中南大學
CENTRAL SOUTH UNIVERSITY

本科毕业设计(论文)

GRADUATION DESIGN(THESIS)

题　　目：	基于自然语言处理和知识图谱的试题知识点自动归纳方法
学生姓名：	刘哲潭
指导教师：	阳旺
学　　院：	计算机学院
专业班级：	大数据 1901 班

本科生院制

2023 年 5 月

基于自然语言处理和知识图谱的试题知识点自动归纳方法

摘要

在线教育的快速发展为中小学生提供了一种新的学习方式。对试题背后隐藏的知识点进行归纳，可以提高学生的学习效率。然而，人工归纳知识点费时费力，自然语言处理(Natural Language Processing，NLP)相关技术的发展使得自动归纳知识点成为可能。本课题致力于研究基于知识图谱和自然语言处理的相关技术实现高中试题知识点提取的方法。首先，借助自然语言处理命名实体识别（Entity Typing）技术结合可自定义的规则匹配算法，提出了一套从半结构化数据中自动构建学科知识图谱的方法。然后，提出了适用于知识图谱的 TF-IDF 匹配算法，用于从已构建好的知识图谱数据库中搜索并评估与试题相关的知识点。本课题所设计的试题知识点归纳系统，对于在线教育领域试题知识点自动化归纳问题有一定的研究意义。

关键词：教育 自然语言处理 知识图谱 命名实体识别 关键词匹配

Automatic induction of knowledge points based on natural language processing and knowledge graph

ABSTRACT

The rapid development of online education provides a new way of learning for primary and middle school students. It can improve students' learning efficiency to summarize the knowledge points hidden behind the test questions. However, manual induction of knowledge points is time-consuming and laborious. The development of Natural Language Processing (NLP) technology makes automatic induction of knowledge points possible. This topic is devoted to the study of knowledge mapping and natural language processing based on relevant technology to achieve the extraction of knowledge points in senior high school. Firstly, a new method of automatic mapping of subject knowledge from semi-structured data is proposed by Entity Typing and customizable rule matching algorithm. Then, a matching algorithm of TF-IDF suitable for knowledge map is proposed, which is used to search and evaluate knowledge points related to test questions from the established knowledge map database. The knowledge point induction system designed by this subject has certain research significance for the automatic induction of knowledge points in the field of online education.

Key words： Education NLP Knowledge graph Entity typing Keyword matching

目录

第 1 章　绪论

1.1　选题背景——AI 在国内外教育行业的发展

人工智能（AI）技术已经广泛应用于教育行业，有潜力去解决当今教育中的一些重大的挑战并促进教学实践的创新。人工智能的概念可以应用于各种教育场景下，例如：可以用于抄袭检测和考试诚信；方便老师根据学生的特点提出个性化的教学方式；对课程质量做出更准确的反馈，如果许多学生回答错误，人工智能可以专注于学生遗漏的特定信息或概念，因此教育工作者可以在材料和方法上进行有针对性的改进，学生也可以得到更及时的反馈[1]。

人工智能技术在教育领域的应用已经在国际上得到了广泛的重视。教科文组织致力于支持会员国利用人工智能技术的潜力以实现《2030 年教育议程》，同时确保其在教育领域的应用以包容和公平为核心指导原则 [2]。

在国内，政府对 AI+教育同样有相当高的重视程度。我国的“教育信息化”在推进程度上确实称得上先行一步，而“人工智能+教育”作为教育信息化的升级版，也正在被教育界内外所追捧。国务院先后将 AI+教育写入《教育信息化十年发展规划》、《新一代人工智能发展规划》、《教育信息化“十三五”规划》、《教育信息化 2.0 行动计划》、《中国教育现代化 2035》等多项指导文件之中，并在印发后指出 AI+教育的重点发展方向为利用大数据驱动知识学习，构建以学习者为中心的智能学习环境。

对于人工智能赋能教育产业（Artificial intelligence in education，AIED）的研究，国内外形成了智能导学系统（ITS）、人工智能教育机器人、机器学习、学习模型、智慧学习[3]等研究热点。

同时在疫情期间，线上教育成为主流教学模式。据 CNNIC 最新报告[4]显示，截至 2020 年 6 月，我国在线教育用户规模达 3.81 亿。疫情期间在线教育行业的日活跃用户数量从平日的 8700 万上升至春节后的 1.27 亿，升幅达 46%，新增流量主要来自三、四、五线城市。截至 2020 年 6 月，三线及以下城市在线教育用户占整体的 67.5%，同比提高 7.5 个百分点。线上教育的普及为 AI 技术在教育领域的大规模应用提供了坚实的基础。

目前 AI 领域的主流方向有自然语言处理（Natural Language Processing，NLP）、计算机视觉（Computer Vision，CV）、强化学习（Reinforcement Learning, RL）。目前在

计算机视觉方向上，OCR（Optical Character Recognition，光学字符识别）技术被广泛地用于搜题软件，例如：作业帮、小猿搜题等。

试题知识点抽取是自然语言处理领域的一个难点。目前的方法有基于规则、基于机器学习、基于深度学习的方法。基于规则的方法的优点是简单易懂，但缺点是需要人工设计规则，对于复杂的试题难以有效处理。另外一些研究者采用了基于机器学习的方法，通过训练模型来自动地学习试题中的知识点。这种方法的优点是可以自动地处理复杂的试题，但缺点是需要大量的标注数据，且模型的性能受到数据质量和数量的限制。近年来，一些研究者提出了基于深度学习的方法，如使用卷积神经网络（CNN）、循环神经网络（RNN）等。这些方法在某些情况下可以取得很好的效果，但需要大量的数据和计算资源。总之，试题知识点抽取是一个具有挑战性的问题，目前的研究还有很大的发展空间。

1.2 国内外研究现状

试题知识点自动化提取的目的是从试题中提取出与解题相关的必要知识，这些知识一般存储在现有的知识库当中，知识库的实现一般为知识图谱或 NoSQL 数据库。知识库存储了必要的知识实体（Entity）和知识实体之间的关系（Relation），通过知识实体之间的联系可以挖掘出关联的知识。已有的研究在初等数学教育领域构建了基础概念知识图谱与数学知识点体系图谱，并使用 TransE 得到了知识图谱的向量表示[5]。也有的研究将知识点标注的问题视作一个多分类问题，使用深度学习结合注意力机制训练了一个多分类模型[6]。除此之外，还有有研究者考虑使用 K-Means 等聚类算法试题自动提取与存储[7]。以上研究用到了自然语言处理和知识图谱的相关技术，以下小结对自然语言处理和知识图谱的研究现状做一些简要的介绍。

1.2.1 自然语言处理研究现状

自然语言处理(NLP)是一个研究和应用领域，探索如何使用计算机来理解和操纵自然语言文本或语音来做有用的事情。自然语言处理研究人员的目标是收集人类如何理解和使用语言的知识，以便开发适当的工具和技术，使计算机系统理解和操作自然语言，以执行所需的任务。自然语言处理的基础学科包括计算机与信息科学、语言学、数学、电气与电子工程、人工智能与机器人、心理学等。NLP 的应用包括机器翻译、自然语言文本处理和摘要、用户界面、多语言和跨语言信息检索(CLIR)、语音识别、人工智能和专家系统等多个研究领域[7]。

自然语言处理是由理论驱动的用于自动分析和表示人类语言的计算技术的集合。

然而，与人类相当的文本自动分析需要机器对自然语言有更深入的理解，这离现实还很遥远。有很多 NLP 的例子，如在线信息检索、聚合和问答，主要是基于依赖网页文本表示的算法，在一定程度上也是基于 NLP。这些算法在检索文本(IR)、将文本分成部分、检查拼写和单词级分析方面非常出色，但在句子和段落级分析方面并不成功。因此，当涉及到解释句子和提取有意义信息的问题时，这些算法的能力仍然非常有限。

自然语言处理有很多难点：

首先，语言是一组有限长度的句子，用有限的字母集构成，或者就语言语法而言，它们是用有限的符号词汇构成的。由于字母表集（汉字字典）是有限的，句子的长度也是有限的，所以在一种语言中句子集也是有限的。然而，在的许多理论研究中，通常认为语言是无限的，因此一些句子的大小也可能是无限大的，而对于具有实际用途的特定语言，则把研究局限于有限集。这是因为，用来处理语言的计算机有限的内存，和有限的处理能力。[7]

其次，语言是没有规律的，或者说规律是错综复杂的，语言是可以自由组合的，可以组合复杂的语言表达。语言是一个开放集合，我们可以任意的发明创造一些新的表达方式。语言存在歧义和隐含信息，需要结合上下文和常识来理解。语言存在多种风格和变体，比如方言、俚语、口语等。另外，NLP 也面临着数据稀缺、标注质量、迁移学习等挑战[8]。

随着大数据、机器学习和深度学习等技术的不断发展，自然语言处理领域也取得了重大进展。在语言建模领域，神经语言模型(Neural Language Model, NLM)已成为语言建模的主流方法，其中最著名的就是 GPT 和 BERT 模型。使用词向量可以有效地解决语言中的歧义和多义性问题，目前 Word2Vec 和 GloVe 是最常用的词向量模型。机器翻译是自然语言处理的一个重要应用，它可以将一种语言的文本自动翻译成另一种语言的文本，近年来，基于神经网络的机器翻译(Neural Machine Translation, NMT)已成为主流方法，并且在各种翻译任务中都取得了优秀的效果。文本分类是自然语言处理的另一个重要应用，它可以将文本分类到不同的类别中。目前，基于卷积神经网络(Convolutional Neural Network, CNN)和循环神经网络(Recurrent Neural Network, RNN)的文本分类模型是最常用的。信息抽取是从文本中自动抽取结构化信息的一种技术，目前，基于模板匹配和基于神经网络的信息抽取模型是最常用的。

1.2.2 信息抽取技术研究现状

信息抽取 (IE)是从非结构化或半结构化的机器可读的文档和其他电子表示源中自

动提取结构化信息的任务。在大多数情况下，该活动涉及通过自然语言处理(NLP)处理人类语言文本。在多媒体文档处理中的许多任务中，如自动注释和从图像、音频、视频、文档中提取内容，可以看作是信息提取，由于这个问题的难度较大，目前的 IE 方法专注于狭窄的受限领域[9]。

知识点自动提取需要用到信息抽取技术从不同来源、不同结构的数据中抽取出可用的知识单元，包括实体、属性和关系，并以此为基础，形成一系列高质量的事实表达，为上层模式层的构建奠定基础。它是实现自动化构建大规模知识图谱的重要技术。

信息抽取主要包括的子任务如下：

（1）命名实体识别（Named entity recognition，NER）

命名实体识别目的是识别中文文本中实体的边界和类别。NER 通过使用特定领域的现有知识或从其他句子中提取的信息，识别已知的实体名称(人员和组织)、地名、时态表达式和某些类型的数值表达式[10]。一个更简单的任务是实体检测，它的目的是在没有任何关于实体实例的现有知识的情况下检测实体。例如，在处理句子“M. Smith 喜欢钓鱼”时，命名实体检测将表示检测到短语“M. Smith”指的是一个人，在不一定需要有关于 M. Smith 的任何关联知识下就能做出判断。

（2）指代消解（Coreference resolution）

一般在语言学及我们日常用语当中，在下文采用简称或代称来代替上文已经出现的某一词语，语言学中把这种情况称为“指代现象”，也即是指代。将代表同一实体（Entity）的不同指称（Mention）划分到一个等价集合（指代链，Coreference Chain）的过程称为指代消解[11]。例如“国际商业机器公司”和“IBM”指的是同一个现实世界实体。如果拿“ M. Smith 喜欢钓鱼”和“但是他不喜欢骑自行车”这两个句子来说，“他”和“M. Smith”指代的是同一个人，因此也属于同一个实体。

