基于 MOOC 视频字幕和学习数据的自动评测模型和算法研究

（申请清华大学工学硕士学位论文）

培 养 单 位：计 算 机 科 学 与 技 术 系
学　　科：计 算 机 科 学 与 技 术
研　究　生：马　琳
指 导 教 师：马　昱　春　副　教　授

二〇一九年六月

Research on Automatic Evaluation Model and Algorithm based on MOOC Video Subtitles and Learning Data

Thesis Submitted to

Tsinghua University

in partial fulfillment of the requirement

for the degree of

Master of Science

in

Computer Science and Technology

by

Ma Lin

Thesis Supervisor : Associate Professor Ma Yuchun

June, 2019

摘 要

慕课是近几年出现并且发展非常迅速的一种在线学习课程，它将教育资源以视频、博客、文档等形式通过互联网传播给每一个人，使得学习变得更加容易和方便，在高等教育领域影响较大。

MOOCs 学生数量的大规模增加使得其评测反馈系统的自动化变得越来越重要，因此本文围绕 MOOCs 评测系统中出题和组卷两个任务的自动化开展了一系列的研究，其中主要工作和创新点包括：

- 本文创新性地提出了一种基于 MOOCs 视频字幕的自动出题算法。先基于维基知识图谱对视频字幕进行结构化，提取出感兴趣的知识实体及其相互关系，构建课程知识图谱；然后用基于模版的出题算法从课程知识图谱中自动生成测试题目。在公开数据集 SimpleQuestions 上的实验结果表明，我们提出的基于模版的自动出题算法可以有效地改善题目的准确性和可理解性，其中准确率比其他方法提高了 6.99%（评价指标为 BLEU）。
- 本文创新性地提出了一种由 MOOCs 学生平时学习数据所驱动，基于遗传算法的自动组卷模型。在该模型中，我们先根据学生的平时学习数据预测其学习效果，进而决定试卷的整体难度要求；然后用遗传算法从题库中找到合适的题目组合，使得其难度、知识点分布、题型分布等都基本满足目标约束。在 MOOCs 课程上的实验结果表明，预测模型的平均绝对误差在百分制的情况下为 10 分左右，并且算法可以在多个目标约束下成功完成组卷任务。

综上，本文设计并实现了一套基于 MOOCs 视频字幕和学生学习数据的自动评测模型，为 MOOCs 评测反馈系统的自动化、自适应化作出了贡献。

关键词：慕课；知识图谱；自动出题；自动组卷；遗传算法；成绩预测

Abstract

MOOC is a kind of online course that has emerged and developed very rapidly in recent years. By sharing educational resources to everyone in the form of video, blog and document on the Internet, it makes learning easier and more convenient. Hence, MOOC has made a great impact on higher education.

The large-scale increase of MOOC learners makes the automation of evaluation system more and more important. Therefore, we carry out a series of researches on how to automatically generate questions and automatically compose test papers in this paper. Our work and innovations mainly include:

- We innovatively propose an algorithm of automatically generating questions from MOOCs video subtitles. Firstly, we use wiki knowledge base to extract facts of interest and visualize them in the form of knowledge graph. Secondly, we propose a novel template-based method to generate questions from knowledge graph. The experimental results based on the public dataset SimpleQuestions show that our proposed template-based method outperforms other competitive methods by 6.99% in terms of BLEU.
- We innovatively propose an algorithm of automatically composing test papers with the support of the learners' usual learning data. We firstly determine the overall difficulty of test paper according to learners' usual learning data, and then use genetic algorithm to generate test paper considering multiple constraints. The experimental results based on a MOOC course show that the mean absolute error of prediction model is roughly around 10 points on 100 points scale and we can successfully achieve the intelligent composition of test papers with various objectives optimized.

In summary, we propose and implement an automatic evaluation model based on MOOCs video subtitles and learners' learning data, which contributes to the automation of MOOCs' evaluation system.

Key words: MOOCs; Massive Open Online Courses; Knowledge Graph; Automatic Question Generation; Automatic Composition of Test Paper; Genetic Algorithm; Performance Prediction

目 录

主要符号对照表

MOOCs	大规模开放在线课程
ETS	美国教育考试服务中心
CPR	校准学生互评系统
NLP	自然语言处理
LSTM	长短期记忆网络
RNN	循环神经网络
GRU	门控循环单元
WIKIDATA	维基百科知识图谱
WQS	WIKIDATA 数据库查询服务
RDF	资源描述框架
SPARQL	RDF 查询语言和数据获取协议
CRF	条件随机场
TF-IDF	词频-逆文档频率
KBQA	基于知识图谱的问答任务
BLEU	双语评估替换
MAE	平均绝对误差

第 1 章　绪论

1.1　研究背景

MOOCs 是一种近几年出现且发展非常迅速的在线学习课程。它的全称是大规模开放在线学习课程（Massive Open Online Courses），旨在通过视频、博客、文档等互联网媒介将课程内容免费传播给每一个人，从而打破以往优秀教育资源仅掌握在少数人手里的局面，实现高校教育的大众化、多元化。它的出现是互联网发展在教育行业中引发的一次课程改革。自 2012 年美国三大优秀的 MOOCs 平台 edX[①]，Udacity[②]，Coursera[③]相继发布之后，一股 MOOCs 研究的热潮就开始席卷全球各地。在我国，除包括清华大学、北京大学、复旦大学、上海交通大学等在内的一大批知名高校纷纷加盟之外，学堂在线[④]、华文慕课[⑤]等众多优秀的 MOOCs 平台也都发展的风生水起。因此针对 MOOCs 在实践过程中遇到的一些问题开展研究是非常必要且有意义的。

传统课程和 MOOCs 课程最本质的区别在于学生数量上的巨大差异，前者选课人数一般在几百人左右，最多不过 1000 人，而后者的选课人数却一般都在几万、几十万左右。如此大的数量差异使得传统课程的评测模式在 MOOCs 中几乎失效。在教育行业，及时有效地对学生的学习成果进行评测，并将结果反馈给学生是非常重要的一个过程。从教师角度来讲，它可以帮助老师及时调整教学方案，并且有针对性地为部分需要的学生提供额外辅导，从而提高教学成果；从学生角度来讲，它可以对学习用功努力的学生给予肯定，也可以对那些学习开始懈怠的学生敲响警钟，最重要的是，它可以帮助学生找到学习过程中的盲点，从而在之后的学习过程中可以有针对性地进行学习，提高效率和成果。因此对于 MOOCs 而言，自动化的评测反馈系统是其必不可少的一个组成部分。

传统的 MOOCs 自动评测反馈流程如图1.1所示，主要包括三个阶段。首先是教师准备阶段。教师作为课程的管理者需要在开课前一定时间内，预先准备好本讲的课程视频和测试题库，并从题库中挑选出一些合适的题目作为课后测试题，用以检验学生是否完成视频内容的学习。然后是学生学习阶段。开课后学生通过课程视频进行学习，并且需要在规定时间内完成对应的测试题，与此同时，其所有

① https://www.edx.org
② https://cn.udacity.com
③ https://www.coursera.org
④ www.xuetangx.com
⑤ http://www.chinesemooc.org

的学习轨迹都会在平台上产生相应的数据记录，包括什么时候登入过、什么时候退出、什么时候观看过课程视频、观看时间、什么时候完成了测试题以及测试题得分情况等等。我们将学生学习过程中产生的一系列数据统称为学习数据。最后是评测反馈阶段。截止日期过后，教师需要对学生当前的学习成果进行评分，其中除主观题之外，其余工作都由系统自动完成，包括客观题批改、学生学习数据统计汇总等等。根据评分结果，教师需要及时调整教学方案，对部分学生重点关注提醒，以帮助其更好地完成教学目标。

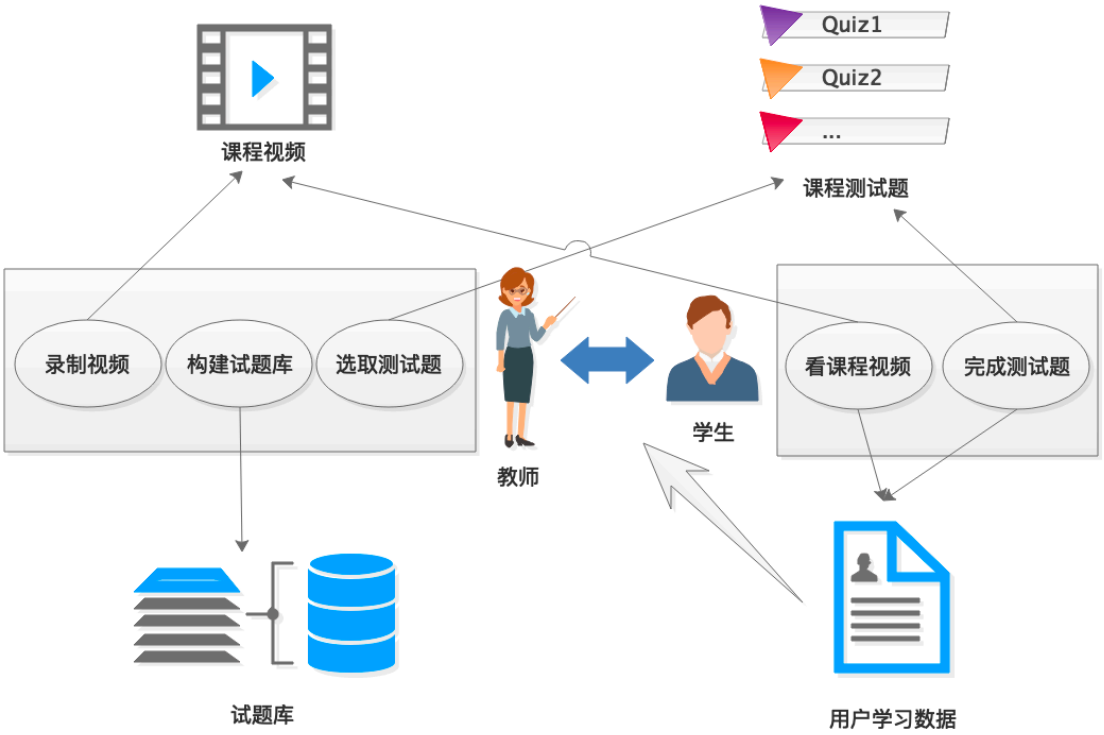


图 1.1　传统的 MOOCs 自动评测反馈流程图

通过分析整个过程，我们可以发现当前 MOOCs 的自动评测反馈系统中主要存在以下三个问题：

1. 教师需自己手动完成出题和组卷任务，工作量较大。
2. 组卷任务中，题目的难易程度完全依赖于出题人的经验和知识水平，且知识点、题型的覆盖程度没有定量的衡量标准。
3. 试卷参数无法针对学习者的学习情况进行自适应调整，以实现个性化学习。

因此，本文提出了一套完整的基于 MOOCs 视频字幕和学生学习数据的自动评测模型，如图1.2所示。在该模型中，我们可以从 MOOCs 课程内容中自动生成测试题，并根据学生日常的学习数据自适应地决定试卷的难易程度，然后自动从

题库中找到满足题型、知识点及难度约束的题目组合用作课后测试题，从而实现个性化学习。MOOCs 的出现，使得教学内容的载体变得越来越多样化，除教材之外，常见的还有视频、博客、学习笔记等，但因为在 MOOCs 中视频是最主要的教学内容载体，因此本文的自动出题算法仅考虑如何从视频字幕中生成题目。

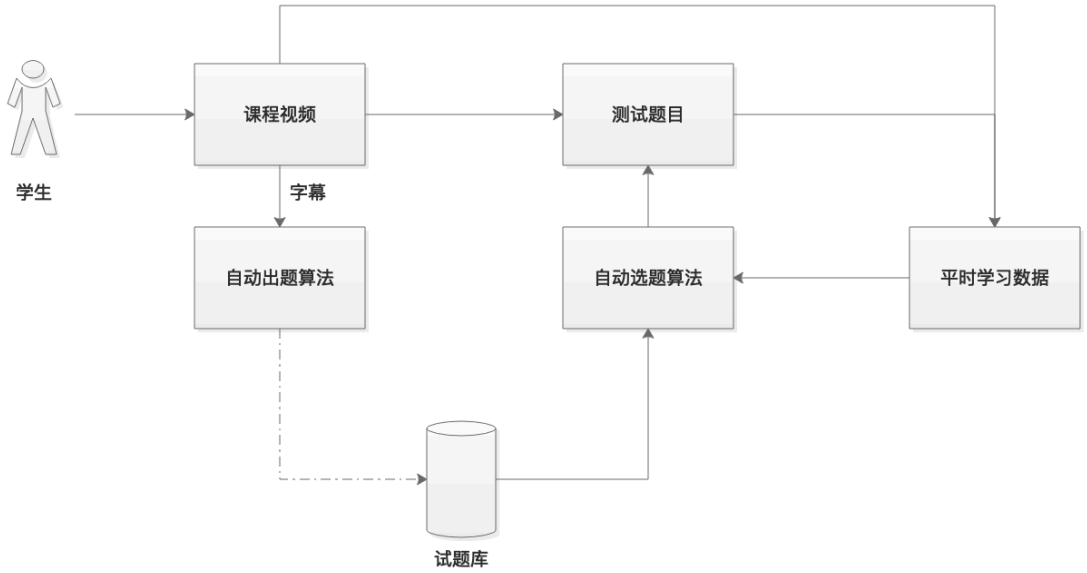


图 1.2　基于 MOOCs 视频字幕和学生学习数据的自动评测模型

1.2　主要工作

本文主要围绕 MOOCs 评测反馈系统中，出题和组卷两个核心任务的自动化开展了一系列的研究，其中主要工作和贡献点包括：

- 提出并实现了一套完整的基于 MOOCs 视频字幕和学生学习数据的自动评测模型。模型从 MOOCs 视频字幕出发，自动生成测试题目，并根据学生的平时学习效果自适应地决定选题目标（主要是决定选什么难度的题），然后用组卷算法从题库中抽取出符合目标约束的题目组合作为课后测试题。相较于传统的 MOOCs 自动评测流程，我们提出的模型更加自动、智能且符合个性化需求。
- 在基于 MOOCs 视频字幕的自动出题任务中，我们设计了一套基于 WIKIDATA 知识图谱的自动出题流程。在该任务的实现过程中，我们主要有两个贡献点：
  - 将维基知识图谱（WIKIDATA）引入到了 MOOCs 的学习过程中，从而帮助实现课程内容的结构化，为其他一些应用提供了可能，例如基于维基知识图谱的课程内容纠错、课程助手问答系统等。

  - 提出了一套基于模版的出题算法，用以从课程知识图谱中自动生成题目。在公开数据集 SimpleQuestions 上的实验结果表明，该算法可以有效地改善生成问题的准确性和可理解性，其中准确率比基于序列神经网络的算法提高了 6.99%(BLEU)。
- 在自动组卷任务中，我们设计了一套由学生平时数据驱动、基于遗传算法的自动组卷流程，同样，在该任务的实现过程中，我们也主要有两个贡献点：
  - 提出了一种基于群体划分的学习效果预测模型，相较于直接从平时数据预测学习效果的预测模型，群体划分结合数据过滤的预测方式可以有效降低预测误差。实验结果表明在百分制的情况下，它可以将平均绝对预测误差降低至 10 分左右。
  - 设计并实现了一套基于遗传算法的自动组卷模型，实验结果表明，它可以有效完成组卷任务，提高组卷的成功率。

1.3　论文组织

本文一共包含五章内容，大致组织结构如下所示：

- **第1章：** 绪论。本章主要从 MOOCs 研究的意义、自动评测反馈系统对于 MOOCs 的重要性、以及当前 MOOCs 自动评测反馈系统中存在的问题三个方面进行了简单的介绍，从而说明本文的研究背景和研究意义。在此基础上，进一步阐述了本文工作的主要贡献点以及文章的组织结构。
- **第2章：** 相关工作。包括 MOOCs 相关工作介绍；自动出题、自动组卷相关算法介绍以及 WIKIDATA 相关概念、工具介绍。
- **第3章：** 基于 MOOCs 视频字幕的自动出题模型。主要包括方法介绍和实验结果。
- **第4章：** 基于 MOOCs 学习数据的自动组卷模型。主要包括方法介绍和实验结果。
- **第5章：** 总结与展望。本章对论文的所有工作进行了总结，并指出了当前模型和方法中存在的一些问题以及未来可能的改进方向。

第 2 章　相关工作

2.1　MOOCs 介绍

MOOCs 是近几年兴起且发展极其迅速的一种课程形式，尤其在高等教育领域更为普及。它打破了以往教育资源仅被少部分人占有的局面，将教育资源以视频、网站、博客等形式通过网络传播给每一个人。与传统的老师和学生面对面的教学模式相比，MOOCs 这种教学方式最显著的特点有：

- **学生数量的大规模增加**。论文[1] 中指出，传统课堂，即便是大学那种大规模的演讲大厅，人数也最多不过 1000 来人；但对于 MOOC 来说，学生人数动辄就在几万、几十万甚至更高的数量级上。这种人数上的剧增可以说是 MOOCs 和传统课程各方面都不一样的本质原因。
- **教育理念的变化**。传统教育旨在通过一系列教学活动使学生完成一定的教学目标。与传统的教育理念不一样的是，MOOCs 仅以传播知识为主。这种定位使得 MOOCs 的学生群体非常多样化，大部分人仅以选择性了解为目的。

MOOCs 的出现是网络科技发展的产物，它引导人们开始思考在互联网极速发展的 21 世纪，与之相适应的教育模式到底应该是什么样子的。因此自 2012 年它开始活跃于高等教育领域之后，越来越多的学者致力于与它相关的研究工作中。本节接下来会对和 MOOCs 相关的研究工作做个简单的介绍和分析。

2.1.1　MOOCs 相关研究概述

当前，和 MOOCs 相关的研究大致可以被分为三大类。第一类是与课程模式设计有关的理论分析研究。这类研究大多是由教育学家、课程指导老师等发起的。第二类是与学生特征、学习数据有关的预测分析研究。例如，MOOCs 当前非常常见的一个现象是：开课前会有非常多的人注册课程，随着时间的推移，越来越多的人会渐渐不参与任何课程活动，仅有一少部分人能坚持到最后，完成所有内容并取得证书的人则更少。因此一些研究是希望通过分析学生特征和学习模式来研究影响学生退课的因素，从而帮助解决当前困扰 MOOCs 很大的高退课率问题。除退课、通过预测之外，这类研究还有一个很常见的工作是课程推荐，即根据学生特征为其推荐合适的课程，从而实现个性化课程定制。第三大类则是和 MOOCs 相关的一些工具开发，用以辅助整个 MOOCs 学习过程。下面是分类列举的部分研究工作：

