基于知识图谱和深度学习的企业智能风险预警系统

**摘要（可以最后写）**

**一 前言 （参考周温丁的报告和一些论文）**

**1.1 背景意义（1-2页）**

**1.1.1 项目背景**

Google、IBM等国际巨头公司已经将人工智能技术渗透到各种产品，国内金融行业也将人工智能技术应用于多个场景。随着国内双创政策的推动和对人工智能产业的投资拉动，人工智能技术在金融行业的应用会越来越广泛[19]。国内一些互联网巨头以及金融机构已经在人工智能技术应用在金融行业领域发力，比如阿里巴巴旗下的蚂蚁金服设有专门的科学家团队，专门从事机器学习与深度学习等人工智能领域的前沿研究并在蚂蚁金融的业务场景下进行一系列的创新与应用；交通银行的智能网点机器人“交交”采用人工智能技术对客户进行业务办理指引；平安集团下设平安科技人工智能实验室，大规模研发人工智能金融应用。

现阶段，人工智能技术在金融行业的应用非常广泛。

1.  智能客服

智能客服主要是整合集团、公司对外的客户服务通道，提供多模式融合的在线智能客服，通过实时语音识别、实时语义理解，掌握客户需求，自动推送客户特征、知识库等内容。

通过电话客服渠道、网上客服、APP、短信、微信以及智能机器人终端与客户进行语音或文本的互动交流，理解客户业务需求，语音回复客户提出的业务咨询，并能根据客户语音导航至指定业务模块。对传统按键式菜单进行改造，用户使用自然语音与系统交互，实现菜单扁平化，提升用户满意度，减轻人工服务压力，降低运营成本。电话客服不再受限于菜单，可开展全业务的语音导航服务。

2.  反欺诈

反欺诈是风控中非常重要的一道环节。基于大数据的反欺诈的难点在于如何把不同来源的数据（结构化，非结构）整合在一起，并构建反欺诈引擎，从而有效地识别出欺诈案件（比如身份造假，团体欺诈，代办包装等）。而且不少欺诈案件会涉及到复杂的关系网络，这也给欺诈审核带来了新的挑战。知识图谱，作为关系的直接表示方式，可以很好地解决这两个问题。 首先，知识图谱提供非常便捷的方式来添加新的数据源，这一点在前面提到过。其次，知识图谱本身就是用来表示关系的，这种直观的表示方法可以帮助我们更有效地分析复杂关系中存在的特定的潜在风险。

反欺诈的核心是人，首先需要把与借款人相关的所有的数据源打通，并构建包含多数据源的知识图谱，从而整合成为一台机器可以理解的结构化的知识。在这里，我们不仅可以整合借款人的基本信息（比如申请时填写的信息），还可以把借款人的消费记录、行为记录、网上的浏览记录等整合到整个知识图谱里，从而进行分析和预测。这里的一个难点是很多的数据都是从网络上获取的非结构化数据，需要利用机器学习、自然语言处理技术把这些数据变成结构化的数据。

3.  不一致性验证

不一致性验证可以用来判断一个借款人的欺诈风险，这个跟交叉验证类似。比如借款人张三和借款人李四填写的是同一个公司电话，但张三填写的公司和李四填写的公司完全不一样，这就成了一个风险点，需要审核人员格外的注意。

再比如，借款人说跟张三是朋友关系，跟李四是父子关系。当我们试图把借款人的信息添加到知识图谱里的时候，“一致性验证”引擎会触发。引擎首先会去读取张三和李四的关系，从而去验证这个“三角关系”是否正确。很显然，朋友的朋友不是父子关系，所以存在着明显的不一致性。

不一致性验证涉及到知识的推理。通俗地讲，知识的推理可以理解成“链接预测”，也就是从已有的关系图谱里推导出新的关系或链接。比如在上面的例子，假设张三和李四是朋友关系，而且张三和借款人也是朋友关系，那我们可以推理出借款人和李四也是朋友关系。

4.  组团欺诈

相比虚假身份的识别，组团欺诈的挖掘难度更大。这种组织在非常复杂的关系网络里隐藏着，不容易被发现。当我们只有把其中隐含的关系网络梳理清楚，才有可能去分析并发现其中潜在的风险。知识图谱，作为天然的关系网络的分析工具，可以帮助我们更容易地去识别这种潜在的风险。举一个简单的例子，有些组团欺诈的成员会用虚假的身份去申请贷款，但部分信息是共享的。下面的图大概说明了这种情形。从图中可以看出张三、李四和王五之间没有直接的关系，但通过关系网络我们很容易看出这三者之间都共享着某一部分信息，这就让我们马上联想到欺诈风险。虽然组团欺诈的形式众多，但有一点值得肯定的是知识图谱一定会比其他任何的工具提供更佳便捷的分析手段。

5.  异常分析（Anomaly Detection）

异常分析是数据挖掘研究领域里比较重要的课题。我们可以把它简单理解成从给定的数据中找出“异常”点。在我们的应用中，这些”异常“点可能会关联到欺诈。既然知识图谱可以看做是一个图（Graph），知识图谱的异常分析也大都是基于图的结构。由于知识图谱里的实体类型、关系类型不同，异常分析也需要把这些额外的信息考虑进去。大多数基于图的异常分析的计算量比较大，可以选择做离线计算。在我们的应用框架中，可以把异常分析分为两大类：静态分析和动态分析。

（1）静态分析

所谓的静态分析指的是，给定一个图形结构和某个时间点，从中去发现一些异常点（比如有异常的子图）。下图中我们可以很清楚地看到其中五个点的相互紧密度非常强，可能是一个欺诈组织。所以针对这些异常的结构，我们可以做出进一步的分析。

（2）动态分析

所谓的动态分析指的是分析其结构随时间变化的趋势。我们的假设是，在短时间内知识图谱结构的变化不会太大，如果它的变化很大，就说明可能存在异常，需要进一步的关注。分析结构随时间的变化会涉及到时序分析技术和图相似性计算技术。

6.  失联客户管理

除了贷前的风险控制，知识图谱也可以在贷后发挥其强大的作用。比如在贷后失联客户管理的问题上，知识图谱可以帮助我们挖掘出更多潜在的新的联系人，从而提高催收的成功率。

现实中，不少借款人在借款成功后出现不还款现象，而且玩“捉迷藏”，联系不上本人。即便试图去联系借款人曾经提供过的其他联系人，但还是没有办法联系到本人。这就进入了所谓的“失联”状态，使得催收人员也无从下手。那接下来的问题是，在失联的情况下，我们有没有办法去挖掘跟借款人有关系的新的联系人，而且这部分人群并没有以关联联系人的身份出现在我们的知识图谱里。如果我们能够挖掘出更多潜在的新的联系人，就会大大地提高催收成功率。举个例子，在下面的关系图中，借款人跟李四有直接的关系，但我们却联系不上李四。那有没有可能通过2度关系的分析，预测并判断哪些李四的联系人可能会认识借款人,这就涉及到图谱结构的分析。

7.  智能搜索及可视化展示

基于知识图谱，我们也可以提供智能搜索和数据可视化的服务。智能搜索的功能类似于知识图谱在Google, Baidu上的应用。也就是说，对于每一个搜索的关键词，我们可以通过知识图谱来返回更丰富，更全面的信息。比如搜索一个人的身份证号，我们的智能搜索引擎可以返回与这个人相关的所有历史借款记录、联系人信息、行为特征和每一个实体的标签（比如黑名单，同业等）。另外，可视化的好处不言而喻，通过可视化把复杂的信息以非常直观的方式呈现出来，使得我们对隐藏信息的来龙去脉一目了然。

8.  精准营销

一个聪明的企业可以比它的竞争对手以更为有效的方式去挖掘其潜在的客户。在互联网时代，营销手段多种多样，但不管有多少种方式，都离不开一个核心-分析用户和理解用户。知识图谱可以结合多种数据源去分析实体之间的关系，从而对用户的行为有更好的理解。比如一个公司的市场经理用知识图谱来分析用户之间的关系，去发现一个组织的共同喜好，从而可以有针对性的对某一类人群制定营销策略。只有我们能更好的、更深入的（Deep understanding）理解用户的需求，我们才能更好地去做营销。

9.  产业链分析

产业链建模分析主要给证券、基金行业的行业研究员、基金经理、量化投资者提供分析框架。由于产业链本身就是各产业部门之间的技术经济关联，并依据特定的逻辑关系和时空布局关系客观形成的链条式关联关系形态，所以天然比较适合于用知识图谱这种图结构来展示。

知识图谱应用于产业链分析能够帮助行业研究员和基金经理透析整个行业、领域的技术与经济传导关系，实现概念发掘、事件影响分析、行情分析。目前市场上做这一部分的机构和公司逐渐多了起来，不过我觉得难点很多。核心问题是覆盖整个产业链上下游的公司、产品、事件因子异常庞大，其本质问题转化成了一个数据质量问题——即数据的完整程度决定了产业链知识图谱的完整程度。举例来说，假设我要对零售行业像苏宁云商这种个股进行产业链分析，需要关注企业链、价值链、供需链和空间链四个维度，从产品研发、原材料与设备、仓储管理、市场营销、销售物流、渠道分销、终端服务等各个环节的数据情况。这还只是零售行业的一个细分领域，一旦涉及到全行业，数据的运营、持续更新问题难度就更大。所以数据运营也成为产业链建模核心难点。

10.担保链分析

担保链是多个企业在向金融机构融资时，通过互相担保、连环担保、联合担保等担保关系链条形成的特殊利益共同体。企业之间的债权债务复杂隐蔽，一旦某些企业发生违约就会引起区域或行业的系统性风险，所以目前银行业对担保违约现象的量化评估很困难。而且这种担保链条存在传染特性，任何一个企业出问题，风险都会传播，所以担保链风险也是银行信贷业务管理中的难点和痛点。

担保链形成的网状结构较为复杂，但基本有迹可循，可通过担保网络的类别形成如下几类：单向担保、互联互保、循环担保、复杂担保、融资担保、平台担保等。而分析网状结构的方法主要是社交网络分析：即网络中每个节点可以看作借款人和担保人，网络中的每条边作为担保关系和担保方向。

模型：针对商业银行构建担保链的方法主要是Modularity模型，也就是Community Detection，反映了群体(子网络)内部相对外部的连接密度，通过迭代式算法不断划分，最终形成担保网络。

分析：主要方法可以归纳为：结合不同担保网络的风险系统，通过违约历史，识别风险最大的担保链。然后，通过担保网络的风险指标，量化核心企业的风险指标，识别风险最大的企业。