（3）关系抽取（Relationship extraction）

关系抽取指的是抽取实体之间存在的关系，例如“M. Smith 在 IBM 工作”，则实体“M. Smith”和“IBM”的关系可以用一个三元组< M. Smith , work in, IBM>表示，关系提取任务需要对文本中的语义关系提及进行检测和分类。该任务与信息提取(IE)非常相似，但 IE 还需要去除重复关系(消歧义)。

对于从文本中提取关系三元组的任务通常需要结合以上三种任务共同完成，这个三个任务可以是流水线（pipeline）的方式依次进行。最近的研究也有使用一个深度学习模型一次性完成以上三个任务。

1.2.3 知识图谱研究现状

知识图谱是一个用于表示实体、概念和它们之间关系的图形化知识表示方式。它是一种人工智能技术，可以帮助计算机理解世界的语义和知识，并支持自然语言处理、数据挖掘、机器学习和推理等任务。

可以从“知识”和“图”的角度理解知识图谱的定义。图是一种结构，它相当于一组对象，其中一些对象对在某种意义上是相关的[12]。知识的概念则有点模糊，基于智能体遵循理性原则的假设(后来细化为有限理性的概念(Simon 1957)，包括“最优”决策的成本)，我们将知识描述为通过感知它为实现某些目标所采取的行动，从这个意义上说，知识是由观察者从外部赋予给这个智能体的，在内部，“知识”被编码在符号层面[13]。

知识图谱通常包括三个要素：实体、属性和关系。实体指具体的事物或概念，属性是实体的特征或描述，关系描述实体之间的连接和相互作用。例如，一个人可以是一个实体，他的姓名、年龄、性别等属性可以描述他，而他与其他人之间的家庭关系、工作关系等关系可以描述他与其他实体之间的联系。知识图谱可以帮助计算机系统更好地理解和应用知识，从而支持自然语言理解、智能搜索、智能推荐、智能问答等应用场景。

知识图谱是一个非常大的语义网络（Semantic Net），它集成了各种不同的信息源来表示关于特定话语领域的知识，语义网络和关联数据（Linked Data）的研究导致了许多开放数据集，这些数据集现在大多被重新命名为知识图谱。这些知识图通常是跨领域的并且许多开放的知识图谱都来自维基百科，因为它包含了分布在多个领域的大量事实知识。一些知识图的构建过程也得益于非结构化语料库和词典[6]。目前，一些常见的开放知识图谱有：DBpedia、Freebase、YAGO、NELL、Wikidata。

同时，也有许多商业公司开发了它们自己的私有知识图谱用于实现它们自己的应用程序。例如：Cyc 是历史最悠久的人工智能项目之一，也是一个常识性的知识库，它最早发行于 1984 年。Cyc 知识库包含 150 万个概念、2 亿个一般公理和针对各个领域(如医疗保健、交通运输和金融服务)的特定领域扩展。Facebook 的 Entity Graph 由 Facebook 维护，并用于支持图形搜索功能。它最初于 2010 年推出，包含有关 Facebook 用户的信息，即用户的个人资料信息、兴趣和用户之间的联系，Facebook 的知识图谱包含了 50 亿个概念，可以通过 Facebook Graph API 访问。谷歌的知识图谱在 2012 年推出，最初的目的是提升谷歌搜索引擎的搜索结果。

总的来说，目前围绕知识图谱的动态来自经济部门，而不是科学界，这源于这样一个事实：即成功的电子营销和电子商务在旅游业和其他领域的价值分配方面变得越来越重要[13]。而 Wahlster 在他的一次演讲中指出，正确结合归纳和演绎，基于知识的方法是未来人工智能研究的关键挑战[14]。

对于知识图谱在在 NLP 领域的应用，百度[15]和清华大学[16]相关的研究者都尝试将知识图谱得的实体融入到 BERT 模型当中。两者使用方式各有不同，清华 ERNIE 对 BERT 的结构进行了调整，而百度的 ERNIE 1.0 则没有改变 BERT 的整体结构，只是对 BERT 的 mask 机制进行了调整，百度随后提出了 ERNIE 2.0[17]，引入了持续多任务学习的概念（Continual Multi-task learning），并在大部分 NLP 任务上取得了更好的效果。

1.3 主要的研究内容

本课题主要研究使用自然语言处理相关的技术并构建知识图谱实现一个实体知识点抽取系统。系统接受指定学科的试题作为输入，能够从题干中抽取关键信息，通过查找本地知识库获取相关知识点，对其排序并作为输出。对于该系统的实现主要有三个难点：

（1） 如何利用半结构化的文档自动化地构建知识图谱

（2） 如何从输入的试题题干中抽取关键信息并找到对应的知识点

（3） 如何从对候选的知识点集合排序以选出最匹配的知识点

关于以上三个问题的讨论于解决，本文将在第 3 章和第 4 章展开论述。

1.4 论文的组织结构

论文的组织结构设计如下：

第一章为绪论，主要介绍了本课题的研究背景以及国内外对于自然语言处理和知识图谱相关技术的研究现状。

第二章介绍了自然语言处理和知识图谱相关的技术和方法。

第三章详细阐述了学科知识图谱的构建方法。

第四章阐述使用关键词在知识图谱上匹配的知识点算法。

第六章介绍系统的设计与实现。

第七章对系统的功能进行测试。

1.5 本章小结

本章首先介绍了课题的研究背景以，然后阐述了国内相关的试题知识点抽取相关的

研究形况以及自然语言处理和知识图谱相关的研究情况。最后对总结了本课题研究的内容和论文的组织结构。

第 2 章　相关技术和理论

2.1 自然语言处理相关技术

2.1.1 Transformer

Transformer[12]是 BERT 和 ERNIE 模型的基础，在自然语言处理领域，Transformer 已成为一种非常强大和流行的模型，广泛应用于机器翻译、文本生成、问答系统等任务中。Transformer 由 Google 在 2017 年首次提出。与传统的循环神经网络（RNN）或卷积神经网络（CNN）不同，Transformer 使用注意力机制来捕捉序列中的依赖关系，从而提高了模型的性能和可扩展性。

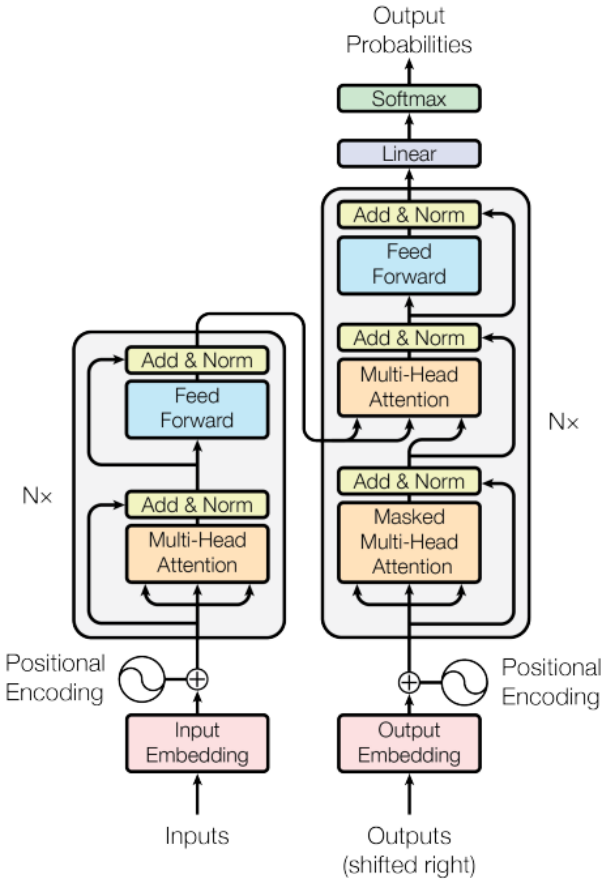

图 3-1 Transformer 结构

Transformer 模型由编码器和解码器两部分组成。编码器将输入序列转换为一系列高级特征表示，解码器则使用这些特征表示来生成输出序列。每个编码器都由多个相

同的层（Transformer 层）组成，每个 Transformer 层包含两个子层：多头自注意力机制和前向神经网络。而解码器使用了自回归方式输出，它有三层，相比于编码器在中间加入一层多头自注意力机制以融合编码器的输出和解码器上一步输出的 output。

多头自注意力机制通过计算输入序列中各个位置之间的注意力权重来捕捉序列中的依赖关系。前向神经网络则通过多层感知器对每个位置进行单独的变换，以产生高级特征表示。

单头注意力计算公式如下：

$$Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2-1}$$

输入的Q,K的维度是d_k，输入V的维度是d_v。

多头注意力的计算方式如下：

$$MultiHead(Q,K,V) = Concat(head_1,\dots,head_h)W^O \; where\; head_i$$
$$= Attention(QW_i^Q, KW_i^K, V\,W_i^V) \tag{2-2}$$

其中：$W_i^Q, W_i^K \in R^{d_{model}\times d_k}$，$W_i^V \in R^{d_{model}\times d_v}$ $and\; W^O \in R^{hd_v\times d_{model}}$

在 Transformer 中，所有子层的输出都通过残差连接和层归一化进行组合，以避免梯度消失和梯度爆炸问题。此外，Transformer 中还使用了位置编码来表示序列中各个位置的相对位置关系。

2.1.2 BERT

BERT（Bidirectional Encoder Representations from Transformers）是 Google 在 2018 年发布的一种预训练语言模型，它是一种基于 Transformer 的深度双向编码器，是一种无监督的学习方法，可以从大量无标注的文本中学习到通用的语言表示。

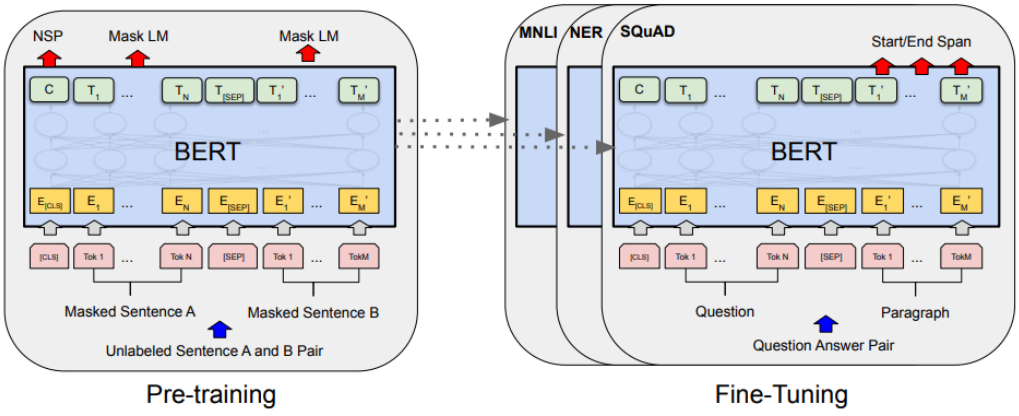

图 3-2 Bert 的预训练和微调

与传统的语言模型不同，BERT 的设计中使用了双向 Transformer 编码器，可以从左到右和从右到左同时处理输入，这使得它可以捕捉到文本中的上下文信息。此外，BERT 在预训练过程中使用了两个任务：掩码语言模型(Masked Language Model, MLM)和下一句预测任务(Next Sentence Predict, NSP)。掩码语言模型任务是将输入文本中的某些单词掩盖，并要求模型从上下文中预测被掩盖的单词。下一句预测任务则要求模型判断两个输入句子是否是连续的。

在预训练完成后，BERT 可以用来微调（fine-tuning）各种下游任务，如文本分类、命名实体识别、语义相似度等。最简单的做法就是在 BERT 后加一个全连接层，将 BERT 编码输出得到的嵌入表示作为下一层的输入。与传统的基于卷积神经网络和循环神经网络的模型相比，BERT 在各种自然语言处理任务中都表现出了更好的性能和更好的泛化能力。

