战, 抽取上市公司中的股权交易关系; 工程组成; 先验数据导入; 待抽取文章导入; 工作流; 文章数据预处理, 对文章数据进行NLP解析, 为后续特征抽取做准备; 候选实体抽取; 候选实体对生成; 特征抽取; 样本打标; 因子图构建; 因子图定义; 吉布斯采样; 权重学习; 其他配置文件; 迭代调试; 总结; 开放域关系抽取; IE的发展趋势; 主要系统; 第一代OpenIE系统; challenges, 关系不一致不准确, 提取的关系不包含有效信息; 第二代OpenIE系统, 更深入研究了句子的语法特性; 更多进展; OpenIE的应用, 直接回答问题, 用作其他NLP任务的特征; Part 2, 知识挖掘; 实体链接简介, 给定一篇文本中的实体指称(mention), 确定这些指称在给定知识库中的目标实体(entity); 实体链接基本流程; 输入-实体指称识别-候选实体生成-候选实体消歧-输出链接; 示例1, 百科型知识库; 示例2; 示例3; 总结; 知识规则挖掘; statistical schema induction--主要方法; 关键规则挖掘ARM; 统计关系学习; 根据已知三元组对未知三元组成立的可能性进行推测, 可以应用于知识补全; 基于图的方法, 将连接两个实体的路径作为特征来预测其间可能存在的关系; 路径排序方法--path ranking algorithm(PRA); 知识图谱表示学习; 表示学习的意义, 建立统一的语义空间, 语义可计算; 知识图谱表示学习, 实体预测和推理, 关系推理; TransE; 最小化势能最稳定; 克服transE的缺陷, TransH, TransR; 属性如何表示; 分而治之; 路径的表示学习; 性能比较--三元组分类; 性能比较--链接预测; 加入规则的表示学习; 多模态的表示学习; 基于知识图谱图解钩的表示学习, 哪些数据可以用来描述实体--实体周围的实体、从一个实体到这个实体的连通路径; 总结和挑战; (2020-9-29)
7. P5, 知识存储; Part 1, 从一个例子开始; 音乐知识图谱schema定义; 图谱数据生成; 音乐图谱数据生成器; 图谱存储工具--图数据库; 开源数据库--Apache Jena; 安装Apache Jena; 启动服务; Fuseki访问界面; 增加数据库; 创建数据库; 导入数据; 准备查询; 查询例子; 查询结果; 查询例子; 查询结果, 计算某一个专辑的曲目; 查询例子; 更新例子; Part 2, 图数据库介绍; 图数据库分类; 开源图数据库介绍; 开源数据库--RDF4J; 开源图数据库--gStore; 商业图数据库介绍; Virtuoso; Allgrograph; Stardog; 原生图数据库介绍; Neo4j; Neo4j--数据结构; Neo4j--优点; Neo4j--安装(docker); Neo4j--官方web前端; 数据导入; 数据存储; 查询数据; Movie graph示例, 查询两个节点间的最短路径; 原生图数据库--OrientDB; 原生图数据库--Titan; Benchmark; 图数据库vs图计算; Part 3, 图数据库实现细节; conversion of SPARQL to SQL; is that all; dictionary for strings; indexes for commonly used triple patterns; why sort order matters for joins; RDF-3x, compression scheme for triplets; compression effectiveness vs efficiency; back to the example query; selectivity estimation; handling updates; differential updates; example, columnstores; property tables; challenges and opportunities; (2020-9-30)
8. P6, 知识融合; 目标, 融合各层面的知识; 等价实例, 等价类, 等价属性; sameAs关系的发现是重中之重; 来源于不同知识库的同一实体; 知识图谱的构建经常需要融合多种不同来源的数据; 实体对齐是知识图谱融合的主要工作; 中文百科中的等价实例; 概念层知识融合; 跨语言知识融合; 知识在线融合; Google knowledge vault; 各种相关的名词术语; 知识融合的主要技术挑战; 数据质量的挑战, 数据规模的挑战; 知识融合竞赛--OAEI; OAEI-2017 schedule; OAEI-tracks; 知识融合的基本技术流程; 本体对齐+实体匹配; 数据预处理; 语法正规化, 数据正规化; 记录链接; 属性相似度, 实体相似度; 怎样计算属性相似度; 属性相似度--编辑距离; 最小编辑距离Levenshtein distance; 