- **与 MOOCs 课程模式相关的理论研究**。论文[2] 分析了传统课堂、混合式课

堂、翻转课堂和在线课程几种课程设置下学生的学习效果。混合式课堂是一种将传统的面对面教学和线上教学相结合的课程模式，翻转课程则是混合式课程的一种特定形式。在翻转课程设置里，课程由线下课堂和线上教学两部分构成，教学内容（视频以及测试题）一般会提前上传到线上。学生在参加实体课程之前，需要先在线上学习课程内容，这阶段的时间安排可以由学生自己决定。线下实体课程则会组织一些讨论，或由专业教学人员就学习、作业中遇到的问题提供答疑。论文中作者将 90 个研二的学生随机分配到四种课堂中（传统课堂、翻转课堂、混合式课堂、线上课堂），发现翻转课堂的学生学习效果更好，同时也发现翻转课堂对学生的自我激励和自我效能信念会产生积极的影响。论文[3] 中指出 MOOCs 可以分为 cMOOCs，xMOOCs, quasi-MOOCs 三种模式，而论文[4] 则分别分析了 cMOOCs 和 xMOOCs 两种课程模式的特点和不足。cMOOCs 是 MOOCs 的最初形态，又叫基于联通主义学习理论的 MOOCs，它强调创造分布式的社交网络学习模式。在这种模式下，老师在共享课程视频、学习材料之后，学生基于社交网络构建学习社区，互帮互助，完成课程内容。这种模式较为理想化，对学生的自制力、积极性都有较高的要求。相比之下，xMOOCs 则更接近传统教学，它也叫基于行为主义学习理论的 MOOCs，翻转课堂、混合式课堂都属于此类 MOOCs。由前面的介绍我们可以发现，这种课堂虽然可以更好的提高学生学习效果，但它并不完全贴近 MOOCs 的设计初衷，即无法实现大规模学习。从某种意义上来来说，它是高校传统教育和在线教育相融合的一个产物。

- **与 MOOC 学生特征、学习数据有关的预测分析**。论文[5] 通过分析 MOOCs 中学生特征与退课率之间的关系，指出 MOOCs 学生退课的原因主要有课程不符合预期、学习体验差、中途才加入课程等因素。论文[6] 同样研究的也是 MOOCs 较高的退课率问题，作者提出了一个隐马尔可夫预测模型，可以根据学生当前学习活动来预测其未来是否会退课。论文[7] 则是通过在开课前发放信息收集问卷，从而探索影响学生选择退课或继续完成课程的因素，例如学生职业和课程内容的相关性、之前的 MOOCs 学习经历等。论文[8] 基于用户行为发掘其潜在兴趣，然后联合人口学特征以及课程之间的先修关系在学堂在线平台上搭建了一个课程推荐系统。
- **与 MOOCs 相关的工具开发**。当前主流的 MOOCs 平台有很多，常见的有学堂在线、edX、Udacity 等，不同平台的数据格式一般是不一样的。论文[9] 中作者开发了一个用于整合不同平台的数据的工具，MoocViz，从而大大节省了不同平台的研究人员花在数据格式转换上的时间。美国加利福尼大学则研发

了一个用于学生互评的系统——校准学生评测系统（Calibrated Peer Reviews），简称 CPR^{TM} [10]。该系统可以通过样例作业对学生赋予一定的权重，从而提高学生互评结果的准确率。

2.1.2　MOOCs 评测方法概述

传统的教学活动主要包括三个重要组成部分：老师教学、学生学习、评测反馈。其中评测反馈对于检验学生当前学习成果、指导其之后的学习过程以及老师之后的教学方案设计是非常关键的。但因为 MOOCs 学生数量太大，教师很难像传统课程一样，对学生的学习成果进行及时有效地评测反馈，因此很多研究开始专注于如何在 MOOCs 中构建有效的评测反馈机制。不幸的是，现有的 MOOCs 评测反馈方法都有其各自的局限性，目前还没有哪一种 MOOCs 评测方法是完全有效且没有弊端的。当前大多数 MOOCs 所采用的评测反馈方法包括但不限于以下几种[1]：

1. **自动评测**。一些 MOOCs 课程会在每个课程视频后面提供一些对应的测试题，用以检测学生是否看过课程视频并且掌握了其中的内容。学生的答案由机器自动进行打分，并将打分结果反馈给学生和老师。这些测试题一般以选择和填空等客观题为主。这种评测方法的不足主要在于以下三点：
    - 一是适合课程类型有限，不适用那些课程目标无法通过试卷量化的课程。譬如一些课程，它的课程目标可能是希望学生课后能设计一件作品、完成一份报告或者解决一个实际问题。
    - 二是适用题型有限，目前该方法适用的题型仅包括填空、选择、判断等客观题，而主观题的自动打分较为麻烦，一般不涉及。
    - 三是为每个视频增加对应的测试题会增加老师的工作量，如果考虑到试题质量的话则更具挑战性。
2. **利用课程论坛进行反馈**。在一些课程中，老师会选择在课程论坛内发布帖子，对常见的一些问题进行讲解说明。这种方法最明显的弊端是无法考虑到每个人的需求。
3. **混合式教学**。前文提到混合式教学是一种线上线下结合的课程模式，学生先在线上看视频学习，然后在线下实体课再针对学习中遇到的问题找老师答疑。这种模式虽然可以给学生提供反馈，但线下课程容量和传统课程差不多，违背了 MOOCs 最初的设计初衷，无法做到大规模推广。同时线下实体课一般是收费的，这也加大了其推广的难度。
4. **自动文章打分系统**。对于一些写作课程，可以用已有的文章打分系统去对学

生写的文章进行打分，例如英文写作课可以用美国教育考试服务中心 (ETS) 的 e-rater[①]去进行打分。这种方法的局限性一是在于仅适合部分写作类课程；二是在于打分系统只能从语法角度对文章进行考查，无法更进一步考查学生的抽象总结、表达能力。edX 已经宣布将自动文章打分系统应用到平台的一些课程中[10]。

5. **学生互评**。学生互评（Peer Review）包含学生作业互评和学生论坛互评两部分。它是当前 MOOCs 中应用最广、且被认为最符合 MOOCs 教学模式的评测方法[1]。学生互评最基础的模型中，作业互评是指每个学生的作业都会被分给几个不同的学生，然后由他们基于老师给定的评分标准来对该作业进行评分，最后作业的分数取所有人打分的平均值或中值；论坛互评则是学生将问题发到讨论区，然后由其他学生进行回答，从而获得反馈。可以发现这种模式的优点是：作业类型不像自动打分系统一样，局限在选择、填空等客观题，学生可以上传各种类型的作品；同时其适用的课程类型也没有限制，不像文章打分系统仅适用于部分写作课程。但尽管如此，基础的学生互评模型还是存在一些问题，其中最大的弊端就是评测结果的可信度难以保证，这主要是因为打分的人和写作业的人一样，都是才刚开始学习课程内容，很容易因为知识不够而误判。除此之外，学生互评的另一个缺点是会增加学生的工作量，影响学生的学习体验。论文[5] 中就指出一些学生就是因为互评环节的引入最后选择退课。因此，很多 MOOCs 相关的研究都是讨论如何解决学生互评的可信度问题，比较成功且已经实际投入使用的是美国加利福尼亚大学研发的 CPR^{TM} 系统。在 CPR^{TM} 系统中，学生首先需要对特定的几份测试作业进行打分，测试作业老师也会给出一个分数，每个学生会根据他对测试作业的打分和对应的老师打分之间的接近程度被赋予一定的权重，和老师打分越接近，学生的权重也就越大，最终作业的学生互评分数由各位评分人给的分数和评分人的权重共同决定。

2.1.3　小结

本节先是对 MOOCs 做了一个简单的介绍，然后对 MOOCs 相关的研究进行了一个分类讨论，可以发现现有的研究以预测分析、理论研究为主，相对缺乏与课程自动化评测相关的方法和系统研究。接着通过对各种 MOOCs 常用的评测方法进行讨论分析，可以发现当前每种评测方法都有其一定的局限性和弊端，因此如何在 MOOCs 中构建有效的评价反馈机制仍然是一个极具挑战的问题。MOOCs

① http://www.ets.org/erater

数以万计的学生群体使得其评测反馈系统的自动化是必需的，因此本文主要围绕 MOOCs 评测反馈系统的自动化开展了一系列的研究。

2.2　自动出题相关方法介绍

自动出题（Automatic Question Generation）是 NLP 领域一个较为重要且很有挑战的问题，旨在根据输入自动生成一些语法正确、语义正确且和输入相关的问题。自动出题算法主要的应用场景有：

1. 在在线教育行业（MOOCs）中，可以用自动出题算法从教学内容（例如教材书、博客等）中自动生成测试题，以检测学生是否阅读并理解了该教学内容。
2. 在各种问答、对话系统中，可以用自动出题算法去生成一些问题，从而继续整个聊天过程。

自动出题的输入来源多种多样，常见的可以分为两大类：一类是文本输入；一类是结构化数据库或知识库输入。输入形式不一样，常用的出题算法也就不一样，本节接下来会分别介绍这两种输入形式对应的常见自动出题算法。

2.2.1　基于文本的自动出题算法

基于文本的自动出题任务是指从给定文本中生成内容相关且无语法、语义问题的题目，从而去测试读者是否阅读了该内容，并且理解和掌握了其中的信息[11]。当前基于文本的自动出题算法研究方向主要集中在从教学文本（educational text）、或者是信息文本（informational text）中生成事实类问题（factual question）。这类文本的特点是主要以陈述事实信息为主，而不是表达个人意见或进行逻辑推理[11]。图2.1是从 SQuADv2.0 的验证集中抽取的一个例子。SQuAD 数据集[12] 全称 Stanford Question Answering Dataset，是斯坦福大学发布的机器阅读理解问答数据集，共包含从 500 多篇维基百科文章中标注的 10w+ 个问题-答案对，其中答案可以直接从文章中找到，是文章的一个片段（segment）。

基于文本的自动出题算法可以给定答案，也可以不给定答案，前者是后者的一个子问题，不给定答案的情况下算法需要找到所有可能成为答案的候选短语并依次生成相对应的问题。当前从文本中自动出题的方法主要有两大类，一类是基于预定义好的语法转换规则去生成问题，另一类则是基于序列神经网络去生成问题。下面对这两种方法分别进行介绍。

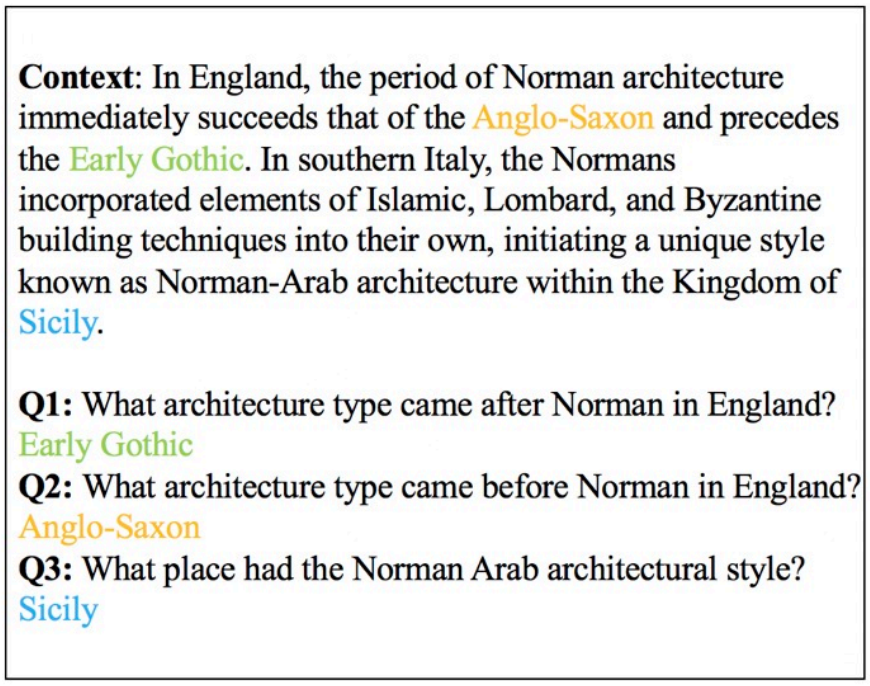


图 2.1　SQuAD 数据集样例[12]

2.2.1.1　基于规则的文本自动出题算法

本节以论文[11][13][14]提出的方法为例来对基于规则的文本自动出题算法进行说明，其他论文中的方法大致思想基本与其一致。算法输入为一段文字，输出为一个有序的 WH 问题列表，且该论文提出的算法不需要输入答案。这里 WH 问题指的是疑问代词前缀为“WH”的问题，例如“What's your name?”就是一个 WH 问题。

图2.2为算法的整体流程图，可以发现算法共包含三步：源文本简化、候选问题生成、问题打分。

- **源文本简化**。这一步是希望在保留源文本语义的情况下，能够借助 NLP 工具（例如词性标注工具、语法分析工具等）将嵌套的复杂源文本进行简化压缩（Sentence Simplification），从而使得之后生成的问题更容易被接受。不被接受的问题指的是有语法错误无法被人理解的问题。常见的简化规则有：删除同位语、删除括号内的内容、删除句子级别的附属修饰短语等等。文本简化压缩之后，我们还需要将文本中出现的代词用其真正指代的实体进行替换。这是因为问题一般是独立的一句话，没有上下文，如果问题中含有代词的话读者一般无法理解，例如“When was he eleted as the president of USA?”问句中，he 在没有上下文的情况下不清楚到底指代的是谁。代词替换需要用到 NLP 领域里常用的共指（coreference）识别技术（例如斯坦福的自然语言处理包

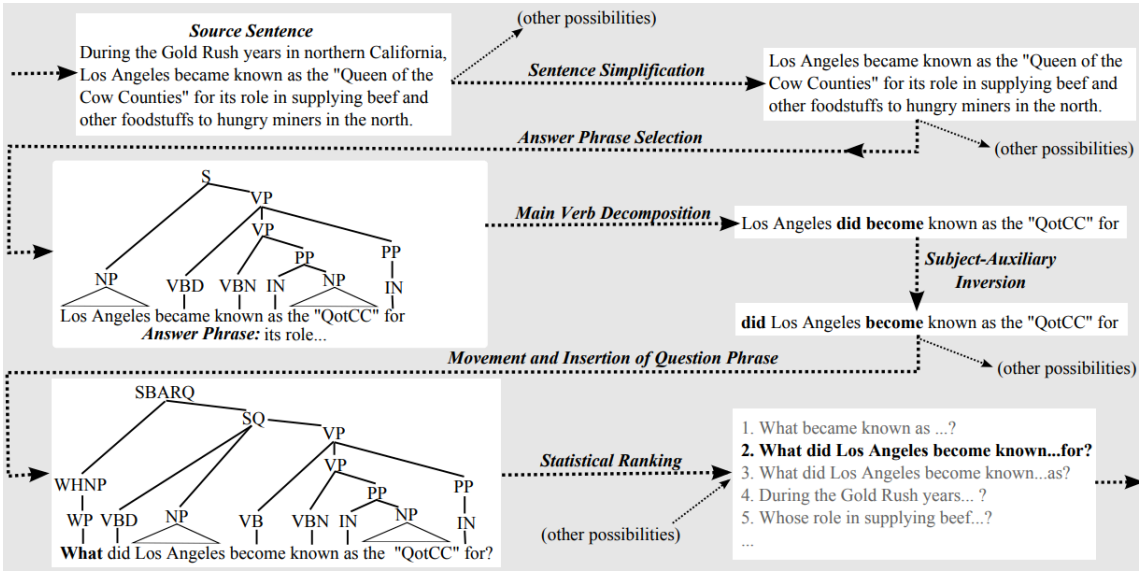


图 2.2　基于规则的自动出题算法流程[14]

中就含有相关工具），两个短语是共指关系，指的是它们在真实世界中指向同一个实体。

- **候选问题生成**。首先第一步，我们需要从完成过简化和代词替换的源文本中找到所有可能的答案短语（Answer Phrase Selection），当然如果是给定答案的出题算法则不需要这一步。因为答案短语在生成问题的过程中需要被替换成对应的 WH 疑问词或疑问短语，然后移动到句首（这一过程称为 WH 移动），所以这里需要设置一些语法规则去定义出那些不可进行 WH 移动的短语，从而避免其被选中为候选答案，例如句子“Bob studied how bird fly.”中 bird 就是一个不可进行 WH 移动的单词，对它进行 WH 移动得到的问句为“What did Bob study how fly?”很明显，这不是一个正确的句子。第二步，我们需要继续找到句子的主要动词并将其进行分解（Main Verb Decomposition）。主要动词指的是主句的动词，而不是分句的动词。分解指的是将那些不是系动词和情态动词的动词分解成对应的 do + 动词原形的形式，例如“studied”分解之后就变为“did study”。第三步，助动词倒装（Subject-Auxiliary Inversion），也就是颠倒助动词和主语的顺序。第四步，用合适的疑问代词或疑问短语替换答案短语并进行 WH 移动，将其移动到句首处（Movement and Insertion of Question Phrase）。答案到疑问短语的替换规则示例：类型标记为人（PERSON）或组织（ORGANIZATION）的名词短语对应的疑问代词为 who；类型标记为地理位置（LOCATION）的名词短语或含 in，at，on，over 这些介词的介词短语对应的疑问代词为 where。
- **问题打分**。得到所有候选问题之后，我们需要对问题的质量进行打分，并依

次输出分数较高的前几个问题作为算法的最终输出。问题打分器是从训练数据中训练得到的一个逻辑回归分类器，问题被分为可接受和不可接受两种，回归器输出问题可接受的概率，介于 0 ~ 1 之间，值越大，说明问题越容易被接受。

可以发现，整个流程较为繁琐复杂，且其支持的问题类型有限、扩展性低。因此出现了另一类自动出题算法，基于神经网络的自动出题算法。

2.2.1.2　基于神经网络的文本自动出题算法

这种方法的出现得益于神经网络在解决机器翻译问题上取得了极大的突破。可以发现从文本中自动出题任务和机器翻译任务从本质上来说是一样的，都是从一个文本序列生成另一个文本序列，且期望生成的文本序列和目标文本尽可能地相似。

问题定义：给定输入文本 X 是长度为 M 的单词序列 $(x_1, x_2, \cdots, x_M)$，Q 是长度为 L 的单词序列 $(q_1, q_2, \cdots, q_L)$，这里 L 为提前设置的最大输出长度，也就是最长问题长度。输出序列 Y 是长度为 L 的单词序列 $(y_1, y_2, \cdots, y_L)$，模型的目标是学习公式2-1中的条件概率分布函数，使得 Q 是输出序列 Y 的极大似然估计。

$$P(Y|X, \theta) = \prod_{t=1}^{L} P(y_t|X, y_1, y_2, \cdots, y_{t-1}, \theta). \tag{2-1}$$

