基于大纲规划的 AI 写作系统

摘 要

随着人工智能技术的不断发展，自然语言生成技术广泛地应用于文本写作。本文提出了一种基于倒排索引与 Hash 编码的文本检索模型，先利用倒排索引实现快速筛选，再利用 Hash 编码进行精准检索，实现基于文本检索的大纲规划。经过机器评估，在包含超过 265 万条大纲数据的大纲库中，该模型的检索时间控制在 500 毫秒以内，经过人工评估，检索大纲的主题相关性强，内容丰富连贯。相比没有大纲规划的文本生成文章，本文在大纲规划后所生成文章内容更丰富，主题性更强。基于本文大纲规划模型与 GPT-2 文本生成模型，本文构建了一个 AI 写作系统，以文章生成为核心功能，同时集成文本摘要等技术，形成了一个从用户灵感发现到文章产出的一站式写作平台。

关键词：文本检索；文本生成；文本摘要；写作系统

An AI writing system based on outline planning

Abstract

In recent years, the application of AI-assisted writing technology in real life has become increasingly widespread, and natural language generation technologies such as text generation have become hot topics in academia and industry. In this thesis, we propose a text retrieval model based on inverted index and Hash coding. First, the inverted index is used to achieve fast screening, and then the Hash code is used to achieve accurate retrieval, and the outline planning based on text retrieval is realized. After machine evaluation, the retrieval time of the model is controlled within 500 milliseconds in the outline library containing more than 2.65 million outline data. After manual evaluation, the retrieved outline is highly topic-related and rich and coherent. Compared with the text generation without outline planning, the content of the generated article after outline planning is richer and the topic is stronger. Based on this model and the GPT-2 text generation model, we designed an AI writing system with article generation as the core function and integrating text summarization and other technologies, creating a one-stop writing platform from user inspiration discovery to article production.

Keywords: Text retrieval, Text generation, Text summarization, Writing system

目 录

1 绪论

1.1 课题背景与意义

在当今的人工智能时代，互联网、移动设备和社交媒体等技术的广泛普及，使得海量的数据可以被快速地收集、存储和分析，为人工智能技术的发展提供了基础。同时，随着机器学习，深度学习等各种算法以及计算机技术的不断进步，人工智能技术已经被广泛应用于如人脸识别，自动驾驶，机器博弈等各个领域。

自然语言生成(Natural Language Generation, NLG)作为人工智能应用的一个分支，近年的发展不可谓不火热。互联网上能获取到的海量数据成为了驱动自然语言生成发展的基础，伴随着大规模的预训练模型（如 GPT、BERT 等）的出现，以及人类对人机交互愈加迫切的需求，自然语言生成技术得到了极大的飞跃。而诸如智能客服、医疗报告、广告文案等多领域的应用需求，也使得自然语言生成技术的研究和应用在快速发展的过程中面临越来越多的挑战。可控文本生成（Controllable Text Generation, CTG）是自然语言生成的子任务，通过对文本生成模型进行一些控制，使得模型在某些具体的下游任务中的表现更好，以适应不同领域的应用需求。可以看出，随着数据、模型以及应用场景的不断丰富和发展，自然语言生成技术在将来仍是人工智能领域一个重要的任务。

AI 写作系统是基于人工智能技术的自动文本生成系统，随着自然语言生成技术的快速发展，尤其是大规模预训练模型的兴起，文本生成的准确性和流畅度得到了大幅度的提升，AI 写作系统的发展也有了坚实的基础。现如今市面上已经有不少的 AI 写作系统，其中最火热的便是 ChatGPT，针对多种下游任务都能取得不错的效果；当然也有诸如 Hey Friday、写作猫等专攻文章写作的平台。在当前社交媒体和互联网高度普及的时代，内容创作的产业发展潜力极大，AI 写作系统所带来的智能化辅助，能够大大提高内容创作者的生产效率和文本质量，从而推动内容创作产业的发展。AI 写作系统在不断创新和完善，正为人类带来更加高效、准确、创新和自然的文本输出体验。

AI 写作系统是当前自然语言处理领域的热门应用之一，常见的写作系统在文本生成过程中往往会出现一些问题，如冗余和重复的内容，不连贯的段落和主题，模板痕迹较明显等。如何让计算机生成出更流畅，更符合人们预期的内容，是当前的研究的热点。