BERT 的出现极大地推动了自然语言处理领域的发展，也启发了许多后续的预训练模型的研究和开发，如 GPT、RoBERTa 等。

2.1.3 ERNIE

在 BERT 提出之后，百度和清华大学的研究者均在 2019 年分别提出了 ERNIE（命名一样，以下分别用清华 ERNIE 和百度 ERNIE 加以区分）。

清华 ERNIE 考虑加入知识图谱(KGs)，知识图谱可以提供丰富的结构化知识事实，以更好地理解语言。清华 ERNIE 对 BERT 的结构进行了修改，加入了知识图谱的嵌入表示，并设计了 aggregator 结构用于融和知识图谱嵌入表示的信息：

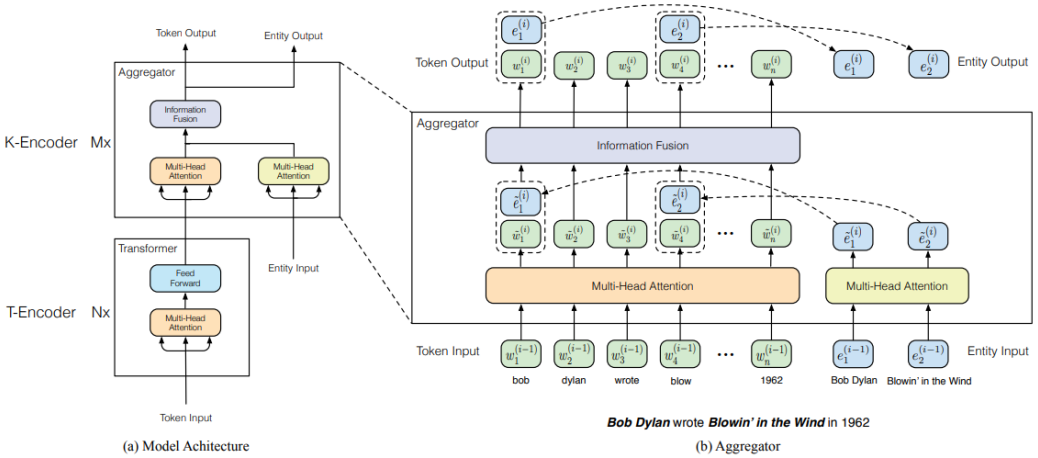

图 3-3 清华 ERNIE 结构

它主要由 T-Encoder 和 K-Encoder 两部分构成，T-Encoder 就是一个 Transformer 编码器，计算方式如下：

$$\{\boldsymbol{w_1},\ldots,\boldsymbol{w_n}\} = T-Encoder(\{w_1,\ldots,w_n\}) \quad (2\text{-}1)$$

K-Encoder 由许多个 aggregator 构成，aggregator 用于融合 T-Encoder 编码器输出和由 TransE 转化的嵌入表示的知识图谱，首先使用了两个注意力模块分别对编码器输出和知识图嵌入进行编码：

$$\{\tilde{w}_1^{(i)},\ldots,\tilde{w}_n^{(i)}\} = MH-ATT(\{w_1^{(i-1)},\ldots,w_n^{(i-1)}\}) \quad (2\text{-}2)$$

$$\{\tilde{e}_1^{(i)},\ldots,\tilde{e}_n^{(i)}\} = MH-ATT(\{e_1^{(i-1)},\ldots,e_n^{(i-1)}\}) \quad (2\text{-}4)$$

然后对两者信息进行互相集成：

$$h_j = \sigma\left(\tilde{W}_t^{(i)}\tilde{w}_j^{(i)} + \tilde{W}_e^{(i)}\tilde{e}_k^{(i)} + \tilde{b}^{(i)}\right) \quad (2\text{-}3)$$

$$w_j^{(i)} = \sigma\left(\tilde{W}_t^{(i)}h_j + b_t^{(i)}\right) \quad (2\text{-}5)$$

$$e_k^{(i)} = \sigma\left(\tilde{W}_e^{(i)}h_j + b_e^{(i)}\right) \quad (2\text{-}6)$$

为了简单起见，第 i 个 aggregator 的操作可以表示如下：

$$\{w_1^{(i)},\ldots,w_n^{(i)}\},\{e_1^{(i)},\ldots,e_n^{(i)}\} = Aggregator(\{w_1^{(i-1)},\ldots,w_n^{(i-1)}\},\{e_1^{(i-1)},\ldots,e_n^{(i-1)}\}) \quad (2\text{-}7)$$

最后使用了 dEA、MLM 和 NSP 之和作为预训练最终的损失函数。

百度 ERNIE 没有对模型结构进行任何修改，采用的就是完全的 BERT 模型。作者想办法对 masked language model 进行修改，使得其分层次地对 token 进行 mask。

如下图所示，分别在 Basic、Entity 和 Phrase 层面进行 Mask。在 Entity 层面中 Harry Potter 是一个实体必须整个 Mask，在 Phrase 层面对短语进行 Mask，例如对 a series of 这个短语 Mask

Sentence	Harry	Potter	is	a	series	of	fantasy	novels	written	by	British	author	J.	K.	Rowling
Basic-level Masking	[mask]	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	J.	[mask]	Rowling
Entity-level Masking	Harry	Potter	is	a	series	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]
Phrase-level Masking	Harry	Potter	is	[mask]	[mask]	[mask]	fantasy	novels	[mask]	by	British	author	[mask]	[mask]	[mask]

图 3-4 百度 ERNIE Mask 策略

2.2 知识图谱相关技术

2.2.1 图数据库

图数据库是一个使用图结构进行语义查询的数据库，它使用节点、边和属性来表示和存储数据。与关系型数据库相比，图数据库可以更好地处理高度连接的数据集，例如社交网络、地理信息、生物网络、推荐系统等等，因为它们可以轻松处理节点之间的多种关系，而不需要使用复杂的联接操作。

图数据库和关系数据库之间最显著的区别是，图数据库将数据之间的关系存储为数据，而关系数据库则以不同的方式推断数据之间的关系。关系焦点在数据表的列之间，而不是数据点之间。

图数据库通常提供了一些特殊的数据类型和算法，例如遍历、聚合、路径搜索等，这些操作可以帮助用户更好地理解和分析图形数据集。此外，一些图数据库还提供了可视化工具和交互式查询界面，以帮助用户更好地探索和发现数据集中的模式和趋势。

常见的图数据库包括 Neo4j、ArangoDB、OrientDB、JanusGraph 等，关于现有的图数据库之间的差异已有了一些总结[18]。

2.2.2 信息抽取

实体关系抽取（Entity and Relation Extraction，ERE）是信息抽取的关键任务之一。实体关系抽取分为实体抽取和关系抽取两个子任务，如何协调好这两个任务是当前的热点研究方向。实体关系抽取的方法主要有 pipline 和联合抽取。

Pipeline 依次进行实体抽取和关系抽取。Pipeline 方法易于实现，这两个抽取模型的灵活性高，实体模型和关系模型可以使用独立的数据集，并不需要同时标注实体和关系的数据集。但存在误差积累、实体冗余交互缺失的缺点[19]：

联合模型将两个实体抽取和关系抽取统一建模，可以进一步利用两个任务之间的潜在信息，以缓解错误传播的缺点。相比于传统的 Pipeline 方法，联合抽取能获得更好的性能。联合抽取的难点是如何加强实体模型和关系模型之间的交互，比如实体模型和关系模型的输出之间存在着一定的约束，在建模的时候考虑到此类约束将有助于联合模型的性能[19]。现有联合抽取模型总体上有两大类[20]：

（1）共享参数的联合抽取模型

共享参数的模型通过共享输入特征或者内部隐层状态实现联合抽取，由于使用独立

的解码算法，导致实体模型和关系模型之间交互不强。已提出的基于共享参数的模型有依存结构树[21]、指针网络[22]、多头选择机制等[23]。

（2）联合解码的联合抽取模型

为了加强实体模型和关系模型的交互，复杂的联合解码算法例如整数线性规划等被提出来了。在联合解码情况下需要对子模型特征的丰富性以及联合解码的精确性之间做权衡。如果设计精确的联合解码算法，往往需要对特征进行限制。相关的工作有：基于位置注意序列标记的实体和重叠关系联合提取[24]、基于新标记方案的实体和关系联合提取[25]。

2.3 本章小结

本章主要介绍了自然语言处理和知识图谱相关的技术原理。首先介绍了 Transformer 的结构原理，然后进一步介绍以 Transformer 为基础的 BERT 模型，以及对 BERT 改进的 ERNIE 模型。然后详细介绍了知识图谱相关的技术，包括图数据库、图嵌入技术和用于构建知识图谱的信息抽取技术。

第 3 章　学科知识图谱构建

3.1 知识图谱的构建动机

考场上考生由于受限于考试时间往往不会一字不漏地读完整道试题，而是关注于题干中的某些**关键词**，并由此迅速地联想到与之相关的**知识点**。对于简单的试题，在题干中可能就给出了与之相关的知识点的提示信息，例如在图 3-1 中的例题中，通过题干和选项中的关键信息“电离”、“溶液”、“强电解质”、“弱电解质”等关键词在“弱电解质”这个知识点下频繁出现，由此可以定位到该知识点，如图 3-2 所示。

2．已知：①$Al(OH)_3$ 固体的熔点为 300 ℃，电离方程式为 $H^+ + AlO_2^- + H_2O \rightleftharpoons Al(OH)_3 \rightleftharpoons 3OH^- + Al^{3+}$；②无水 $AlCl_3$ 晶体易升华，溶于水的电离方程式为 $AlCl_3 = Al^{3+} + 3Cl^-$；③熔融状态的 $HgCl_2$ 不能导电，稀溶液具有弱的导电能力且可作为手术刀的消毒液。下列关于 $Al(OH)_3$、$AlCl_3$ 和 $HgCl_2$ 的说法正确的是(　　)

A．均为强电解质　　B．均为弱电解质

C．均为离子化合物　　D．均为共价化合物

图 3-1 知识点直接在题干中出现

考点一　弱电解质的电离平衡

知识梳理 · 夯基础

1．弱电解质

(1)概念

电解质 —在水溶液里— 全部电离 → 强电解质，包括强酸、强碱、大多数盐等；部分电离 → 弱电解质，包括弱酸、弱碱、水等

(2)与化合物类型的关系

强电解质主要是大部分离子化合物及某些共价化合物；弱电解质主要是某些共价化合物。

图 3-2 在图 3-1 中的试题对应的知识点，图片中对应的关键词用红色下划线标记

然而，为了提高难度以考察学生对所学知识的理解程度，难度较大的试题所蕴含的关联知识点一般不会在题干中明确出现，但是还是会在题干中给出与之关信息。例如，

图 3-3 所示的北京地区的一道高考试题考查的知识点是“外因对电离平衡的影响”，虽然题干中并没有明确出现“弱电解质”、“电离平衡”这些词，但是通过检索“醋酸”、“CH_3COOH 溶液”“CH_3COONa”、“NaOH”、“$NaCO_3$”等关键词，发现其大多数在“外因对电离平衡的影响”这一知识点下出现，所以还是能找到隐含的知识点。

2．(2020·北京，11)室温下，对于 1 L 0.1 mol·L^{-1} 醋酸溶液。下列判断正确的是(　　)

A．该溶液中 CH_3COO^- 的粒子数为 6.02×10^{22}

B．加入少量 CH_3COONa 固体后，溶液的 pH 降低

C．滴加 NaOH 溶液过程中，$n(CH_3COO^-)$与 $n(CH_3COOH)$之和始终为 0.1 mol

D．与 Na_2CO_3 溶液反应的离子方程式为 $CO_3^{2-}+2H^+=\!=H_2O+CO_2\uparrow$

图 3-3 知识点隐藏于题干中，但是仍然可以通过关键词挖掘隐含的知识点

3．外因对电离平衡的影响

以 0.1 mol·L^{-1} CH_3COOH 溶液为例，填写外界条件对 $CH_3COOH=CH_3COO^-+H^+$　$\Delta H>0$ 的影响。

