摘要

知识图谱是通过保存实体和实体间关系来实现语义搜索的数据库。金融知识图谱是通过将公司、管理层、新闻事件以及使用者个人偏好都表示为实体，发现其间的联系，让对金融数据的搜索更加高效，进一步能为投资者提供针对性的投资建议。对公司来说，金融知识图谱有利于提高风控、反欺诈、获客等能力。本文简述了金融知识图谱从知识提取、本体存储、分析推理到知识利用的流程。

背景

如同吴文俊先生在《人工智能及其应用》序中所提，脑力劳动机械化将为社会效率带来不可估量的提升。目前（2016）A股和新三板上许多财务报告已经使用XBRL（可拓展的商务报告语言）进行规范化，便于计算机处理和呈现人们想关注的重点，从而消除了价值判断过程中阅读大量原始材料的负担。
然而 XBRL 基于 XML（可拓展的标记语言），是较早期的技术成果，无法很好地承载语境、关系等语义信息。例如当需要把子公司 A 的收入汇总到总公司的总收入里时，除了在子公司 A 的账内说明以外，还需要在总公司的帐里再次进行冗余的陈述:「总公司的总收入包括子公司A的收入」，而这种冗余有时又会导致不一致性。而在进行基本面分析时金融数据源则反而显得单薄，需要手动去查找总公司有几个子公司、其收入几何、总公司有没有负面新闻、子公司有没有负面事件等等，并将其与汇率、利率、政策等世界知识联系在一起，因为 XBRL 无法自动帮我们联想到一次基本面分析所有维度上的信息。

理想的情况是，当我们对一个公司或有价证券产生兴趣时，有一个系统能自动整合网上公开信息和使用者拥有的私有信息，不仅仅是金融报告，还包括 911、熔断机制、舆论走向等世界知识，对有兴趣的实体进行基本面分析，或对某个实体集合进行量化分析。这样的系统被称为「语义网」或「知识图谱」，其构建方法从上世纪六十年代就开始了研究，其研究目标是利用海量开放数据发现现实世界中各实体间的关系并从中推理出有实用价值的知识。

知识图谱与机器学习的关系

目前有许多金融从业者已经接纳机器学习方法为他们工作的一部分，需要说明的是，知识图谱技术与机器学习技术有以下相似之处:

1. 都使用海量标注数据集
2. 都以替代人类进行分析实体特征为目标
3. 知识图谱中需要用到机器学习，机器学习也需要知识存储

但它们相异之处在于:

1. 知识图谱不需要训练
2. 知识图谱可以容忍比较「脏」的异构数据
3. 知识图谱推理的中间结果很容易让人类理解

有哪些公司已经入行

2005年由来自 University of Southampton （语义网的核心研究机构之一）的核心成员成立于英国的 Garlik 公司收集公共网络上的个人信息，并对信息盗用提供报警服务，后被信用记录公司 Experian 收购，用于提供信用分析服务。

2004年在硅谷开始运作的 Palantir 结合客户提供的信息源与公共网络上的事件提供对海量数据的研究分析，其金融数据分析平台 Palantir Metropolis 可以对金融数据进行复杂搜索、可视化编辑、关联发现等操作。

文因互联是2013年在北京成立的公司，希望通过金融知识图谱和金融语义搜索帮助投资者对标A股、获取新三板信息、挖掘交易数据的背后信息（如异常交易等）、发现投资机会。

其他领域中的知识图谱

2003年，美国国防部先进研究项目局启动了 CALO（能自我学习和组织的认知助理）计划，旨在研发出一款能协助军方指挥官完成信息处理和办公任务的虚拟助手软件，投资 1.5 亿后，这个军方项目为 Siri 的诞生奠定了基础。在研发过程中，siri 就能判断一场会议中与会人员的身份特征，并自动分发材料，同时能在其中一个成员请假时决定是否需要取消这次会议。

Google 于2012年推出其整合入搜索的知识图谱，由于互联网上的搜索近半是关于某个实体的，含有5亿个以上实体和35亿条以上事实的 Google 知识图谱能在很多情况下直接向搜索者返回精确定位的知识，例如「how long is a marathon」会直接返回「42.195 kilometres」。

建造流程

知识图谱建造的流程一般是: 脏数据 -> 干净数据 -> 文档树/表 -> 图谱 -> 本体 -> 逻辑
对于知识图谱的建造范式有多种看法，有的着重于推理能力，有的着重于知识的表示能力，有的着重于工程实现。实际操作过程中我们在考虑成本的同时，一个典型的建造过程如下:

1. 首先信息抓取的系统要能在只替换少量代码的情况下适配大量异构、不断更新的数据源
2. 从股转中心、证监会、微博、文档OCR等渠道抓取的信息是肮脏（带噪音）的，需要有比较宽容的方法能对不同的数据进行清洗
3. 清洗过的数据根据应用需要，得格式化到一定的程度，同时根据成本限制保留非格式化的部分
4. 装载格式化数据到图谱里，根据数据内容或数据特性添加它与其他数据的关系
5. 从数据中抽象出本体，与人类拥有的世界知识相对应
6. 从关联中得到逻辑，这步成本比较高昂，根据应用可简化