基于文本的自动出题算法中，基础网络一般为带注意力机制的编码-解码网络。我们接下来会先介绍基础的编码解码网络模型[15]，如图2.3所示，然后举例说明在自动出题任务中常用到的一些改进方法。

- **Embedding 层**：将输入的每一个单词映射到一个低维的空间向量中，从而解决传统的 one-hot 向量无语义、稀疏、维度高难训练的问题。方便起见我们直接假定 x_i 是 Embedding 之后的向量表示。
- **编码器（Encoder）：** 用双向的 LSTM[16] 网络将输入文本信息提取到一个固定长度的上下文向量 C 中。LSTM 全称为长短期记忆网络（Long Short Term Memory），是一种更适合处理有长期依赖的循环神经网络结构[17]。传统的 RNN 重复结构中，当前时刻的隐状态输出由上一时刻的隐状态和当前时刻的输入经过线性变化加 tanh 激活函数得到，因此当时间跨度较大的时候，隐状态就会丢失以前的信息，而增大隐状态向量维度来保存更远以前的信息会增大网络训练的难度。LSTM 则在传统 RNN 的基础上，引入门和细胞状态

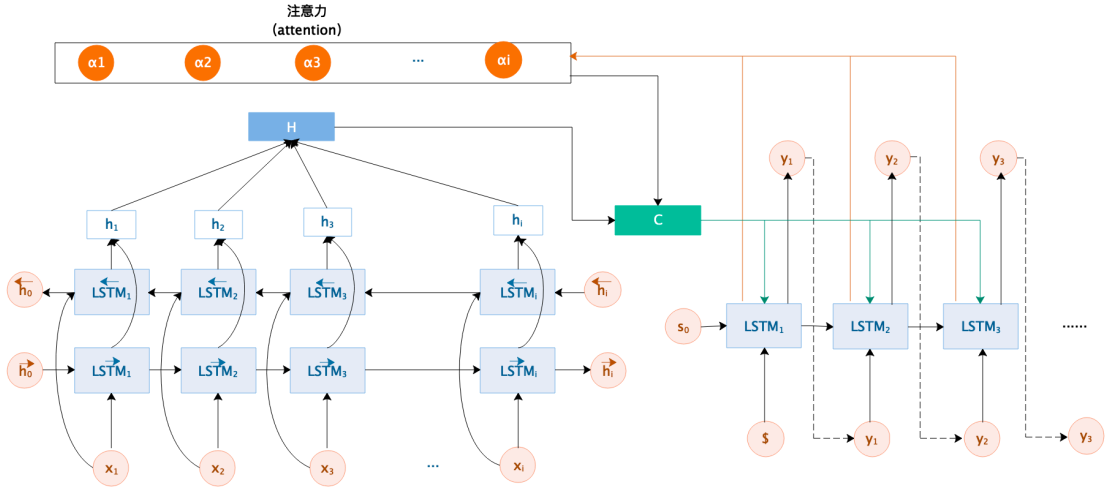


图 2.3　基于注意力机制的 Encoder-Decoder 模型

两个概念来解决 RNN 长期依赖难的问题。LSTM 共包含三个门，遗忘门、信息门和输出门。遗忘门根据当前时刻的输入和上一时刻的隐状态从细胞状态中删除特定信息，相反信息门则是根据同样的内容从细胞状态中加入特定信息。输出门根据上一时刻的隐状态和当前时刻的输入从更新之后的细胞状态中决定当前时刻的隐状态输出。三个门的顺序依次为遗忘门、信息门和输出门，每个门大致都由输入的线性变换加激活函数构成。双向 LSTM 需要将输入文本按从前到后和从后到前两个方向进行信息提取，每个单词对应的隐状态是两个方向得到的隐状态拼接在一起之后的结果。设 $\overrightarrow{h_j}$ 为第 j 个单词由前向 LSTM 单元得到的隐状态向量，$\overleftarrow{h_j}$ 为第 j 个单词由后向 LSTM 单元得到的隐状态向量，其各自计算方法见公式2-2：

$$\begin{aligned} \overrightarrow{h_j} &= \overrightarrow{bi-LSTM}(x_j, \overrightarrow{h_{j-1}}) \\ \overleftarrow{h_j} &= \overleftarrow{bi-LSTM}(x_j, \overleftarrow{h_{j+1}}) \end{aligned} \tag{2-2}$$

每个输入单词对应的隐状态向量为

$$h_j = \begin{bmatrix} \overrightarrow{h_j} \\ \overleftarrow{h_j} \end{bmatrix} \tag{2-3}$$

最终得到的上下文向量 C 为每个输入单词对应的隐状态向量的加权和，$C = H\alpha$，其中 $H = [h_1, h_2, ..., h_M]$，α 为权重系数，权重系数根据注意力机制由输出单词计算得到。

- **基于注意力机制的解码器（Attention-based Decoder）：** 基础解码器是一个

单向的 LSTM 单元，每个单元的输入为上一个输出单词，起始为一个代表开始的占位符。也就是说解码器 LSTM 的隐状态向量 $s_t = LSTM(s_{t-1}, y_{t-1})$，初始时刻的隐状态为编码器得到的上下文向量。这种情况下每一个时刻用于决策输出哪一个单词的上下文是一样的，而注意力机制则强调在不同时刻，决定输出哪一个单词对应的上下文应该是不一样的，需根据过去的输出序列来决定对应的上下文向量，也就是根据 s_{t-1} 来决定。从算法上来说就是用 s_{t-1} 更新用于决策 t 时刻输出单词的上下文权重系数 α_t，并计算出需要依赖的上下文 C_t[15]，然后将上下文向量信息整合到用于输出的向量 e_t 中，$e_t = \sigma(W[s_t^{\mathrm{T}}, (H\alpha_t)^{\mathrm{T}}]^{\mathrm{T}} + b)$。最后 e_t 经过一个 softmax 函数映射到单词表大小的一个向量中，代表每个单词可能用于输出的概率。$p = softmax(We_t + b)$。这里 W 和 b 都是神经网络的参数矩阵，前文中在计算输出向量 e_t 时也有用到相同的符号，但这两处不是同一个参数矩阵，此处为简便起见没有区分。

- **损失函数：**损失函数一般用交叉熵，也就是使得输出序列中每个位置上和 Q 序列该位置对应的单词的输出概率尽可能大，其他单词的概率尽可能小。

为使得上述带注意力机制的编码解码模型能更好的去解决自动出题任务，很多研究者对其进行了一些特定的改进，常用的一些经典改进手段包括但不限于以下几点：

1. 丰富输入层特征，使得编码器的输入不仅包括输入的单词语义上的特征（即常规的 word embedding），还包括语言学特征[18]、词汇特征[19][20]、答案位置特征[19][20]、命名实体标签信息[20] 等等。
2. 用 GRU（Gated Recurrent Unit）单元去构建编码器[19]。GRU 是一种 LSTM 结构的变体，相较于 LSTM，GRU 只有两个门，更新门和重置门，且 GRU 没有细胞状态。其中更新门用于决定历史信息保留到现在的程度，重置门决定当前输入和历史信息如何联合得到当前时刻的隐状态输出。可以发现，如果更新门为 0，重置门为 1 的话，GRU 失去对历史信息的记忆，退化为原始的 RNN 循环单元。与 LSTM 相比，GRU 的优点是结构更简单，参数更少，更容易学习，与此同时，它还保留着 LSTM 可以解决长期依赖问题的特性，因此是一种非常常用的 LSTM 结构变体。
3. 引入复制机制（copy mechanism）[19][18]。解码器会得到目标语言单词库（也就是由训练集中所有问题的单词构成的词表）的单词概率分布，选择概率最大的作为输出。复制机制则是以一定的概率选择从源文本中直接复制一个单词作为当前输出。决定是否从源文本复制并且选择复制源文本中哪一个单词，都是由当前注意力机制下算出的上下文权重系数分布决定的。这么做的原因

是自动生成问题任务中，问题中的单词很多都是和源文本一致的，引入复制机制可以很明显的提高预测准确率。

4. 引入段落级别的编码器[21]。源句转换为问句的时候，将源句所在段落的整体信息用段落编码器提取出来，并和句子编码器得到的信息一起去解码输出问题序列。
5. 论文[22] 将答案从输入分本中分离出来，并将输入文本中的答案用特殊字符进行替换，随后各自进行编码。实验表明可以取得很好的效果，其中答案编码器主要提取答案信息，输入文本编码器则用来提取答案位置信息、上下文信息。

2.2.2　基于知识图谱的自动出题算法

基于文本的自动出题算法主要在教育行业应用较多，而聊天系统、问答系统中的自动出题算法则大部分是基于知识图谱的。本节主要来介绍基于知识图谱的一些经典出题算法。与基于文本的出题算法一样，基于知识图谱的出题算法也主要有两大类：基于模版的算法和基于序列神经网络的算法。

- **基于编码解码神经网络的方法**。和基于文本的出题算法一样，基于知识图谱的经典出题算法也是用带注意力机制的序列神经网络，区别只是前者的输入序列为上下文文本，而后者则是知识图谱中的基本三元组。知识图谱是由很多实体结点和它们之间的关系组成的，因此其基本格式可以描述为一个三元组 $\{s, p, o\}$，其中 s 为主语，p 为谓词、关系，o 为宾语。复杂一点的格式中，可以对主语和宾语做修饰限定，构成五元组，其处理方式和三元组的处理方式完全一致，因此为简化起见，本文后续均以基本三元组来描述知识图谱。论文[23] 中直接将主语、谓词、宾语按序输入到序列神经网络中，然后输出对应的问题，该模型中默认将主语作为问题的答案，所以不需要额外标记答案信息。如果生成宾语为答案的问题，则只需要将主语和宾语顺序颠倒一下，并将谓词替换为相应的逆关系即可，例如文学作品到其中的文学人物的关系为主要人物（characters），它的逆关系即为人物到作品的关系出场作品（present in work）。论文[24] 则是通过从知识图谱中提取出所有可能的问题关键字-答案对，并将问题关键字依次输入到神经网络模型中得到对应的问题序列。为了保证生成的问题有唯一解，需要限定抽取问题关键字-答案对的规则，主要有两条：如果要让主语做答案，则要求主语到宾语之间的关系对于宾语来说是唯一的，也就是说知识图谱中不存在另一个实体也有一条指向宾语的边，其属性和主语指向宾语的一样；反之如果要让宾语做答案，则要求对于主语

来说，到宾语的属性取值也是唯一的，即没有另一个实体使得主语到它的关系和主语到宾语的关系一致。如果宾语为答案，问题关键字为 $\{s, p\}$，如果主语为答案，则问题关键字为 $\{p, o\}$。论文[25] 提出了基于模版的序列神经网络方法，该模型解码器输出的不是最终的问题，而是和问题相关的模版，其中和话题（答案）相关的部分用特殊符号替换，之后只需将其替换成输入对应的话题（答案）即可。论文[26] 提出了双编码器模型，第一个编码器称为事实编码器（fact encoder），它的输入为知识图谱中的 $\{s, p, o\}$ 三元组，另一个编码器为文本编码器（textual encoder），它的输入为三元组对应的上下文文本输入，其中谓词对应的上下文应该从训练集中和它相关的问题中获得，例如关系出生地 “place of birth”，它的主语一般为人，该谓词对应的上下文文本为 “[O] is birthplace of [S]”，主语和宾语对应的上下文文本为知识图谱中定义的实体类型。

- **基于模版的方法**。论文[23] [25] [27] 中采用的 baseline 方法都是基于模版的方法。基于模版的出题算法主要包括两个步骤：一是生成模版库，二是模版选择。在生成模版库阶段，我们需要给训练集中出现的每一个关系生成所有可行的模版库。生成某一个关系对应的模版库的具体方法为：找到训练集中该关系对应的所有问题，然后用字符串匹配的方法找到每一个问题对应的话题实体（topic entity），即问题答案，并用特定的占位符替换即可。给定一个事实三元组，该算法从和它相关的模版库中随机选取一个作为模版，并用输入话题的信息（答案）替换占位符即可，论文[23] 将它和基于神经网络的方法进行了对比，可以发现虽然后者性能有所提升，但效果并不是很明显。

2.2.3　小结

本节主要讨论了不同输入形式下常见的出题算法，无论输入为文本还是知识图谱，常见的算法都包括两大类：基于规则模版的传统方法和基于编码解码模型的序列神经网络算法。基于文本的出题算法对输入文本的质量要求比较高，一般为信息文本或教学文本，为使得生成的问题更准确，常常还需要对输入文本进行一些简化处理。基于知识图谱的出题算法里，神经网络算法虽然准确率比基于模版的方法高，但没有非常显著的准确率提升，相反基于模版的方法却更简单、速度更快。

除可以按输入形式划分之外，还有很多和自动出题算法相关的研究。例如按题型来分的话，有的论文专门研究如何生成填空题，有的则主要研究如何生成选择题，本节讨论的论文中生成的题型为 WH 型问答题，论文[28] 研究的主要是如何

生成填空题，论文[29][30] 则研究的主要是如何生成多选题。多选题题目组成包括题干和选项两部分，基于知识图谱的出题方式在构造错误选项时有明显的优势，可以根据知识图谱中实体之间的连接关系去构造错误选项，也可以根据实体相似度去构造错误项。除按题型划分外，还可以按领域划分，不同领域有其专有的一些实体类型和对应的问题形式，论文[30] 研究的就是如何根据不同领域的实体类型定制不同的出题策略从而自动生成题目。

2.3　自动组卷相关方法介绍

自动组卷问题可以被形式化描述为：给定试题库，算法需以一定的选题策略从试题库中选择若干试题，使得得到的试题组合尽可能满足预先定义好的约束，常见的约束包括难度、题型、知识点以及分数等。可以看出，这是一个多目标约束下寻求最优解的问题，常见的一些选题策略有：随机抽取法、回溯递归法、遗传算法等等，接下来我们先简要介绍一下各个算法的算法流程。

2.3.1　随机选取法

论文[31] 采用随机抽取法来生成试卷，具体流程见算法1。该论文中组卷约束包括总分、题目类型、难度以及知识点，知识点以章节来代替。算法首先按一定的策略将约束进行量化，使得约束细化为需要从题库中抽多少分的选择题、多少分的填空题（约束矩阵 T），每一章需要抽多少分简单题、多少分难题（约束矩阵 KD) 这种形式，然后从题库中随机进行抽取。可以发现该算法最大的问题的是约束太强，极易导致组卷失败，因此论文作者提出可以在现有强约束的基础上加入近似匹配，提高组卷成功率。具体做法为当某个 T_{1k} 或者 KD_{ij} 已经非常小了 (可接受的误差范围内)，可能只有 1 分或者 2 分，题库中根本不存在这样的题时，可以稍微调整原来的约束矩阵，使得组卷过程继续下去。尽管如此，综合观察我们可以发现，随机选取法虽然实现简单，但算法在生成量化目标的时候大大减少了候选解的个数，使得组卷成功率较低，性能较差。

2.3.2　回溯递归法

回溯法[32] 是随机抽取法的一个改进版本，在随机抽取法中，算法之所以容易失败是因为目标之间的相互干扰以及量化目标时的绝对导致。在回溯递归法中，有两处可以回溯的地方：一是量化目标，如果在某一个量化目标的指导下组卷失败的话，可以对其进行调整回溯，继续组卷过程，例如刚开始给某一章难题分的比例较多，而题库中相应的题目较少时，组卷势必会失败，这时应该降低该章难题

Algorithm 1: 基于随机抽取法的组卷流程

Input: 题型、难度、知识点（章节）、总分约束；

Output: 满足约束的试题组合

% 根据约束生成量化矩阵 T 和 KD；

1. 根据总分和题型约束，得到题型约束矩阵 T，该矩阵维度为 [1, 试题库所有题型个数]，T_{1k} 代表需要从题库中抽取 k 题型的题目总分；

2. 根据总分和难度、章节约束，得到章节难度约束矩阵 KD，该矩阵维度为 [所有章节数（知识点个数），所有难度等级个数]，同样，KD_{ij} 代表需要从题库中抽取第 i 章难度为 j 的题目总分；

while $(\exists i, j, k, s.t.(T_{1k} \neq 0) and (KD_{ij} \neq 0))$ **do**

从试题库中找到所有未被选择，且题型为 k，难度为 j，属于第 i 章的试题集合 Q;

if $Q.count == 0$ **then**

printf（“组卷失败”）;

break;

end

从 Q 中搜索一道分值不大于 T_{1k} 和 KD_{ij} 的试题 q；

if 能找到试题 q **then**

$T_{1k} = T_{1k} - score(q)$;

$KD_{ij} = KD_{ij} - score(q)$;

标记 q 的状态为已选择；

else

printf（“组卷失败”）;

break;

end

if $(\forall i, j, k, (T_{1k} == 0) and (KD_{ij} == 0))$ **then**

printf（“组卷成功”）;

break;

end

end

比例，相应增加其他章的难题比例。第二个需要回溯的地方是在选题的时候，我们知道算法1中满足约束的 Q 集合中的题目往往不只一个，当选中某道题无法继续之后的组卷过程时，可以回溯试试选择另一道题。可以看出，回溯法的引入大大增加了候选解空间，使得组卷成功率明显提高，但与此同时，也大大增加了算

法执行的时间。因此该算法仅适用于题库较小的情况，无法有效解决大题库上的组卷任务。

2.3.3　遗传算法

遗传算法是一种常见的寻找最优解的算法，它是一种模拟自然界生物进化过程的最优解搜索算法。其以实现简单，可以并行计算处理、收敛速度快，同时可以找到全局情况下的最优解等优点，常被用于解决自动组卷任务[33]。用遗传算法解决组卷任务基本的流程为：首先我们需要明确搜索任务的解是什么，并对其进行编码，方便之后的计算操作过程。在组卷任务中，我们的最终解是试卷，也就是若干试题的组合，因此我们需要对试卷进行编码。可行的编码方案有很多种，例如我们可以对试题库的题目进行排序编号，然后将试卷编码为长度为试题库题目总数的二进制串，每一个位置上的二进制值代表对应编号的试题选择状态，0 代表未选择，1 代表选择。第二步我们需要设置解个体的适应函数值，也就是个体的进化方向。进化过程中（也就是搜索过程中），适应函数值越高的个体其携带的信息越容易被保留下来，反之适应函数值越低的个体则越容易被淘汰。在组卷任务中，适应函数值一般为试卷的各项指标与目标约束之间的差异，差异越小，个体越有可能是最优解，适应函数值应该越高。第三步设置终止条件。一般终止条件为在一定的误差范围内找到最优解、或者迭代层数达到一定的上限。在完成上述的准备工作之后，算法开始正式进行迭代搜索。第四步：随机生成若干候选试卷解构成初始群体，也就是第一代解。第五步：计算群体中每个个体的适应函数值，判断算法是否达到终止条件，如果达到了则跳出整个迭代过程，反之继续。第六步：通过遗传操作生成下一代群体，遗传操作包括选择、交叉、变异。选择是指以一定的概率（选择概率）直接复制父代群体中的部分个体到下一代。例如如果选择概率为 20%，交叉概率为 80%，群体个数为 100 的话，下一代群体中 20 个个体直接是父代群体中已有的个体，80 个个体是由父代个体经过交叉和变异生成的新个体。选哪些个体进行复制和交叉与个体的适应函数值有关，适应函数值越高，其携带的信息越容易保留到下一代中，也就是选中去复制或交叉的概率越高。为使得下一代个体的信息更丰富，一般复制到下一代的个体不参与交叉操作。交叉是指根据个体的适应函数值依次从父代中选择“父”“母”两个个体，然后随机选中一个位置将其编码二进制串截断，并交换截断位置之后“父”“母”双方的编码信息。交叉可以看作是群体之间遗传信息的重新组合过程，但仅靠交叉来生成新的解个体会使得搜索方向陷入局部最优解，因此交叉之后我们需要以一定的概率引入变异操作，使得算法可以跳出局部最优，在全局寻找最优解。变异是指随机选择一个