改变条件	平衡移动方向	$n(H^+)$	$c(H^+)$	导电能力	K_a
加水稀释	向右	增大	减小	减弱	不变
加入少量冰醋酸	向右	增大	增大	增强	不变
通入 HCl(g)	向左	增大	增大	增强	不变
加 NaOH(s)	向右	减小	减小	增强	不变
加 CH_3COONa(s)	向左	减小	减小	增强	不变
升高温度	向右	增大	增大	增强	增大

图 3-4 在图 3-3 中的试题隐含的知识点

以上例子说明：在解题的过程中只要找准了题干中的关键信息，通过到关键词和知识点之间的关系“联想”到可能的知识，就能定位试题考查的知识点。通过知识图谱对关键词和知识点之间的联系进行建模，可以由关键词找到与之关联的知识点集合作为候选集，并对候选集中的知识点排序。

在当前主流的教学辅导都对考试考查的知识进行了精炼的总结。通常一个章节或一

个小结的标题便是对接下来下知识的总结。知识点与知识点之间存在层级，知识点之间的层级一般由目录结构组织，同时考虑对知识点用一个关键词的集合描述。通过知识图谱建立知识点与知识点、知识点与关键词之间的关系，便可以通过关键词索引对应的知识点。

3.2 构建知识图谱的挑战

本课题用到的数据是学科教学辅导文档，以 word 的方式保存。本课题研究并实现一个试题知识点归纳系统，需要以学科知识图谱作为支撑。考虑从半结构化的文档中抽取知识点以及关键词并构建知识图谱，然后设计特定的基于知识图谱的查找算法实现对于试题的知识点匹配。对于本课题构建的试题知识点归纳系统，知识图谱的构建尤为重要，为基于关键词的知识点查找算法奠定基础。

然而，为试题知识点归纳系统构建知识图谱需要解决以下几个问题：

（1）知识图谱建模

知识图谱一般由实体（Entity）和关系（Relation）构成，实体是知识图谱的节点，关系是知识图谱中的边。实体和关系一般都可以附带属性或标签，如图 3-5 所示。标签通常用于将实体分组，如果不进行分组的话，就会在所有的实体中查找，从而大大降低了查询速率。对于本课题的知识点归纳系统，如何对知识点之间的层级关系、对关键词和知识点之间关联关系建模非常必要。

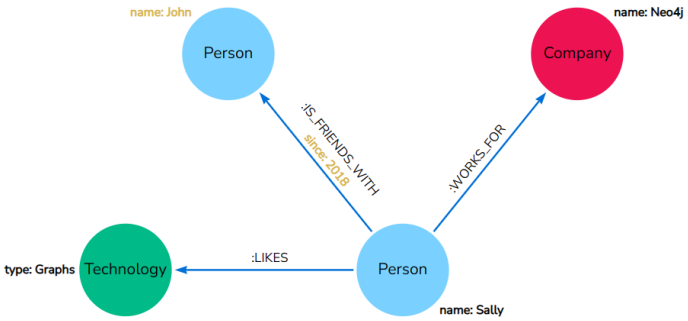

图 3-5 实体、关系的标签和属性

（2）构建知识图谱的方法

通过文本构建知识图谱，需要从文本中抽取出实体和关系。最直接的方法是手工标记实体三元组$< Head\ Entity, Relation, Tial\ Entity >$，但是这样工作不仅繁琐、耗时巨

大，人工关联记录也非常容易出错。

第二章探讨了使用深度学习自动地提取实体、关系的方法。然而，基于监督学习的方法需要的训练数据集需要大量的人工数据标注才能达到较好的效果，对于少量的文本数据效果不佳。

基于规则的实体抽取方法是通过人工定义一些实体或者关系的抽取规则，从文章中抽取出三元组信息，这种方法的重点和难点在于如何定义规则。基于规则的实体抽取方法最大优点是不需要数据训练。此外，基于规则的抽取方法只返回规则定义中的结果，不会出现其他超出规则的结果。然而，规则设计的本身具有的复杂性以及设计的规则可能未能覆盖所有的实体或者关系使得基于规则的方法难以实现。在本课题中，不同的教学体系、不同的书本编排方式不同，固定规则的难以实现实体、关系的抽取。本课题考虑设计一个系统能够根据需求灵活地添加规则，可以实现从不同体系的资料中抽象实体关系并构建知识图谱，从而解决这个问题。

3.3 基于实体识别和模式匹配的知识图谱自动构建

3.3.1 学科知识图谱关系模型

学科知识图谱为每一个学科构建一张知识图谱，在本节中定义学科知识图谱中的实体和关系，具体的实现方式为 Neo4j 图数据库存储实体和实体之间的关系。

实体在 Neo4j 图数据库表现为一个数据点，实体的类别由标签（Label）区别，同时实体通常附带一些属性。

在学科知识图谱中的实体分为两种：

（1）知识点：是对某一个知识的精炼概括，来源于文档中的标题段落或者具有标题功能的图片文字，使用标签 Knowledge 表示实体。

（2）关键词：通常于标题段落下，从描述知识点的文字段落中摘取（关于知识点和描述该知识点的文字段落的划分将在 3.3.2 节讨论），用标签 KeyWord 表示实体。

表 3-1 Knowledge 属性表

属性名	描述	数据类型
knowledge_id	在数据库实例中唯一确定 knowledge 的编号（主键）	Integer
name	知识点名称	String
level	知识点层级	Integer
subject	知识点所属的学科	String

表 3-2 KeyWord 属性表

属性名	描述	数据类型
keyword_id	在数据库实例中唯一确定 keyword 的编号（主键）	Integer
content	关键词内容	String
subject	关键词所属的学科	String

学科知识图谱中实体之间的关系有两种：

（1）对于知识点和知识点之间的关系，通常一个知识点可以从属于（Belong to）一个层级（level）更大的父知识点，知识点之间是树状的关系，用$< A, belong_to, B >$表示知识点 A 从属于知识点 B。

（2）一个知识点 i 需要用一个关键字集合S_i描述，关键字集合S_i与知识点 i 相关（Related to）。如果知识点 i 有子知识点集合，则用其子知识点集合的关键字集的并集表示，即：

$$S_i = \bigcup_j S_j \quad < j, belong\_to, i > \tag{3-1}$$

由此，一个描述某个知识点的关键字集是以它为根节点的子树的叶子节点的关键字集合的并集。知识点和关键词之间的关系可以用三元组$< C, related_to, B >$,其中 C 是关键词，B 是知识点。关键词实体在知识图谱中是**唯一**的，所有的关键词实体在知识图谱中构成一个集合。

知识点之间的关系不包含任何额外属性,使用二元组$< i, j >$表示 knowledge_id 为 i 的知识点从属于 knowledge_id 为 j 的知识点。而知识点与关键词之间的关系使用三元组$< m, n, c >$表示 keyword_id 为 m 的关键词从属于 knowledge_id 为 n 的知识点，其中 c 表示关键词 m 在与之关联的知识点 n 的文本中出现的次数。

学科知识图谱模型可以用图 3-6 表示，假设知识点集合为 U_T，U_T 中的元素的关系按照 level 属性自顶向下构成了一棵树状结构。假设关键字集合为 U_K，学科知识图谱模型忽略了 U_K 元素之间的关系，U_T 和 U_K 的元素之间构成多对多关系，即一个知识点需要用一个关键字集（U_K 的子集）描述，同时一个关键字也可以从属于不同的知识点。在学科知识图谱模型中如果父知识点和它的字知识点的关键词集包含了同一个关键词，

那么只有其子知识点与该关键词存在连接，这样可以避免知识图谱中的连接数过多。

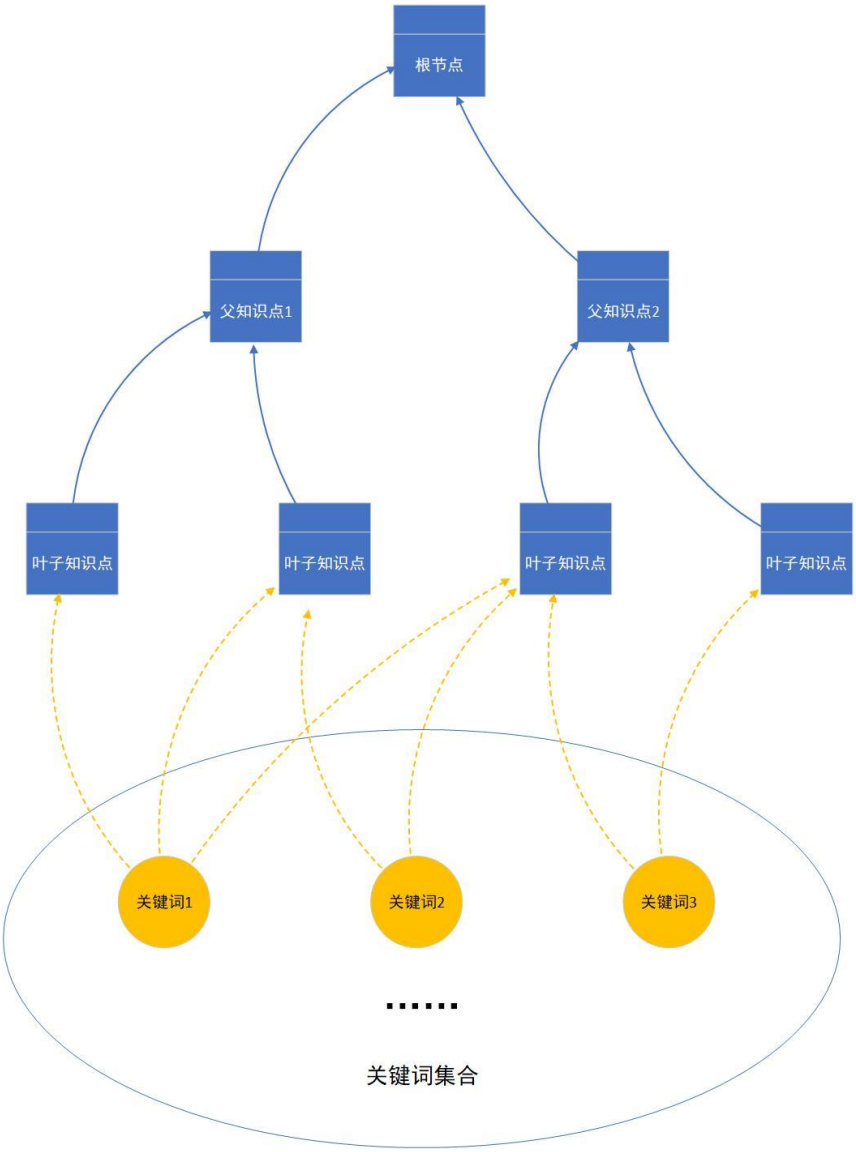

图 3-6 学科知识图谱结构图

3.3.2 文档解析方式

系统的主要数据来源为以 Word 形式存储的学科辅导资料。不同类型的辅导资料会有不同的编排格式，考虑到系统需要兼容不同的教学资料体系，其内部的提取规则应该可以通过配置文件进行更改。Word 文档中的内容除了纯粹的文本之外还有图片和表格，图片和表格中通常蕴含了丰富的文字信息，需要对其特殊处理。本小节主要介绍对于文档的一般解析方式。

首先，使用 python 的 mammoth 模块将 docx 文档转化为 HTML 文档（python-docx 没有提供 API 获取正文中的图片位置，所以没有考虑使用 python-docx 直接解析文档）。

转化后的 word 文档段落、正文、图片、表格分别变成 p 、h、img、table 标签，统一使用 **Tag 标签**表示，用 BeautifulSoup 包解析 HTML 文档可以得到一个包含上述 HTML Tag 标签的列表，根据 Tag 的 name 属性可以区分不同的标签。