实践过程中应该考虑到，每一步都应该对实际应用做出贡献并得到回报，以防止过高的成本导致项目提前终止。

在接下来的文段中，我将展开以上建造流程，并详细叙述从数据中抽象出本体的过程。

数据获取

金融数据源

经过 Tim Berners-Lee 等人的努力，目前美国有许多格式化的数据源，而中国则多在脏数据阶段。现阶段我们对数据的获取有以下几种情况:

巨潮网: http://www.cninfo.com.cn/cninfo-new/index 披露深沪上市公司公告信息和市场数据，数据之间没有关联，格式为 PDF 或网页
数据获取方式为通过爬虫爬取文件后，解析 PDF、HTML 或 OCR 得到无结构文本。

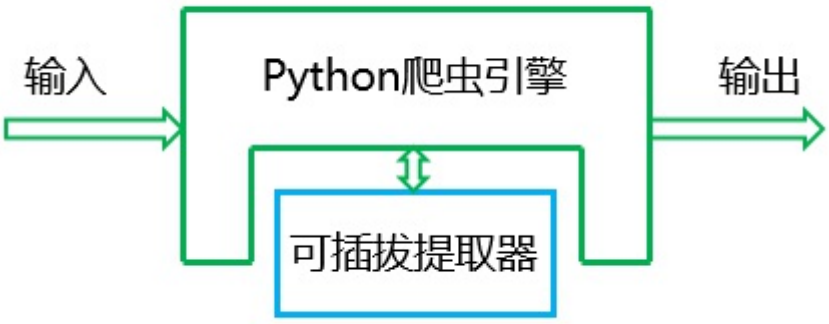
雅虎金融: http://finance.yahoo.com/ 上市公司的长期表现数据，数据之间关联较少，格式为 HTML
数据获取方式为调用网站使用的 API 或爬虫模拟浏览器获取格式统一的无语义数据。

美国政府公开数据: http://catalog.data.gov/group/finance3432 整合了美国各政府机构不涉密的数据，例如SEC法规年鉴、共同基金报告、公共公司破产记录等，提供一站式下载，数据之间没有关联，格式有 csv、text、PDF、HTML、Excel、XML
数据获取方式为爬取下载链接后，对每种格式分别处理得到结构化的异构数据。

美国证券交易委员会: https://www.sec.gov/Archives/edgar/xbrlrss.all.xml 参与证券交易的企业状况和证券情况，数据之间关联较少，格式为 XBRL 数据获取方式为直接抓取数据端点，得到结构化带一定语义的同构数据。

爬虫框架

为了适应大量各有特性的数据源，爬取数据的爬虫应是模块化的。参考 GooSeeker 项目中的 Python 爬虫，其模拟登陆、页面爬取、数据存储、多线程或分布式等复杂的引擎功能是不变的，针对不同数据源用图形界面进行设置，自动生成适配数据源的可插拔提取器，从而较低成本地适应异构数据源。



提取器包含的内容可以有:

- 用 CSS 选择器或正则表达式匹配数据源中的元素，并将匹配到的元素填入特定的格式（Schema）内
- 用包装器(Text Wrappers)学习命名实体
- 运用生物信息学中的基序发现算法来发现网络请求中相似的部分，从而让爬虫自动抓取列表类型的数据
- 用知识图谱中的本体验证信息抓取规则的正确性

其在金融数据获取方面有经典的使用案例:

> 中国进出口银行项目：集搜客网络爬虫负责抓取中文财经、金融、证券和经济报告类网站内容，而爬行范围、时间安排和其它管理指令是由整个IT系统的其他软件模块发出。

数据清洗

目前（2016）常用的语义数据结构化表示方案有「JSON-LD」和「RDF」。

其中 JSON-LD 采用柏拉图式二元论: 每个数据源对数据项的命名相当于是「真正对命名」的一种方言，所以它提供一种方法，将各个数据源对数据项的命名映射到一个统一的本体库中，从而统一数据，而且考虑到同一能指在不同语境下有不同的所指，它也提供了描述语境的方法。

RDF 采用三元组: 数据被表示成「主语-谓语-宾语」的形式，每个词都有自己的命名空间以避免重名，其优势在于三元组天生就是图（Graph）的定义，便于使用图论推理和存储。当需要更高的表达能力，又不关心完备性、可判定性时，可以无缝升级到 OWL（网络本体语言），获得定义词汇之间的关系、类与类间的关系、属性与属性之间的关系之能力。
其他格式如「Microdata」、「KIF」、「RIF」都有其缺憾，从能搜到的教程数量就能看出其使用率低于前述两种语言，故不展开。

不论是什么结构化数据用的语言，其目标都是为了节省后续步骤的代码量:

> 利用语义Web技术，每个服务都会暴露出"这是一个人（Person）"这一语义，我们只需要编写一次代码，理解人是什么。而且可以跨多种服务复用代码。

在获得了原始数据（Raw Data）后，为了结构化数据，我们有多种见仁见智的做法:

直接结构化表示

我们已经看到，有的数据源自身一定一定程度上格式化完毕，但因为格式化程度不高无法被我们的应用所理解。我们要做的就是将其映射到我们私有的数据结构上，赋予其被理解的可能。而如果我们使用的私有结构化表示使用了 `http://schema.org/` 等世界统一的语境，也就意味着它不再「私有」，而是和世界上其他同语境的数据互联在了一起。

由于 RDF 本身就是一种 XML，有的 XML 数据源如 XBRL（可扩展的金融报告语言）数据就能直接转化成 OWL 和 RIF 的组合，从而表现出财年等本体，并提供自动查账等推理。

使用 JSON 传送数据的数据源，例如雅虎金融提供的 API 接口:

```
https://query.yahooapis.com/v1/public/yql?
q=select%20*%20from%20yahoo.finance.quotes%20where%20symbol%20in%20(%22BHP.AX%22)&format=json&diagnostics=true&env=store%3A%2F%2Fdatatables.o
```

返回的数据就可转化为 JSON-LD。

在转化过程中还是需要一定程度的人工干预，例如根据常识我们会将股票交易代码（stock trading symbol）这一项和 CIK（Credit Suisse Asset Mgmt Fd）这一项用 `owl:sameAs` 来连接，从而在未来的分析中能获得更广的视角。
因此这一步应该与爬虫的可插拔提取器更紧密地结合起来，在人工设置对结构化数据的爬取时就设置好数据源数据项到私有结构化表示的映射。

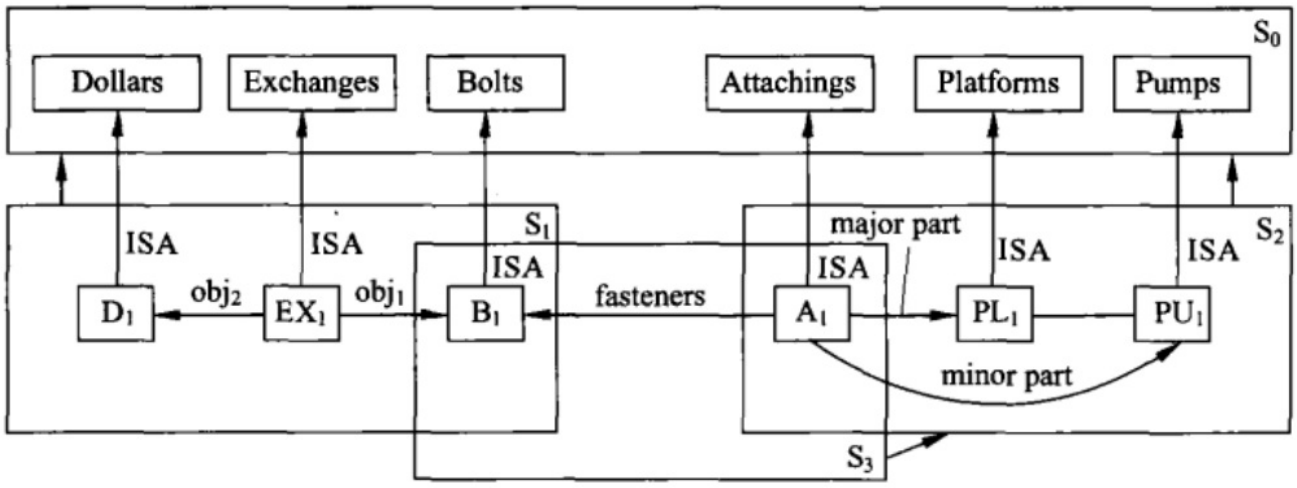
知识图谱反馈

自然语言处理有许多较为成熟的解决方案，可用于处理数据源为无结构文本或结构特别松散的情况。其层次为: 语音分析或光学字符识别 -> 词法分析 -> 句法分析 -> 语义分析 -> 语用分析。在分析金融数据时，达到语义分析的层次即可识别出文本中的本体，达到语用分析层次可为推理打下基础，但有更高的成本。未达到语义分析的层次时，文本和其他数据将不是「互联」的，但可以通过关键词搜索到。

为实现语义分析，我们可以考虑分区语义网络:

> 「接着，把水泵固定到工作台上。螺栓就放在小塑料袋中。」第二句中的螺栓，应该理解为是用来固定水泵的螺栓。因此如果在理解全句时，把需用的「螺栓」置于「焦点」，则全句的理解就不成什么问题了。

在上例中的那句话里，我们需要知道和「固定」有关的知识，因此注意力从下图中的 S0 区域转移到 S2 区域，而为了发现第二句话中和「螺栓」有关的知识，注意力移动到 S3，因此发现「螺栓」和「固定」之间的关系。而当我们的关注点是「美元」和「汇率」时，又会有不同的注意力移动路径。