位置，更改其对应的遗传信息，也就是对应位置编号的试题选择状态。通过遗传操作得到与父代个体数一样的群体之后，回到第五步，开始下一轮迭代或者跳出整个迭代过程。

可以看出，遗传算法的性能和其参数设置（选择、交叉、变异概率）、编码设置、适应函数设计等密切相关，因此为使得算法更容易收敛，找到全局范围内的最优解，很多论文在基础的算法上进行了改进使得其更适用于组卷任务。例如论文[34] 提出了基于整数编码和自适应遗传算法的组卷流程，这里自适应是指选择概率、交叉概率和变异概率都是动态可变的。

2.3.4　小结

本节我们主要对常见的几种组卷算法进行了简要的介绍，可以发现随机抽取法实现简单，但组卷失败率较高；回溯递归法组卷成功率较高，但因为需要遍历所有候选解，算法执行时间较长，仅适用于小试题库，无法完成大题库的组卷任务；遗传算法以其实现简单、可以并行计算处理、收敛速度快、同时可以找到全局情况下的最优解等优点常被用来处理组卷任务。与此同时，遗传算法参数的设计、遗传操作的设计、编码设计等都会对算法性能产生较大的影响，因此在实际组卷过程中，需根据目标约束仔细考虑和设计。

2.4　WIKIDATA 相关工作介绍

WIKIDATA[①]是由维基媒体基金会支持的一个维基项目，旨在希望能为其他维基项目或者个人提供结构化数据支持。它主要有以下四个特点：

1. 它允许任何人、任何公司对其数据进行下载，并且在不需要任何版权的情况下按自己的需求随意使用、修改。
2. 它的数据录入可以由不同用户、甚至机器协同完成。同时，它为每个实体提供到外部数据库的链接，从而使得原先不同数据库的数据被链接到一起，提高知识的丰富性。
3. 它支持多种语言，目前其已经可支持近 300 种。
4. 它的数据都有固定的结构模式，方便计算机“理解”和处理。

因为本文第3章的自动出题算法需要借助 WIKIDATA，所以本节先对 WIKIDATA 及相关工具进行一个简要的介绍，包括数据结构、数据获取方式以及实体相似度的计算方法。

① https://www.wikidata.org/wiki/Wikidata:Introduction

2.4.1　WIKIDATA 数据结构

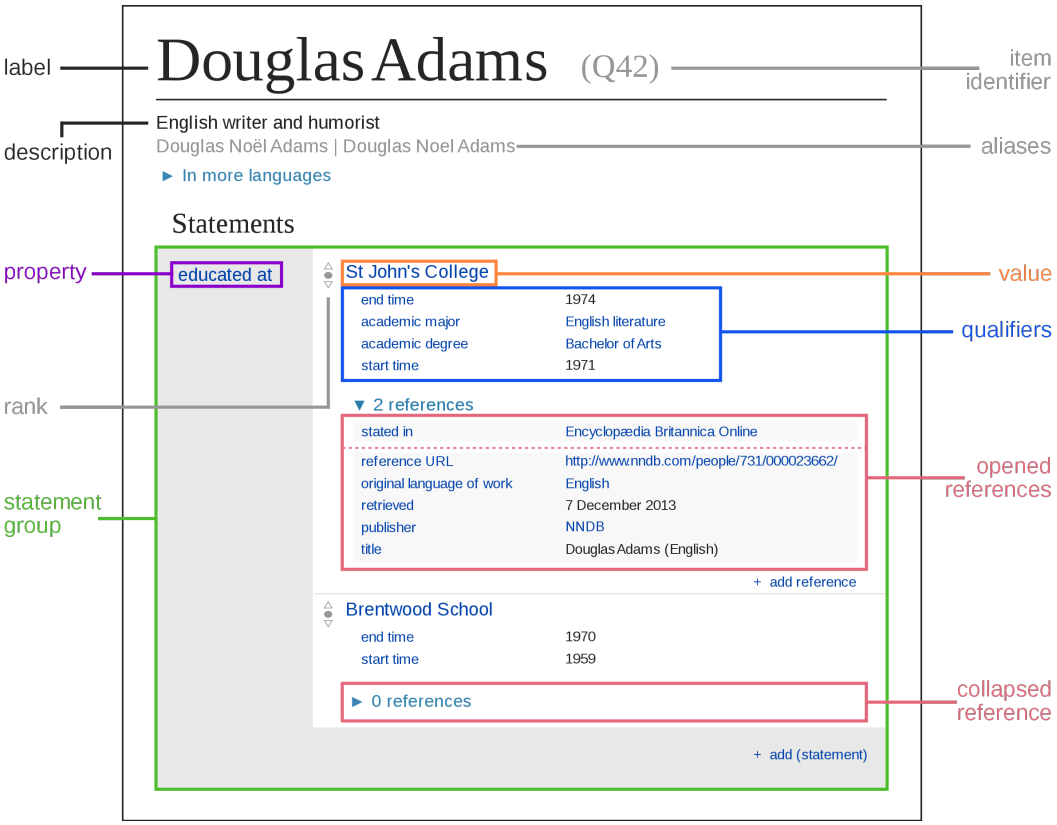


图 2.4　WIKIDATA 数据库项目结构图[35]

WIKIDATA 数据库主要由很多个项目（item）组成，截止目前共包含 5000 多万个项目。如图2.4所示，每个项目都包括固定格式的基本信息、按属性分组的语句（statement）描述以及与该实体相关的外部站点链接（sitelinks）。其中基本信息包括唯一的项目标识符（item identifier，一般以 Q 开头）、多语言项目标签（label）、多语言描述信息（description）以及任意数量的多语言别名（aliases）。例如项目 Q42 中文基本信息有：标签为道格拉斯·亚当斯，对其的描述为英国作家，他的别名有亚当斯。语句则主要描述项目的详细特征以及和数据库中其他项目的关系，每条语句由属性（property）和对应的属性值（value）组成。可以发现，如果将项目对应到知识图谱中的结点的话，属性则对应图谱中的边，现在 WIKIDATA 数据库中已经有 6000 多种不同的属性。与项目类似，每一个属性也包括其基本信息和语句，只是属性的标识符一般以 P 开头。常见的属性值类型主要有字符串、数量值、时间、地理位置或者其他项目，而且一个属性往往可以包括多个属性值，例如图2.4中属性 P69，在哪里接受教育（educated at），这个属性的主体一般为人，对应的属性值可以包括多个大学项目，比如 Q691283 剑桥大学圣约翰学院，Q4961791

雁林学校。可以发现，虽然我们可以很方便地用简单的属性-值对来表达项目信息，但这在很多时候往往是不够用的，因此为了进一步提供更清晰、准确的信息表达，对于每一个属性-值对，可以选择性地引入一个或多个限定符（qualifiers）、引用（reference）或排序（rank）去进一步地对语句进行扩展，注释或上下文化，这里限定符和引用的形式和语句很类似，也由属性和属性值组成。例如图2.4中道格拉斯 · 亚当斯（Q42）在剑桥大学圣约翰学院（Q691283）接受教育（P69）这条语句就可以用两个限定符，开始时间（P580）和结束时间（P582）来进一步描述他在剑桥大学圣约翰学院上学的时间。此外，有一些语句是专门用来将项目链接到外部数据库的（例如库和归档使用的权限控制数据库），这些属性统一称之为标识符（identifier）。项目的特殊附加链接（sitelinks）则用来将其连接到 Wikipedia，Wikisource 和 Wikivoyage 等其他维基媒体网站。

2.4.2　WIKIDATA 数据获取方式

WIKIDATA 的数据获取分为在线和离线两种方式，离线获取首先需要将 WIKIDATA 整个数据库下载下来并将其导入到本地数据库，然后基于本地服务器搭建搜索服务。这种获取方式的优点是速度快，稳定性好，尤其当出现大量访问请求的时候，在线搜索查询往往会发生堵塞，不能实时返回结果，而离线查询则会稳定、快速地返回查询结果。但缺点是实现复杂，且本地数据库需要定期进行更新维护。在线获取主要是利用 WIKIDATA 提供的一些在线 API 和查询工具实现，这种获取方式最大的优点就是实现简单，但往往会受限于网络情况、请求频率等因素。常用的 WIKIDATA 在线数据获取方式主要有以下几类：

1. **Mediawiki API**[①]。 Mediawiki API 是维基百科、WIKIDATA 的在线访问接口，用户通过向 Web 服务器发送 HTTP 请求，可以实现登录 WIKI、访问数据并发布更改等功能。在本论文中，我们主要用到了两个接口，一个是 wbsearchentities 动作接口，该接口根据用户输入的查询关键字返回对应实体基本信息，接口返回格式为 JSON，例如搜索 Fudan，该接口返回的部分相关实体数据格式如图2.5所示。其中返回数据中较为重要的是实体 id 和 match 字段信息，id 需要在后续的处理中被用来获取实体详细信息，match 字段则用来说明实体和搜索关键字的匹配信息，图中 type 为 aliase，表明该实体有别称和关键字匹配。另一个是 wbgetentities 接口，该接口根据用户输入的实体 id，以 JSON 格式返回对应实体的详细信息。所以利用 Mediawiki API 获取 WIKIDATA 数据的方式一般为：先调用第一个接口确定关键字对应的实

① https://www.mediawiki.org/w/index.php?title=API:Main_page/zh&oldid=2988378

体 id，然后调用第二个接口获取实体详细信息。


```
{
        "repository":"",
        "id":"Q45533699",
        "concepturi":"http://www.wikidata.org/entity/Q45533699",
        "title":"Q45533699",
        "pageid":46701664,
        "url":"//www.wikidata.org/wiki/Q45533699",
        "label":"Li Qidan",
        "description":"Qing dynasty person CBDB = 77592",
        "match":{
                "type":"alias",
                "language":"en",
                "text":"Fudan"
        },
        "aliases":["Fudan"]
}
```


图 2.5　Mediawiki API 中 wbsearchentities 接口返回数据示例

2. **WIKIDATA Query Service (WQS)**。基于 SPARQL 语言的 WIKIDATA 数据库查询服务，是目前关联数据查询最主流的方式。SPARQL 是一种类似数据库 SQL 语言的 RDF 查询语言，本论文中我们主要用它来构建题目的错误选项，主要选取数据库中和正确答案相关的实体作为错误选项。
3. **WIKIDATA Client**[①]。WIKIDATA client 是用户自发编写的一个 python 包，主要是利用 Mediawiki api 中 wbgetentities 接口获取实体详细信息，并对返回数据进行了初步的格式解析。

2.4.3　WIKIDATA 实体相似度

WIKIDATA 实体的相似度计算可以仿照两个单词计算相似度的方法。计算两个单词相似度最常用的方法就是词向量模型，词向量模型通过大量的语料训练，将单词的语义信息保存在一个低维向量中，从而根据向量之间的距离来计算单词的相似度，例如一个很常见的例子是：$\overrightarrow{king} - \overrightarrow{man} + \overrightarrow{woman} = \overrightarrow{queen}$，从国王的信息中减去男人的特征，再加上女性的特征之后即等于王后的词向量表示。同理，将知识图谱中的实体利用词向量模型得到其含有上下文信息的向量表示后，便可以得到实体之间的相似度，为后文基于知识图谱的自动出题提供支持。

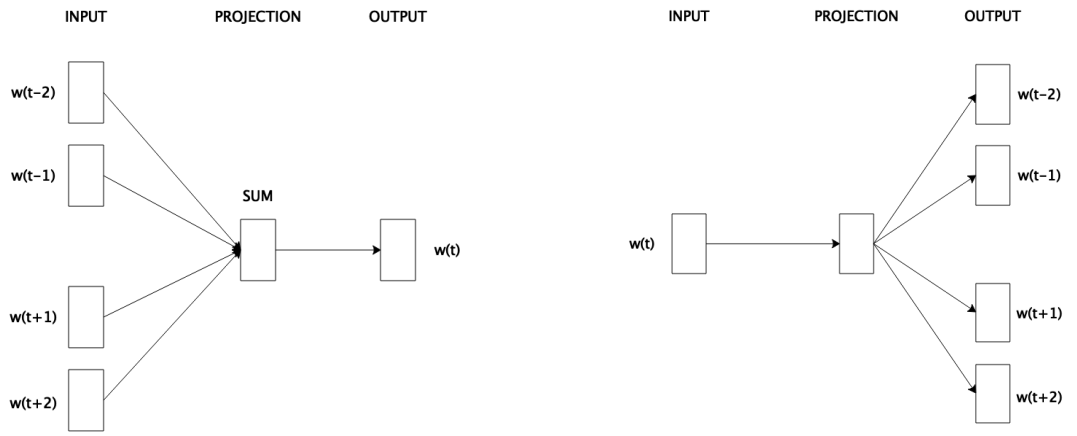
2.4.3.1　词向量模型

词向量模型一般有两种不同形式：一种是 CBOW 模型，另一个种是 Skip-Gram 模型，如图2.6所示。给定一个单词 $w(t)$ 和它对应的上下文单词序列 $c(t)$，CBOW 模型是用 $c(t)$ 去预测 $w(t)$，而 Skip-Gram 模型则反过来，用 $w(t)$ 来预测 $c(t)$。我们这里仅以 CBOW 模型为例来对整个网络结构进行说明。模型总共包括三层。输

① https://pypi.org/project/Wikidata/

入层为某一语料中单词 $w(t)$ 的对应的上下文单词序列 $c(t)$，例如“My mom drink coffee everyday?”语句中，如果上下文窗口大小为 4，则单词 drink 的上下文单词序列为（My，mom，coffee，everyday），每个单词原始的词向量可以设为随机值。映射层为输入层的若干个词向量之和。输出层利用变换矩阵和 softmax 函数将映射层得到的向量转化为词典中所有单词的概率值。训练目标为期望训练样本中特征词 $w(t)$ 对应的 softmax 概率最大。相对应地，Skip-Gram 模型的训练目标则变为期望训练样本中特征词 $w(t)$ 的上下文单词序列 $c(t)$ 的联合概率更大。在实际实现过程中，一般会引入两种不同的加速方法：

- 用层级 softmax 代替线性 softmx。当词典规模较大时，线性 softmax 是非常浪费时间的，层级 softmax 用哈夫曼树来代替映射层到输出层的映射。假设树根到某一单词的路径长为 L，则获得该单词的输出概率仅需做 L 次二分类即可，然后利用对数似然函数进行极大似然估计求取参数。哈夫曼树中高频词更靠近根部，到树根的路径长度更短，因此相比线性 softmax 过程而言，层级 softmax 的速度明显会更快。
- 负采样技术。负采样加速技术中，每一个训练样本仅被用来更新部分网络权重，从而降低梯度下降过程中的计算量，实现网络加速。以 CBOW 模型为例，除特征单词 $w(t)$ 之外的所有单词都称之为负样例，我们每次仅从所有负样例中随机抽取若干个和正样例特征单词一起去更新网络权重，而不是使用所有单词的概率值。



(a) CBOW 模型　　(b) Skip-Gram 模型

图 2.6　词向量模型[36]

2.4.3.2　训练数据

WIKIDATA 支持用户下载其所有数据，我们将所有数据下载之后，需要对所有实体的语句（Statements）进行分析，提取出所有宾语也是实体的语句，最终得到数据格式形如（Q3772823，P1441，Q8265）的训练语料库，其中单词为实体或属性 id。然后用词向量的训练方法在该训练数据集上进行训练得到实体 id 对应的词向量表示。

2.4.3.3　相似度计算

得到实体的嵌入式向量表示后，实体之间的相似度计算可以用词向量之间的距离或夹角来表示。论文[37] 在包含 60 多万个实体和 700 多个属性的训练集上进行训练，并根据训练得到的实体向量提供了两个和实体相似度有关的 API 接口，一个是给定一个实体 id，接口从所有实体中返回前若干个最相似的实体。另一个则是给定两个实体 id，接口返回两个实体的相似度。本文后续和 WIKIDATA 实体相似度有关的计算均是通过调用这两个接口得到。

2.4.4　小结

因为在本文第3章提出的基于 MOOCs 视频字幕自动出题算法中，需要用 WIKIDATA 来对视频字幕进行结构化，因此本节主要对 WIKIDATA 知识图谱中的数据格式、数据获取方式以及实体相似度计算方式进行了一个简要的介绍。

第 3 章　基于 MOOCs 视频字幕的自动出题模型

3.1　引言

本章我们主要介绍基于 MOOCs 视频字幕的自动出题算法，整体的算法流程如图3.1所示。首先借助维基知识图谱 WIKIDATA 对字幕文本进行结构化，从中提取出感兴趣的知识实体及实体之间的关系，构建课程知识图谱。然后基于课程知识图谱去自动生成测试题目。

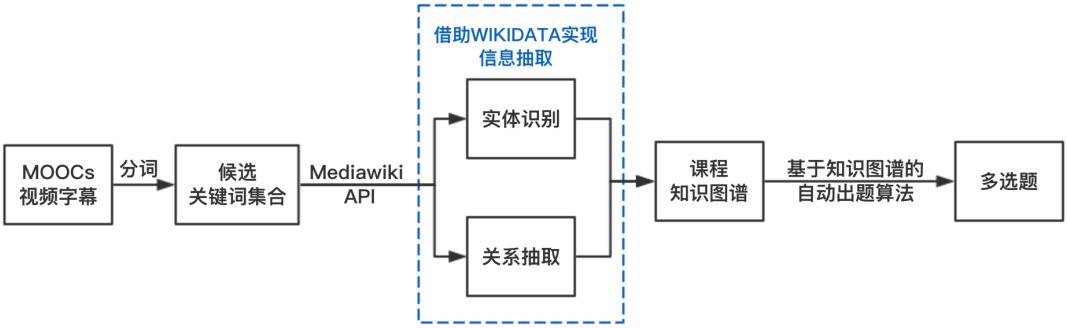


图 3.1　基于 MOOCs 视频字幕的自动出题模型

我们知道常见的自动出题任务都是根据上下文找到给定答案的最佳提问形式。上下文数据一般来自课本、维基百科或者学习网站，图2.1展示的是基于文本的自动出题任务中常用的一个数据集 SQuAD 的数据样例，其文章来源于维基百科。可以发现这类文本的特点是：有明确的断句和标点符号；以陈述事实的信息文本为主；内容简练；措辞严谨。而视频字幕呢，一般是演讲者的解说文字，主要特点是：口语化；信息冗余；无标点符号和语义断句。演讲者在陈述一个事实的同时往往会穿插一些不相关的内容，例如自己的观点、经历等，这些内容在使得字幕信息变得冗余的同时，也会把相关上下文的跨度变大，给上下文的抽取增加难度。与此同时，字幕文本一般按时间进行分段，无标点符号及语义上的断句，从而进一步增大有效上下文提取的难度。