针对不同的 HTML 标签采用不同的处理方式获取文字内容形成一个**文字段落（Text Paragraph）**：

（1） 对于 p 或 h 标签，直接获取文字

（2） 对于 img 标签，读取图片中文文字内容，方法是使用 OCR 技术识别图中的文字

（3） 对于 table 标签，将每个单元格中的文字（如果单元格是图片内容，按照第（2）条的方法转化为文字）再用分隔符将每个单元格中的字符相连。

使用上述规则将从 docx 文档得到的 **Tag 标签列表**转为为一个**文字段落列表**。接着基于定义好的规则对 Text Paragraph 解析以生成知识点和关键词。

图 3-7 文档解析流程

3.3.3 从文字段落中抽取知识点和关键词

本小结讨论如何根据HTML Tag列表以及Text Paragraph列表提取知识点和关键词。

（1）知识点匹配

知识点一般在文档中以标题段落或者正文段落（Tag 中的 name 属性为 h 或者 p）出现，一般带有固定格式的前缀，如图 3-9 所示。由此只需要定义匹配的模式和方式，使用正则表达式从 Text Paragraph 中匹配前缀或者查找特定模式便可判断 Text Paragraph 是否是一个知识点。正则表达式的匹配模式和匹配方式需要有用户通过配置文件自行配置。图 3-8 列举了用户定义的匹配规则列表，配置文件为 Json 格式。

系统使用匹配模式定义如下（系统中所有的模式匹配方式都是以下方式之一）：

- 前缀匹配“match_prefix”：从 Text Paragraph 中匹配特定的前缀
- 模式查找“search_pattern”：从 Text Paragraph 中查找特定的模式

知识点匹配规则可以配置的属性在表 3-3 中阐述。


```
{"id": 0,"level": 2,"match_pair": {"match_mode": "match_prefix","pattern": "第\\d+讲\\s"}, "cut_prefix": true},
{"id": 1,"level": 3,"match_pair": {"match_mode": "match_prefix","pattern": "考点.\\s"}, "cut_prefix": true},
{"id": 2,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "知识梳理·夯基础"}, "hidden": true},
{"id": 3,"level": 5,"match_pair": {"match_mode": "match_prefix","pattern": "\\d+(\\.|\\. )(\\s)*"}, "cut_prefix": true,
{"id": 4,"level": 4,"match_pair": {"match_mode": "search_pattern","pattern": "易错易混"}, "hidden": true},
{"id": 5,"level": 4,"match_pair": {"match_mode": "search_pattern","pattern": "深度思考"}, "hidden": true},
{"id": 6,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "归纳总结"}, "hidden": true},
{"id": 7,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "归纳总结"}, "hidden": true},
{"id": 8,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "练后反思"}, "hidden": true},
{"id": 9,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "规律方法"}, "hidden": true},
{"id": 10,"level": 4,"match_pair": {"match_mode": "match_prefix","pattern": "方法技巧"}, "hidden": true}
```


图 3-8 知识点匹配规则配置列表

第 37 讲　弱电解质的电离平衡

【复习目标】 1.了解电解质的概念，了解强电解质和弱电解质的概念。2.理解电解质在水中的电离以及电解质溶液的导电性。3.理解弱电解质在水中的电离平衡，能利用电离平衡常数进行相关计算。

考点一　弱电解质的电离平衡

知识梳理 · 夯基础

1. 弱电解质

(1)概念

电解质 —在水溶液里— 全部电离 → 强电解质，包括强酸、强碱、大多数盐等
电解质 —在水溶液里— 部分电离 → 弱电解质，包括弱酸、弱碱、水等

(2)与化合物类型的关系

强电解质主要是大部分离子化合物及某些共价化合物；弱电解质主要是某些共价化合物。

2. 弱电解质的电离概念

图 3-9 在图 3-8 的规则下匹配的知识点用红色高亮标出，其中的黄色字体为规则匹配到的模式，知识点的描述文本（没有考虑图片，实际上图片中的文字也会被识别）用紫色高亮标出

表 3-3 知识点匹配规则属性表

属性名	描述	数据类型
id	唯一确定知识点匹配规则的编号	Integer
level	匹配到的知识点层级	Integer
match_pair	<match mode，pattern >二元组	Dict
Hidden（可选）	被隐藏的知识点不被加入到知识图谱，父知识点继承关键词集	Bool
cut_prefix（可选）	是否删除匹配前缀	Bool

（2）关键词抽取

关键词是从描述知识点的文本中抽取得到的实体。如果一个 Text Paragraph 匹配为知识点，则其“描述知识点的文本”在此定义为：该 Text Paragraph 段落后、下一个匹配为知识点的 Text Paragraph 前的所有 Text Paragraph（最后一个知识点的描述文本为其后所有的 Text Paragraph）。

对于从 Text Paragraph 中抽取关键词，使用命名实体识别和正则表达式匹配相结合的方式，分别是：使用命名实体识别抽取常见的中文实体，使用自定义的正则表达式识别学科专有的实体（如化学式、公式）。

命名实体识别（Named Entity Recognition，NER）是 NLP 中一项非常基础的任务，用于具有特定意义的实体，主要包括人名、地名、机构名，以及时间、数量、货币、比例数值等文字。在本系统中使用百度 PaddleNLP 的 TaskFlow 中命名实体模块，对于输入的中文文本可以分解并识别出其定义的命名实体[26]。此外，TaskFlow 也支持配置专用的字典。

除了中文名词之外，文本中还存在的一些难以提取的专有名词（例如：化学学科中的化学式）无法被识别，在本系统中采用手动配置正则表达式匹配的方式提取。例如，图 3-13 中配置的两段正则表达是可以用来匹配文本中的化合物化学式和离子化学式。不同的表达式会可能会匹配到重叠的结果，使用最长前缀匹配保留最长的匹配结果。例如：离子化学式“Al3+”经过图 3-13 中的两个表达式分别匹配到了“Al3”和“Al3+”两个结果，取最长的“Al3+”最为最终结果。

```
"keyword_extract_patterns": [
  {
    "match_pair":
    {
      "match_mode": "search_pattern",
      "pattern": "((([A-Za-z]+\\\\d*)+)|(\\\\\\([A-Za-z]+\\\\d*)+\\\\\)\\\\d))+'"
    }
  },
  {
    "match_pair":
    {
      "match_mode": "search_pattern",
      "pattern": "((([A-Za-z]+\\\\d*)+)|(\\\\\\([A-Za-z]+\\\\d*)+\\\\\)\\\\d))+((\\d?(\\+|+))|(\\d?(\\-|−)))\\d?"
    }
  }
]
```


图 3-10 配置的正则表达式匹配化学式，上面的用于匹配化合物，下面的用于匹配离子式

此外，并不是所有的识别得到的命名实体都是必要的，系统支持配置关键词忽略模式 ，匹配方式同样遵循前缀匹配（match prefix）和 模式查找（search pattern）的方式。关键词匹配的效果将在第五章展示。

3.3.5 文档读取控制

并不是文档中所有的内容都需要抽取知识点或者关键词并加入到学科知识图谱中，例如在教辅文档中有相当一部分是练习题及其答案解析，它不是学科知识图谱构建需要关注的内容，由此需要对文档的读取进行控制，忽略不感兴趣的文本段落或者在不需要读取的部分跳过。系统控制方式采用和知识点匹配一样的方式：用正则表达式匹配文本段落，用户需要定义正则表达式模式匹配字典，一旦文本段落匹配到了目标模式，便执行相应的操作，在系统中定义的控制操作有以下四种:

（1）Ignore: 忽略此文本段落

（2）Skip：跳过本段落和接下来的段落，具体做法是将 skip 控制变量（bool 型）变量置为 True。

（3）Stop：将 skip 控制变量置为 False，即在这里停下

（4）End：结束文档读取

以 Skip 为例，列举了相应正则表达式的设置方式，如图 3-11 所示

```
"skip_signs": [
  {
    "id": 12,
    "match_pair": {
      "match_mode": "match_prefix",
      "pattern": "递进题组·练能力"
    }
  },
  {
    "id": 13,
    "match_pair": {
      "match_mode": "match_prefix",
      "pattern": "题组.\\s"
    }
  }
],
```

图 3-11 Skip 操作的正则表达式设置方式

3.4 知识图谱的构建算法

在 3.3.1 小节中阐述了学科知识图谱的结构，使用 Neo4j 构建知识图谱支持预定义好数据模式然后从 csv 文件中导入数据，这需要提供相应的实体表和实体之间的关系表。本系统的数据模型共有 Knowledge 和 KeyWord 两种实体以及 belong_to 和 related_to 两类实体关系，构建学科知识图谱需要以下四张表：

表 3-4 构建学科知识图谱需要的表

表名	描述	属性
knowledge_table	包含所有 knowledge 实体的表	见表 3.1
keyword_table	包含所有 knowledge 实体的表	见表 3.2
knowledge_relations	包含所有 belong_to 关系的表	见 3.3.1 小节的关系定义
keyword_relations	包含所有 related_to 关系的表	见 3.3.1 小节的关系定义

知识图谱构建算法输入 docx 文档（一般是多个）并输出上述四表，得到对应的实体属性表和实体关系表后，使用 Neo4j 的数据导入工具构建知识图谱。

表 3-5 构建学科知识图谱算法

算法 3-1 生成构建知识图谱的关系表

输入：文档集合 D，学科名$subject$

输出：构建学科图谱的四张表

1: 初始化知识点栈 S

2: 初始化关键词集合 U

3: 初始化表：$knowledge_table, knowledge_relations,$

```
    keyword_table, keyword_relations
root = ConstructKnowledge(subject, 0, subject)// 初始化第一个 Knowledge 实体
WriteTable(knowledge_table, root)      // 写入到 knowledge_table 中
S.push(root)   // 压栈
for docx file d in D:
    // 假设 ParseDocx 函数解析文档 d 生成 Text Paragraph 列表
    Text Paragraph List L = ParseDocx(d)
    skip = false      // 设置 skip 控制变量
    for Text Paragraph t in L:
        if MatchPattern("ignore", t):
            Continue
        if MatchPattern("skip", t):
            skip = true
        if MatchPattern("stop", t):
            skip = false
        if MatchPattern("end", t):
            break
        if skip:
            continue
        // 如果匹配到知识点MatchKnowledge返回对应层级，否则返回-1
        level = MatchKnowledge(t)
        parent = S.top()
        if level ≥ 0:
            while parent.level ≥ level:
                S.pop(), parent = S.top()
            child = ConstructKnowledge(t, level, subject)
            r = ConstructRelation("belong_to", parent, child)
            WriteTable(knowledge_table, child)
            WriteTable(knowledge_relations, r)
        else:
            // 从 Text Paragraph 获取关键词集
            keywords, count_dict = ExtractKeyword(d)
            for keyword in keywords:
                if keyword in U:
```

```
                    U.add(keyword)
                    WriteTable(keyword_table, keyword)
                r = ConstructRelation("related_to", parent,
                    keyword, count_dict[keyword])
                WriteTable(keyword_relations, r)
return knowledge_table, knowledge_relations,
    keyword_table, keyword_relations
```

在算法 3-1 中需要使用一个栈保存当前的知识点路径。ConstructKnowledge函数依次接受 knowledge 的 name、level、subject 属性构造一个 knowledge 实体，MatchPattern 函数收控制信号和一个 Text Paragraph 判断是否满足正则表达式匹配类型。ConstructRelation接受关系类别、关系元组属性作为参数构造一个关系对象。WriteTable函数的作用是将一个实体或者关系对象写入到对应的表中

3.5 本章小结

本章从构建知识图谱的动机出发，先阐述了构建知识图谱的难点，然后比较了各种知识图谱构建方法，最后提出了基于模式匹配和实体识别的知识图谱构建方法，其中重点论述了学科知识图谱的数据模型、知识点和关键词的提取方法、文档读取控制流。最后总结并归纳了解析 docx 文档并构建知识图谱的算法。