动态规划计算最小编辑距离; Dice距离; Jaccard系数; 属性相似度--基于向量的相似度; tf-idf similarity; 怎样计算实体相似度--机器学习方法; 聚合, 加权平均, 手动制定规则, 分类器; 聚类, 层次分类, 相关性聚类; 层次聚类; 相关性聚类; Canopy + K-means; 知识表示学习--知识嵌入; 实体与向量之间的关系; 怎样计算实体的相似度, 知识表示学习; 知识嵌入, TransE模型; 实体与向量之间的关系; 如何链接实体, 找最近的链接; 分块; 动机, 降低时间复杂度; 关注数据规模问题; 常用的分块方法; 基于Hash函数的分块; 邻近分块; 负载均衡; 典型知识融合工具简介; 本体对齐--Falcon-AO; 相似度组合策略; 语言学可比性, 结构可比性, 映射单元集成, 映射单元选取算法; Falcon-AO分块; 本体划分, 概念间的结构亲近性计算; 本体划分算法; 本体分块的构建; 本体对齐; Graph match for ontology, 使用RDF二部图来表示本体; 实体匹配--Dedupe, 是一个用于模糊匹配, 记录去重和实体链接的python库; 指定谓词集合&相似度函数, 找到最优谓词集合来分块, 训练逻辑回归模型; 实体匹配--Limes; 知识库-样本选取(样本之间距离尽可能大)-三角不等式过滤-相似度计算-序列化-知识库; 实体匹配--Silk; 知识库-预匹配-链接-过滤-输出; 典型案例简介; zhishi.me中的实体融合; 等价实体; 解决方案; workflow--挖掘等价属性; workflow--挖掘匹配规则; the wrapper算法; E-step; M-step; 似然函数; evaluation--precision; evaluation--新发现的matches; results; 网页访问等价实例; 典型案例二, OpenKG的链接百科; 链接结果; 详细结果; 链接方法; LIMES实战演练; (2020-10-7)
9. P7, 知识推理; OWL本体语言; 最规范(W3C), 最严谨(描述逻辑), 表达能力最强(一阶谓词逻辑的子集); RDF语法, 三元组; 描述逻辑系统; 概念, 关系, 个体; TBox--泛华的知识; ABox--具体个体的信息; KG可满足的, 概念可满足的; 算子, 交, 并, 非, 存在量词, 全称量词; 描述逻辑与OWL词汇的对应; 知识推理任务; 可满足性(TBox), 分类(TBox), 实例化(ABox); 可满足性, 本体可满足性, 概念可满足性; 分类, 找出概念标签; OWL本体推理; 分类的例子; 实例化, 计算属于某个概念或关系的所有实例的集合; 实例化的一个例子; 本体推理方法与工具介绍; 基于Tableaux运算的方法; 构建Abox, 检测可满足性; Tableaux运算的正确性, 基于Herbrand模型; 相关工具简介; FaCT++; Racer; Pellet; HermiT; 基于逻辑编程改写的方法; 可以根据特定的场景定制规则, 以实现用户自定义的推理过程; Datalog语言; 原子(Atom), 规则(Rule), 事实(Fact); Datalog推理举例; 相关工具介绍; KAON2; RDFox实践; 本体(Tbox), 数据(Abox), 自定义规则; 基于一阶查询重写的方法; Step 1, 重写为Datalog查询; Step 2, 将数据库关系表达式映射成Datalog原子; Step 3, 将从SPARQL以及数据库重写过来的Datalog规则整合进行查询; Ontop; 基于产生式规则的方法; 产生式系统=事实集合 + 产生式规则集合 + 推理引擎; LHS, 条件的集合, 各条件之间是且的关系; RHS, 动作的序列, 执行时依次执行; 推理引擎; 模式匹配(用每条规则的条件匹配当前WM, RETE算法), 解决冲突(随机选择), 执行动作; 相关工具介绍; Drools; Jena; RDF4j; GraphDB; Drools实践; 输入; 自定义规则; Drools工程结构; 代码示例(Java); 基于并行技术的方法; 相关工具简介; RDFox, DistEL, DRAOn, WebPIE; 实践展示; 使用Jena推理; 构建本体, 添加推理机, 上下位推理, 针对类别的推理(类别补全), 不一致检测; (2020-10-7)