结合 MOOCs 字幕的特点，我们可以发现直接采用经典的文本出题算法来完成字幕出题任务是非常困难的，且最终生成的问题质量也会较低。因此本文决定先采用维基知识图谱去结构化字幕文本，从中提取出感兴趣的知识实体及实体间的关系，然后进一步基于知识图谱去完成自动出题任务。这样做的缺点是会丢失部分无法结构化的信息，优点是会使得课程内容更加趋于结构化，为很多其他课

程功能的自动化提供可能，例如知识点总结、知识点索引、外部学习资源链接等；与此同时它也可以丰富自动出题的方法，譬如可以从开源的外部题库中检索和本课程内容相关的题目。

接下来我们会分成三个小节进行详细介绍，第3.2节介绍信息提取，也就是如何从视频字幕中提取知识实体，构建课程知识图谱；第3.3节介绍如何基于知识图谱去自动生成题目；第3.4节为实验部分。

3.2　基于 WIKIDATA 的 MOOCs 视频字幕知识图谱构建

知识图谱是由很多实体（节点）和它们之间的相互关系（边）组成的，因此知识图谱构建一般需要解决实体识别和关系识别两个子问题。这两个问题都是近年来自然语言处理的热点研究问题——序列标注问题。调查发现当前该问题的解决流程大致是：

- 第一步，定义需要识别的实体或关系类型；
- 第二步，采集语料库，并进行数据标注；
- 第三步，特征采集，可以人工定义特征模版，也可以采用神经网络自动提取特征；
- 第四步，用条件随机场（Conditional random field, CRF）模型获得概率最高的标注结果。

这种解决流程在实际使用中主要有以下几个问题：一是时间开销巨大。无论是数据集的采集还是标注都是一个极其耗费人力的过程，而且标注质量难以控制，继而会影响最终的实验结果；二是可扩展性低。开源的公开数据集实体类型、关系类型较少，对于每门课程而言，基本需要从零开始去构建、标注自己的训练集，因此构造课程知识图谱的起步代价过大，实现难度高。为了使得每门课程无需额外精力就可以构建一个最基础的知识图谱，我们必须尽可能多地搜集包含各个领域知识的语料库，并且在这些语料中定义和标注尽可能多的实体类型、关系类型，使得其可以服务于任意学科。维基百科和维基知识图谱刚好就符合这样的需求。

维基百科是全球最大的网络百科全书，除自由可编辑之外，它最大的特点就是内容丰富。它不仅收录百科知识，连一些非学术的热门事件也都有记录，因此维基百科语料库完全可以满足我们在内容广度方面的需求。同时，Freebase，WIKIDATA 都是类似于维基百科的大型维基知识图谱，其中已经定义了数以万计的实体类型和关系类型。与此同时，它也允许网民自由编辑，便于增加新的实体和关系类型。因此，本文选择用维基知识图谱作为基础知识库来帮助完成实体识别和关系提取。Freebase 已在 2014 年宣布关闭，并将所有数据迁移到了 WIKIDATA，因此本文选

择 WIKIDATA 作为基础知识库，与 WIKIDATA 相关的介绍参见第2章内容。

利用 WIKIDATA 从 MOOCs 视频字幕中提取实体并构建知识图谱的伪代码见算法2，其中输入为一段 MOOCs 视频的字幕，记为 S，需要识别的实体类型集合，记为 T；目标输出有三个，一是实体集合，记为 *Entities*，其中存储字幕中出现并且类型属于 T 的所有实体信息；第二个是关系集合，记为 *Relations*，其中存储 *Entities* 中任意两个实体的关系；最后一个是属性集合，记为 *Properties*，其中存储 *Relations* 集合中出现的所有属性详细信息。

Algorithm 2: 基于 WIKIDATA 从 MOOCs 视频字幕中构建知识图谱

Input: MOOCs 视频字幕文本 S; 需要识别的实体类型集合 T

Output: 字幕 S 中出现并且类型属于 T 的所有实体信息，记为 *Entities*;
Entities 中任意两个实体之间的关系，记为 *Relations*;
Relations 中出现的所有属性信息，记为 *Properties*

1 获取所有候选的单词集合，*WordSets*
2 遍历 *WordSets* 集合，对其中每一个 *word*，调用 Mediawiki API 的 wbsearchentities 接口得到所有可能的实体 id，然后根据 id 调用 Mediawiki API 的 wbgetentities 接口或利用 WIKIDATA Client 获取实体的详细信息
3 解析实体类别，将所有类型属于 T 集合的实体存入 *Entities*
4 遍历 *Entities*，解析实体语句（Statements），提取 *Entities* 中的每一个实体和剩余实体的关系，并将其存储在 *Relations* 中
5 利用第二步的方法，根据属性 id 获取所有 *Relations* 中出现的属性信息，保存在 *Propertities* 中。
6 实体过滤消岐。删除图 (*Entities*, *Relations*) 中所有节点总数小于给定阈值的连通子图。
7 **return** *Entities*, *Relations*, *Propertities*

输入字幕文本 S，首先第一步需要得到所有可能的候选单词集合。如果字幕是英文，可以采用 n-gram 法轻易地得到所有候选单词，但如果字幕是中文的话，则首先需要进行分词。当前已经有很多优秀的中文分词工具可供我们直接利用，如 jieba，SnowNLP，THULAC，NLPIR 等，本文选择用 jieba 分词工具来完成中文分词任务。

第二步，对于每一个候选词，调用 Mediawiki API 的 wbsearchentities 动作接口，得到所有和候选词匹配的实体 id，并调用 wbgetentities 动作接口或利用 WIKIDATA Client 得到实体的详细信息。

第三步，解析实体类别。WIKIDATA 中实体的类别是由 P31(instance of) 和 P279(subclass of) 两个属性确定的。我们需要过滤掉所有类型不属于 T 的实体，并且保存所有符合要求的实体信息，存入 $Entities$。这一步是利用实体类别限制做的第一次实体过滤、歧义消除。

第四步，关系抽取。解析 $Entities$ 中每一个实体关联的语句（Statements）。如果实体 E_s 的某条语句是描述它和 $Entities$ 中另一个实体 E_o 的关系的话，则 $Relations[E_s][E_o]$ 就等于该语句对应的属性。

第五步，利用第二步的实体信息获取方法，获取所有 $Relations$ 中出现的属性的详细信息，存入 $Properties$ 中。

最后一步，实体过滤消岐。这一步过滤消岐所依据的原理是：一节课程里涉及到的知识点大多都是相互关联的。给定最小连通阈值 $THRE$，删除图（$Entities$，$Relations$）中所有节点总数小于 $THRE$ 的连通子图。

为了能够更直观展示提取到的知识实体及其之间的关系，我们利用 D3.js 实现了知识图谱的可视化。它是一个专门用来做可视化的 js 函数库，本文关于知识图谱的可视化是借助其中的力导向图来实现的，具体的可视化结果参见实验部分3.4。

3.3　基于课程知识图谱的自动出题算法

知识图谱是一个以实体作为节点、关系作为边的有向图。例如，在 WIKIDATA 中，实体红楼梦（Dream of the Red Chamber）到实体曹雪芹（Cao Xueqin）之间就有一条边作者（author）。可以发现，{红楼梦（Dream of the Red Chamber），作者（author），曹雪芹（Cao Xueqin）} 这样一个三元组其实就可以表示一条完整的信息，因此我们称这样的三元组为一个事实（fact），其中红楼梦（Dream of the Red Chamber）为主语（subject），作者（author）为谓词（predicate），曹雪芹（Cao Xueqin）为宾语（object），知识图谱实质上是由很多个三元组事实构成的集合。

基于课程知识图谱的自动出题流程见图3.2。本文自动出题生成的题目类型为多选题，其题目构成包括两部分：题干和选项。因此基于知识图谱去自动生成多选题需要依次解决两个问题：构造错误选项、生成题干。

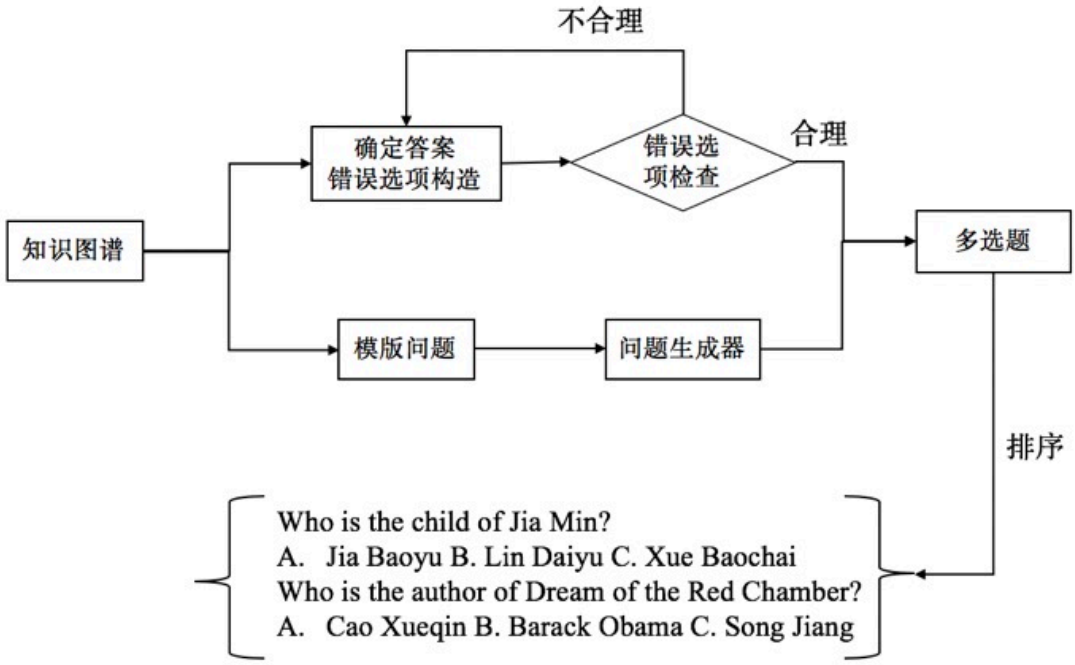


图 3.2　基于课程知识图谱的自动出题流程图

3.3.1　错误选项构造

错误选项的构造方法主要有两大类：一类是利用图谱中实体之间的关系进行构造；另一类则是利用实体之间的相似度来进行构造。前者我们可以利用 WIKIDATA 提供的 query service（WQS）从其数据库中检索和答案相关的实体作为错误选项，例如可以选择和正确答案类型一致的实体作为错误选项，并优先选择课程视频中已经出现的同类型实体，检索语句示例如图3.3所示，该图展示的 SPARQL 语句表示从 WIKIDATA 中选取 100 个类型（P31，instance of）为人类（Q5，Human）的实体。后者我们可以利用第2.4节中介绍的 WIKIDATA 实体相似度计算方法选择和答案最相似的实体作为错误选项。

```
SELECT ?instance_of ?instance_ofLabel WHERE {
  SERVICE wikibase:label { bd:serviceParam wikibase:language "[AUTO_LANGUAGE],en". }
  ?instance_of wdt:P31 wd:Q5.
}
LIMIT 100
```

图 3.3　WQS 检索语句示例

错误选项生成之后我们需要对其合理性，即错误性进行检查。以答案为宾语为例，如果错误选项和主语之间有关系，且关系和正确选项与主语之间的关系一致，则说明如果没有其他关于答案的限制要求的话，错误选项也是一个正确的答

案，这显然是不合理的，需要重新构造。这里我们选择检查答案和错误选项与主语（三元组中除答案之外的另一个实体）之间的关系，而不是通过检查主语针对该关系的宾语列表中是否包含错误选项来进行合理性排查，原因是 WIKIDATA 中包含一种类似举例说明的一对多关系，此时宾语列表并没有包含所有正确的解，例如三元组 {红楼梦（Q8265），出场人物（P674），林黛玉（Q3178958）、贾宝玉（Q8428650）、薛宝钗（Q8045205）} 中，我们知道红楼梦的出场人物远不止林黛玉、贾宝玉、薛宝钗三个，但因为人太多这里并没有全部列出来，如果此时我们构造的错误选项为王熙凤（Q3772823）的话，她本身也是红楼梦的一个重要人物角色，但是仅通过检查红楼梦的出场人物列表是排查不出的，所以我们需要反过来检查王熙凤和红楼梦之间的关系出场作品（P1441）是否和林黛玉与红楼梦之间的关系一致。

3.3.2　题干生成

基于模版的方法从知识图谱中自动生成题目主要包括三个基本过程：一是生成候选模版集；二是从候选模版集中选择目标模版；三是模版内容替换。

- **候选模版集：**这一步我们和传统基于模版的方法[23] 一样，都是用训练集中所有和输入关系一致的三元组作为候选模版。
- **模版选择：**这一步传统基于模版的出题方法的选择策略为随机选取。而我们的策略则是选择和输入最相似的三元组作为模版，谓词一样的两个三元组的相似度等价于两个主语和两个宾语之间的相似度，如何计算 WIKIDATA 中两个实体的相似度见2.4.3.3。这么做的原因是即使对于同一个谓词，如何去问问题还是会和主语（或者宾语）的类型有关，如果仅采用随机选取的方式的话，很有可能会选中一个不合适的模版。例如对于三元组 {Edward clarke cabot, notable work, Uptown Theater} 来说，合理的问题应该是 “Edward Clarke Cabot 设计了什么建筑（what building did Edward Clarke Cabot design?）”。而如果选择三元组 {曹雪芹（Cao Xueqin）, 重要作品（notable work）, 红楼梦（Dream of the Red Chamber）} 做模版的话，模版问题是 “曹雪芹写了什么书（What book did Cao Xueqin write?）”，因此我们会生成问题 “Edward Clarke Cabot 写了什么书（What book did Edward Clarke Cabot write?）”，这显然是不合理的。不同动词（write v.s. design）和不同修饰词（book v.s. building）的使用恰好体现出了实体类型的不一致性。因此我们的模版选择策略为从候选模版集中选择和输入三元组最相似的三元组事实作为模版去生成题干描述。
- **模版替换：**输入三元组其实是由两个实体加一个关系组成的，一个实体作为

答案，不出现在题干中，另一个实体则需要出现在题干中用来生成问题，我们分别称之为答案实体和问题实体。传统基于模版的出题方法仅替换问题实体的内容，例如对于输入三元组 {Lady Gaga, place of birth, Manhattan} 来说，如果选取的模版问题是“哪个足球运动员出生在都柏林（Which football player was born in Dublin）”的话，那么传统基于模版的方法仅将问题题干中的都柏林（dublin）替换为曼哈顿（manhattan）。很明显这种替换是远远不够的，因为我们知道 Lady Gaga 是一个歌手，而并非运动员。因此我们的替换策略是在传统模版内容替换的基础上，引入实体的描述信息，从而增加问题的准确性和可理解性。这里实体描述信息既包括答案实体的，也包括问题实体的，前者主要用来提高问题的准确性，后者则主要以注释的形式来增强问题的可理解性。例如对于三元组 {Lady Gaga, place of birth, Manhattan} 来说，我们的模版替换策略最终生成的题目为“Which singer player was born in Manhattan(borough of New York City, New York, United States)?”，其中歌手（singer）是答案描述信息，括号内的内容美国纽约市的一个自治行政区则是问题实体注释信息。

3.3.3 题目重要度排序

最终生成的题目需要按其重要程度进行排序。题目的考点是和答案密切相关的，因此我们用答案在字幕中的重要程度来衡量题目的重要性，这里我们选择用 $TF-IDF$ 算法来衡量答案在字幕中的重要性。$TF-IDF$ 是一种常见的关键词算法，计算方法见公式3-1：

$$\begin{aligned} TF-IDF(w,d) &= TF(w,d) * IDF(w,d) \\ TF(w,d) &= \frac{\text{单词}w\text{在文档 d 中出现的总次数}}{\text{文档}d\text{的总单词数}} \\ IDF(w,d) &= \log \frac{\text{语料库总文档数}}{\text{语料库中包含}w\text{的文档总数}} \end{aligned} \tag{3-1}$$

由其计算公式我们可以看出，对于某一特定文档，$TF-IDF$ 一般会认为整个语料库内不常出现的文档高频词对于该文档更重要。

3.4 实验

实验部分共包括两个实验，第一个实验是在公开数据集 SimpleQuestions 上去验证我们提出的基于模版的出题算法在知识图谱自动出题任务上的性能，第二个

实验则是在真实的 MOOCs 视频字幕上去验证我们提出的出题算法的效果。

3.4.1　数据集

SimpleQuestions[38] 是由 Facebook 研究人员构建并发布的数据集，用于解决基于知识图谱的问答任务（Knowledge Base Question Answering, KBQA）。数据集由若干个事实-问题对组成，如表3.1所示。

表 3.1　SimpleQuestions 数据集示例

what netflix genre does Thoughtcrimes belong to?
(Thoughtcrimes, genre, drama film)
who was involved in the battle of long island?
(Battle of Long Island, participant, Thomas Mifflin)
Which time zone does strasburg belong to?
(Strasburg, located in time zone, Central Time Zone)
who is the director of dying changes everything?
(Dying Changes Everything, director, Deran Sarafia)
Where was nicholas crafts's place of birth?
(Nicholas Crafts, place of birth, Nottingham)

其中三元组事实里下划线部分为问题答案。原始数据集是基于 Freebase 知识图谱的，共包含 10 万多个问题（见表3.2）。然而 Freebase 早在 2014 年就宣布关闭，并将所有数据都迁移到了 WIKIDATA，所以本论文相关方法都是基于 WIKIDATA 实现的，我们需要将原始 SimpleQuestions 数据集中的实体映射到 WIKIDATA 中。论文[39] 给出了 SimpleQuestions 数据集实体映射到 WIKIDATA 的版本，可以发现，映射之后数据集规模较原始版本缩减了很多，共包含 2 万多个问题（见表3.2）。后续实验都是基于 SimpleQuestions（WIKIDATA）数据集的。