第 4 章　知识点匹配算法

4.1 知识点匹配算法的目标和作用方式

试题归纳系统的功能是对输入的试题文本找到知识图谱中与之最相关的知识点。首先需要抽取试题文本中的关键词，抽取方式和 3.3.3 小节中的关键词抽取方式一致。对于从试题中抽取的关键词，需要设计一种算法从现有的学科知识图谱中找到与之关联的知识点形成**知识点候选集**并计算候选集中知识点的得分，取得分最高的若干知识点作为最终结果。知识点匹配算法接受一个关键词集合$S_{keyword}$作为输入，输出候选的知识点集合和对应的分数。

4.2 适用于知识图谱的 KF-IKF 匹配算法

传统的 IF-IDF 算法常用于挖掘文章中的关键词，需要计算关键词在文档中词频 TF（Term Frequency）和关键词在语料库文档中的逆词频 IDF（Inverse Document Frequency）。而在学科知识图谱中没有了文档的概念，同一学科所有的文档都通过抽取知识点和关键词构建了一张学科知识图谱。通过关键词匹配可以在学科只是图谱上找到与关键词相关叶子知识点，顺着叶子知识点可以找到层级更低的知识点，这些与关键词有直接关联（与叶子知识点直接连接）或者间接关联（相关叶子知识点的祖先知识点）的知识点构成了**知识点候选集**。知识点匹配算法需要对**知识点候选集**中的知识点进行评估。

受 IF-IDF 算法的启发，提出了适用于知识图谱的 IF-IDF 匹配算法用于评估与关键词集合$S_{keyword}$关联的知识点候选集的分数。该算法的核心思想是使用知识点的概念替换 TF-IDF 算法中的文档概念，使用了新的方式计算关键词关于知识点的词频和逆词频。为了与 TF-IDF 算法区分，分别使用 KF（Keyword Frequency）和 IKF（Inverse Knowledge Frequency）表示。

KF 的计算方式如下：

$$KF(i,j) = \frac{connection\_num(i,j)}{\sum_k connection\_num(k,j)} \tag{4-1}$$

其中 i 为 KeyWord 实体的 id，j 为实体 Knowledge 实体的 id，$connection\_num(i,j)$ 表示知识图谱中 i 与 j 的连接计数 c，在 3.3.1 为 related_to 关系中的属性。k 是所有与 j

有连接关系的关键词。

IKF 的计算方式如下：

$$IKF(i) = log\left(\frac{D}{knowledge\_num(i)}\right) \quad (4\text{-}2)$$

其中，D 为知识点总数，$knowledge_num(i)$为与关键词 i 关联的知识点数目。

知识点 j 的分数计算方式为

$$score_j = \sum_i KF(i,j) \times IKF(i) \quad (4\text{-}3)$$

4.3 分层的知识点评估

知识点的层级（level）体现了知识点覆盖的关键词的范围大小。在 3.3.1 小节的学科知识图谱模型中，知识点之间的层级关系体现为一颗树，在树底端的叶子节点是层级最大（知识点的层级越大，越靠近叶子节点）的知识点，它直接与关键词连接。虽然上层的知识点也会有与关键词的连接，但是由于 3.4 小结中的构建规则，层级越大的知识点描述的知识越细分，直接连接的关键词也越多，而它的父知识点则不与其子知识点的关键词直接连接。父知识点的关键词集合是与其直接连接的关键词集合和其子知识点的关键词集合的并集。

分层的知识点分数计算是在计算KF和IKF时按照层级对知识点划分，分别为每一层级的知识点计算$score$并排序。在计算过程中，需要将子知识点的关键词连接映射到父知识点中。关键词关于知识点的映射方式建立在一张**关键词-知识点映射表**上。在系统中将关键词-知识点映射表用 kw_kn_table 表示，它记录了当前的关键词和知识点之间的连接关系，kw_kn_table 的表头的及其含义在表 4-1 中列举出。对于一个关键词集合 $S_{keyword}$，在 neo4j 中 kw_kn_table 可以通过查询知识图谱构建。

为了分层地计算知识点的评估分数，给定计算层级 l，需要将 kw_kn_table 转换为只含有 kn_level 为 i 的关系表 kw_kn_level_table，具体的转换方式为：找到 kw_kn_table 中所有 kn_level 大于 i 的行，替换 kn_id、kn_name、kn_level 为原知识点第一个 kn_level 小于或者等于 i 的祖先知识。最后把具有相同的 kw_id 和 kn_id 的行合并，relation_count 相加。

表 4-1 kw_kn_table 表头属性

表名	描述	类型
kw_id	KeyWord 实体的 id	Integer
kw_name	KeyWord 实体的名字,对应 content 属性	String
kn_id	Knowledge 实体的 id	Integer
kn_name	Knowledge 实体的名字,对应 name 属性	String
kn_level	Knowledge 实体的层级,对应 level 属性	Integer
relation_count	KeyWord 与 Knowledge 连接数，对应 related_to 关系的 count 属性	Integer

表 4-2 分层知识点评估算法

算法 4-1 分层知识点评估算法

输入：关键词集合$S_{keyword}$，$S_{keyword}$的大小为 m，

输出：各个层级的知识点得分表d

1: 查询知识图谱获得 kw_kn_table

2: 通过 kw_kn_table 中知识点的层级初始化列表 level_list

3: 初始化分层知识点得分表字典d

4: **for** level **in** level_list:

5: 由 kw_kn_table 获取层次为 level 的 kw_kn_level_table

6: 基于 kw_kn_level_table 构造矩阵词频矩阵$\boldsymbol{KF} \in \boldsymbol{R^{m\times n}}$，n 为关联知识点数目

7: 基于 kw_kn_level_table 构造逆词频向量$\boldsymbol{IKF} \in \boldsymbol{R^{m}}$

8: 计算知识点分数向量$\boldsymbol{score = KF^{T} IKF}$，$\boldsymbol{score} \in \boldsymbol{R^{n}}$

9: $d[level] = \boldsymbol{score}$

10: **return** d

4.4 本章小结

本章主要介绍知识点匹配算法，提出了适用于知识图谱的 IF-IDF 匹配算法，并将其重新命名为 KF-IKF 匹配算法。本章着重阐述了 KF 和 IKF 的计算方式以及分层的知识点分数计算算法。

第 5 章　系统设计与实现

5.1 功能需求分析

本系统的核心功能是通过构建知识图谱实现实体知识点的自动归纳，系统的需求可以分为三大类：学科知识图谱构建功能、学科知识图谱管理功能、试题知识点匹配功能。

（1）学科知识图谱构建功能

学科知识图谱构建功能是实现系统可用性的基础功能，为了构建学科知识图谱，需要配置相应的配置规则，从已有的 docx 文档中抽取知识点和关键词实体及其之间的定义的关系。最后将数据导入到 neo4j 数据库中。

（2）学科知识图谱管理功能

为了方便对学科知识图谱进行管理，系统必须提供对于知识点和关键词的查询、删除等接口。考虑到对知识图谱某个知识点或关键词实体或者关系的更改会打乱原有的学科知识图谱结构，应该对学科知识图谱的修改功能做限制甚至不提供单独的修改知识图谱的接口。

（3）试题知识点匹配功能

试题知识点匹配功能是系统最核心的功能也是系统最终的实现目标，试题知识点匹配功能要求用户向系统发送实体文本，系统需要返回试题匹配的知识点列表。系统除了返回的知识点之外，还需要返回知识点的评估分数，

5.2 系统功能模块设计

针对 5.1 的功能需求，系统设计为三个功能模块，本小节对三个功能模块的设计进行详细阐述。

（1）知识图谱构建模块设计

通过配置相关规则，自动化地从文档构建学科知识图谱。规则的配置需要用户配置两种配置文件，分别是全局配置文件和文档配置文件。

由于构建知识图谱的原始文可能由不同编排规则我 docx 文档组成，不同类型的文档可能存放在文件系统不同目录下，全局配置文件为每一类文档配置全局的解析规则，包括每一类文档存放的目录、每一类文档的文档配置文件路径和该类文档的名字识别的

方式（正则表达式匹配文件名）。而文档配置文件用于匹配没一类文档的知识点匹配规则、文档控制模式匹配规则等属性。所有的配置文件以 Json 的格式存储。

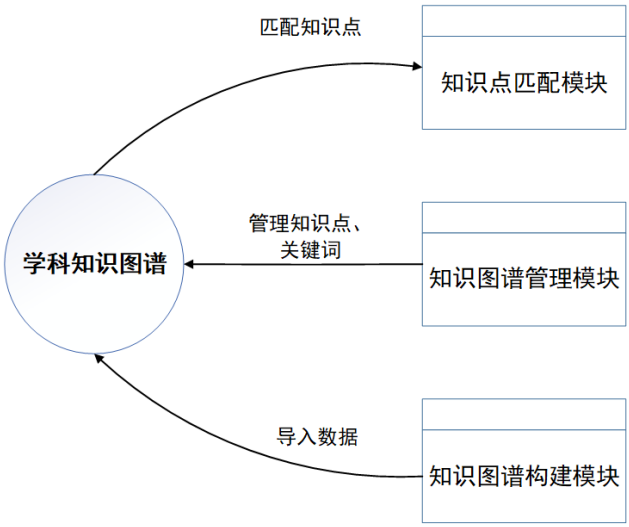

图 5-1 功能模块设计

（2）知识图谱管理模块设计

知识图谱管理模块需要向用户提供对学科知识图谱中的知识点和关键词的管理接口，系统为知识点的管理提供了 4 个功能接口，为关键词的管理提供了 3 个功能接口。

对于知识点系统提供和 3 个与知识点相关的查询接口和一个知识点删除接口。

知识点查询接口包括：通过知识点的 knowledge_id 查找知识点、通过层级（level）属性查找知识点和查找某个知识点所有的祖先知识点。对于知识点的删除接口，首先对于需要删除的知识点必须是叶子知识点，因为如果一个知识点还有子知识点，删除它会导致其子知识点悬挂，然后需要用户给定知识点的 knowledge_id，删除一个叶子知识点会产出该知识点实体及其所有的关系，系统返回删除是否成功。

关键词的管理功能与知识点的管理功能相近，包括：通过关键词的 keyword_id 查找关键词、通过学科名称查找所在学科下的所有关键词、根据关键词的 keyword_id 删除关键词实体及其关联的关系。

知识图谱管理模块的没有提供单独的知识点或者关键词增加模块和修改模块，因为考虑到知识图谱的数据来源于对文档的解析，人为地增加一个实体或者修改实体的属性会破坏知识图谱的正常结构不利于管理，例如人为修改知识点的层次属性会导致知识点层次混乱。

（3）知识点匹配模块设计

知识点匹配模块为用户提供系统最核心的功能——试题知识点归纳功能的接口，需要用户配置的参数有试题文本、学科名、每层推荐知识点的数目。知识点匹配模块的实现原理在第四章已经详细阐述。

5.3 系统实现

5.3.1 学科知识图谱的构建、存储及其管理模块的实现

第三章对学科知识图谱模型的原理及其构造算法进行了详细论述。本小节主要介绍学科知识图谱的构建、存储和管理模块的一些具体实现。

学科知识图谱的构建模块由配置解析器、文档解析器和数据表生成脚本构成。配置解析器首先解析用户配置的系统全局配置文件（全局配置文件存放在系统根目录的 config 目录下）实例化一个 GlobalConfig 对象，然后由全局配置文件中的文档配置文件目录找到并读取文档配置文件并实例化一个 DocConfig 对象，GlobalConfig 对象 DocConfig 包含获取各种配置属性的方法。文档解析器按照算法 3-1 依次读取并解析 DocConfig 配置的目录中的文档，把各个文档的解析结果依次写入到表 3-4 中的数据表中。数据表生成脚本在本地执行，调用配置解析器和文档解析器生成数据表。