识别：分析风险集中度较高的核心企业和风险扩散度较高的核心企业。结合多维度的借贷和担保信息,如风险等级、担保类型、贷款流向、行业分布等，多方面了解担保风险情况。

**1.1.2 项目意义**

在风险防范领域，传统模式下事后的、手动的、基于传统机构性数据的风险防范模式已经不能满足金融科技新业态的需求，金融机构也应与时俱进的采用大数据、人工智能等新兴技术进行风险管控，减少人力支出，完成风险的主动识别与控制，有效降低风控合规成本，增强风控合规能力。

本课题基于知识图谱与深度学习事件驱动的证券公司风险监测预警机制展开研究。基于深度学习与知识图谱的前沿理论与方法，构建出企业主体知识图谱和关联图谱，结合风险舆情分析，通过深入研究事件驱动对公司自营以及融资融券、股票质押等业务的风险传导机制以及国内外股票市场的运行规律，有效的对风险新闻事件进行自然语言处理，结合深度学习技术可以对风险事件做准确的分析和判别。构建科学合理、操作性强的适合券商内部风险监测预警机制，提升公司的风险管理水平，促进公司业务的健康、快速发展，从而有效的防范系统性金融风险的发生。

我们的系统结合了知识图谱，可实现多企业的多径推理扩展，知识图谱可以关联企业内部信息以及企业与企业间的关联信息，形象化展示企业间的关联信息。其次当获取到一条风险新闻后由关联的单一企业可在知识图谱中进行逻辑推理，快速定位多个相关联企业。提高业务人员的分析效率。

其次，在对于新闻的风险判断中，我们引入了机器学习，深度学习可通过预训练获得高效、精准的判别模型，在后期新闻事件分析判断中可实现判别精准度高、分析覆盖面广、实时计算度快的功能。

最后，为了应对互联网上数据量爆炸的新闻事件，引入了大数据技术，Hadoop集群可以实现大数据的存储和计算，利用spark内存级别的快速读写技术可实现海量数据实时存储及读取。对于获取到的新闻数据可进行多路并行处理来提高处理新闻事件的单一时间，实现实时化处理。

**1.2 技术发展趋势和应用现状（2-3页）**

**1.2.1 知识图谱**

“知识图谱（Knowledge Graph）”的概念由Google公司在2012年提出，是其用于提升搜索引擎性能的知识库。知识图谱的出现是人工智能对知识需求所导致的必然结果，但其发展又得益于很多其他的研究领域，涉及专家系统、语言学、语义网、数据库，以及信息抽取等众多领域，是交叉融合的产物而非一脉相承[6]。

知识图谱本质上是语义网络，是一种基于图的数据结构，由节点和边组成。在知识图谱里，每个节点表示现实世界中存在的“实体”，每条边为实体与实体之间的“关系”。知识图谱是关系的最有效的表示方式。知识图谱就是把所有不同类型的信息连接在一起而得到的一个关系网络，他提供了从“关系”的角度去分析问题的能力[7]。

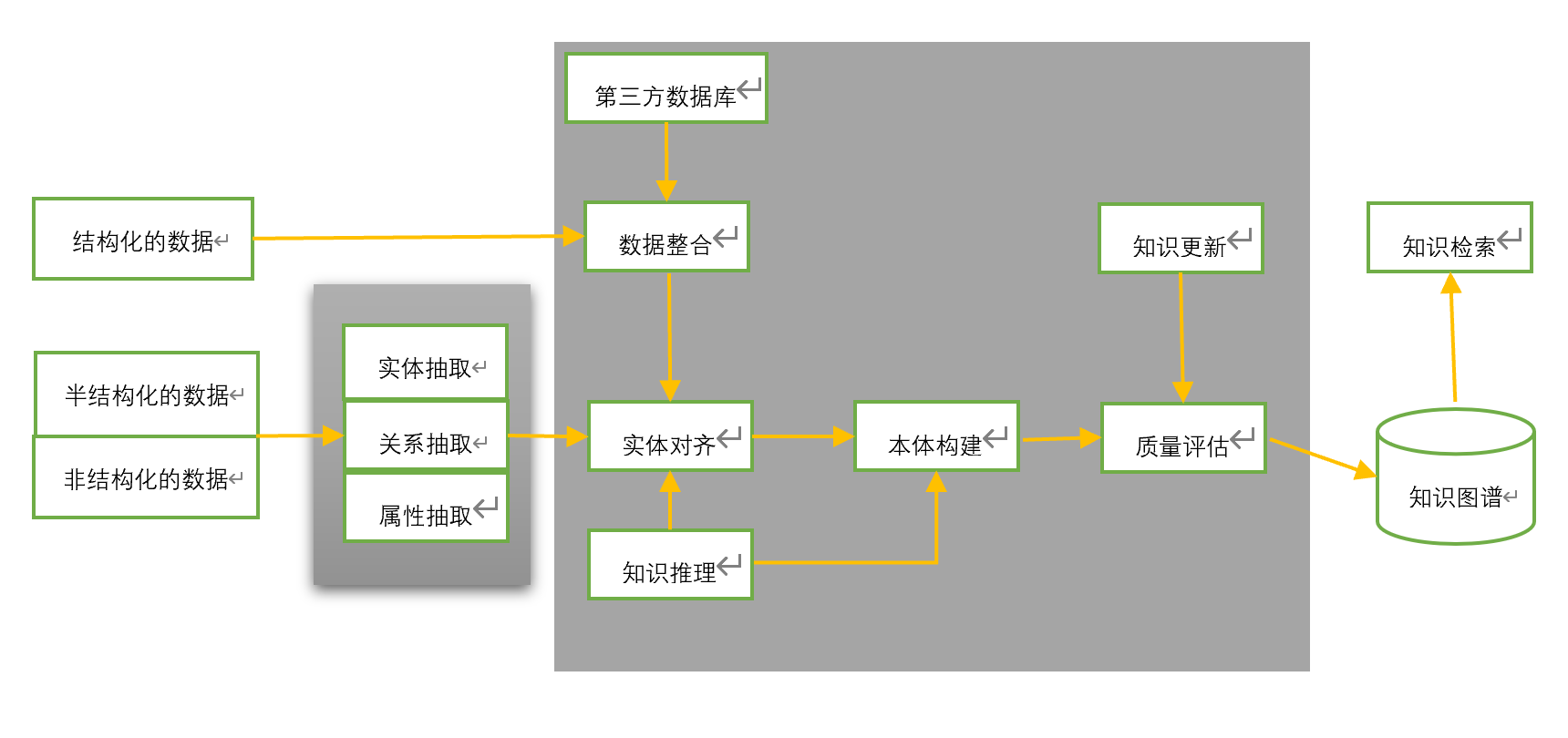
尽管知识图谱本质上来源于语义网，然而当前环境下的知识图谱却有着新的特点——海量规模，这也是为什么知识图谱技术能够展现出巨大潜力和活力的原因所在。传统语义知识图谱的构建是以任务驱动型为导向的自顶向下方式驱动的，是依赖领域专家人工的对领域概念和术语体系进行建模，如CYC、WordNet等。因此数据规模非常有限，WordNet只包含大约71万个三元组，Cyc包含大约160万个三元组。然而，在当前数据爆炸的时代，随着知识抽取技术的不断完善和数据来源(文本、图像、传感器、视频等)的不断丰富，知识图谱的规模正在呈爆炸式增长。以开放关联数据(Linked OpenData,即LOD)项目为例，2009年语义Web中包含RDF三元组数目约为44亿条，2010年该数目增长至130亿条，而2011年9月统计的RDF三元组总数已超过310亿条，截止到2016年11月，LOD项目上发布了包含总共超过1300亿的RDF三元组[8]。当前知识图谱已从传统的小规模本体性知识演化为大规模海量的事实性知识。从某种程度上来说，知识图谱已经等同于海量知识图谱的概念。

知识图谱主要有自顶向下(top-down)与自底向上(bottom-up)两种构建方式。自顶向下指的是先为知识图谱定义好本体与数据模式，再将实体加入到知识库。该构建方式需要利用一些现有的结构化知识库作为其基础知识库，例如Freebase项目就是采用这种方式，它的绝大部分数据是从维基百科中得到的。自底向上指的是从一些开放链接数据中提取出实体，选择其中置信度较高的加入到知识库，再构建顶层的本体模式。目前，大多数知识图谱都采用自底向上的方式进行构建，其中最典型就是Google的Knowledge Vault。

知识图谱的架构主要包括自身的逻辑结构以及体系架构，

知识图谱在逻辑结构上可分为模式层与数据层两个层次，数据层主要是由一系列的事实组成，而知识将以事实为单位进行存储。如果用（实体1，关系，实体2）、（实体、属性，属性值）这样的三元组来表达事实，可选择图数据库作为存储介质，例如开源的 Neo4j、Twitter 的FlockDB、JanusGraph 等。模式层构建在数据层之上，主要是通过本体库来规范数据层的一系列事实表达。本体是结构化知识库的概念模板，通过本体库而形成的知识库不仅层次结构较强，并且冗余程度较小。

知识图谱的体系架构是指其构建模式的结构，如下图所示：



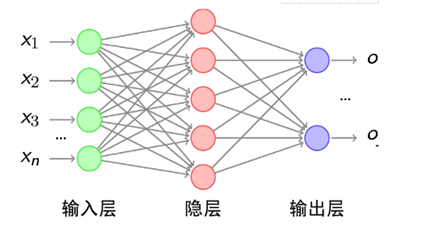
金融风险的复杂性与多样化使风险监管的知识图谱建设迫在眉睫。知识图谱作为金融科技领域语义理解和知识搜索的基础技术，为金融科技服务于实体经济提供了深度技术，提供了金融领域知识的提取和融合，以及进行分析、推断和决策的依据，能促进金融科技信息技术与开放知识领域的融合，并可以为金融行业的风险评估、预测、反欺诈、精准营销、智能搜索等提供技术支撑。因此，越来越多的金融机构及企业在探索和构建金融领域知识图谱，为金融领域应用决策提供更加精准可靠的依据。

2018年证监会关于企业知识图谱的建设，即广泛应用于中央监管信息平台的上市公司画像、拟上市公司画像和中央监管私募机构画像，则是通过关联关系识别面向企业层面的风险防范，如非法集资、财务管理风险等，使金融风险管理适应于金融科技产品、服务、行业的发展成为了可能。

现阶段，知识图谱和人工智能技术在金融行业的应用非常广泛，如智能客服智能搜索及可视化展示、反身份作假和群体欺诈、异常分析、失联客户管理、产业链及担保链分析、用户画像与情感分析以及智能知识库等等。

**1.2.2 深度学习**

深度学习的概念由Hinton等人于2006年提出，其前身为神经网络。2016年DeepMind公司所开发的AlphaGo击败围棋世界冠军李世石之后，深度学习这一概念逐渐被人认可。通过深度学习技术，可有效地处理机器翻译、语音识别、图像分类以及自然语言处理等机器学习的问题。深度学习的基本原理是通过神经网络结构进行低维到高维的映射，从而能够对数据的特征进行提取，最后能够得到数据的特征表示。典型的深度学习模型见下图。