表 3.2　SimpleQuestions 数据集统计信息

数据集版本	训练集	验证集	测试集
SimpleQuestions(Freebase)[38]①	75910	10845	21687
SimpleQuestions(WIKIDATA)[39]②	14695	2013	4249

① 基于 Freebase
② 基于 WIKIDATA

3.4.2　评测标准

本文采用 BLEU[40] 去衡量生成的问题与人工标注的问题之间的相似性，进而评测自动出题算法的性能。BLEU 是 2002 年由 IBM 研究人员提出的用于评测机器翻译质量的一个方法，主要是通过对比机器生成的序列和若干个人工标注的参考序列之间的相似度来衡量机器翻译的质量。计算方法见公式3-2。BLEU 计算公式由两部分组成，一个是 BP，也称短句惩罚（Brevity Penalty），另一个是多元 n-gram 的加权平均准确率，w_n 为 n-gram 的权重，P_n 为 n-gram 准确率。之所以采用多个 n-gram 的加权平均来表示匹配的准确率是因为，如果仅采用 1-gram 词来衡量生成的翻译结果和参考翻译结果之间的相似性，则评价结果往往会受很多停用词的影响而不准确，例如生成的翻译结果中含有很多 in、the 这样的常用词但其翻译质量并不高的时候，如果参考翻译结果中也含有大量的相同停用词，评价结果就会不准确。同样地，如果仅使用较大的 n-gram(一般我们 n 取 1，2，3，4)，例如 4-gram 词来衡量翻译结果的时候，则只有当其和参考翻译结果完全一致时 BLEU 值才会较高，但我们知道翻译本身就没有固定的答案，可以有多种形式，严格等于参考翻译结果显然不能很好地反映翻译的质量。因此 BLEU 用多个 n-gram 的加权平均准确率来评价翻译结果的准确率。具体每个 n-gram 的准确率 P_n 的计算公式中，$Count(n-gram)$ 为某个 n-gram 单词在生成序列中出现的次数，某个 n-gram 单词的 $Count_{clip}(n-gram)$ 值等于 min(它在生成序列中出现的次数，它在所有参考翻译序列中的最大出现次数)。BP 为短句惩罚因子，给定一个参考长度 r，如果机器生成的结果长度 c 小于 r 的时候，就会产生一个惩罚系数，但这并不意味着机器生成的结果越长 BLEU 值越大，因为如果机器生成的结果越长的话，P_n 的分母就会越大，P_n 就会越小。

$$
\begin{aligned}
BLEU &= BP * exp(\sum_{n=1}^{n=N} w_n \log P_n) \\
BP &= \begin{cases} 1 & if\ c > r \\ e^{1-\frac{r}{c}} & if\ c \le r \end{cases} \\
P_n &= \frac{\sum_{C \in Candidates} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in Candidates} \sum_{n-gram' \in C'} Count(n-gram')}
\end{aligned}
\tag{3-2}
$$

3.4.3　基于公开数据集的自动出题对比实验

本实验我们在 SimpleQuestions(WIKIDATA) 数据集上测试我们提出的基于知识图谱的自动出题算法的性能，并和论文[23] 中的方法进行对比。实验结果如表3.3所示。可以看出我们提出的算法 BLEU 值较其他几种算法明显更高，此处其

表 3.3　基于知识图谱的自动出题算法性能比较

方法	BLEU
baseline[23]	31.36
SP Triples[23]	33.27
MP Triples[23]	32.76
SP Tripples TransE++[23]	33.32
MP Triples TransE++[23]	33.28
ours	**40.31**

他几种算法的 BLEU 值来自论文[23]。该论文中用来评测的数据集为 SimpleQuestions(Freebase)，但是鉴于我们的整个实现过程都是基于 WIKIDATA 的，所以只能在 SimpleQuestions(WIKIDATA) 数据集上验证。由表3.2可知两个数据集的规模存在一定的差异。虽然 SimpleQuestions(WIKIDATA) 测试集规模较小，但其训练集相较于 SimpleQuestions(Freebase) 而言更小，而训练集的大小对于基于模版的方法、机器学习算法来说是至关重要的，因此在这种情形下我们的算法相较于其他算法还是有 +6 左右的性能提升，这说明我们提出的基于模版的出题算法对于知识图谱自动出题任务来说是有效的。

表3.4列出了我们在 SimpleQuestions 数据集上生成的几个问题样例，其中 Fact 为知识图谱中的三元组事实，下划线部分为答案，斜体部分为关系谓词；Human 为人工标注的问题；Template 为我们的算法选中的模版问题；ours 为算法最终生成的问题；而 BLEU 则是算法生成的问题与人工标注的问题之间的相似性。通过对 BLEU 值较低 (BLEU < 0.40) 的几个样例进行分析我们发现，大部分都是误判，误判的原因主要有以下两种：

1. 为提高问题的可理解性，额外引入的题干注释信息使得生成的问题和人工标注问题之间的相似度降低，例如表3.4中第 1 个、第 2 个、第 3 个样例都有这样的注释信息。
2. 语言的多样性使得针对同一个事实，可以有多种不同的提问方式，例如表3.4中第 1 个、第 4 个样例。

通过从测试集中随机抽取 20 个负问题样本（BLEU < 0.40）进行分析我们可以发现，80%（16 v.s. 20）的负样本都是由上述原因导致的误判，其中有 17 个问题受到额外注释信息的影响，13 个问题受到语言多样性的影响（一个问题可能受到多个因素的影响）。这也就说明算法生成的问题大部分都是合理、正确的，从而进一步证明了我们提出的基于模版的出题算法在解决知识图谱自动出题任务上的有效性。

表 3.4　基于 SimpleQuestions 数据集生成的问题示例

输入三元组、人工标注问题以及我们选取的模版和生成的问题	**BLEU**
Fact: (the diamond arm, -*screenwriter*-, leonid gaidai) **Human:** Which films has leonid gaidai written? **Template:** what movie did sergei godrov write? **ours:** what movie did leonid gaidai (soviet film director) write?	0.22
Fact: (beatrice hsu, -*cause of death*-, cardiac arrest) **Human:** how did beatrice hsu die? **Template:** how did chitti babu die? **ours:** how did beatrice hsu (taiwan actor) die?	0.71
Fact: (joseph vijay, -*religion*-, catholicism) **Human:** What religion does vijay practice? **Template:** what religion does shilpa shetty practice? **ours:** what religion does joseph vijay (tamil film actor broad term for beliefs associated with several christian churches social reformer) practice?	0.20
Fact: (andy lippincott, -*creator*-, garry trudeau) **Human:** what American cartoonist is the creator of andy lippincott? **Template:** who created the character bizarre? **ours:** who created cartoonist andy lippincott?	0.27
Fact: (velvet sidewalk, -*record label*-, K Records) **Human:** Which label is some velvet sidewalk signed to? **Template:** who label does terry evans work for? **ours:** who label does some velvet sidewalk work for?	0.13

3.4.4　基于 MOOCs 视频字幕的自动出题实验

本实验的目的在于检验我们提出的基于 MOOCs 视频字幕的自动出题模型在真实 MOOCs 视频字幕上自动出题的效果。实验用到的视频字幕有两个，第一个取自学堂在线课程《大学国文——北宋至现代文学赏析》第二十三章《红楼梦》片段赏析，第二个取自学堂在线课程《我国周边 14 个陆地邻国基本情况介绍》第一讲《吉尔吉斯斯坦共和国基本情况介绍》。对于第一个视频，我们限定需要识别的实体类型主要包括文学作品和人物，对于第二个视频，我们限定需要识别的实体类型主要包括国家、城市和人物。

首先以《红楼梦》片段赏析为例来说明和展示从字幕中提取出的知识图谱的可视化效果。图3.4展示了《红楼梦》赏析一节中人物、作品之间的关系图，其中每个节点为一个实体，鼠标悬浮在它上面时可以在旁边显示它的一些基本实体信息，点击时则可以跳转至其对应的维基百科页面，方便学生进一步扩展学习。

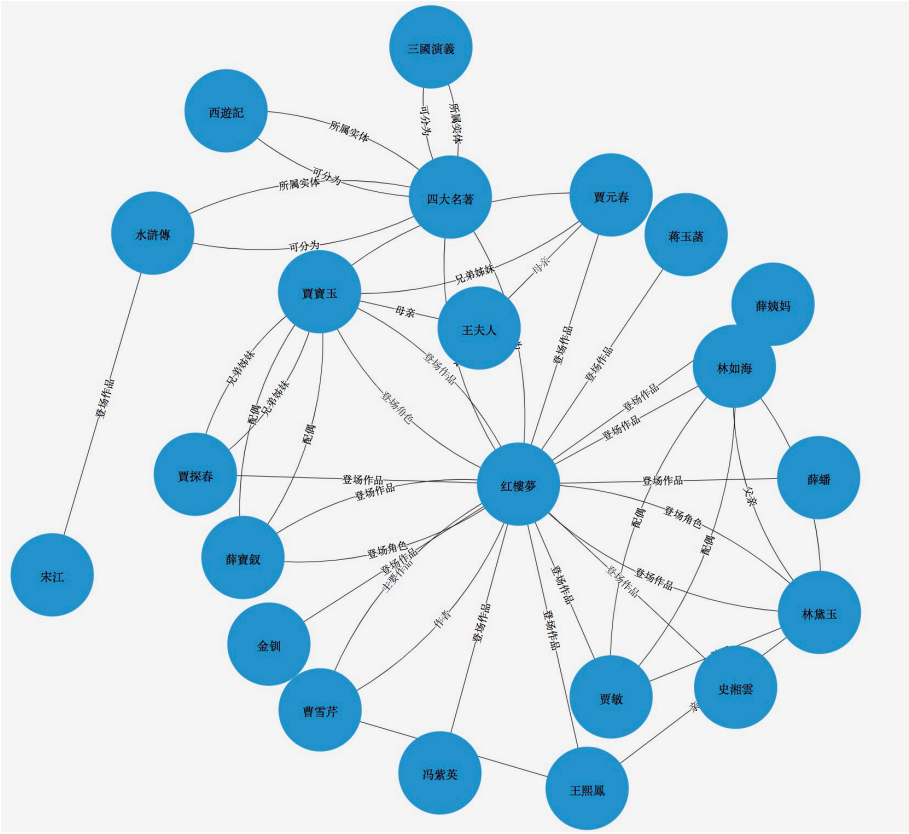


图 3.4　基于 MOOCs 视频字幕的课程知识图谱可视化示例

最终生成的问题示例见表3.5和表3.6，它们分别是从《红楼梦》片段赏析视频以及《吉尔吉斯斯坦共和国基本情况介绍》视频字幕中生成的多选题样例，其中字体加粗的选项代表正确答案。这里需要解释一下为什么课程视频是中文的，但生成的问题却是英文的。原因有两点：(1) 目前没有中文的知识图谱问题模版库；(2) WIKIDATA 是支持多语言的。所以我们可以基于 WIKIDATA 从中文字幕中抽取出知识实体及其相互之间的关系，然后利用 WIKIDATA 获取实体英文信息，进而在已有的英文问题模版库上生成测试题目。实验结果表明，两个视频各自都只有一个样例题干描述有误，同时从《吉尔吉斯斯坦共和国基本情况介绍》视频字幕中生成的问题里，第五个测试样例会因为额外的注释信息而泄漏了正确答案。综合来看可以发现，除少数题干描述不准确外，大部分生成的问题描述都是合理的，且没有题目有选项问题，这也就说明我们提出的基于 MOOCs 视频字幕的自动出题算法是可行且有效的。与此同时，问题实体注释信息的引入在公开数据集上可以帮助我们更好地理解问题题干，但在 MOOCs 视频字幕的自动出题任务中，可能会面临泄漏答案的风险，因此需要有选择性地加以使用。

表 3.5　基于《红楼梦》片段赏析视频字幕生成的多选题示例

多选题示例		标注
Q1: Who is the son of Jia Min?		
A. Jia Baoyu	**B. Lin Daiyu**	题干错误
C. Xue Baochai	D. Jia Yuanchun	
Q2: Which work is a part of Four Great Classical Novels?		
A. Dream of the Red Chamber	B. Natural History	✓
C. Uncle Tom’s Cabin	D. Life, the Universe and Everything	
Q3: What’s a structure designed by Cao Xueqin (Chinese writer during the Qing dynasty)?		
A. Water Margin	**B. Dream of the Red Chamber**	✓
C. Natural History	D. Uncle Tom’s Cabin	
Q4: Who is the author of the book Dream of the Red Chamber?		
A. George Washington	B. Douglas Adams	✓
C. Cao Xueqin	D. Barack Obama	
Q5: Who created Wang Xifeng (character in the classic Chinese novel Dream of the Red Chamber)?		
A. Song Jiang	**B. Cao Xueqin**	✓
C. Barack Obama	D. George Washington	

表 3.6　基于《吉尔吉斯斯坦共和国基本情况介绍》视频字幕生成的多选题示例

多选题示例		标注
Q1: Which country does Kyrgyzstan(country in Central Asia) share border with?		
A. People’s Republic of China	B. North Korea	✓
C. United States of America	**D. Tajikistan**	
Q2: What’s the official language of Kyrgyzstan(country in Central Asia)?		
A. Russian	B. English	✓
C. French	D. Armenian	
Q3: What’s the captical of Kyrgyzstan(country in Central Asia)?		
A. Bishkek	B. Osh	✓
C. Washington, D.C.	D. Zagreb	
Q4: Name a city on Kyrgyzstan(country in Central Asia) in the western pacific ocean?		
A. Osh	**B. Batken Region**	题干错误
C. Chuy Region	D. Alabama	
Q5: What’s the higher classification of osh region(region of Kyrgyzstan)?		
A. Tashkent	**B. Region of Kyrgyzstan**	泄漏答案
C. Zagreb	D. Buenos Aire	

3.5　小结

本章详细介绍了如何从 MOOCs 视频字幕中自动生成测试题目。算法借助维基知识图谱 WIKIDATA 先对字幕文本进行结构化处理，从中提取出感兴趣的知识实体及其之间的关系，构建课程知识图谱，然后基于知识图谱再去用基于模版的方法自动生成测试题目。实验部分有两个，一是用公开数据集 SimpleQuestions 对我们提出的自动出题算法的性能进行测试，实验结果表明，我们提出的自动出题算法可以有效提高最终得到的问题的准确性和可理解性。第二个实验以《红楼梦》片段赏析和《吉尔吉斯斯坦共和国基本情况介绍》两个真实的 MOOCs 视频为例来检验我们的自动出题算法出题的效果，实验结果表明，大部分生成的题目都符合预期，仅有少量题目题干描述不准确。与此同时，课程知识图谱的可视化结果也可以清晰、直观、准确地反映出知识实体之间的关系，符合我们最初的预期。

第 4 章　基于 MOOCs 学习数据的自动组卷模型

4.1　引言

本章主要介绍如何以学生为主体，根据其过去一段时间的学习表现，自适应地从题库中为其抽取难度合适的题目组合。这种根据学生能力来自适应地生成不同的测试试卷的做法主要有以下两点好处：

1. 我们知道对于每一次测验，只有当题目整体难度系数和考试的人水平差不多的时候，检测效果才比较好，太难或者太简单的题目都不能很好测试出答题的人的知识水平。但以往，测试题目的难度一般是由老师或者课题组的几个老师来决定，这对老师的经验和知识水平有很高的要求。我们根据学生的学习表现来自适应地决定测试难度水的做法，相较于传统方法更加合理、科学。
2. 可以实现个性化学习过程，改善学生的 MOOCs 学习体验。

图4.1展示了基于 MOOCs 学生平时数据自动从题库中选题的整体流程。从图中可以看出，整个流程包括两个部分：首先需要根据学生的平时数据对学生的学习效果进行预测，进而调整选题约束中的平均难度系数；然后在选题约束的指导下，用合适的策略从试题库中选出符合要求的题目组合用于 MOOCs 课程的日常测验和考试。

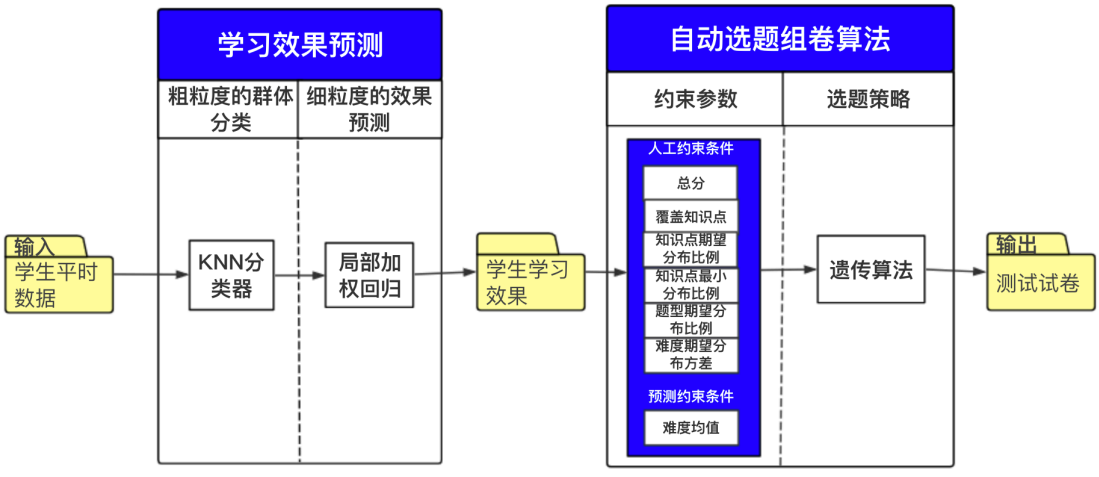


图 4.1　基于 MOOCs 学生平时数据的自动组卷流程图

本章接下来会对流程图4.1中的每一个模块进行详细介绍，第4.2节介绍如何根据 MOOCs 学生的平时学习表现预测其学习效果；第4.3节介绍如何从试题库中找

到符合约束的题目组合；第4.4小节为实验部分，会基于真实的 MOOCs 课程数据分别检验预测模型和组卷模型的性能。

4.2　基于群体划分的 MOOCs 学习效果预测模型

预测问题是机器学习算法最善于解决、也最常被用来解决的一种问题。一般处理模式为：首先进行特征提取，这可以由机器自己完成（例如神经网络就是机器自己提取特征的一个过程），也可以人工进行设计；然后构造训练集及特征到待预测特征之间的映射模型，并根据训练集中的数据进行模型参数学习；最后针对给定的测试样例，我们将和它相关的特征输入到上一步训练得到的模型中，即可得到最终的预测输出。