我们将在本文后面的部分看到，存储本体时我们对于超图（Hyper-Graph）的使用与此是异曲同工的，因此经由知识图谱内的知 ，我们可以反过来支援数据清洗阶段的处理过程。

命名实体识别

金融数据中有许多日常用语中出现率较低的专名，例如经纪机构名、股票代码、衍生品等，其涌现是持续不断的，很难通过人工规则编写来处理。现在常用的方法有「条件随机场（CRF）」、「最大熵隐马尔科夫」、「隐马尔科夫」等序列标注模型。 主要的处理思想有:

- 辅以字典法，利用已有的种子实体
- 将人名、地名、公司名放在不同的阶段处理
- 辅以其他句法分析层次的词性标注

当发现未登录词，即可向知识图谱中添加新的实体，通过知识图谱对自然语言处理的反馈为其他过程提高效率。

文本聚类

为无结构文本附加结构的一个思路是打 Tag，例如两条微博「今儿大盘真不错」、「买进了一万股」，我们可以从语义上判断它们具有相似性，进而在知识图谱中赋予它们相关性。
目前实用的技术主要有 LDA 与 Word Embedding。

LDA（隐狄利克雷分布）是一种无监督的主题模型，通过每个词语在大量文档中的出现概率学习出某个主题中出现某个文档的概率以及某个词语在某个主题中出现的概率。最终可以为一个文本添加多个主题标签，有利于对无结构而且最终也不需要完全结构化的文档添加与其他数据的关联。

Word Embedding（词嵌入）是一种把词语变成向量然后比较向量方向得到词之间相似性的无监督方法，通过词的相似性可以得到文本的相似性，而且其结果具有类比的性质（如男人-撸管 = 女人-柔道）能保留文档中的一些语义，对于知识图谱中无结构数据的关系添加有好处。

生物信息学算法

生物信息学主要是关于发现大量基因、蛋白质数据中的模式，从而为生物研究提供思路的学科。
一个典型的生物信息学问题是发现果蝇是如何免疫病原体的感染的:

> 果蝇有一小部分的免疫基因，这些基因在果蝇基因组中通常处于休眠状态，但是当机体受感染后，这些基因会以某种方式激活。当激活这些基因后，它们会合成一些破坏病原体的蛋白质，通常能治愈感染。许多免疫基因在起始位点的上游都有像 TCGGGGATTTCC 基序样的序列，这些称为 NF-kB 结合位点的短序列就是激活免疫的调控基序。假设我们并不知道这基序的样子，也不知道它的位置，我们可以通过构造基因的剖面矩阵，通过检索树等概念来找出这一模式。

通过将文本数据视为线性的数据，可以与基因相类比，在多个事件中频繁具有的模式可能就是一种欺诈、一种操纵，将它们识别出来对后续的推理分析有好处。

此外生物信息学还能解决信息在传播过程中发生流变后，如何找出新信息是由哪些原始信息合成的问题，在处理多数据源带来的无结构文档时是一个可尝试的思路。

利用现存本体库

开放网络上有许多语料库、词典和本体库，如 DBPedia、CYC、HowNet、Enterprise Ontology 等，对它们的利用对降低成本是很有必要的。

Cyc 知识库提供了大量世界知识，能让知识图谱拥有人类的日常生活常识水平，知识使用 CycL 表示，最高支持二阶谓词。
HowNet 使用特有的 Knowledge Base Markup Language 表示出知识树，经过格式转换后对于本体之间层次关系的建立有好处。
Enterprise Ontology 使用企业行动与过程、法律实体和有组织的单位、策略、营销，对于谓词本体的构造有参考价值。

一个低成本的利用方法是直接导入已有本体，并通过一些更抽象的本体将相似的本体连接在一起。
进一步可以通过「转文因」等方法重组出新的本体: 筛选出几个本体所拥有的实体、属性，重新组合出新的本体，并与原来的本体保持联系。

本体存储

本体存储的格式与应用有一定关系，但也有其独立于应用的规律。
实际应用中的需求，例如查询影视传媒相关定增的平均市值和融资市盈率、VR 产业的上游状况、某企业是否有持股平台及其近期投资等，需要能表示公司本体、瞬间和持续性事件谓词、某个时间点或片段内的数据、本体层次关系、事件信度、事件与其他事件的融贯程度。

我们用超图中的超边来表示本体、谓词，用图论中的节点来存储实体、量化数据、事件的元数据，同时也用节点来表示元数据节点的标签、位点。
在向知识图谱添加知识的过程中图会发生变化，为了表示信念（旧有知识）和新加入的知识之间的融贯程度，我们将超边再视作节点，与表示新添加知识的超边同属于一个更高阶的「融贯超边」。

图论数据库与超图

超图（Hyper-Graph）是对传统图论的扩展，无向超图中的边的定义是节点的笛卡尔积，有向超图更复杂一些，但本文中我们更关心工程实现，因此对有向超图和图上操作的定义将采用声明式声明式图论语言（Cypher）给出。