深度学习本质上是构建含有多个隐含层的机器学习架构模型，通过大规模数据进行训练，得到大量更具代表性的特征信息。从而对样本进行分类和预测，提高分类和预测的精度。这个过程是通过深度学习模型的手段达到特征学习的目的。深度学习模型和传统浅层学习模型的区别在于：

1. 深度学习模型结构含有更多的层次，隐含层节点的层数通常在5层以上，有时甚至包含多达10层以上的隐藏节点；
2. 明确强调了特征学习对于深度模型的重要性，即通过逐层特征提取，将数据样本在原空间的特征变换到一个新的特征空间来表示初始数据，这使得分类或预测问题更加容易实现。
3. 和人工设计的特征提取方法相比，利用深度模型学习得到的数据特征对大数据的丰富内在信息更有代表性。

深度学习算法打破了传统神经网络对层数的限制，可根据设计者需要选择网络层数。它的训练方法与传统的神经网络相比有很大区别，传统神经网络随机设定参数初始值，采用BP算法利用梯度下降算法训练网络，直至收敛。但深度结构训练很困难，传统对浅层有效的方法对于深度结构并无太大作用，随机初始化权值极易使目标函数收敛到局部极小值，且由于层数较多，残差向前传播会丢失严重，导致梯度扩散，因此深度学习过程中采用贪婪无监督逐层训练方法。即在一个深度学习设计中，每层被分开对待并以一种贪婪方式进行训练，当前一层训练完后，新的一层将前一层的输出作为输入并编码以用于训练；最后每层参数训练完后，在整个网络中利用有监督学习进行参数微调。

深度神经网络的训练过程主要是先逐层预训练初始化深度神经网络每一层的参数，然后通过监督学习对整个深度结构神经网络进行微调。

训练过程如下：

1. 将第一层作为一个自编码模型将第一层作为一个自编码模型，采用无监督训练，使原始输入的重建误差最小；
2. 将自编码模型的隐单元输出作为另一层的输入；
3. 按步骤2迭代初始化每一层的参数；
4. 采用最后一个隐含层的输出作为输入施加于一个有监督的层（通常为输出层），并初始化该层的参数；
5. 根据监督准则调整深度结构神经网络的所有参数。

**1.3 本系统的创新点和特色（1-2页）**

人工智能与知识图谱作为认知智能的关键技术，正成为智能金融中新的研究热点。这两种技术在金融领域的应用场景有高度的重合，往往是互为依托、互为补充。基于此，本课题项目基于深度学习、大数据计算及知识图谱等多项目前主流的技术手段设计并实现了对于证券企业公司的风险监测预警系统。建立了企业以及关联企业的公司主体知识图谱以及结合深度学习算法对新闻风险事件及时准确的作出分析判决预警。从而防范公司风险，避免由于证券公司风险引起系统性风险的发生。

**1.3.1 知识图谱展示企业完整信息**

近年来，随着我国上市公司数量快速增加，金融市场规模持续扩大，风险也在积聚。因此，为了及时发现市场主体的潜在风险，维护市场稳定，切实保护中小投资者利益，金融行业一线监管迫切需要加强监控力度和敏锐度。目前，金融机构已经把公司风险的防范与把控提升到非常重要的高度，实现公司内部风险信息监测预警已经成为金融行业的热点问题。

在风险防范领域，传统模式下事后的、手动的、基于传统机构性数据的风险防范模式需要大量人力的投入并且效率不高，另外其数据也缺乏多样性，所以已经不能满足金融科技新业态的需求。因此，金融机构应与时俱进地在传统监管模式的基础上，借助大数据、人工智能等新兴金融科技进行风险管控，减少人力支出，完成风险的主动识别与控制，有效降低风控合规成本，增强风控合规能力，从而寻求监管突破。

大数据时代的开启为海量数据资源的融合与智能解析提供了完整的新路径。面向不同数据源、不同数据结构和不同数据类型的企业风险信息的集成、融合及其深度挖掘，可以使已知数据中的隐含知识得以最大化的释放，从而对公司的智能监管进行赋能，更好地认识与识别金融行业公司风险。

根据上述要求，知识图谱技术首先将公开的上市公司和互联网信息进行抽取整理，然后研究其中各种实体的概念信息以及相互之间可利用的关联、隶属关系等，实现相关实体概念表述以及关联关系的可视化，从而建立公司主体及关联方主体知识图谱。最终，将建好的知识图谱与深度学习模型相结合，提出一些规则模型和概率模型等分析方法，构建公司内部风险监测预警机制，从而更有效、更准确地、更快速地防范公司风险，为公司提供科学的投融资决策及提升公司的风险管理水平，促进公司业务的健康、快速发展，避免引起系统性风险的事件发生。

**1.3.2 大数据集群系统高效存储处理新闻数据**

在互联网信息高速发展的时代，传统的数据存储处理系统已经不能够胜任并有效的处理当前系统所需数据量。大数据并不单纯代表数据量积累的足够多，大数据既包含结构化数据也包括非结构化数据，而且是以数量巨大、变化率高的形式存在。

本课题搭建spark数据集群为应对大量的新闻数据到来。Spark 允许将中间输出和结果存储在内存中，节省了大量的磁盘 IO。Apache Spark 使用最先进的 DAG 调度程序，查询优化程序和物理执行引擎，实现批量和流式数据的高性 能。同时 Spark 自身的 DAG 执行引擎也支持数据在内存中的计算。

**1.3.3 深度学习结合知识图谱精准风险预警**

传统的预警预警系统主要由基于关键词、基于指标、基于情感倾向这三类方法来实现。但都有各自的不足之处。

基于关键词的舆情预警就需要事先由专业人员给出相应的关键词并组成关键词词库，而词库的丰富度直接决定了预警系统是否可以准确的判断事件的风险类型。当出现某些未知类型的事件此系统就无法给出正确的判断。

基于指标的舆情预警系统需要根据新闻的传播度、转发量来判断事件的风险特征，这显然有悖于风险监控的实时性的要求，对于企业的有关新闻应该第一时间对其作出判断并作出相应的处理。

基于情感倾向类的舆情预警系统需要分析出事件中的情感倾向，但是在金融证券行业，大多数的新闻表述都是客观角度出发，这就导致很难获取到其中的情感特征。因此此类系统在企业新闻舆情分析方面效果并不明显。

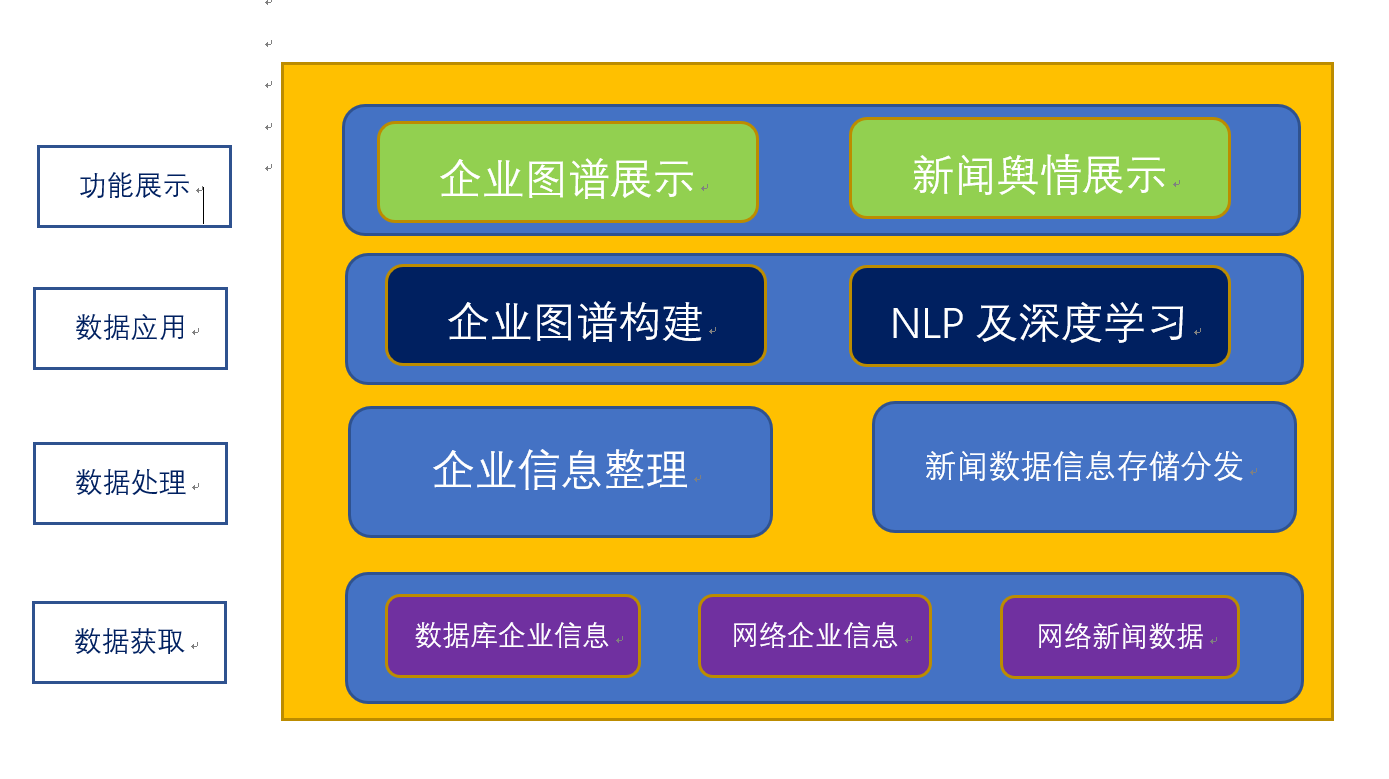
基于深度学习与知识图谱的前沿理论与方法，通过深入研究事件驱动对公司自营以及融资融券、股票质押等业务的风险传导机制以及国内外股票市场的运行规律，构建科学合理、操作性强的适合券商内部风险监测预警机制，提升公司的风险管理水平，促进公司业务的健康、快速发展。