学科知识图谱中的实体和关系使用 Neo4j 图数据库存储，首先需要在 Neo4j 图数据库中创建如图 5-2 所示的数据模式以实现图 3-6 中的学科知识图谱模型，关于 Neo4j 中实体及其关系的定义参照 3.3.1 小节中 Knowledge、KeyWord 的属性表以及 belong_to 和 related_to 的关系定义。

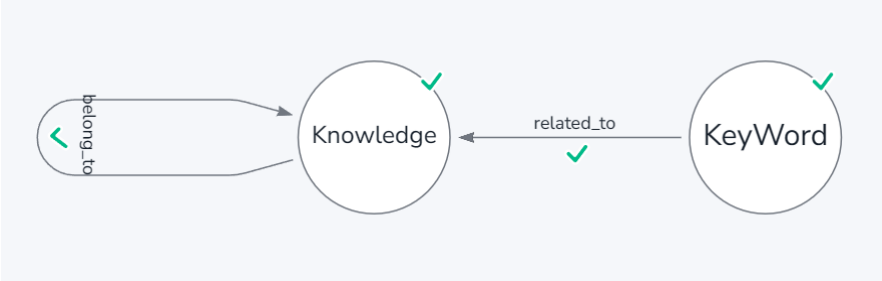

图 5-2 Neo4j 数据模式

学科知识图谱的管理模块的功能使用 Neo4j 的专用查询语言 Cypher 实现，简单的查询（如根据 id、学科名或者其他属性查找知识点或者关键词实体）直接使用 Cypher

的 Match 查询语句即可。管理模块中复杂的功能拆解为若干个简答查询，例如查询一个知识点所有的祖先知识点需要递归地查询一个知识点的父知识点。管理模块的删除操作实现原理是使用 Cypher 中的“detach delete”操作删除匹配的实体及其所有的关系，对于知识点在执行删除操作时规定只能删除叶子知识点，系统在执行删除操作之前会对要删除的知识点进行检查，如果不满足条件，则删除失败。对于关键词实体，删除操作没有限制

5.3.2 系统服务端实现

本系统只提供服务端的实现，服务端接受用户发送的知识点关管理功能和试题知识点匹配功能的请求，返回查询结果或者操作执行结果。系统服务端基于 python Flask 架构实现一个 web 服务器， 所有的请求均使用 Http GET 请求方式，参数设置在 url 中，返回的数据为 Json 格式，如果传递表格则使用 pandas DataFrame 表格转为 Json 字符串，前端再将 Json 格式的数据还原为 pandas DataFrame 数据对象。

5.4 本章小节

本章节着重介绍了系统的功能需求，并以功能需求为基础介绍了系统的三大功能模块，其中知识图谱构建模块和知识点匹配模块的原理分别在第三章和第四章已经进行了详细的阐述。

第 6 章　系统的功能测试

6.1 知识图谱构建模块功能测试

6.1.1 关键词提取测试

关键字抽取不仅是构建知识图谱的一个重要过程，还用于知识点匹配模块从试题中抽取用于匹配的关键词集。系统提供了 extraact_keyword 函数作为关键词抽取的统一的 API 接口，它接受字符串作为输入，使用 PaddlePaddle TaskFlow 模块进行中文实体识别并读取用户配置的匹配规则识别特殊实体，最后输出抽取的关键词。以如下试题为例，抽取试题的关键词：

2．已知：①$Al(OH)_3$ 固体的熔点为 300 ℃，电离方程式为 $H^+ + AlO_2^- + H_2O = Al(OH)_3 = 3OH^- + Al^{3+}$；②无水 $AlCl_3$ 晶体易升华，溶于水的电离方程式为 $AlCl_3 = Al^{3+} + 3Cl^-$；③熔融状态的 $HgCl_2$ 不能导电，稀溶液具有弱的导电能力且可作为手术刀的消毒液。下列关于 $Al(OH)_3$、$AlCl_3$ 和 $HgCl_2$ 的说法正确的是(　　)

A．均为强电解质　　B．均为弱电解质

C．均为离子化合物　　D．均为共价化合物

图 6-1 抽取测试使用的试题

抽取的结果如图 6-2 所示，可见 paddlepaddle TaskFow 模块成功地提取了中文实体，同时在自定义匹配规则的加持下成功地识别了化合物和离子化学式。

```
from knowledge_graph.key_word import extract_keyword
question  = '2．已知：①Al(OH)3固体的熔点为300 ℃，电离方程式为H++AlO-2+H2O==Al(OH)3==3OH-+Al3+；②无水AlCl3晶体易升华，溶于水的电离方程式为AlCl3===Al3++3Cl-；③熔融状态的HgCl2不能导电，稀溶液具有弱的导电能力且可作为手术刀的消毒液。下列关于Al(OH)3、AlCl3和HgCl2的说法正确的是(　　) A．均为强电解质　　B．均为弱电解质 C．均为离子化合物　　D．均为共价化合物'
key_words = extract_keyword(question)
print(key_words)
```

在 274ms 中执行，30 May at 13:24:36

```
{'Al3+', '熔点', '消毒液', '固体', '熔融', '电离方程式', 'B', '导电', '已知', '晶体', '说法正确', '强电解质', 'HgCl2', '升华', 'Cl-', 'Al(OH)3', '弱电解质', '作为', '无水', 'H', '300 ℃', '共价化合物', '离子化合物', 'AlCl3', 'H2O', 'OH-', 'AlO-2', '手术刀', '溶液'}
```

图 6-2 抽取得到的关键词集合

6.1.2 知识图谱构建提取测试

知识图谱的创建需要在服务端本地配置好全局配置文件和相关的文档配置文件再

执行知识图谱创建脚本得到创建知识图谱所需的数据表，本次测试通过执行构建脚本解析化学学科的 102 份文档，得到了构建知识图谱所需 4 张表格

knowledge_table.csv 和 keyword_table.csv 部分内容如图 6-3 所示

knowledge_table.csv 记录了抽取的知识点，共计 669 个

图 6-3 keyword_table.csv 和 title_table.csv 内容

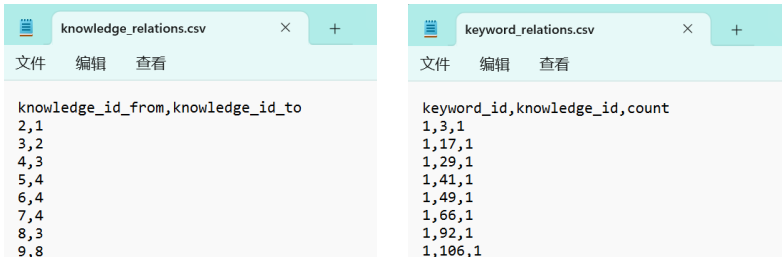

图 6-4 itle_relations.csv 和 keyword_relations.csv 内容

title_relations.csv 和 keyword_relations.csv 分别记录了知识点之间、关键字和知识点之间的关联。其部分内容如图 6-4 所示

本次测试再 Aura 云中创建 neo4j 图数据库实例，在控制面版中创建好如图 6-5 所示数据模式，上传 csv 文件后点击“run import“执行数据导入，最后随机查询学科数据库得到如图 6-6 所示的一部分子图.

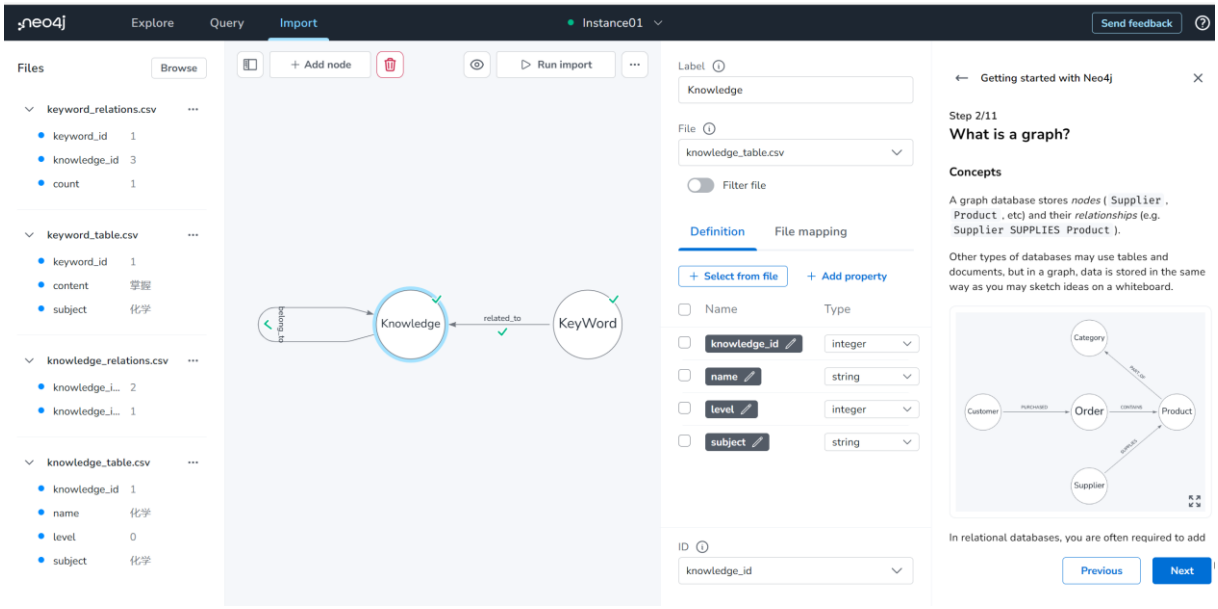

图 6-5 Aura neo4j 创建数据模式

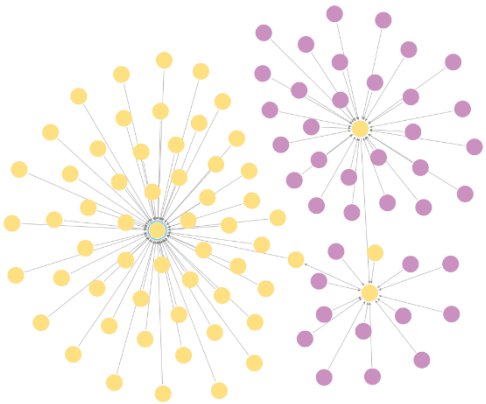

图 6-6 学科数据库的子图，黄色为知识点，紫色为关键字

6.2 服务端功能测试

使用 Flask 框架实现的服务端主要接受用户的 Http 请求实现知识图谱管理模块和知识点匹配模块的功能

服务端测试操作系统平台为 Windows 11。首先使用 pip install -r requirements.txt 安装依赖包，在 powershell 中切换到项目目录下的 server 目录，然后执行以下命令在本地启动服务器：python -m flask --app flask_server run

6.2.1 知识图谱管理模块功能测试

本小节以知识点的查询和删除功能为例，对知识图谱管理模块的部分功能进行测试。首先客户端通过构造特定的 url 向服务端发送请求，请求格式为：

<服务器地址：功能接口标识：参数>

例如，假设服务器运行在本地端口 5000，以查询 knowledge_id 为 2 的知识点信息，构造的 url 格式为：

http://localhost:5000/query_by_id/?type=knowledge&id=2

发送请求，得到查询结构如图 6-7 所示，图中的“raw response”为收到服务器返回的 Json 原始数据，“return_result”为 true 说明服务器成功返回了结果（不论结果如何），result 为查询得到的表单，为 DataFrame 转化为 Json 格式的数据，要正常显示需要再将其转化为 DataFrame 对象，转化后的结果在“query result”下方。

```
raw response:
{"return_result": true, "result": "{\"kn_id\":{\"0\":2},
\"name\":{\"0\":\"\\u4ece\\u5316\\u5b66\\u5230\\u5b9e\\u9a8c\"},
\"level\":{\"0\":1},\"subject\":{\"0\":\"\\u5316\\u5b66\"}}"}
query result:
   kn_id    name  level subject
0      2  从化学到实验      1      化学
```