当我们将超边视作高维球体、节点处在球体内，有向超图中的超边可以具有梯度、散度、旋度等描述维度，可以用两个节点之间的正势描述出度，负势描述入度。例如具有散度的一条超边可以描述一种从一个源节点（Source）指向多个节点的关系，并且能简单地定量描述出源节点到其他节点的出度，类似地，具有梯度的超边可以描述对对多关系，具有旋度的一条超边可以描述出「A男喜欢B女，B女喜欢C男，C男喜欢D女，D女喜欢A男」的关系。

有向超图的表述能力比一般有向图更适合金融场景，例如对供应链中各企业本体的连接就可以选用带散度的有向超图，势的方向由上游企业指向下游企业。当供应关系发生变化，可以在「融贯超边」内使表示旧的供应关系的超边以 100% 的出度指向新的供应关系。

等价类与多层图

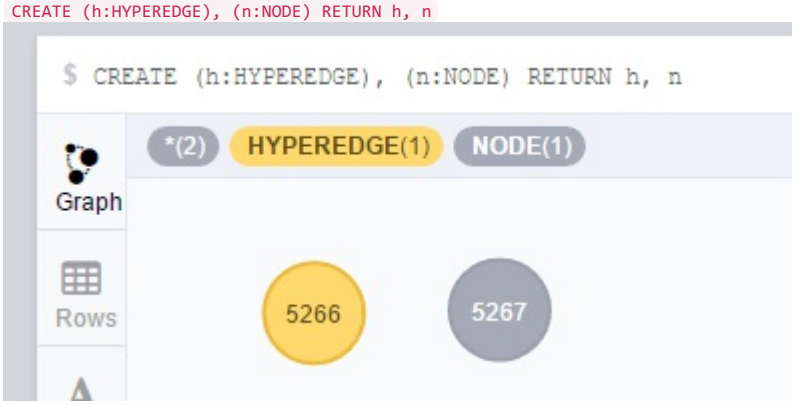
同一个本体在不同语境中可能会有不同的用词，例如一些媒体翻译成「帕兰提尔」的公司，有的时候只被用「估值第四高的那家公司」来指代，而它在其他语境中被直接称为「Palantir」。我们将多个词语节点用一个超边连接起来，用这个连接多个实体的超边来表示这些实体的本体，从而实现不同数据源中实体的融合。在引用数据时，一般可将一条超边视为一个等价类，在需要更细粒度的推理时，可用有向超边中各点之间的势表现其「符合这个本体的程度」。

而由于观察角度的不同，一个实体也可以属于多个本体，例如「贝克汉姆」除了可以和「碧咸」同属于一个「Beckham」超边以外，还可以属于一个「足球运动员」超边，而「Beckham」超边又可以看做一个节点，属于「足球运动员」超边。即一个节点可以属于多个超边，而超边又可以再看做节点，从而实现关系的融合。

有向超图的描述性定义

超边的定义:

```
CREATE (h:HYPEREDGE), (n:NODE) RETURN h, n
```



可以看到超边和节点除了标签外都是相同的。

向具有 id 为 id1 的超边添加 id 为 id2 的实体节点，用边上的信息说明一个实体在什么语境下属于一个本体:

```
MATCH (h:HYPEREDGE {id: {id1}}), (n:NODE {id: {id2}}) CREATE (h)-[r:HAS {context: "http://json-ld.org/contexts/shanghaiTech.jsonld"}]->(n) RETURN h, n
```

其意义在于描述了节点所代表的实体属于超边所代表的本体。

使具有 id 为 id1 的超边属于另一个 id 为 id2 超边:

```
MATCH (h1:HYPEREDGE {id: {id1}}), (h2:HYPEREDGE {id: {id2}}) CREATE (h2)-[:HAS]->(h1) RETURN h1, h2
```

其意义在于使多个具有相似性的本体能被抽象，得到一个更高层的本体。

通过 PageRank-like 创造优先级

PageRank 是 Google 用于评估互联网上页面重要性的图论算法，其主要参考点是有向图中图的入度和出度，并在多次迭代后最终给出每个节点的权重。
节点权重可用于计算给定一个节点的情况下出现另外某一个节点的后验概率，也可用于表示整个知识图谱中任意一个节点的先验概率。得出的概率信息可用于反馈机器学习过程，也可用于推理过程。

数据-本体关联

为保证效率，图论数据库中我们不存储文档、量化数据等文件。
JSON-LD 格式的量化数据内，各数据项使用图论数据库中存储的本体的 URI 来描述。
此处使用的是面向人类的写法:

```
{
    "@context": "http://URI-to-GraphDataBase/",
    "@type": "financeData",
    "publisher": "crawler-uuid",
    "headline": "TSLA-option-april-1st",
    "url": "http://URI-to-ThisFile/",
    "keywords": "option, TSLA",
    "optionList": [
      {
      "@type": "optionList",
      "strike": 100,
      "ContractName": "TSLA160401C00100000",
      "last": 0.66,
      "bind": 0.65,
      "ask": 0.67,
      "change": -0.79,
      "%change": "-54.48%",
      "Volume": 13184,
      "OpenInterest": 13490,
      "iv": "34.57%"
      }
    ]
}
```