**二 系统框架**

给出一个总图

图中分层，分模块介绍各自在系统中的功能

**2.1 系统总框架**

****

系统总体框架如上图所示，分为四个层面：

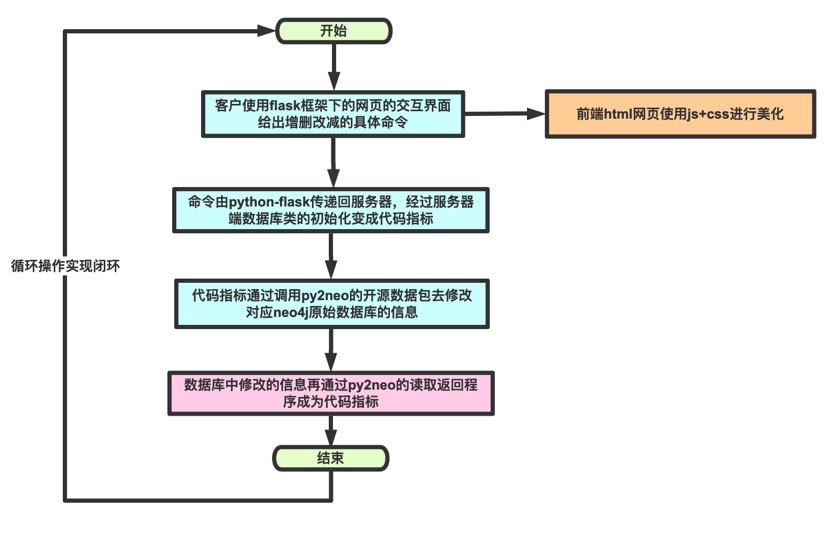
1. 最底层为数据获取层，主要功能为获取外部数据信息，包括企业信息、新闻事件信息。其中企业数据信息由两部分构成，分别事已有的数据库中获取和网页爬虫获取结构化及半结构化信息。新闻事件信息主要由爬虫工具实时监测重点新闻发布网站发布的头条最新新闻并获取。
2. 第二层为数据处理层。该层主要功能为对获取到的数据信息做处理。对企业信息筛选整理，提取其中有效的属性结构化信息作为企业知识图谱创建的数据来源。对于新闻事件数据信息及时的做存储及分发。存储到大数据集群系统中可以持久化数据信息，快速的数据分发能力可以极大的提高对于新闻事件的实时监控能力。
3. 第三层为数据应用层。功能主要分为对企业数据信息构建完整的企业知识图谱及企业关联图谱。NLP自然语言处理技术可以有效的提取新闻中关联到的企业、行业等信息，通过这些信息可以通过知识图谱的搜索推理功能可以关联到更多的企业主体。强大的深度学习技术可以很好的判别新闻舆情的类型及等级，更好的辅助用户，提高对企业的了解和监控能力。
4. 第四层为功能展示层。功能展示层可以将企业的主体知识图谱及关联图谱形象化的展示在网页上，让用户清楚的了解的企业的内部及外部信息。新闻舆情展示界面可以将每一条新闻舆情事件展示给用户。让用户时刻可以了解到所关注企业的舆情情况。

**2.2 知识图谱框架**

**2.2.1 知识图谱构建框架**



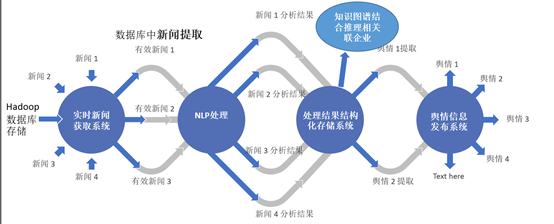
在公司主体知识图谱的构建过程中，第一步的爬虫所获取的上市公司的信息来源较为复杂，多数是半结构化、非结构化的文本信息，例如自营持仓主体的基本属性包括公司介绍、高管、集团、股东、债务、行业、上游、下游、概念、主营业务等属性；而最重要的是后三步，即知识单元的抽取、知识单元间关系的识别以及知识图谱的绘制：首先将原始数据按照BMEO规则进行标注，使用Python将标注好的数据处理成字标注的形式，将每一个字和标注转换成id的形式，供模型训练使用，并使用Pytorch框架通过命名实体识别，将得到的数字id根据之前的对应关系转换回汉字，稍做处理就可以获得文本中的实体；然后通过定义实体之间的关系，将该段文本转化为词向量表示，计算句子中每个词距离两个实体在句子中的距离，把距离转换成距离向量，与词向量合并，输入Pytorch模型进行训练；最后就可以得到关于上市公司知识图谱的三元组结果，存入Neo4j图数据库中，就得到了上市公司主体知识图谱。

****

**2.2.2 知识图谱功能实现框架**

上述提到的所用实现的功能和系统我们都是使用了flask框架作为交互通道，Web Application Framework（Web应用程序框架）或简单的Web Framework（Web框架）表示一个库和模块的集合，使Web应用程序开发人员能够编写应用程序，而不必担心协议，线程管理等低级细节。Flask自由、灵活，可扩展性强，第三方库的选择面广，开发时可以结合自己最喜欢用的轮子，也能结合最流行最强大的Python库。我们使用flask框架把以上提到的功能在服务器上进行部署，方便所有公网上的用户访问服务器对应端口对于网站进行访问从而完成对应的操作。这样以来，用户通过访问服务器对应的端口，通过flask框架与服务器进行交互，flask引出的html网页在post的条件下，request.form.get得到对应的信息反馈给服务器，服务器运行python脚本对neo4j数据库中的数据库的数据进行增删改查，然后用可视化的方法再回馈给用户，整体上实现了一个闭环操作，我们运用了neo4j这样一个强大的数据库平台构造企业知识图谱储存信息作为后台，使得交互又方便，可靠性又强。

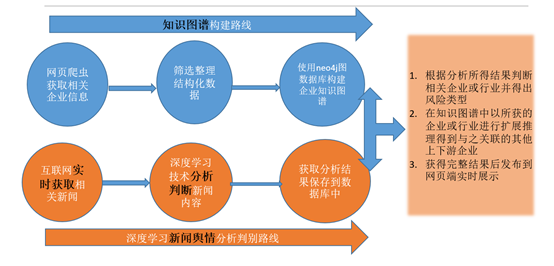
**2.3 新闻数据处理框架**

****

新闻数据处理系统功能实现主要分为如上图所示四个部分：

1. 实时新闻获取系统：此系统通过获取外界新闻并对新闻信息进行存储分发筛选，提取其中有效的新闻分发到下一系统中进行NLP分析处理。
2. NLP处理阶段通过预先给出的数据新闻以及专家提供的风险类型标签，利用Bert深度神经网络训练得出判别模型；NLP处理阶段为核心技术对于新闻中出现的相关企业、证券、股票或行业进行采用关键词向量匹配算法提取；NLP处理阶段为核心技术对于新闻中出现的相关企业、证券、股票或行业进行采用关键词向量匹配算法提取
3. 存储系统将处理结果以信息流的方式存储到Hadoop数据系统中，等待下一步结果信息提取
4. 舆情信息发布系统将信息发布到网页端，在网页端工作的业务人员可以第一时间了解到风险舆情并作出相应的处理

**2.4 知识图谱结合舆情舆情框架**

****

本课题采用知识图谱结合深度学习来对舆情做预警判断。知识图谱与舆情分析的结合体现在上图中：根据分析所得结果判断相关企业或行业并得出风险类型；在知识图谱中以所获的企业或行业进行扩展推理得到与之关联的其他上下游企业

**三 关键技术及实现 （结合之前你们写的三个部分）**

NLP处理，知识图谱、web框架构建、深度学习等算法

知识图谱及深度学习、自然语言处理在系统中的应用，如何构建公司主体图谱及关联、深度学习、自然语言处理应用什么算法对事件进行判定，如何进行模型学习等技术要点

**3.1 NLP处理算法框架及实现**

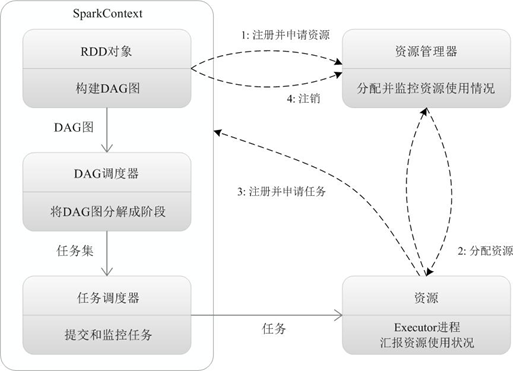
**3.1.1 spark大数据存储分发**

Spark是基于内存计算的大数据并行计算框架，可用于构建大型的、低延迟的数据分析应用程序。本课题为了应对每天高流量的数据流采用了spark大数据计算框架来实现对于高密度数据的处理。Spark计算框架主要有如下几个特点：

* 运行速度快：Spark使用先进的DAG（Directed Acyclic Graph，有向无环图）执行引擎，以支持循环数据流与内存计算，基于内存的执行速度可比Hadoop MapReduce快上百倍，基于磁盘的执行速度也能快十倍；
* 容易使用：Spark支持使用Scala、Java、Python和R语言进行编程，简洁的API设计有助于用户轻松构建并行程序，并且可以通过Spark Shell进行交互式编程；
* 通用性：Spark提供了完整而强大的技术栈，包括SQL查询、流式计算、机器学习和图算法组件，这些组件可以无缝整合在同一个应用中，足以应对复杂的计算；
* 运行模式多样：Spark可运行于独立的集群模式中，或者运行于Hadoop中，也可运行于Amazon EC2等云环境中，并且可以访问HDFS、Cassandra、HBase、Hive等多种数据源。

Spark的基本运行流程如下图4-1所示：

1. 当一个Spark应用被提交时，首先需要为这个应用构建起基本的运行环境，即由任务控制节点（Driver）创建一个SparkContext，由SparkContext负责和资源管理器（Cluster Manager）的通信以及进行资源的申请、任务的分配和监控等。SparkContext会向资源管理器注册并申请运行Executor的资源；
2. 资源管理器为Executor分配资源，并启动Executor进程，Executor运行情况将随着“心跳”发送到资源管理器上；
3. SparkContext根据RDD的依赖关系构建DAG图，DAG图提交给DAG调度器（DAGScheduler）进行解析，将DAG图分解成多个“阶段”（每个阶段都是一个任务集），并且计算出各个阶段之间的依赖关系，然后把一个个“任务集”提交给底层的任务调度器（TaskScheduler）进行处理；Executor向SparkContext申请任务，任务调度器将任务分发给Executor运行，同时，SparkContext将应用程序代码发放给Executor；
4. 任务在Executor上运行，把执行结果反馈给任务调度器，然后反馈给DAG调度器，运行完毕后写入数据并释放所有资源。

****

**3.1.2 NLP技术处理新闻数据**

**3.1.2.1 对企业主体的识别提取**

企业主体识别采用关键词识别的方式进行分析提取。对于完整的标题做分词切词得到一列词列表，由企业知识图谱中导入的企业名称和企业别名等等信息对分词后的结果进行比对筛选最终获取到相对于的企业主体。