图 6-7 查询 id 为 5 的知识点结果

对于其他的功能接口表示，表 6-1 列举了其名称和参数。

表 6-1 功能接口参数表

接口名	参数描述
query_by_id	Type:区分查询实体为”knowledge”或”keyword”，id:查询实体的 id
query_by_level	只查询知识点，level:知识点实体的层次，subject:学科名
query_parent	查询实体的所有祖先节点参数与 query_by_id 相同
delete	删除一个实体，参数与 query_by_id 相同
question	匹配试题知识点，question:试题字符串，subject:学科，row_limit:每个层次结果数目限制

测试 query_parent 功能，构造 url：

http://localhost:5000/query_parent/?type=knowledge&id=255

查询结果如下：

```
query result:
                name  level subject
254   氧化还原反应方程式的配平      2      化学
253       化学物质及其变化      1      化学
1               化学      0      化学
```

图 6-8 query_parent 查询结果

测试 query_by_level 功能，构造 url：

http://localhost:5000/query_by_level/?level=2&subject=化学

查询结果如下：

```
query result:
                      name  level subject
3              化学实验基础知识和技能      2      化学
17               物质的分离和提纯      2      化学
29               物质的量及相关概念      2      化学
41  一定物质的量浓度溶液的配制  溶解度的应用      2      化学
```

图 6-9 query_by_level 查询结果

测试 delete 功能，构造 url

http://localhost:5000/delete/?type=knowledge&id=5

执行结果如下：

```
raw response:
{"return_result": true, "result": {"update": true, "nodes_deleted": 1,
"relationships_deleted": 74}}
```

图 6-10 delete 执行结果

为了验证删除的正确性，再次查询 id 为 5 的知识点，结果图 6-12 所示，可见返回了空结果，功能测试成功。

```
raw response:
{"return_result": true, "result": "{\"kn_id\":{},\"name\":{},
\"level\":{},\"subject\":{}}"}
query result:
Empty DataFrame
Columns: [kn_id, name, level, subject]
Index: []
```

图 6-11 delete 执行结果严重

6.2.2 知识点匹配模块功能测试

使用图 6-1 中的试题，“subject”为“化学”，row_limit 限制为 5 向系统接口 question 发送请求，如图 6-12 所示

```
[2023-05-31 12:06:12,314] [    INFO] _internal.py:224 - 127.0.0.1 - - [31/May/2023 12:06:12] "GET /?question=2. 已知：①Al(OH)3固体的熔点为300+℃，电离方程式为H + +AlO - 2 +H2O%3D%3DAl(OH)3%3D%3D3OH - +Al3 + ；②无水AlCl3晶体易升华，溶于水的电离方程式为AlCl3%3D%3D%3DAl3 + +3Cl - ；③熔融状态的HgCl2不能导电，稀溶液具有弱的导电能力且可作为手术刀的消毒液。下列关于Al(OH)3、AlCl3和HgCl2的说法正确的是(    )+A．均为强电解质++%09B．均为弱电解质+C．均为离子化合物++%09D．均为共价化合物&subject=化学&row_limit=5 HTTP/1.1" 200 -
```

图 6-12 服务器接收到 question 请求

最终得到的知识点匹配的完整结果如图 6-13 所示，共显示了层级 1、2、3、5 知识点的推荐结果（层级不是连续的，因为有一部分模式匹配到的知识点不是系统感兴趣的内容，在图 3-8 中设置”hidden”属性为 true，没有将其匹配到的知识点加入到知识图谱中，即隐藏了该知识点）。由于 row_limit 参数设置为 5，每个层级取 score 最大的 5 条。可见层级越大，知识点越细分，它的评估分数越大，该算法倾向于匹配更加准确的知识点。

```
knowledge level:5
knowledge score:
        score              name
280  3.911302        电离方程式的书写原则
267  3.780100  物理变化与化学变化的区别与联系
330  3.760262              弱电解质
416  3.676301             电极的判断
278  3.676301           电解质及其分类
knowledge level:3
knowledge score:
        score          name
548  2.484907       共价键及其参数
311  2.441559     化学键及化合物类型
277  2.146922        电解质及电离
569  2.094856  晶体类型与微粒间作用力
329  2.038751     弱电解质的电离平衡
knowledge level:2
knowledge score:
        score              name
302  1.243750          原子结构  化学键
565  1.195081           晶体结构与性质
532  1.093183           原子结构与性质
328  0.956391         弱电解质的电离平衡
154  0.932626  镁、铝、铜及其化合物  金属冶炼
knowledge level:1
knowledge score:
        score          name
301  1.243750   物质结构、元素周期律
564  1.195081  物质结构与性质(选考)
531  1.093183  物质结构与性质(选考)
327  0.956391     水溶液中的离子平衡
153  0.932626       金属及其化合物
```

图 6-13 知识点匹配结果

6.3 本章小结

本章主要介绍知识图谱的构建结果和系统测试结果。在 6.1 小节对知识图谱的构建功能进行测试，主要测试了关键词抽取功能和然后根据第三章的方法构建数据表，最后导入到 Aura Neo4j 数据库实例中。在 6.2 小节对系统服务端测试，主要测试了知识图谱管理模块的功能和试题知识点匹配模块的功能。

第 7 章　总结与展望

7.1　工作总结

本课题研究并构建一个试题知识点归纳系统，提出了使用知识图谱匹配试题知识点的方法。课题的研究围绕如何构建知识图谱和怎么利用知识图谱匹配关联的知识点这两个问题开展研究。

对于构建知识图谱，首先构建知识图谱的模型，然后从不同的文档中提取知识点和关键词，构建了基于规则匹配的知识点提取机制和基于中文试题识别与规则匹配结合的关键词抽取机制。系统实现了使用配置文件为每一类文档自定义一套匹配规则，抽取知识点和关键词。最后将抽取的知识点和关键词导入到 Neo4j 数据库中以构造学科知识图谱，并实现了一个知识图谱管理模块实现对知识点和关键词的查询等操作。对于如何利用知识图谱做知识点的匹配，使用改进了 TF-IDF 算法的对不同层级的知识点进行评估。

在系统的实现上，基于 5.1 小节的需求分析构建了 5.2 小节中的学科知识图谱构建模块、管理模块和知识点匹配模块，并基于 Flask 框架上构建了一个服务器处理用户对于学科知识图谱的管理请求和试题知识点匹配请求。

7.2　未来展望

随着文档级别的信息抽取技术精度提高，可以考虑使用实体关系抽取模型直接从文档中抽取实体和关系从而更容易地构建知识图谱。对于知识点匹配，考虑使用图嵌入等技术利用知识图谱的信息，结合更强大的 NLP 模型对试题知识点预测将可能是更好的方式。

结束语

在调研完成之后，我按照预想的技术路线进行探索。虽然一开始的设想很美好，但是真正实现起来发现还是困难重重，不得已需要对原来的方案进行修改，为此还需要不停地去探索行的可行的路线，在此过程中我需要逼迫自己学习新的东西。经过不断探索最终形成一套可行的方案。这就是研究的过程，通过本次毕业设计我更加深刻地理解了研究的意义。最后感谢阳旺老师给予我的帮助和指导。

参考文献

[1] University of San Diego Online Degrees. 43 Examples of Artificial Intelligence in Education [EB/OL]. (2023-5-16) [2023-5-16] https://onlinedegrees.sandiego.edu/artificial-intelligence-education/.

[2] UNESCO. Artificial intelligence in education [EB/OL]. (2023-5-16) [2023-5-16] Artificial intelligence in education | UNESCO.

[3] 陈颖博，张文兰．国外教育人工智能的研究热点、趋势和启示[J]．开放教育研究，2019，25(04)：43-58. DOI：10.13966/j.cnki.kfjyyj.2019.04.005.

[4] 中国教育网讯．CNNIC最新报告：我国在线教育用户规模达3.81亿[EB/OL]．(2023-5-16) [2023-5-16]．https://www.edu.cn/info/zyyyy/zxjy/202010/t20201010_2020613.shtml.

[5] 高蜜．基于知识图谱的初等数学特征提取及其应用研究[D]．电子科技大学，2021. DOI：10.27005/d.cnki.gdzku.2021.004584.

[6] 王晓璐．面向K12数学的试题知识点自动标注研究[D]．华东师范大学，2022. DOI：10.27149/d.cnki.ghdsu.2022.002287.

[7] Chowdhary K R, Chowdhary K R. Natural language processing[J]. Fundamentals of artificial intelligence, 2020: 603-649.

[8] 知乎．自然语言理解难在哪儿？[EB/OL]．(2023-5-16) [2023-5-16]．https://zhuanlan.zhihu.com/p/96801863

[9] Wikipedia. Information extraction. [EB/OL]. (2023-5-16) [2023-5-16]. https://en.wikipedia.org/wiki/Information_extraction

[10] Wikipedia. Named-entity recognition. [EB/OL]. (2023-5-16) [2023-5-16]. https://en.wikipedia.org/wiki/Named-entity_recognition

[11] 李恒.基于深度学习的中文指代消解关键技术研究[D].武汉.华中科技大学.2020.

[12] Wikipedia. Graph (discrete mathematics). [EB/OL]. (2023-5-16) [2023-5-16]. https://en.wikipedia.org/wiki/Graph_(discrete_mathematics)

[13] Fensel D, Şimşek U, Angele K, et al. Introduction: what is a knowledge graph?[J]. Knowledge graphs: Methodology, tools and selected use cases, 2020: 1-10.

[14] Professor Wahlster's Keynote: 30 Jahre DFKI—Von der Idee zum Markterfolg, at the event "30 Jahre DFKI—KI für den Menschen", Berlin, Oktober 17, 2018.

[15] Sun Y, Wang S, Li Y, et al. Ernie: Enhanced representation through knowledge integration[J]. arXiv preprint arXiv:1904.09223, 2019.

[16] Zhang Z, Han X, Liu Z, et al. ERNIE: Enhanced language representation with informative entities[J]. arXiv preprint arXiv:1905.07129, 2019.

[17] Sun Y, Wang S, Li Y, et al. Ernie 2.0: A continual pre-training framework for language understanding[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8968-8975

[18] 公众号“图数据库技术”. 图数据库选型. [EB/OL]. (2023-5-16) [2023-5-16]. https://mp.weixin.qq.com/s/_kr-E6t2bnCkguBHtU-1vg

[19] 知乎. NLP中的实体关系抽取方法总结. [EB/OL]. (2023-5-16) [2023-5-16]. https://zhuanlan.zhihu.com/p/77868938

[20] 基于深度学习的联合实体关系抽取. [EB/OL]. (2023-5-16) [2023-5-16]. http://www.czsun.site/publications/thesis.pdf

[21] Miwa M, Bansal M. End-to-end relation extraction using lstms on sequences and tree structures[J]. arXiv preprint arXiv:1601.00770, 2016.

[22] Katiyar A, Cardie C. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees[C]//Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2017: 917-928.

[23] Bekoulis G, Deleu J, Demeester T, et al. Joint entity recognition and relation extraction as a multi-head selection problem[J]. Expert Systems with Applications, 2018, 114: 34-45.

[24] Dai D, Xiao X, Lyu Y, et al. Joint extraction of entities and overlapping relations using position-attentive sequence labeling[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33(01): 6300-6308.

[25] Zheng S, Wang F, Bao H, et al. Joint extraction of entities and relations based on a novel tagging scheme[J]. arXiv preprint arXiv:1706.05075, 2017.

[26] Github. TaskFlow. [EB/OL]. (2023-5-16) [2023-5-16]. https://github.com/PaddlePaddle/PaddleNLP/blob/develop/docs/model_zoo/taskflow.md#%E5%91%BD%E5%90%8D%E5%AE%9E%E4%BD%93%E8%AF%86%E5%88%AB