10. P8, 语义搜索; 文档检索vs数据检索; 语义模型; 语义搜索分类; 语义搜索-流程图; 搜索模式趋向一致; 语义web--数据web; 利用链接数据进行搜索; 难点1--可扩展性; 难点2--异构性; 难点3--不确定性; 最佳实践; 存储和索引(Semplore); 深入逻辑结构--索引虚拟文档; 为什么增量索引; 增量索引--处理当前索引; 基于块的索引扩展; 谨慎选择块的大小; 索引构建vs索引更新; 排序和索引; 如何找到复杂查询的答案; 不同的查询种类(结构化查询); 不同查询的响应时间(毫秒); 排序原则; 如何将排序紧密结合到基本操作中; 从DBpedia收集的混合的查询数据集; 在QS3和QS4上的实验结果; 高效和可扩展的数据web搜索; 为图结构数据(RDF)建立结构化索引; 使用结构索引做结构匹配; 评估; 在多数据源、多存储的场景下搜索; Hermes--Pay-as-you-go数据web搜索框架; 知识融合工作流程; 知识融合主流方法分类; 知识融合的主要挑战; 两阶段知识融合来解决效率瓶颈; 索引(indexing); 分块(blocking); 聚类(clustering); 迭代式数据整合--分块和聚类的效果; 联合查询处理; 数据预处理; 评估; 查询翻译; 查询和结果优化; 结论; 混合语义搜索系统; 混合搜索数据模型; 混合搜索, 查询和数据模型; DB和IR的轻量级集成; 系统架构(CE^2); 查询分解和执行; 答案合并; 初步结果; 原生混合搜索系统--挑战; 结论; 可用性; 交互范式; 一种基于本体的查询解释的通用方法; 将关键词解释为合取查询; 步骤1-将关键词映射为本体实体; 步骤2-发掘本体实体之间的连接; 步骤3-从连接中导出DL合取查询; Top-k关键词查询--工作流程; 摘要图生成(graph summarization); 关键词映射和摘要图扩充; Top-k图探索; 将查询图映射到合取查询; 评估-效果; 评估-性能; 查询解析--理解用户的需求; web数据图摘要生成; 关键词映射和图扩充; Top-k图计算; 打分排序; 评估; 分面搜索系统; 使用语义数据进行分面搜索; 分面浏览, 简单还是困难; 查询时间和相关的分面计算; 高级分面搜索特征; 动态的值分区; 分面排序; 语义搜索路线图; 参考文献; 相关应用--facebook graph search; FGS--查询构建; 实践展示; 实践展示--功能; 实现步骤--确定索引方式; 实现步骤--数据格式转换; 实现步骤--导入ElasticSearch; 属性同义词扩展; 实现步骤--查询解析及构造; 实现步骤--查询解析; 实现步骤--用户查询分类; 实践--进阶; (2020-8-5)
11. P9, 知识问答; 交互方法的转变; 问答系统; IBM Watson; 问答系统历史; 根据问答形式分类; KBQA应用; 答题机器人; 测评数据集--QALD; QALD任务1; QALD任务2; QALD任务3; QALD的评测指标; 评测数据集; 知识问答简单流程和分类; 基于符号表示(传统)的知识库问答; 基于分布式表示(DL)的知识库问答; NLI&QAs; 基本概念与术语; 概念一, 问句短语; 概念二, 问题类型; 概念三, 答案类型; 概念四, 问题主题; 概念五, 问答来源类型; 概念六, 领域类型; 概念七, 答案格式; 问答质量的评估原则; 答案的评估; 答案的处理; 问答任务的复杂性; 问答系统的基本组件; 基于知识图谱的问答, 基本需求; 技术挑战, 如何将问题映射到答案; 自然语言问题与知识图谱之间的鸿沟; 映射自然语言表达式到知识图谱元素词汇; 不同的知识表示增加了映射难度; 关系或属性隐含表述; 知识库如何支持多语言问答; 数据质量和异构性; 分布式和互联数据; 回答性能和可扩展性; KBQA挑战总结; 基于模板的方法; TBSL; 动机; 基于模板的问答; 示例, who produced the most films; TBSL架构; Step 1, template generation--linguistic processing; 示例, who produced the most films; step 2, template instantiation--Entity identification and predicate detection; 示例, who produced the most films; step 3, query ranking and selection; TBSL主要缺点; 模板是否可以自动生成; GUINT架构; 方法与主要贡献点; 模板生成--问句依存分析; 为utterance构建query; 通过ILP将问题与query进行对齐; 词典L构建; 模板定义与生成; 模板匹配和实例化; 候选query排序; 复杂问题处理; 实验结果; 模板匹配方法的优缺点; 基于语义解析的方法; 传统语义分析方法; 基于符号表示(传统)的知识库问答--Semantic Parsing; 问句的形式化表示; 基本步骤; 两个关键问题; 语义分析--资源映射; 语义分析--逻辑表达式; 利用logic form表示问题语义; 语义解析经典方法; 传统语义解析vs弱监督语义解析; 应对大规模KB的语义解析; 文本(问题)映射到KB的若干挑战; 应对文本(问题)映射到KB的若干挑战; 语义解析--问题定义; (2020-8-6)