4.2.1　特征选择

在 MOOCs 学习效果预测问题中，我们首先需要将学习效果进行量化作为待预测特征。期末考试是传统课程用来评测学生一学期的学习效果最常用的指标，因此我们也选择用期末考试成绩来量化学习效果。一般常用的 MOOCs 学生特征包括：

1. 性别、工作、教育程度、婚姻状态等人口学特征（demographic feature）；
2. 学生平时的课程活动参与特征，例如是否看过课程视频、看视频学习的时间、频率特征、讨论区发帖数量、回帖数量等；
3. 学生平时的作业成绩特征。

虽然人口学特征、课程活动参与情况特征都与学生的学习效果有关，但很显然，和期末成绩最相关的还是学生的平时作业成绩。同时人口学特征的可信度没有什么保证，很多注册学生也没有提供相关信息，因此前期为了简化整个预测模型，我们仅选取和期末成绩最相关的平时作业成绩特征，去预测学生的学习效果。

4.2.2　预测模型

因为成绩都是连续的值，所以我们选择用回归模型来预测学生学习效果。本论文中，我们共选取了五种不同的常见回归模型，并在实验部分4.4.2对比了各个模型的效果。这里我们选择的五种回归模型分别为：

- **M5 模型树：** M5 模型树[41] 是一种分段线性回归的树模型。模型树的构建过程是一个递归构建的过程，递归的终止条件为：某一个结点的样本数足够少或者根据公式4-1计算出的父节点和孩子节点的目标值标准差变化低于某个阈值。如果某个结点的数据集不满足终止条件，则我们需要根据某一特征

值将数据集划分为两个子集，然后分别在两个子集上继续递归构建对应的子树。这里选择用哪个特征、怎么划分的决策过程中，我们需要枚举所有可能的特征划分方式，然后基于公式4-1找到最优的划分方法，也就是标准差变化最大的划分方式。其中 T 为父节点数据集，T_i 为根据某一特征属性划分之后的其中一个子集，sd 为集合预测目标值的标准方差。

$$\Delta error = sd(T) - \sum_i \frac{|T_i|}{T} sd(T_i) \tag{4-1}$$

模型树构建完成之后，针对每个叶子结点需要用最小二乘法得到对应的线性拟合函数，进入某一叶子结点的样例需用其对应的线性函数得到预测值。为避免过拟合和预测函数的不连续性，一般还需要进行剪枝及平滑操作。

- **支持向量回归（SMOreg）：**它的基本思想和支持向量机分类一致，都是找一个决策面。在回归模型中，模型的损失函数包括两部分，一个是正则项，一个是所有数据点到决策面的距离，当距离小于某一阈值时可以忽略不计。支持向量机分类和回归模型的参数求解问题是一个凸二次规划问题，我们选择用 SMO（Sequential minimal optimization）算法[42] 进行求解，因此后文中我们用 SMOreg 来表示支持向量回归模型。
- **线性回归（LR）：**线性回归（Linear Regression）是一种最为常见的简单回归模型。假设预测值为 y，特征为 $x = (x_1, x_2, \ldots, x_n)$，模型为 $y = \theta\mathbf{x} + b = \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n + b$。参数求解使用最小二乘法，也就是找到使 $Loss = \sum_i (\theta_1 x_{i1} + \ldots + \theta_n x_{in} + b - y)^2$ 最小的 θ_i 和 b 作为最终的模型参数。最小二乘法可以直接得到解析解，即方程组

$$\begin{cases} \frac{\partial Loss}{\partial \theta_1} = 2\sum(\theta_1 x_{i1} + \theta_2 x_{i2} + \quad \ldots + \theta_n x_{in} + b - y)x_{i1} = 0 \\ \frac{\partial Loss}{\partial \theta_2} = 2\sum(\theta_1 x_{i1} + \theta_2 x_{i2} + \quad \ldots + \theta_n x_{in} + b - y)x_{i2} = 0 \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\ \frac{\partial Loss}{\partial \theta_n} = 2\sum(\theta_1 x_{i1} + \theta_2 x_{i2} + \quad \ldots + \theta_n x_{in} + b - y)x_{in} = 0 \\ \frac{\partial Loss}{\partial b} = 2\sum(\theta_1 x_{i1} + \theta_2 x_{i2} + \quad \ldots + \theta_n x_{in} + b - y) = 0 \end{cases} \tag{4-2}$$

的解。

- **局部加权回归（LWR）：** 局部加权回归（Locally Weighted Regression）是在线性回归的基础上，给每个数据点增加权重，从而使得损失函数转变为 $Loss = \sum_i w_i(\theta x - y)^2$，这里 w_i 为第 i 个数据点的权重，权重越大，则该数据点的预测结果对于最终模型影响越大，反之权重越小，则该数据点对结果的

影响就越小。这里权重 $w_i = exp(-\frac{(\mathbf{x_i}-\mathbf{x})^2}{2\tau^2})$，参数的计算方式同线性回归一样，用最小二乘法计算得到。

- **BP 神经网络（BP）：**将特征作为输入层，经过若干个全连接层（线性变换加非线性激活函数）后，输出预测目标值。损失函数为数据点预测值和真实值之间的平方差的和。然后利用反向传递算法（Back Propagation）得到损失函数对各个参数的梯度，并利用梯度下降法进行参数更新，直至找到最优解。

4.2.3　基于群体划分和数据过滤的学习效果预测改进

实验4.4.2部分的原始学习效果预测实验结果表明，直接从平时成绩预测学生期末考试成绩的话，预测效果较差。基于此，我们提出了两点改进方法：

- **学习群体划分**。第一个改进是将学生按学习效果划分为两个不同的群体，通过与不通过，然后对不同的学习群体构建不同的预测模型。这样整个预测过程就分两步进行，首先根据学生平时成绩进行粗粒度的群体预测，然后用预测群体对应的细粒度的成绩预测模型预测其学习效果。这么做的动机是因为学习效度不同的群体，平时成绩与期末成绩之间的关系是不一样的，各种难度的题目对学生的区分能力以及对预测模型的贡献都是不一样的，如果将其视为一个整体的话，得到的映射关系需兼顾两个群体各自的规律，关系较复杂，致使最终预测结果误差较大。在实验过程中我们可以发现，在第一步群体预测结果完全正确的情况下，整体平均预测误差可普遍降至 10 分左右，原始实验预测误差普遍在 20 分左右（百分制的情况下），这也就说明群体划分对于学习效果预测模型是有效的；然而在第一步预测结果有误差的情况下，算法整体平均预测误差非一致降低，且降低幅度不是很大。通过观察我们可以发现，群体划分确实可以使第一步预测准确的个体第二步预测误差普遍降低至 10 分左右，然而第一步如果预测错误的话，第二步的预测误差相较于直接预测会被放大，致使整体平均误差非一致降低。与此同时，我们可以发现第一步群体预测错误的样例大部分都是躁点，也就是平时成绩和期末成绩严重不符的样例。因此为提高预测模型的准确性，我们需要对数据进行过滤清理。
- **数据过滤**。图4.2以学堂在线《组合数学》课程为例，展示了 MOOCs 学生平时成绩和期末考试成绩之间的分布关系图。可以发现数据存在很多期末成绩与平时成绩严重不符的躁点，包括平时成绩非常高但期末考试成绩较低、以及平时成绩非常低但期末考试成绩比较高的学生样例。这些数据躁点的存在会严重影响预测模型的准确性，因此需要将这部分数据进行过滤，具体过滤

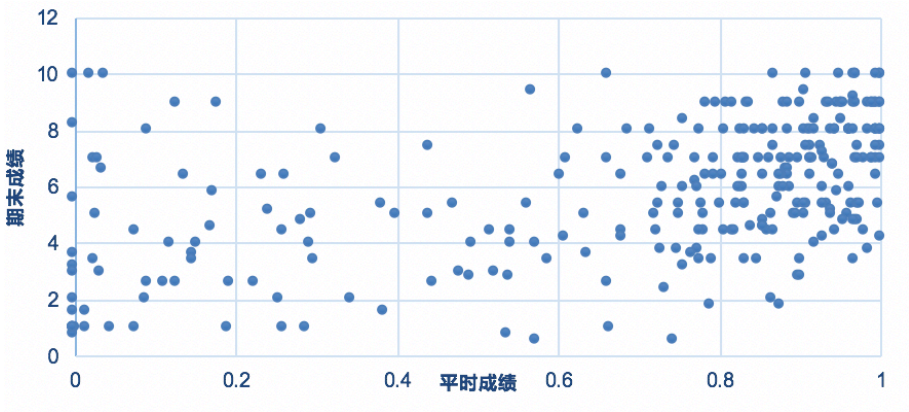


图 4.2　MOOCs 学生平时成绩和期末成绩之间的分布关系图

方式是：按期末成绩通过与否将学生划分为两个群体，并根据每一个群体学生平时成绩的均值设置阈值进行过滤，对于不及格的学生来说，需过滤掉那些平时成绩高于均值的样例；对于及格学生来说，则需过滤掉那些平时成绩低于均值的样例。

4.3　基于遗传算法的自动组卷模型

4.3.1　约束目标

如图4.1所示，我们的选题约束目标主要涉及四个维度：试题总分、题型、知识点以及难度约束。除难度约束外，其余三个约束目标都由教师或者学生人工指定，对于题型和知识点，需指明期望包括的类型及相应的占分比例，每个知识点还需额外给出最小覆盖比例，这是因为一些知识点是必考项，但如果其期望的占分比例较小的时候，最优解很有可能是完全不包含该知识点的，显然这不符合我们的预期。

难度约束包括两个指标，一是试卷平均难度，二是各个难度等级的题分分布。平均难度需要根据学习效果预测结果来决定，学习效果在这里即学习成绩。我们知道学生的测试成绩一般是正态分布的（或者说考试成绩基本是正态分布时，测试才是有效、符合预期的）。因此我们可以用不同成绩段人数的比例来表示相对应难度的考题在试卷中所占的比例。比如试卷共包含 5 个难度级别的题，试卷总分为 100 分，则可以用 0-20 分的人数比例来代替最高难度的考题占分比例，20-40 分次之，依次类推，80-100 分的人数比例来代替最简单的题目占分比例。正态成绩分布的均值为预测的学习效果，方差由学生或老师人工决定。

4.3.2　遗传算法设计

本文采用遗传算法来完成从试题库中自动选题的过程，具体流程如图4.3所示。首先需要定义解空间并进行编码，这里的解空间就相当于自然进化的主体-生物，编码相当于基因。然后我们需要定义解空间上的遗传、交叉、变异操作，这是基因发生变化的原因所在，也是生物进化、自然选择的本质原因。最后我们需要定义适应函数，也就是自然法则，从而对解空间的个体进行淘汰，指导生物进化、也就是搜索的方向。

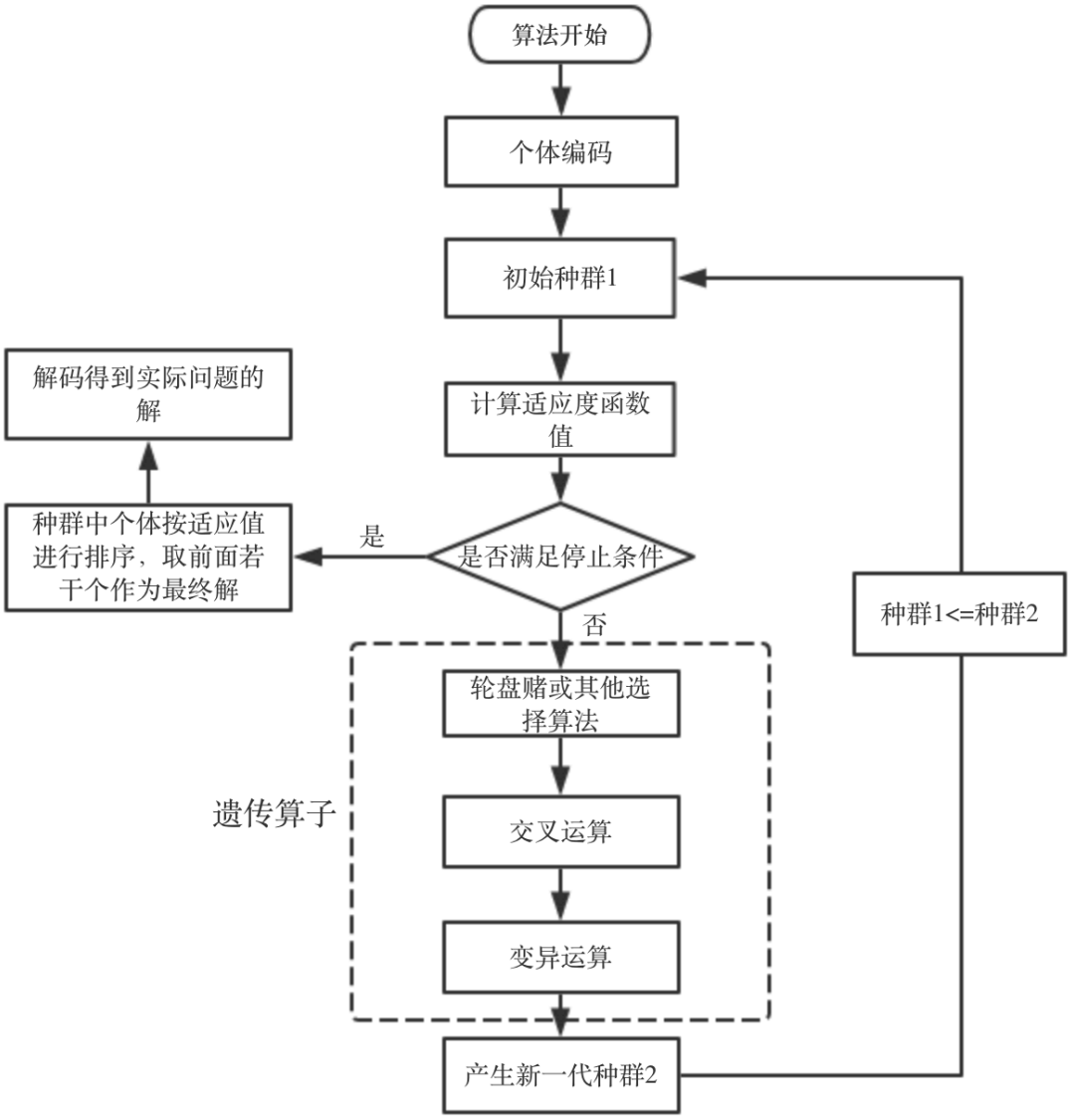


图 4.3　基于遗传算法的自动组卷流程图

- **编码：**我们将试题组合称为试卷，并将试卷作为遗传算法的解个体，解空间就是试题库所有的题目组合形式（一般来说是极其大的）。每个试卷的编码

为一个长为 N 的 01 字符串，其中 N 为试题库题目总数，第 i 位为 0 代表试题库第 i 道题未被选择，1 则代表选择。为方便之后的设计，我们将试题库的题目按题目类型进行排序，例如最前面的是选择，接下来是填空，依次类推。这样反映在编码上就是编码是按题型分段的。

- **初始群体：**这里假定各个题型的题目分数是一样的，则可以直接根据总分约束和题型约束，计算得到每种题型的题应该抽取多少道。这种简化的意义在于可以极大地降低组卷失败的可能性。只要试卷中各个题型的题目总数是够的，算法一定可以成功返回一份总分和题型满足约束的试卷，区别仅在于知识点覆盖程度和难度分布比例是否满足我们的预期。基于该假设，我们在每种题目题数的约束下从试题库中随机生成若干份试卷作为初始群体。
- **适应函数：**每一份试卷的适应函数为其难度分布比例、知识点分布比例与其相应的期望分布比例之间的差值加权求和的倒数，如公式4-3所示。

$$f = \frac{1}{w_d|D - D_{target}| + w_k|K - K_{target}| + w_{ks}\sum_i max(KS^i_{target} - K^i, 0) + C} \tag{4-3}$$

其中，w_d 为难度约束的权重，w_k 为知识点约束的权重，此处均设为 1，w_{ks} 为知识点最小占分比约束的权重，此处设为一个较大的值。D 为试卷的难度分布比例，D_{target} 为难度期望分布比例；K 为试卷知识点分布比例，K_{target} 为知识点期望分布比例；K^i 为第 i 个知识点占分比，KS^i_{target} 为第 i 个知识点期望最小占分比，我们希望所有知识点占分比都能严格大于其对应的期望最小占分比，所以需要将不符合情况的试卷适应函数设的非常小才可以。这里如果 $K^i < KS^i_{target}$，则因为 w_{ks} 特别大会导致适应函数值非常小。C 为常数。

- **遗传算子：**遗传算子依次包括选择、交叉和编译三个操作。父代群体经过这些操作之后，会生成数量与其一致的子代群体，然后重复该过程直至算法触发终止条件。
  - **选择操作**。首先定义父代群体中每个个体能被保留到子代群体的概率，它和个体的适应函数值有关，值越大，保留概率越大。所有个体的保留概率之和为 1。根据选择概率，随机选取若干个个体直接复制到下一代个体中。
  - **交叉操作**。根据个体的适应函数值，依次从父代群体中选取“父”“母”两个个体进行交叉，个体适应函数值越大，选中交叉的概率则越大。具体的交叉过程如图4.4所示。随机在父代个体的编码上选择一个位置（图中红色 1 的位置），找到该位置对应的题型的边界，并交换边界之后两

个父代个体的编码信息。例如题库总共 100 道题，也就是说编码长为 100，共三种题型，1-30 为题型一、30-60 为题型二、60-100 为题型三，如果随机选取的位置为 35，则该位置对应的题型边界为 60，交换两个父代个体 60 位之后的编码信息。

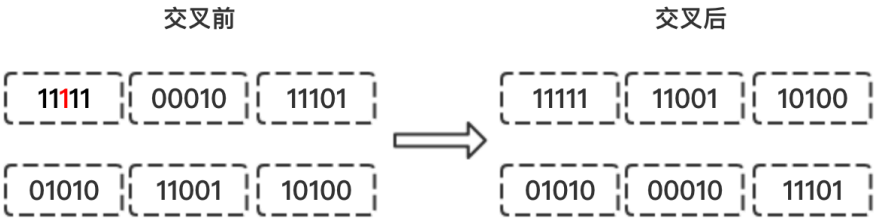


图 4.4　交叉算子

– **变异操作**。对于交叉得到的每个个体，随机从其编码中选取一个位置，以一定的变异概率更改该位置的编码值。这里变异概率一般较小。如果确定发生变异的话，则需要从该位置对应的题型范围内随机选取另一个和它变异之后编码一样的位置，更改该位置编码值，也就是说如果该位置原编码为 1，则变更之后为 0，需要在其对应的题型范围内随机找到一个编码为 0 的位置，将其值变更为 1。这么做的目的是为了不破坏题型约束。