冒号左边的 strike、ContractName 等项将会被自动转换为图论数据库中本体的 URI，从而做到对定义的清晰化。

而无结构文档可以用 Microdata 来进行标记:

```
<div itemscope itemtype="http://URI-to-GraphDataBase">
    Name: <span itemprop="name" >Daniel Butler</span>
    Website: <a href="http://aaa.com" itemprop="url" >aaa.com</a>
    Title: <span itemprop="title" >Head of SEO</span>
</div>
```

这样，我们就实现了文档、数据中的用词到这些词所对应的本体的映射。

使响应速度满足事件驱动的需求:分布式主观文因工程学

分析推理

知识图谱的推理有很多久经讨论的做法，为限制篇幅，在此仅提一点新的思路。

知识利用

报告生成

一个运行良好的知识图谱可能每秒输入的数据量都大于人脑能处理的极限，但人并不关注他的爬虫所爬取的所有事实。当使用者把关注点放在一个企业或行业上，知识图谱应该以文字输出最近的概况，供交易者建立印象。
知识图谱的处理方式可以是找到代表此行业的本体，提取本体下的实体拥有的量化数据计算出查询者要求的指标，通过实体相关的事件分析各时间点行业的情绪状态，从事件中也能粗略分析行业内资产的所有权网和人员关系网，最后根据查询者查询重点的不同进行自动文摘等处理，生成人类可读的短文章。
这一过程涉及到缺失项如何拟合或忽略、怎么调整具有不同信度的信息对最终结果的贡献、如何向人类解释推理过程等问题。

根据量化数据，知识图谱也能自动生成转让说明书、投资意向书、研究报告中的部分章节供人类润色。股转系统官方的反馈意见内有许多关于会计数据与转让说明书不一致性的抱怨，如果容易产生不一致性的章节能由机器自动生成，必能解放许多人工，提升金融市场效率。

自动文摘的方法主要分为抽取原句组合成文摘的 Extractive 和根据语义生成遣词造句有异的 Abstractive 两类，其中前者成本较低，根据应用效果可以尝试向后者迁移。
知识图谱内的数据有以下特征:

- 输入是ranked graph，但每个节点粒度不同
- 所有本体都有先验概率和后验概率，对 decoder 阶段使用注意力机制有利
- 文摘需求都是多文档的
  因此可能对 Abstractive 的实现反而更有利。

一致性检验

对征信、监察来说，数据的不一致能体现出诸多问题。知识图谱在添加知识时已经将常识上相似的数据置于同一本体下，在这些数据之间进行校验可以有针对性地从过去人类容易忽略的角度低成本地发现问题。

如何利用常识库优化机器学习制导的量化策略

使用机器学习的量化策略受限于特征的选取和组合，模型的构建立足于交易员的学识、悟性等世界知识。在知识图谱的世界知识的指导下交易程序能自动处理一些过去无法发现的机会，例如:

> 有家公司在去年以1块钱的低价，定增了大量股份给公司外部的人，后来才发现这是它被外部收购的前兆。只要我们通过知识图谱处理好公司与高管、公司与定增之间的关系，这种信息就可以很容易地在知识图谱中被挖掘出来。

也能使交易程序规避恐怖袭击、笑脸男事件等不直接发生在交易所但能影响人们的情绪和对企业预期的事件。

如何对外用GraphQL开放数据

GraphQL 是一种适于批量获取相关数据的数据端点（EndPoint）描述方法，利于数据的一站式获取。
对比传统的 RESTful 数据端点，GraphQL 有代码量少，易于更新数据获取逻辑，能描述带关系的数据等优点，适用于知识图谱的 API 接口书写。

总结

本文通过调研，人工总结了金融知识图谱行业的现状、方法论和可能产出。限于知识量以及调研时间，部分内容无法展开，限于主题，并未展示工程细节（implementation）而只做描述，但依然希望对于想进入此行业的人士能有普及性的参考价值。在文章的最后我列出了所有参考文献的简介和 URI ，文章中所有对参考的引用都将使用超链接的形式以方便浏览，并提供部分内容的下载源。

参考源:

文因互联博客:

参考类型: 列表
简介: 目前专注于新三板金融数据挖掘的公司的内部博客
URI: http://blog.memect.cn/
列表项:

1.

从XBRL到金融知识图谱
参考类型: 博客
作者: 鲍捷
简介: 金融知识图谱的开放数据源
URI: http://blog.memect.cn/?p=1142

2.

人工智能到底怎么玩？来看人工智能和金融的别样玩法
参考类型: 博客
作者: 严泽徐
简介: 介绍了自动报告生成 人工智能辅助 金融搜索引擎 智能投资顾问等方向
URI: http://blog.memect.cn/?p=1023

3.

语义搜索与金融领域的结合
参考类型: 博客
作者: 王丛
简介: 作者认为将弱关联数据一口气展示出来就已经很棒了，复杂查询还是靠多次查询

URI: http://blog.memect.cn/?p=861

4.