对于已经获取到的企业主体可通过知识图谱的推理计算得到相关联的行业、上下游企业、子母公司等。通过这些推理得到的内容可以由单个企业产生的舆情预警到多企业的预警。

**3.1.2.2 对新闻风险类型判断**

新闻风险类型判断是指通过对新闻标题及内容进行深度学习的模型计算得到相对于的风险类型。在本课题中，通过专家给出的建议，风险类型大致分为如下类型：

1. 信用风险；
2. 市场风险；
3. 经营风险；
4. 法律风险；
5. 高管风险；
6. 子母公司风险；

在前期BERT模型预训练时，通过对于给定新闻标题的风险类型标注得到的预训练数据进行训练可以得到分类精度足够准确的模型。后期在对于新闻标题的风险类型判断中。只需将新闻标题输入到模型中就可以在很短的时间内得到相对于的风险类型。

**3.1.2.3 对新闻风险等级判断**

新闻风险等级判断采用关键词分析的方法来处理，由专家针对于每种风险类型给出相对应的多个关键词，并对不同关键词出现给与不同等级权重。

在经过风险类型判断后得到相应的风险类型并根据指定的风险类型关键词来对新闻标题做匹配分析，根据不同关键词出现的频率和相应的权重做加权计算得到相对应的风险等级。

**3.1.2.4 对企业主体综合指数评分**

对企业主体综合评分是最为重要的一项功能，通过这个功能，用户可以实时了解所关注企业的舆情等级及公司运营的情况。综合指标值反映了企业当天所发生的舆情严重等级。具体计算方法是对于发生过的舆情信息进行加权求和得到最终的综合指标。在专家的建议下赋予不同的风险类型舆情以不同的数值。再根据风险的等级作乘积，最终累计所有发生过的舆情数据信息得到相对应的综合指数。

**3.2 知识图谱算法框架及实现**

如今有多种方式构建知识图谱，人工方式比如Cyc，通过Freebase以及维基数据，或者从大规模、半结构化的数据集类似维基百科、DBpedia、YAGO抽取得来。此外，更多的学者提出了基于结构化或者半结构化的信息抽取系统，在这个基础上产生了NELL、PROSPERA和KnowledgeVault模型。

参考知识图谱的通用构建框架，本课题的公司主体知识图谱的整体搭建流程如图1所示。

具体实现过程如下：

1. 首先，利用网络爬虫技术，爬取搭建知识图谱所需的原始数据；

2. 其次，基于深度学习技术，以长短时记忆网络（LSTM）为语料特征学习模型，建立命名实体识别方案；

3. 另外，在实体关系抽取中，从公开的上市公司研报和公告或者新闻舆情信息中提取素材，结合领域知识和业务需求找到实体之间的关系，为知识图谱提供理论及数据支撑；

4. 最后，通过Neo4j图数据库，将实体与关系存储其中。

**3.2.1 网络爬虫**

网络爬虫又称为网络蜘蛛或网络机器人，是一个程序或脚本，根据一定的规则自动获取网络资源。它已被广泛应用于互联网领域，例如搜索引擎使用网络爬虫来爬取网页、文档甚至图片、音频、视频等资源，通过相应的索引技术组织这些信息，为搜索用户提供查询。传统网络爬虫的工作流程从一个或多个初始网页的URL开始，并从当前网页中连续提取新的URL并将其放入队列中，直到满足系统的某些停止条件。

本课题通过python爬虫工具对于网页中公开的的数据信息（例如与自营以及融资融券、股票质押等业务相关的股票持仓主体以及其关联方相关的新闻、舆情信息等全网信息）进行实时爬取，梳理其中有效的信息并对梳理后的数据信息进行整合，为知识图谱的构建提供原始信息数据。

**3.2.2 实体抽取**

实体抽取又称为命名实体识别，是指通过自动化手段从非结构化文本中识别具有特定含义的某一类实体概念的技术。命名实体识别的准确性直接决定了关系识别（实体关系抽取）的质量，也就影响到知识图谱构建的后续环节，因此是构建知识图谱的一个首要任务。

命名实体识别的任务被定义为：识别出文本中出现的专有名称和有意义的数量短语并加以归类。而命名实体是文本中最基本的信息元素，是正确理解文本的基础，至于其的确切含义，只能根据具体应用来确定。

利用统计学习方法，或者说机器学习方法来解决命名实体识别这样的自然语言处理问题，多年来已经有了较多的解决方案，其中条件随机场模型（CRF）等监督式学习方法虽然已经有了较为稳定的效果，但是这类方法通常需要配合繁琐的人工特征处理过程，且通常只能学习到与相邻词位置比较近的上下文的特征，而无法获取整个句子甚至更长的上下文的特征。因此，本课题所使用的解决方案则是近几年来较为流行的神经网络学习模型或者说深度学习模型。该方案相比较于传统的机器学习方法改进之处在于能够通过复杂的神经网络结构来学习数据的特征，并且可以由低层次至高层次自动地学习特征的层次结构，省去了需要人工介入的特征提取工程，从而可以获得更优秀的效果。

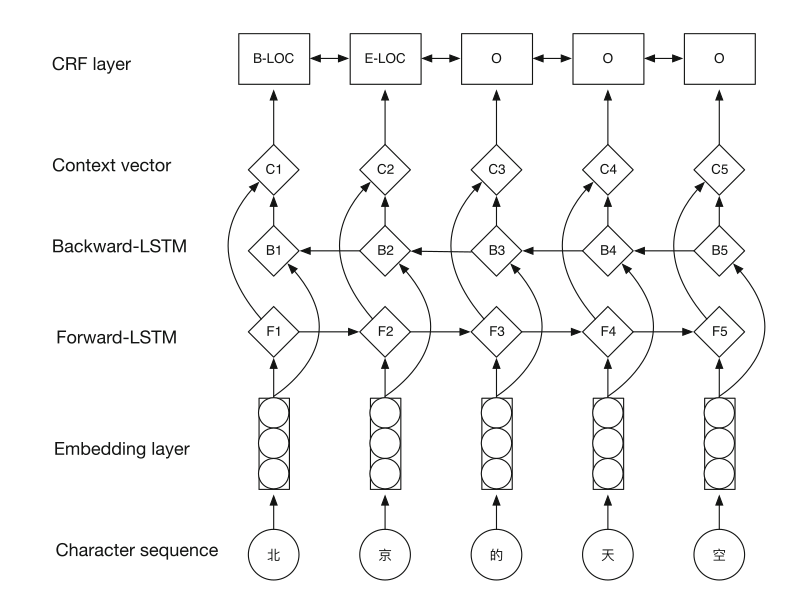
本课题中，命名实体识别的技术路线包括几个部分：

1. 采用嵌入词向量方案进行语言文本表示，使用无监督学习算法从大量未标记文本语料库中提取矢量；

2. 采用的特征学习模型为长短时记忆深度神经网络模型（LSTM）。LSTM是在循环神经网络（RN～N）的基础上改进而来的模型，而循环神经网络对于传统神经网络模型的改进之处就在于其对于时间序列输入的处理。在本课题的应用场景中，特征学习模型是构成双向LSTM网络结构的BiLSTM，该模型可以分别从前往后以及从后往前这两个方向来去学习上下文的特征，然后进行序列信号的记忆和传递；

3. 采用BIEO标注方案表进行数据标注。马尔可夫随机域（CRF）是一种用于标注和切分有序数据的条件概率模型，考察给定输入序列的标注序列的条件概率，因此在序列标注时，在顶层用马尔可夫随机域对BiLSTM的结果进行二次约束可得到更好的结果。

命名实体识别的整体模型结构见图2。

图2 实体抽取总体模型结构

如图2所示，圆形节点是模型的输入层，即Embedding layer（嵌入层）部分；菱形节点代表了长期记忆模型和短期记忆模型的双向网络结构；方块表示马尔可夫随机模型中的随机变量。在模型中要训练的参数包括条件随机场模型中的序列传递矩阵和双向长期和短期记忆模型中的参数，以及每一层中节点和节点的线性权重与偏置的大小。

**3.2.3 关系抽取**

实体关系抽取是自然语言处理（NLP）领域最重要的研究方向之一，其相关工作不仅包含监督模式下的分类学习，也涉及无监督学习的关系发现。其中，大量的工作主要是围绕如何设计出能够更好地描述实体之间句子的特征地办法，从而识别这些关系。

在大多数情况下，关系抽取被视为具有多个分类的有监督问题。有监督的方法能够抽取出有效的特征，然后在准确率和召回率方面是更有优势，因此监督学习方法被证明是解决关系抽取问题的有效方法，并取得了良好的分类效果。然而，传统的监督学习方法严重依赖于特征的质量。近年来，随着代表性学习和深度学习的发展，大量的深度学习技术被应用到自然语言处理中，并取得了许多令人惊叹的成果。

本课题涉及的大多数关系类型可以直接从权威的结构化数据中提取：作为资本市场，可以从自己的结构化数据库中获得提取所有上市公司的法定资本结构，如法定的公司投资关系和股东关系；或从市场关系的角度来看，公司产业链上下游的信息可以从公司披露的年度报告和相关公告中获得。这些信息基本上以半结构化数据的形式获得，利用信息抽取技术可以抽取和存储结构化信息。另外人与人之间的社会关系，最主要的提取方法是从个人在资本市场上的相关简历中获取个人经历，通常以半结构化的数据格式显示在文档中。

本课题采用语句级的粒度，研究一句话中任意两个实体之间是否存在联系。

实体关系抽取算法层次结构主要包括五个部分：输入层、词向量模型训练层、双向LSTM层、attention注意力机制层、关系分类输出层。

整体结构见图3。

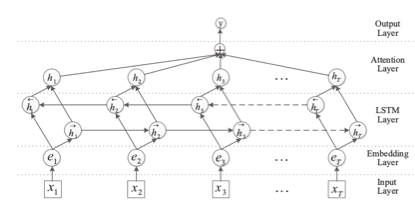


图3 BiLSTM+ATT模型结构图

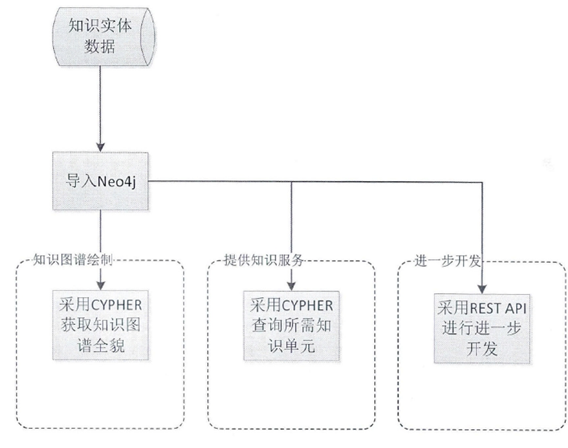
**3.2.4 知识存储**

知识图谱技术解决了知识的获取与整合，但仍需储存于某一介质，用于后续的查询和可视化展示。本课题选用图形数据库存储整合的知识图谱数据，将非结构化数据与半结构化数据，通过命名实体识别和实体关系抽取后，就可以得到结构化数据，再使用资源描述框架（RDF）对数据进行符号性描述，如[头实体]-[关系]->[尾实体]的三元组形式，最后将其上传到图数据库中。