• **终止条件：**这里我们的终止条件有两个：一是连续 10 次最优解适应函数值无变化；二是迭代次数达到预定义上限。

4.4　实验

4.4.1　数据集

预测的训练数据集来源于清华大学的研究生课程《组合数学》，因为它在 edX 和学堂在线都有开设相应的 MOOC 课程，且开放的测试题不完全一样，所以我们用两个平台上的数据独立进行实验，其中 edX 上的平时特征包括 26 个 quiz 成绩和 8 的 homework 成绩；学堂在线上的平时特征包括 25 个 quiz 成绩和 8 个 homework 成绩。

4.4.2　学习效果预测实验

在这一部分，我们首先对比了常见的分类器在 MOOCs 学生平时成绩到是否及格的预测分类问题上的准确率，参与对比分析的分类器有：决策树、k 近邻（kNN）、

朴素贝叶斯（NB）、随机森林（RF）及支持向量机（SVM)。决策树我们采用的是 J48 算法，支持向量机的目标优化函数采用 SMO 算法实现。实验结果见表4.1。

表 4.1　基于 MOOCs 学生平时成绩的群体预测实验结果

MOOCs 平台	J48 (剪枝)	J48 (未剪枝)	1-NN	3-NN	5-NN	NB	RF	SVM
edX	79.37	80.00	82.22	79.37	76.83	71.11	83.49	76.83
学堂在线	80.07	80.76	88.66	77.66	77.66	70.10	87.63	78.70

实验结果表明，随机森林和最近邻算法性能更好，因此基于群体划分的学习效果预测实验中，我们采用 1NN，最近邻算法来做群体预测。在学习效果预测实验中，我们对比了有无群体划分和数据过滤的各个回归预测模型的性能。评价指标采用的平均绝对误差，计算方式为 $MAE = \frac{1}{n}\sum_{i=1}^{i=n}|y_i - \hat{y}_i|$，其中 y_i 为真实值，$\hat{y}_i$ 为预测值，实验结果见表4.2。

表 4.2　基于 MOOCs 学生平时成绩的学习效果预测实验结果

MOOCs 平台	预测方法	M5 模型树	SMOreg	LWR	LR	BP
edX	直接预测	18.403	19.710	19.802	19.352	34.598
	群体划分 + 数据过滤	12.854	13.208	11.779	13.891	16.4
学堂在线	直接预测	23.803	23.136	23.511	22.912	33.414
	群体划分 + 数据过滤	11.284	12.432	10.475	12.161	13.717

可以发现，五种回归算法除 BP 神经网络之外，其余四种算法性能差别不大；加入群体划分和数据过滤后，可以有效地降低预测误差，其中局部加权回归预测误差最小，百分制的情况下，误差在 11 分左右。

4.4.3　自动组卷实验

本实验旨在验证我们设计的自动组卷算法的性能。当前 MOOCs 的试题库一般题量都不大，且试题没有必要的标记信息，包括题型、涉及到的知识点、难度、分值。而基于遗传算法的自动组卷算法一般是在题库较大时才会有明显的优势，构建真实试题库时间、人力耗费较大，所以本实验是在模拟试题库上进行实验的。模拟试题库共包含 3000 道题，覆盖 10 个知识点、4 种题型（选择、填空、判断、问答）、5 个难度等级（简单、较简单、一般、较难、难），每道题的标注由一个随机

程序完成，各个题型的分数一样。现以该模拟试题库上的一次组卷过程为例来展示我们的组卷效果：

- **总分：**目标为 100 分，实际结果为 100 分；
- **题型：**目标为 [选择：填空：判断：问答] = [10%: 20%: 40%: 30%]，实际结果和目标约束完全一样。
- **难度：**目标为 [简单: 较简单: 一般: 较难: 难] = [16%:30%:30%:6%:8%]，实际结果为 [13%:33%:31%:13%:10%]
- **知识点：**目标约束为需覆盖前四个知识点，且期望分布为 [10%:20%:30%:40%]，每个知识点最小占分比例为 [5%, 10%, 15%, 20%]，实际结果为 [9%:18%:28%:45%]。

实验结果表明，最终得到的试题组合各项指标基本符合最初的约束条件，总分和题型在我们的算法设置下是和目标约束完全一致的。整个选题过程可以在很短的时间内完成，由此证明我们设计的组卷算法是可行且有效的。

4.4.4　基于 MOOCs 学生平时数据的预测组卷综合实验

我们本章研究的内容是基于 MOOCs 学生平时成绩去自动从题库中生成一份难度和他平时学习效果相当的试卷。前面两个小实验只是单独就学习效果预测准确率和选题组卷算法的有效性进行验证，并没有将两者结合起来来验证我们整个模型的效果。一个可行的验证方法是：从学期初开始跟踪一部分学生的学习过程，并将其平时数据记录下来，然后在学期末对于其学习效果进行预测，并根据预测结果从题库中用我们提出的组卷算法生成一份测试试卷来对学生进行测验，如果学生最终的平均考试成绩和最开始我们的预测结果是差不多的，则符合实验预期。但这个方法在实际完成过程中有一定的难度和挑战：1. 时间跨度较大；2. 没有真实的大规模标注试题库。因此最终我们想到的一个折衷且可行的解决方法是：把所有平时测试用的题目整合在一起，作为伪考试题库，对于每一道题来说，学生考试的得分用他平时作业里该题的得分来替代。实验步骤为先用局部加权回归模型从学生的平时数据出发得到他的学习效果预测值，然后据此从考试题库（平时测试题库）中自动抽取一部分题，将学生平时的得分加起来得到他的“期末考试成绩”，最后对“期末考试成绩”和预测结果进行对比。实验结果见表4.3。

实验结果表明，用基于 MOOCs 学生平时成绩生成的测试试卷去考 MOOCs 学生，其考试成绩和我们预测的分数是很接近的，误差基本不超过 10 分（百分制）。与此同时，当测试人数较多的时候，两者之间的误差会更小，这也是符合预期的，因为对于单个人的预测来说，预测误差一般有正有负，这样人越多，平均预测误

表 4.3　基于 MOOCs 学生平时数据的组卷实验结果

测试人数	预测分数	期末考试成绩
17	77.59	75.08
16	79.75	71.37
13	73.91	69.23
12	75.94	61.14
6	82.48	69.63

差就更容易互相抵消，准确率就会更高。

4.5　小结

本章主要介绍基于 MOOCs 学生平时数据的自动组卷算法。我们先用回归预测模型对学生的学习效果进行预测，进一步指导选题组卷的目标约束，最终通过遗传算法完成组卷任务。实验结果表明，群体划分和数据过滤可以有效地降低学习效果预测误差，且自动组卷算法可以在总分、难度、知识点、题型等约束下成功完成组卷任务。

第 5 章　总结与展望

本文设计并实现了一套完整的基于 MOOCs 视频字幕和学生平时学习数据的自动评测模型。该模型可以极大地降低老师在评测反馈阶段的劳动量，包括出题和组卷两个过程。与此同时，由学生平时学习数据驱动的自动组卷过程可以解决传统出卷过程中试卷质量完全取决于老师的经验和知识水平的问题。试卷质量主要由试卷整体难度分布和知识点覆盖率来决定。一次好的考试，我们期望的是能尽量包括所有学过且需要掌握的知识点，且越重要的知识点占分比例越高。传统出卷过程中对知识点的控制以直观估计为主，并不精确到具体的占分比例，我们提出的自动出题算法将这些约束目标都进行了量化，从而使得最终得到的试卷参数更加科学、合理。在试卷难度方面，首先我们需要知道试卷难度对于一次考试的重要性。试卷太难，所有人都不会做，试卷太简单，大家分数都很高，显然这两种都不是符合预期的，也就是说考试试卷难度得适中才比较好。可是难易是因人而易的，传统组卷过程中，教师必须根据以往经验和当前学生平常的表现来决定试卷难度，这对于教师来说是一件很有挑战的事情。我们提出的自动组卷模型可以用机器模拟老师的整个决策过程，基于以往数据学习学生平时成绩和期末成绩之间的关系，然后根据当前学生的平时成绩预估其考试成绩，从而决定试卷整体难度。

在基于 MOOCs 视频字幕的自动出题过程中，考虑到字幕文本口语化、信息冗余、无断句等特征，我们借助了维基知识图谱来对其进行知识提取，之后的出题过程也是直接基于提取到的课程知识图谱。这么做的好处有两点：

- 维基知识图谱的引入，使得课程内容更加趋于结构化，同时也为其他一些应用提供了可能，譬如课程内容纠错，课程助手问答系统等。
- 降低了出题任务的难度，基于结构化的知识图谱去出题显然要比从质量不高的字幕文本中出题容易的多。

但同样，这么做也会引入很多问题，其中影响最严重的是生成的题目很有可能和字幕内容相关性比较低。例如第3章的实验部分，课程其实主要是片段赏析，讲发生在人物之间的故事，而我们基于图谱得到的题目却大多是人物之间的关系问题、作品和人物的对应问题。出现这个问题的主要原因是可结构化表示的知识是有限的，很多知识难以用结构化语言来表达。虽然我们最终生成的题目也是合理有效的，但很有可能是不需要看视频就已经知道的内容，这就起不到测验的目的了。因此为了能够增强生成的题目和视频内容的相关性和题目的难度，未来的在

自动出题方面，我们还是需要去研究如何直接从字幕中生成高质量的题目。当然我们也可以考虑不用字幕，直接基于视频去出题，一个可行的方案就是：MOOCs 视频中大部分都是 PPT，或者是 PPT 和人，PPT 中的知识一般是用较为简练的文字表述的。因此我们可以利用文字检测和识别技术将视频中的文字识别出来，然后基于 PPT 文字去出题，比如可以直接将图片中的关键字抠掉，问“下面图片中空白部分应该填入什么?”这种问题。

在基于 MOOCs 学生平时成绩的自动组卷任务中，我们用平时作业成绩去预测学生学习效果，然后根据学习效果决定试卷难度系数，并用遗传算法从题库中找到一组符合约束要求的题目组合。当前阶段，我们仅以跑通整个流程为主，很多地方都做了简化，比如特征仅考虑了平时成绩，模型也用的是简单常见的分类回归器，这里模型简单主要是因为特征较少，未来我们可以考虑引入更多的学生特征来提高整个预测模型的准确率。

综上，本论文设计了一套基于 MOOCs 视频字幕和学生学习数据的自动评测模型，并用实验证明了模型的有效性和可行性。但在具体的一些实现细节中，模型做了一定的简化和假设，未来我们的主要工作是改进现有的一些实现细节，以提高算法整体的性能。同时针对当前出题模型中题目和视频内容相关性较低的问题，未来会继续探索更多可行的适用于 MOOCs 的出题策略。

参考文献

[1] Suen H K. Peer assessment for massive open online courses (moocs)[J]. The International Review of Research in Open and Distributed Learning, 2014, 15(3): 312–327.

[2] Thai N T T, De Wever B, Valcke M. The impact of a flipped classroom design on learning performance in higher education: Looking for the best "blend" of lectures and guiding questions with feedback[J]. Computers & Education, 2017, 107: 113–126.

[3] Daradoumis T, Bassi R, Xhafa F, et al. A review on massive e-learning (mooc) design, delivery and assessment[C]//International conference on P2P, parallel, grid, cloud and internet computing. IEEE, 2013: 208–213.

[4] 韩锡斌, 翟文峰, 程建钢. cmooc 与 xmooc 的辨证分析及高等教育生态链整合[J]. 现代远程教育研究, 2013, 6(4): 3–10.

[5] Onah D F, Sinclair J, Boyatt R. Dropout rates of massive open online courses: behavioural patterns[J]. International Conference on Education and New Learning Technologies, 2014, 1: 5825–5834.

[6] Balakrishnan G, Coetzee D. Predicting student retention in massive open online courses using hidden markov models[D]. Electrical Engineering and Computer Sciences, University of California, Berkeley, 2013.

[7] Greene J A, Oswald C A, Pomerantz J. Predictors of retention and achievement in a massive open online course[J]. American Educational Research Journal, 2015, 52(5): 925–955.

[8] Jing X, Tang J. Guess you like: course recommendation in moocs[C]//Proceedings of the International Conference on Web Intelligence. ACM, 2017: 783–789.

[9] Dernoncourt F, Taylor C, O' Reilly U M, et al. Moocviz: A large scale, open access, collaborative, data analytics platform for moocs[C]//NIPS workshop on data-driven education. 2013.

[10] Balfour S P. Assessing writing in moocs: Automated essay scoring and calibrated peer review. [J]. Research & Practice in Assessment, 2013, 8(1): 40–48.

[11] Heilman M. Automatic factual question generation from text[D]. Language Technologies Institute School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2011.

[12] Rajpurkar P, Zhang J, Lopyrev K, et al. Squad: 100,000+ questions for machine comprehension of text[C]//Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016: 2383–2392.

[13] Heilman M, Smith N A. Question generation via overgenerating transformations and ranking [R]. Technical Report CMU-LTI-09-013, Language Technologies Institute, Carnegie Mellon University, 2009.

[14] Heilman M, Smith N A. Good question! statistical ranking for question generation[C]//Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010: 609–617.

[15] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[16] Schuster M, Paliwal K K. Bidirectional recurrent neural networks[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673–2681.

[17] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C]// Eleventh annual conference of the international speech communication association. 2010: 1045–1048.

[18] Kumar V, Ramakrishnan G, Li Y F. A framework for automatic question generation from text using deep reinforcement learning[J]. arXiv preprint arXiv:1808.04961, 2018.

[19] Zhou Q, Yang N, Wei F, et al. Neural question generation from text: A preliminary study[C]// National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017: 662–671.

[20] Wang Z, Lan A S, Nie W, et al. Qg-net: a data-driven question generation model for educational content[C]//Proceedings of the Fifth Annual ACM Conference on Learning at Scale. ACM, 2018: 7.

[21] Du X, Shao J, Cardie C. Learning to ask: Neural question generation for reading comprehension [J]. arXiv preprint arXiv:1705.00106, 2017.

[22] Kim Y, Lee H, Shin J, et al. Improving neural question generation using answer separation[J]. arXiv preprint arXiv:1809.02393, 2018.

[23] Serban I V, García-Durán A, Gulcehre C, et al. Generating factoid questions with recurrent neural networks: The 30m factoid question-answer corpus[J]. arXiv preprint arXiv:1603.06807, 2016.

[24] Reddy S, Raghu D, Khapra M M, et al. Generating natural language question-answer pairs from a knowledge graph using a rnn based question generation model[C]//Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: volume 1. Association for Computational Linguistics, 2017: 376–385.

[25] Liu T, Wei B, Chang B, et al. Large-scale simple question generation by template-based seq2seq learning[C]//National CCF Conference on Natural Language Processing and Chinese Computing. Springer, 2017: 75–87.

[26] Elsahar H, Gravier C, Laforest F. Zero-shot question generation from knowledge graphs for unseen predicates and entity types[J]. arXiv preprint arXiv:1802.06842, 2018.

[27] Olney A M, Graesser A C, Person N K. Question generation from concept maps[J]. Dialogue & Discourse, 2012, 3(2): 75–99.

[28] Narendra A, Agarwal M, et al. Automatic cloze-questions generation[C]//Proceedings of the International Conference on Recent Advances in Natural Language Processing. 2013: 511–515.

[29] Fattoh I E. Automatic multiple choice question generation system for semantic attributes using string similarity measures[J]. Computer Engineering and Intelligent Systems, 2014, 5 (8): 66–73.

[30] Papasalouros A, Kanaris K, Kotis K. Automatic generation of multiple choice questions from domain ontologies.[C]//International Conference on e-Learning. Citeseer, 2008: 427–434.

[31] 王宇颖, 侯爽, 郭茂祖. 题库系统试卷自动生成算法研究[J]. 哈尔滨工业大学学报, 2003, 35(3): 342–346.

[32] 胡维华, 梁荣华, 江虹. 多目标选题策略研究与应用[J]. 杭州电子科技大学学报 (自然科学版), 1999(2): 36–41.

[33] Song W. Online test paper composition based on genetic algorithm[C]//2018 3rd International Conference on Modelling, Simulation and Applied Mathematics (MSAM 2018). Atlantis Press, 2018.

[34] LU Y, LIU H. Auto-generating examination papers based on integer coding and adaptive genetic algorithm[J]. Computer Engineering, 2005, 31(1): 232–233.

[35] File:datamodel in wikidata.svg[EB/OL]. Wikimedia Commons, the free media repository, 2017. https://commons.wikimedia.org/w/index.php?title=File:Datamodel_in_Wikidata.svg&oldid=244865769.

[36] Mikolov T, Chen K, Corrado G, et al. Efficient estimation of word representations in vector space[J]. arXiv preprint arXiv:1301.3781, 2013.

[37] Nielsen F Å. Wembedder: Wikidata entity embedding web service[J]. Computer Research Repository CoRR, 2017, abs/1710.04099.

[38] Bordes A, Usunier N, Chopra S, et al. Large-scale simple question answering with memory networks[J]. arXiv preprint arXiv:1506.02075, 2015.

[39] Diefenbach D, Tanon T, Singh K, et al. Question answering benchmarks for wikidata[C]// Proceedings of the 16th International Semantic Web Conference. 2017.

[40] Papineni K, Roukos S, Ward T, et al. Bleu: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, 2002: 311–318.

[41] Quinlan J R, et al. Learning with continuous classes[C]//Proceedings of the 5th Australian Joint Conference on Artificial Intelligence: volume 92. World Scientific, 1992: 343–348.

[42] Platt J. Sequential minimal optimization: A fast algorithm for training support vector machines [R]. Technical Report MSR-TR-98-14, Microsoft Research, 1998.

致　谢

衷心感谢我的导师马昱春老师在硕士三年对我在生活和学习方面的精心指导。感谢 EDA 实验室的全体老师和同学们对我在论文写作以及读研期间的帮助和支持，特此致谢。感谢高级机器学习课、数据挖掘课等所有学分课老师的教导，你们传授的知识将使我终生受益。另外，本课题承蒙国家自然科学基金资助，特此致谢。

声　明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签　名：_____ 日　期：_____

个人简历、在学期间发表的学术论文与研究成果

个人简历

1994 年 3 月 4 日出生于甘肃省庆阳市环县。

2012 年 9 月考入清华大学计算机系计算机科学与技术专业，2016 年 7 月本科毕业并获得工学学士学位。

2016 年 9 月免试进入清华大学计算机系攻读硕士学位至今。

发表的学术论文

[1] **Lin Ma**, Yuchun Ma. Automatic Question Generation based on MOOC Video Subtitles and Knowledge Graph.[C]//Proceedings of the 7th International Conference on Information and Education Technology. ACM,2019:49-53. (ICIET 2019 已录用，EI 检索源)

[2] **Lin Ma**, Yuchun Ma. Intelligent Composition of Test Papers based on MOOC Learning Data.[C]//Proceedings of the 10th International Conference on Educational Data Mining. 2017 (ICEDM 2017 已录用，Poster)