互联世界的记忆
参考类型: 博客
作者: 鲍捷
简介: 作者介绍了「文因」，并推荐了一些关于外部记忆存储的参考书目
URI: http://blog.memect.cn/?p=493

5.

Representing Financial Reports on the Semantic Web - A Faithful Translation from XBRL to OWL.In The 4th International Web Rule Symposium (RuleML). (p. 144-152)
作者: Jie Bao, Graham Rong, Xian Li, and Li Ding (2010) 参考类型: 论文
简介: 描述了在转换XBRL金融语义数据到OWL的过程中如何降低冗余和不一致，并提供了金融语义数据的概览
URI: http://blog.memect.cn/wp-content/uploads/2016/07/2010-06-17_XBRL.pdf

6.

即将到来的智能金融军备竞赛
作者: 鲍捷
参考类型: 博客
简介: 介绍了金融机构在引入智能手段做数据分析和交易策略生成时会遇到哪些问题
URI: http://blog.memect.cn/?p=451

7.

模型论，语义信息论和词嵌入
作者: 鲍捷
参考类型: PPT
简介: 作者通过词嵌入近似工程化语义信息论，从而低成本地表达出符号的语义
URI: http://blog.memect.cn/?p=989

8.

活动现场丨语义对话金融——人工智能与价值判断
作者: 韩佩珊
参考类型: 会议记录
简介: 了解一下业界到底缺什么数据
URI: http://blog.memect.cn/?p=83

9.

嘉宾观点丨PALANTIR产品技术解读 参考类型: 演讲记录
作者: 陈利人
简介: 解释了高估值大数据初创企业帕兰提尔的产品
URI: http://blog.memect.cn/?p=85

10.

降低知识图谱的构造成本
参考类型: 博客
作者: 鲍捷
简介: 指出知识图谱要以使用者为中心，不要闷头深挖，还介绍了知识图谱主要的成本因何而来
URI: http://blog.memect.cn/?p=393

11.

讲座分享 | 人工智能与投资价值判断
作者: 鲍捷/段清华
参考类型: 演讲记录
简介: 介绍了中国背景下智能金融的机遇何在
URI: http://blog.memect.cn/?p=348

12.

嘉宾观点|关于金融知识图谱的若干思考
作者: 陈华钧
参考类型: 演讲记录
简介: 我们应该弱化逻辑，机器学习更多偏重计算，知识图谱更多偏重记忆，图挖掘等应该是构建的下一步

URI: http://blog.memect.cn/?p=330

13.

从语义网到知识图谱——语义技术工程化的回顾与反思
作者: 鲍捷
参考类型: 博客
简介: 介绍了知识图谱的历史，搞机器机器润滑的失败了，搞机器人类润滑的成功了;搞推理等理论化的失败了，搞知识关联呈现的成功了
URI: http://blog.memect.cn/?p=81

14.

人工智能正在逐步走进金融领域
作者: 王丛
参考类型: 博客
简介: 介绍了当前使用智能辅助的金融公司
URI: http://blog.memect.cn/?p=73

15.

金融报表数据的语义化
作者: 鲍捷
参考类型: 博客
简介: 介绍了一些论文，关于 XBRL 怎么做金融价值分析，还有语义网为什么做得更好
URI: http://blog.memect.cn/?p=70

18大经典数据挖掘算法小结

参考类型: 列表
作者: Android路上的人
简介: 介绍了18中数据挖掘算法，主要看其中关于图的 PageRank 和 gSpan
URI: http://dataunion.org/11601.html
列表项:

1.

gSpan频繁子图挖掘算法
作者: Android路上的人
参考类型: 博客
简介: gSpan算法的核心就是给定n个图，然后从中挖掘出频繁出现的子图部分
URI: http://blog.csdn.net/androidlushangderen/article/details/43924273

2.

链接挖掘算法之PageRank算法和HITS算法
作者: Android路上的人
参考类型: 博客
简介: PageRank 是谷歌用于计算页面节点权重的算法，可用于知识图谱
URI: http://blog.csdn.net/androidlushangderen/article/details/43311943

图挖掘：社会网络分析和多关系数据挖掘

作者: eric_gcm
参考类型: 博客
简介: 给出了一些用于发掘链接的算法的目录，没有细讲
URI: http://eric-gcm.iteye.com/blog/1940280

Spark+GraphX大规模图计算和图挖掘（V3.0）

作者: 王家林
参考类型: 教程
简介: 介绍基于 Spark 的图挖掘架构，值得参考
URI: http://book.51cto.com/art/201408/450049.htm

语义网学习路线

参考类型: Syllabus
简介: Fork自鲍捷的语义网学习资料表
URI: https://github.com/Learnone/leansemanticweb

集搜客开源爬虫

参考类型: 列表
简介: 一种开源的爬虫，能低成本地抓取异构数据源
URI: http://www.gooseeker.com/index.html
列表项:

1.