图形数据库是一种非关系数据库（NoSQL）。在图数据库中，节点间的节点信息和链路信息构成整个数据集。节点信息类似于传统关系数据库中的表结构，每个节点对应表中的一段信息，而关系是图形数据库的唯一组成，关系数据库对应于表之间的数据关联。因此，与传统的关系数据库相比，图形数据库对节点和节点间复杂关系有更加良好的支持，所以在存储和表示大量关系方面具有更多的优势，成为了知识图谱存储的首选。

本课题的图形数据库采用Neo4j图形数据库，它是世界领先的开源图形数据库，以图形结构的形式存储数据信息，能将实体和关系通过图形的方式显示出来。它可以存储节点、节点属性值、节点之间关系值信息，因此可以使用图结构的自然伸展特性来设计免索引邻近节点便利的查询算法，即图的遍历算法设计。图的遍历是图数据结构所具有的独特算法，即从一个节点开始，根据其连接的关系，可以快速和方便的找出它的邻近节点。这种查找数据的方式并不受数据量的大小所影响，因为邻近查询始终查找的是有限的局部数据，不会对整个数据库进行搜索，所以，Neo4j具有非常高效的查询性能。

知识图谱绘制步骤见图4。

图4 绘制知识图谱流程图

1. 将通过爬虫等方法得到的结构化数据可以直接导入Neo4j数据库；而得到的非结构化数据进行命名实体识别和实体关系抽取后，采用批量导入方法将知识实体与实体之间的关系识别到图形数据库中。

2. 使用CYPHER查询语言对所有节点和关系进行查询，得到知识图谱的全貌。

3. 利用密码语言搜索所需的节点和关系信息，可以为用户提供个性化的知识服务。

4. 通过编程，可以调用Neo4j的REST API接口，从而进一步开发知识图谱接口。

**3.3 Flask web框架构建及实现**

Flask 是一个轻量级的基于python的web框架。Flask建立于一系列的开源软件包之上，这其中 最主要的是WSGI应用开发库Werkzeug和模板引擎Jinja。Flask是一个WSGI应用框架，在进行Flask开发时，不需要关注网络方面的操作，Flask应用的入口是封装过的网络请求包，出口是网络响应，在使用flask框架时仅需要关注这个阶段内的处理逻辑。本课题在网页展示方面采用了flask框架设计搭建网页。

**3.4 深度学习算法框架及实现**

深度学习算法的结构通常非常复杂，包含上百兆的可训练参数。比如2016年由He K等人提出的ResNet网络，最大的一种模型包含了上千层。深度学习算法的复杂网络结构是由有限的基本算子通过各种复杂连接关系搭建起来的，常用的算子包括卷积、池化、全连接等。因此，可以提取出机器学习算法中共性的部分，将其抽象出来，便于反复调用。

**3.4.1 基于数据流图的搭建框架**

基于数据流图（data flow graph）的机器学习编程框架利用节点（node）和边（edge）构造的有向图来描述计算过程。节点可以表示一个运算操作，或者表示一块数据的输入起点或者输出终点，边则表示节点之间的输入/输出关系。数据被表示为多维数组（张量）的形式，可以在这些边上进行传输。通过一个节点时，数据就会作为该节点运算操作的输入被计算，计算的结果则顺着该节点的输出边流向后面的节点。一旦输入端的所有数据准备好，节点将被分配到各种计算设备，完成异步并行的执行运算。

**3.4.2 基于层的搭建框架**

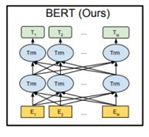
基于层的编程框架为用户提供一组表示各种层（比如卷积层、池化层、全连接层等）的函数作为接口。用户通过反复调用这些层的函数接口构成网络。通过将计算单位限制到层上，库的开发者们可以对各种层进行充分的性能优化，因此可以提供更好的运行效率。Caffe是一种常用的卷积神经网络框架，它采用层的调用方式，用一个prototxt文件对每一个层进行定义和配置，之后程序分析这个文件，获得这个网络结构的信息。

**3.4.3 基于算法的搭建框架**

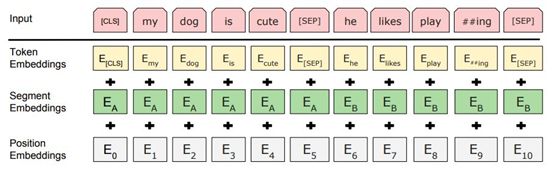
基于算法的编程库提供了大量的机器学习算法，涵盖各种任务和算法。和前两种框架不同，基于算法的编程库不需要构建复杂的网络结构，只需要通过设置库里面提供的算法函数接口中的参数即可完成任务。比如Scikit-Learn机器学习库，它是一种基于Python语言的机器学习库，提供了数据分析和数据挖掘中针对各种任务（包括分类、聚类、数据降维、模型选择、预处理等）的算法，提供了各种不同类别（如k-NN、k-means）的决策树（如C4.5、ID3等）的算法接口，开发者只需要设置接口中的参数，并且将数据传入，就可以得到训练和预测的结果。

**3.4 Google BERT框架**

BERT的全称是Bidirectional EncoderRepresentation from Transformers，即双向Transformer的Encoder。模型的主要创新点都在pre-train方法上，即用了Masked LM和Next Sentence Prediction两种方法分别捕捉词语和句子级别的representation。BERT模型的结构如下图所示：



针对不同的任务，模型能够在一个token序列中明确地表示单个文本句子或一对文本句子(比如[问题，答案])。对于每一个token, 其输入表示通过其对应的token embedding, 段表征(segment embedding)和位置表征(position embedding)相加产生。图2-2是输入表示的直观表示：



具体实现细节有以下几个部分：

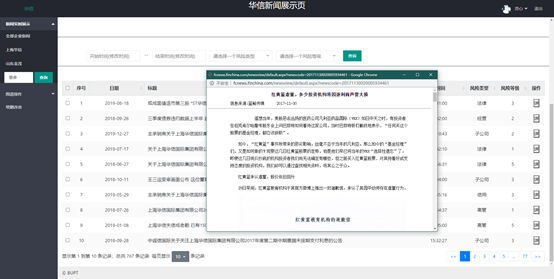
1. 论文使用了WordPiece embeddings含有30000个token，并且使用“##”来拆分词片段
2. 使用了positional embeddings，长达512位，即句子的最大长度
3. 每句话的第一个token总是[CLS]。对应它的最终的hidden state(即Transformer的输出)用来表征整个句子，可以用于下游的分类任务
4. 模型能够处理句子对。为区别两个句子，用一个特殊token[SEP]隔开它们，另外针对不同的句子，把学习到的Segment embeddings 加到每个token的embedding上
5. 对于单个句子仅使用一个Segmentembedding

**四 系统功能（结合之前你们写的三个部分）**

**4.1 新闻舆情分析处理及展示**

**4.1.1 新闻及舆情信息滚动展示**

****

****

新闻标题滚动页面图7-1和图7-2是综合显示系统所获取到的新闻，并实时更新，在新闻获取系统获取到了新的有效的新闻后会动态的插入到新闻列表的首条。此页面列表展示的新闻主要有新闻标题、新闻发生的日期及时间、通过BERT深度学习预测的对应新闻的风险类型、通过关键词匹配识别得到的相关企业主体以及当前风险所对应的风险等级。最后一列的详情按钮可小窗弹出完整的新闻内容。

滚动新闻展示的上方有多个可供选择的下拉框及输入框。通过下拉框及输入框选择的值可以对新闻列表进行筛选，找到所关心关注的新闻。可供选择的内容有日期选择范围、公司企业主体、风险类型以及风险等级。

**4.1.2 单个企业主体新闻舆情展示**

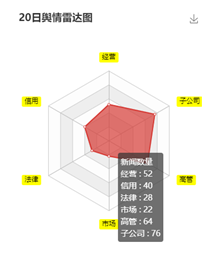
****

单个企业主体界面是在全部企业新闻信息滚动显示界面的基础之上提取其中所关注的当前企业的有关新闻进行展示。另外针对于单个企业系统给出了更多的分析结果及指标。

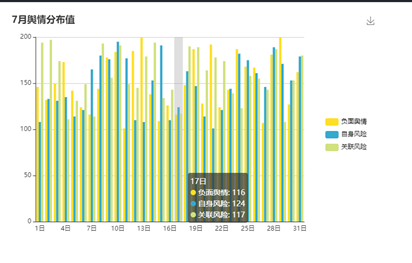
在页面底部是类似于全部企业新闻信息滚动展示界面的新闻滚动显示，区别在于取消了相关联企业的显示及筛选选择。在上方依次展示的是企业当天当前时刻所累积的舆情综合值数值，企业当天发生的不同风险类型时间的数量统计雷达图显示，以及企业当月每天的三种不同的舆情值柱状图显示。

下面是针对于单个企业主体的实时分析结果图：

下图显示的是企业当天风险舆情综合值，这是对于企业的一个综合评价。对于对一个企业做综合评价至关重要。此数值是根据当天企业发生的风险事件由风险类型及风险等级做加权相乘求和得到的数值。用户可根据此值的大小来确定企业的风险大小。