12. P10, 行业知识图谱应用; 谷歌知识图谱, things not strings; 知识图谱助力人工智能应用; 通用知识图谱; 行业知识图谱, Palantir; 行业知识图谱; 行业知识图谱数据的特点; 行业知识图谱应用一览; 行业知识图谱相关项目; 通用知识图谱VS行业知识图谱; 行业KG应用; 金融证券--企业知识图谱; 企业知识图谱应用--企业风险评估; 企业知识图谱应用--企业最终控制人查询; 企业知识图谱应用--企业之间路径发现; 企业知识图谱应用--初创企业融资发展历程; 企业知识图谱应用--上市企业智能问答; 金融证券--金融交易知识图谱; 金融交易知识图谱应用--辅助信贷审核; 金融交易知识图谱应用--反欺诈(1)(2); 金融证券--其他应用场景; 生物医疗--医疗知识图谱; 医疗知识图谱应用--中医药知识平台; 生物医疗--Watson辅助诊断与治疗; 生物医疗--Open PHACTS新药物发现; 图书情报--图情资源知识图谱; 图情资源知识图谱应用--知识导航与资源展示; 图情资源知识图谱应用--知识点推荐与搜索; 图情资源知识图谱应用--图情资源统计; 知识图谱行业应用--其他行业; 行业知识图谱应用挑战; 从数据库发展到大数据时代, 企业希望融合使用全量数据; 企业全量数据应用挑战2, 数据模式动态变迁困难; 企业全量数据应用挑战2, 非结构化数据计算机难以理解; 企业大数据应用挑战4, 数据使用专业程度过高; 企业大数据应用挑战5, 分散的数据难以统一消费利用; 解决方案, 基于行业知识图谱进行数据融合使用; 知识图谱助力企业商业智能; 行业知识图谱生命周期; 知识图谱基础技术规范; RDF; OWL, RDF schema的扩展; SPARQL简介; SPARQL查询图可视化; 本体(ontology)可以填充知识与查询之间的语义间隙; 1, 知识建模; 知识建模关键技术与难点; 2, 知识获取; 知识获取关键技术与难点; 3, 知识融合; 知识融合, 跨语言融合; DBpedia mapping; 识别sameAs关系; knowledge vault, google将建全球最大知识库; 知识融合关键技术与难点; 4, 知识存储; 知识存储关键技术与难点; 5, 知识计算; 关键技术与难点; 6, 知识应用; 关键技术与难点; 行业知识图谱关键技术; LOD2; Stardog; 知识图谱生命周期总结; 使用知识图谱对数据进行抽象建模; 知识建模工具--Protege; 构建一个适用的建模工具(1)(2)(3)(4); 知识获取; D2R工具--D2RQ; D2R工具构建(1)(2); 半结构化行业数据源解析; 包装器配置工具; 包装器示例--专利知识抽取(1)(2); 文本信息抽取, 主要任务; CloseIE和OpenIE; OpenIE; CloseIE典型工具, DeepDive; DeepDive关系抽取过程与总结; 文本信息抽取实践方法; 事件抽取; 知识抽取最佳实践, 多策略学习方法; 多策略学习方法示例(1)(2); 知识融合; 知识融合实践--模式层融合; 知识融合实践--数据层融合; 数据层融合示例--人物实体合并; 知识存储; 图数据存储; 常见的图数据存储--Graph DBMS; 数据存储支持; 数据操作和管理方式; 支持的图结构; 实体和关系表示; 查询机制; Neo4j; 大规模知识图谱存储解决方案; 大规模知识图谱存储最佳实践(1)(2); 知识图谱中的时态信息; 知识图谱时态信息存储; 知识计算; 图挖掘计算; 基于本体的知识推理; 本体知识推理工具--RDFox; 基于本体的知识推理应用实例--冲突检测; 基于规则的推理; 基于规则推理工具--Drools规则定义; 知识应用; 知识应用关键技术; 语义搜索; 实体链接工具--Wikipedia Miner; 现有实体链接工具使用总结; 实体链接的基本方法过程; 基于知识图谱的语义搜索; 语义搜索示例--实体搜索; 知识应用关键技术; 智能问答--基本过程; 智能问答--方法分类; 智能问答方法; 知识问答的最佳实践方法; 基于语义解析的自动问答; 知识应用关键技术; 知识图谱可视化; 图谱可视化基本组件; 基于知识图谱大数据的统一决策支持分析平台; (2020-8-8)
13. -------------------------------------------------------------------------------------------------------------------------
14. 知识图谱本质上就是语义网络，是一种基于图的数据结构; 通用知识图谱; 行业知识图谱; (2020-8-9)
15. What is knowledge base embedding; Knowledge graph embedding aims to embed the entities and relationships of a knowledge graph in low-dimensional vector spaces; (2020-8-12)
16. ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
17. -