Python即时网络爬虫项目: 内容提取器的定义
参考类型: 博客
作者: shenzhenwan10
简介: 介绍了模块化爬虫的理念
URI: http://www.gooseeker.com/doc/thread-1800-1-1.html

2.

集搜客GooSeeker企业版
参考类型: 产品
简介: 说明了开源爬虫框架 GooSeeker 有什么特性
URI: http://www.gooseeker.com/about/enterprise.html

Neo4j图论数据库初级教程

参考类型: 教程
简介: Fork自Neo Technology, Inc.的教程
译者: 林东吴
URI: https://github.com/linonetwo/neo4j-tutorial-Chinese

分析哲学导论

参考类型: 传统书籍
简介: 介绍了古今各流派对于语言知识的表示方法
作者: 黄敏
ISBN: 9787306033574
出版年: 2009-8
URI: http://pan.baidu.com/share/link?uk=873277927&third=0&shareid=938420257&adapt=pc&fr=ftw

语义网技术体系

参考类型: 传统书籍
简介: 介绍了古代和现代语义网的搭建方法
作者: 瞿裕忠 / 胡伟 / 程龚
ISBN: 9787030422132
出版年: 2015-10-1
URI: No PDF available in open web

FaceBook-GraphQL数据端点技术

参考类型: 教程
简介: 介绍了 Facebook 推出的一种适用于图论数据库的数据端点（endpoint）技术
作者: KADIRA
URI: https://learngraphql.com/basics/introduction

Boolean Network Simulator

参考类型: 产品
作者: matthiasbock
简介: 一个命题逻辑的工程实践，用于存储和推理生物信息。不过本文关心的其实是谓词逻辑。
URI: http://rumo.biologie.hu-berlin.de/boolesim/

认知“技能树”的形成

参考类型: 列表
作者: 刘泊荣
简介: 创管学院老师介绍认识的朋友，当时他希望让对一个学科比较懵懂的人也能快速定位到一个学科内他想知道的知识点，所以做了个知识地图，列表中他当时参考的资料有些已经失效了，就不再列出
URI: http://user.qzone.qq.com/649273225/blog/1384500764

列表项:

1.

khanacademy Knowledge Map
参考类型: 产品
简介: 从数数到欧拉公式的知识地图，背后带有习题
URI: https://www.khanacademy.org/exercisedashboard

2.

wikipedia-knowledge-map
简介: 将维基百科内容表示成图，视觉效果不错，但是网站已经挂了，不过类似网站可以用这个关键词在 Github 上找到源码，故列出
URI: http://www.canvasdemos.com/2010/12/17/wikipedia-knowledge-map/

3.

本体库可视化展示
作者: 布衣
简介: 按照二十四史来建立人物、时间、地点、时间、职官本体库，并做了可视化，原文已加密，这是 Fork 版本
URI: http://blog.sina.com.cn/s/blog_4afee0940100txwt.html

4.

webprotege
作者: stanford University
简介: 一个提供下载链接的本体库列表，属于一个轻量级本体论编辑器和知识获取网络工具项目
URI: http://webprotege.stanford.edu/

规则交换格式RIF在表示语义Web规则中的应用及其推理研究

参考类型: 论文
作者: 陈巍
简介: 介绍了现在基本没啥用的 RIF 格式，仅作了解
时间: 2010年
URI: http://cdmd.cnki.com.cn/Article/CDMD-10183-2010109319.htm

The JSON-LD Markup Guide To Local Business Schema

参考类型: 博客
作者: Gene Maryushenko
简介: 介绍了将金融数据转化为 JSON-LD 的过程
时间: 2015年5月28日
URI: https://whitespark.ca/blog/the-json-ld-markup-guide-to-local-business-schema/

人工智能及其应用

参考类型: 传统书籍
作者: 蔡自兴/徐光祐
时间: 2010年5月
简介: 一本人工智能概论级的教科书
ISBN: 9787302068372
URI: https://github.com/likehua/search-engine/blob/master/%E4%BA%BA%E5%B7%A5%E6%99%BA%E8%83%BD%E5%8F%8A%E5%85%B6%E5%BA%94%E7%94%A8%20%20%E7%AC%AC4%E7%89%88%EF%BC%88%E8%94%A1%E8%87%AA%E5%85%B4%EF%BC%89.pdf

生物信息学算法导论

参考类型: 传统书籍
作者: N.C.琼斯
时间: 2007年7月
简介: 一本生物信息学概论级的教科书
ISBN: 9787122001696 URI: https://book.douban.com/subject/2183420/

pure render: ReactEurope 2016 小记 - 下

参考类型: 会议记录
作者: 诚身/Dan Schafer
时间: 2016年7月18日

简介: 通过 GraphQL 建造轻量级的数据接口，将验证身份等操作放在业务逻辑层（也就是调用数据库API的那一层，不管它看起来多简陋）
URI: https://zhuanlan.zhihu.com/p/21616613

rsarxiv: 自动文摘

参考类型: 博客
作者: rsarxiv
简介: 介绍了自动文摘常用技术栈
URI: https://github.com/rsarxiv/rsarxiv.github.io