****

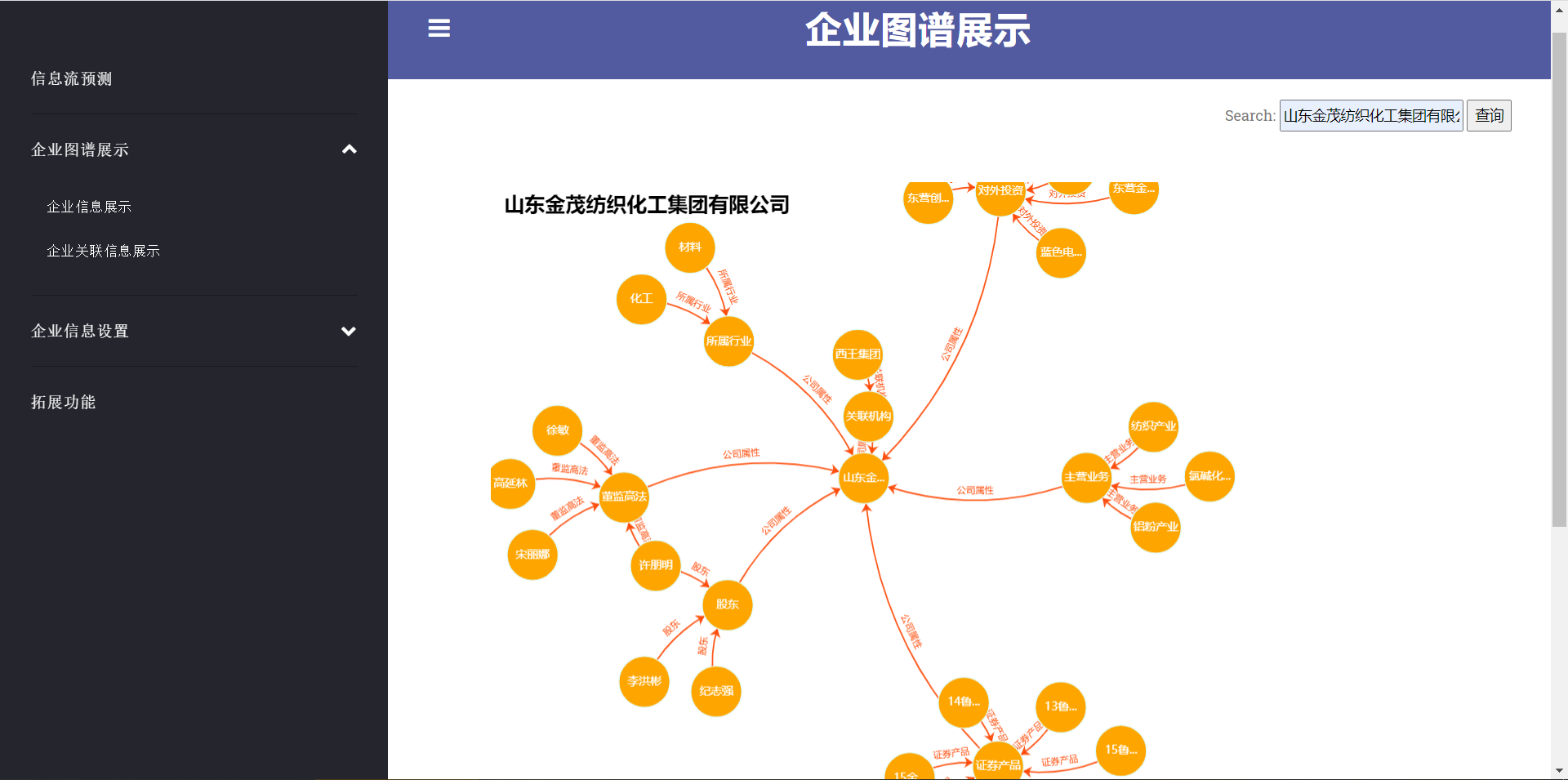
左图显示的是当天企业不同风险类型新闻发生的数量统计，用户可以实时掌握企业主体在哪些方面出现了较多的舆情，在哪些方面需要重点关注和监控。

****

下图显示为每月企业每天的舆情指数值。将舆情指标转化为三种指标，分别是负面舆情、自身风险、关联风险。用户可以清楚的观察到企业主体每月的舆情分布趋势以便对与企业的了解和掌握。

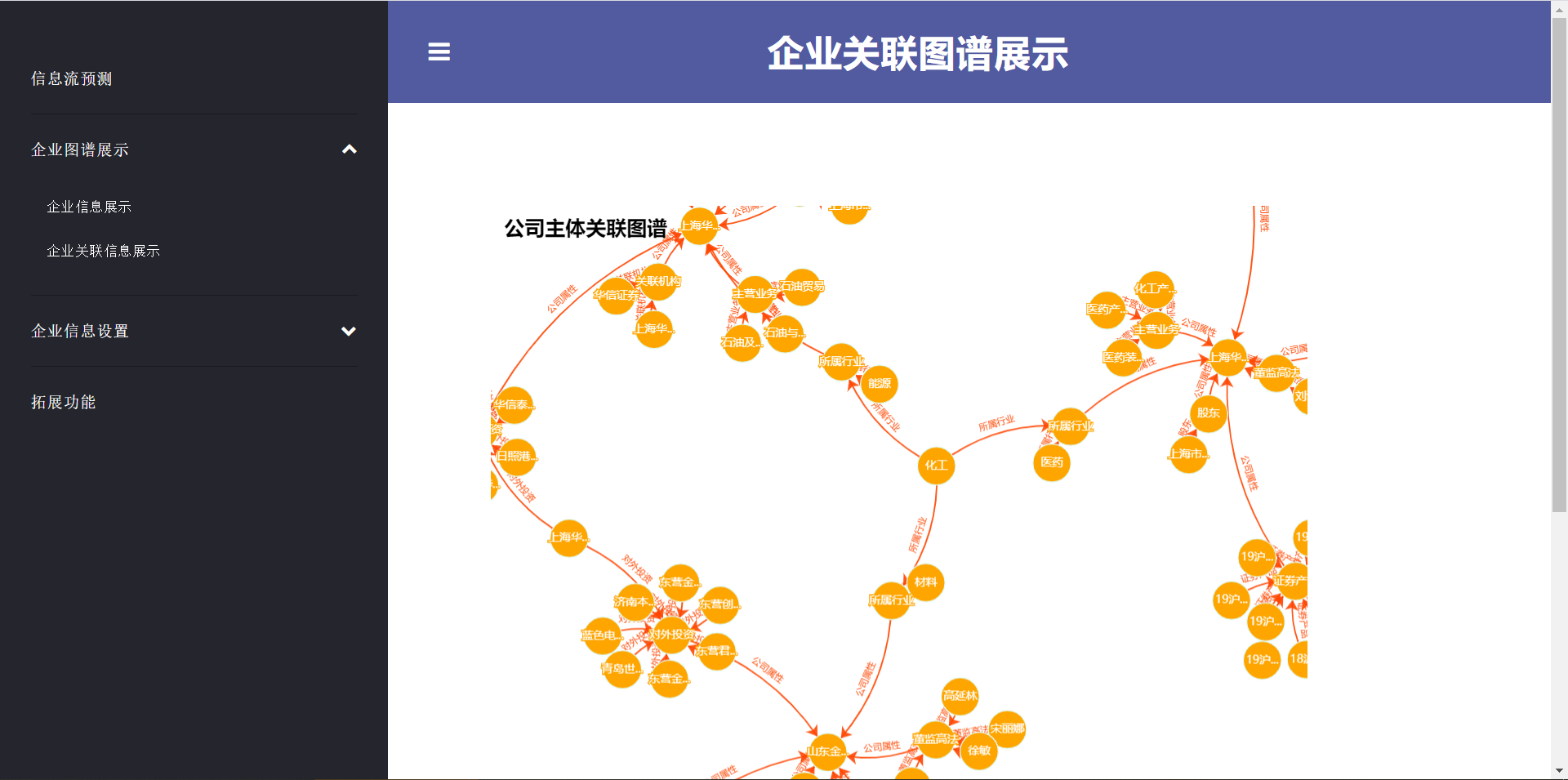
**4.2 知识图谱构建展示**

**4.2.1 单个企业信息展示**

图5 单个企业信息展示

在企业信息展示页面，在搜索框中输入某企业名称，在当前页面得到该公司主体的知识图谱，如图5所示，包括“山东金茂纺织化工集团有限公司”的公司主体、公司属性（例如董监高法、股东、对外投资、证券产品、主营业务、所属行业、关联机构等）以及其属性内容（例如徐敏、化工、西王集团等）。

**4.2.2 关联企业信息展示**

图6 关联企业信息展示

在企业关联信息展示页面，在当前页面得到不同公司主体相互关联的整体知识图谱，如图6所示，不同公司主体“山东金茂纺织化工集团有限公司”、“上海华信国际集团有限公司”以及“上海华谊（集团）公司”共同属于化工行业，且华信和金茂都投资了“上海华投资产管理合伙企业”这个公司。

**4.3 知识图谱应用及展示**

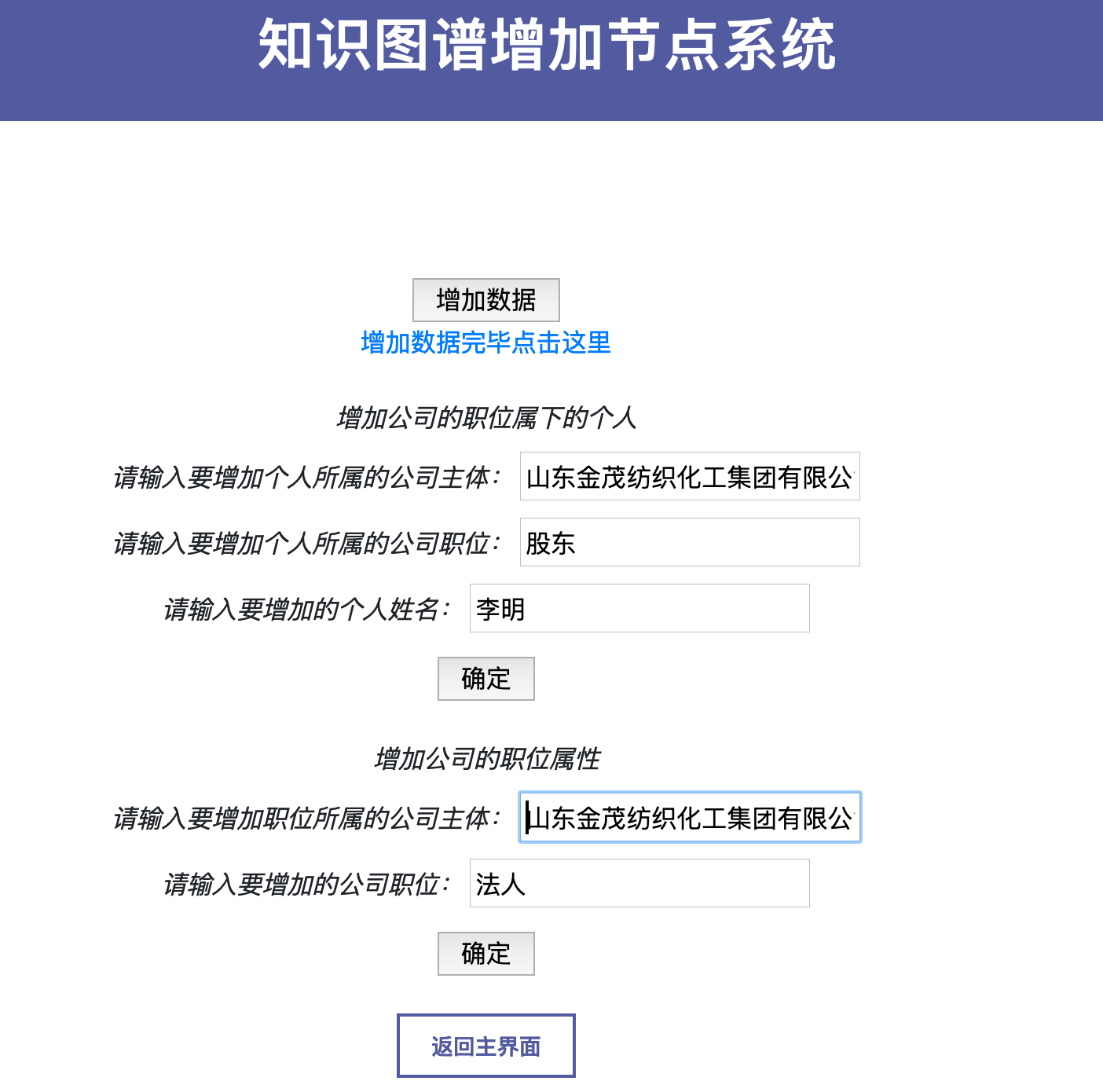
在实现了运用excel表格对于企业公司各类别数据进行导入后，如果我们需要对于知识图谱的每个节点进行一些主观的细小的调整，就需要引入增删改查功能对neo4j数据库中的内容进行即时修改。这里我们使用Python脚本+flask框架控制数据库进行对应的修改。这里我们选择调用py2neo的Python库联动neo4j进行间接控制，而没有选择使用Python脚本书写cypher语言对库进行直接操作。这样我们整个代码会有更好的封装性以及交互性。

首先Py2neo是一个客户端库和工具包，可从Python应用程序内部和命令行使用Neo4j。 该库支持Bolt和HTTP，并提供高级API，OGM，管理工具，交互式控制台，用于Pygments的Cypher词法分析器以及许多其他功能。 与以前的版本不同，Py2neo v5不再需要启用HTTP的服务器，并且可以完全通过Bolt运行。与neo4j官方给出的Python驱动程序相比较，Py2neo提供了更高级别的API和OGM，但是官方驱动程序提供了与集群一起使用的机制，例如自动重试。我们在这次实现增删改查的功能中主要运用了data, database, matching, ogm组件。

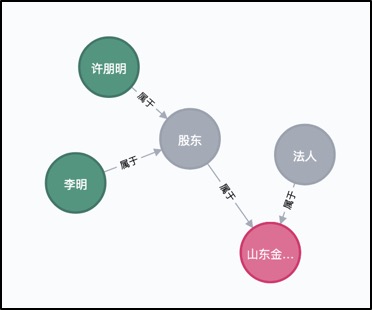
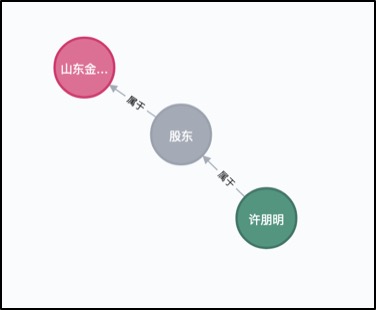
在对于具体的功能实现介绍之前，我们先简单阐述一下使用neo4j数据库对于知识图谱进行构建的框架，框架下除了几个根据行业划分的大型主体节点之外，还有公司主体节点，以及对应公司下属二级职位节点（董监高），三级节点则用于存储对应的人物姓名，每个相关所属的节点之间都有关系相连接，以保证在具体使用Py2neo对于库进行操作的过程中可以成功地检索，在增删改查的功能的实现过程中主要是对于我们所谓的二级职位节点和三级姓名节点进行增删改查操作。

**4.3.1 增加节点系统框架和实现**

对于增加节点这个功能我们根据实际的要求划分出了两个不同的板块，分别是增加二级职位节点以及增加三级姓名节点。前者是根据用户的需求对于企业下属的职位部分进行补充，在实现的过程中先matching用户在前端交互得到公司节点，py2neo.matching模块提供了根据某些条件匹配节点和关系的功能。对于每种实体类型，提供了Matcher类和Match类。Matcher可用于执行基本选择，返回可以评估或进一步完善的Match。NodeMatcher可用于查找满足一组特定条件的节点。通常，可以通过特定标签和属性键值对来标识单个节点。与此同时，允许任何数量的标签和Cypher WHERE子句支持的任何条件。运用graph.create()的方式创建新二级节点并建立关系。其中这里使用到的graph是Py2neo的一个类操作，py2neo.database软件包包含与Neo4j服务器交互所需的类和函数。 其中最重要的是Graph类，它代表Neo4j图形数据库实例，并提供对大部分最常用py2neo API的访问。通过Bolt或HTTP进行整个Neo4j图形数据库安装的访问器。 在py2neo对象层次结构中，数据库包含一个其中发生了大多数活动的图。 目前，Neo4j每个数据库仅支持一个图形。后者是对于企业某职位下的三级姓名节点进行补充。在实现的过程中先matching用户在前端交互得到公司节点以及下属的职位二级节点，运用graph.create()的方式创建新三级节点并建立关系。下图是演示的界面与效果。



图一：增加节点系统页面展示



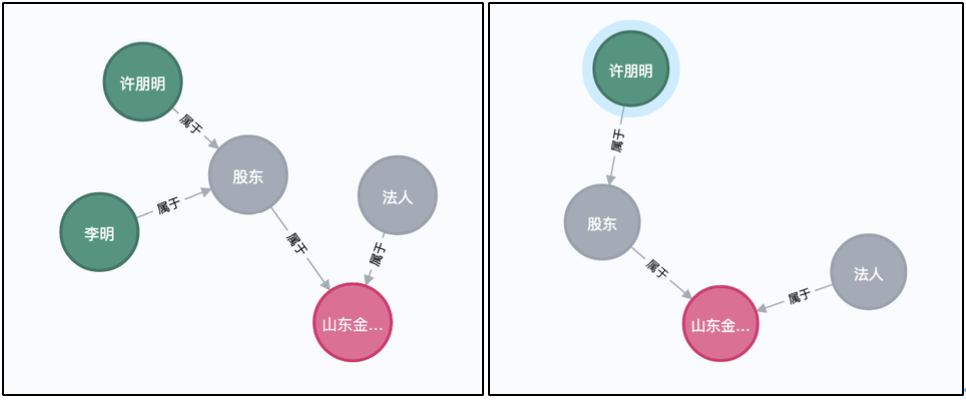
图二：修改前图谱显示及修改后图谱显示

**4.3.2 删除节点系统框架和实现**

删除节点功能主要是对于三级姓名节点的删除，因为实际情况下回存在企业管理人员离职或者转岗的情况，所以我们需要删除对应企业下属职位的对应人物并删除节点关联。这里具体实现方法是调用Py2neo库先matching用户需求删除的节点，再调用graph.delete()进行删除。由于调用py2neo的graph类进行删除操作的时候，节点删除会一并删除所有该节点建立的关系，因此我们在目前的知识图谱的背景应用下，不需要对于关系进行独立的删除操作。当然日后用户有对于知识图谱中关系部分新的应用也很方便解决。下图是演示的界面与效果。



图三：删除节点系统页面展示



图四：修改前图谱显示及修改后图谱显示

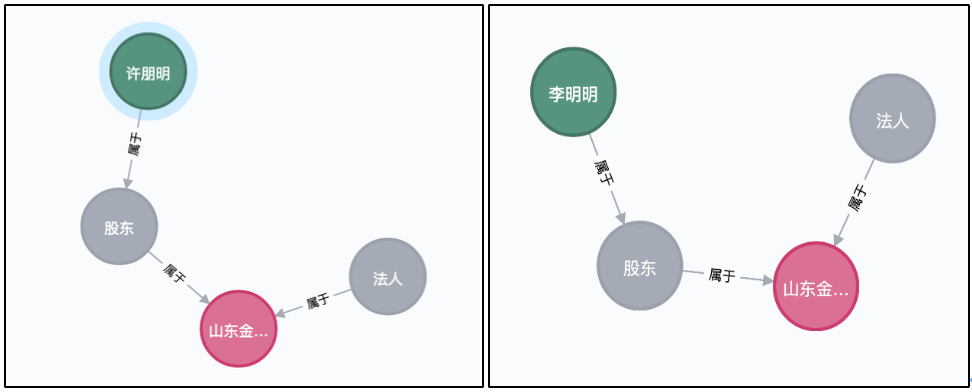
**4.3.3 修改节点系统框架和实现**

修改节点系统功能主要也是对于三级姓名节点进行信息修改，在实际情况下企业的管理人员可能经常会有变动，因此我们需要对于变动的人员进行姓名节点的调整。这里和删除节点系统功能的实现中，我们没有使用用户操作和实现较为简单的单节点匹配，因为在实际的应用中不能排除企业人员重名的情况，所以我们在使用姓名作为检索的这种方式必须要确保精准定位，因此也在前端要求用户必须输入需要修改/删除的三级姓名节点的上属企业节点和职位节点才能精确的匹配到我们需要操作的对应节点。其实这里提到的修改节点操作时前面所谓的删除操作和增加操作的叠加，只不过是我们使用了py2neo的ogm组件使得neo4j数据库与python端的新建类相联系，这样后续操作的可修改性就更强，py2neo中的ogm组件相当于从数据库到python类的一个映射，把neo4j中的对应标签信息作为Python中类的名字，把其下属的一些对应节点，当作类的成员，当然主体成员也是会按照cypher中的对应要素分为Node和Relationship两大类，但是在我们实际的应用汇总会淡化他们之间的区分，而会使用更加贴近用户使用的方法给用户一个更加贴近实际的描述。

py2neo.ogm软件包包含一组用于将Python对象绑定到底层图形数据的工具。类定义扩展了GraphObject，并包括属性和标签定义以及Related对象的详细信息。py2neo OGM框架的核心是GraphObject。 这是所有要映射到图形数据库的类的基类。 每个GraphObject都包装一个节点和一组指向RelatedObjects的指针以及连接它们的关系详细信息。可以像构造任何其他Python对象一样构造GraphObject实例，但也可以从数据库中对其进行匹配。每个实例可能包含代表标签，节点或相关对象的属性。在GraphObject上定义的标签为基础中央节点的标签提供访问器。它显示为布尔值，该值的设置允许打开或关闭标签。标签在API中的显示方式与布尔属性相同。两者之间的区别在于值如何转换为图形数据库。 如果定期对该值进行匹配，则通常应使用标签。次要信息或支持信息可以存储在布尔属性中。下图是演示的界面与效果。



图五：删除节点系统页面展示



图六：修改前图谱显示及修改后图谱显示

**五 总结和展望 （可以最后写）**

**5.1项目总结**

**5.2 项目未来展望**