基于大纲规划的 AI 写作系统可以提高文本生成的准确性和效率。针对生成内容冗余、

段落及主题不连贯等问题，在进行大纲规划后，生成模型能更好地理解用户输入的要求，从而生成更加准确和连贯的文本。其次，基于大纲规划的 AI 写作系统可以增强文本生成的灵活性和多样性。常见的传统 AI 写作系统只能生成特定类型的文本，如新闻报道、产品描述等，且所生成的内容模板痕迹较为明显。而基于大纲规划的方法，让生成模型在大纲的控制下，生成更灵活多样的文本。

研究基于大纲规划的 AI 写作系统，还可以推动自然语言处理领域的研究和发展。这种大纲规划的思想，不仅可以用于写作系统，还可以应用于其他自然语言处理任务，如利用文本的大纲进行文本分类，信息抽取等。

1.2 国内外研究现状

可控文本生成(Controllable Text Generation, CTG)，是 NLG 领域新兴的一个领域，这些技术能更好地满足实际应用中的各种情景限制。自动文本摘要(Automatic Text Summarization, ATS)是自然语言生成任务中另一个应用，主要包括抽取式的摘要和生成式的摘要。文本检索技术在信息化时代中的应用同样很广泛，伴随着搜索引擎走入人们的生活中，文本检索技术也在快速发展。

目前主流的 CTG 应用，Prabhumoye 等人[1]将其概括为五个模块：外部输入控制、序列输入控制、生成器控制、输出控制及训练控制。当然 Prabhumoye 也在文中提出，可以将这几个模块的控制进行结合。Lewis 等人[2]针对于问答系统提出一种将问题和知识进行组合的方式，构造文本生成模型的输入，从而进行回答的生成，这一思想也可以应用于可控文本生成领域，通过组合各种控制信号，来控制文本的生成。Xu 等人[3]利用用户的输入信息，利用 decoder 进行关键词的预测，再结合外部给与的关键词，从知识库中进行知识的抽取，再将得到的关键词和知识作为解码器的输入，从而达到对文本生成的控制效果。Yang 等人[4]根据有限的输入文本，将其映射成一个低维的主题分布，同时训练一个由主题分布重构的解码器模型，从而用得到的主题分布作用于模型得到一些主题相关的词，作为整个故事的框架。Yao 等[5]提出 plan-and-write,利用输入的标题，先对故事的情节进行一个具体的规划，从而生成一个完整的故事。Zhai 等人[6]提出故事的生成需要更详细的细节信息，于是提出根据所给的情节信息设计出大纲，再由大纲设计出每一段之间的联系，从而生成连贯性更强的文章。Rashkin 等[7]也考虑利用给定的大纲，动态规划出每一段的行文情节和思路，从而达到按照大纲写作的效果。

文本摘要的抽取式方法包括主题模型，基于图的方法，基于特征评分的方法等。主题模型有如隐式狄利克雷分布模型(Latent Dirichlet allocation, LDA)[8]等，通过对文本内的文本单元进行抽样分析，构造出每个文本单元的主题概率，从而得出重要的文本单元。基于图的方法主要指由 PageRank 衍生出的 TextRank[9]等系列图算法。 Xu 等人[10]将 TextRank 技术与 Doc2Vec、K-Means 技术进行融合，利用 Doc2Vec 进行文本向量化后，采用 K-Means 实现文本相似度的聚类，再在每个簇中加入 TextRank 进行排序，得到了比 TextRank 质量更高的摘要。李维等人[11]针对藏文摘要抽取的任务，提出将语料库信息以词向量融入到 TextRank 方法中，再进行迭代对句子进行打分，选取出分值最高的句子重新排序作为文本的摘要。谷莹等人[12]针对评论文本，先进行聚类，对文本进行主题的划分，然后用 Word2Vec 模型获取句子的向量化表述，并根据句子间的语义相似度进行图模型的构建，最后利用加权图进行摘要的抽取。基于特征评分的方法，主要包括提取原文的特征，如词频，句子相似度等方法来判断文本中某词或某句是否属于摘要，这种方法简单，速度快，但是效果往往不是很好。Ferreira 等人[13]分析了 15 种基于文本特征的句子评分的算法，对抽取式的文本摘要进行定量和定性的评估，其中的特征包括词频、TF-IDF、大写字母、专有名词、词共现、句子长度、句子位置等众多特征信息。Wang 等[14]提出了 9 中启发式的方法，包括冗余句子删除法、基于完整摘要的句子评分、基于摘要句子的句子评分等，为抽取式摘要来构造近似理想的抽取和上界，同时也用 6 中评分方法和 5 中语料库来证明所提出方法的有效性。

文本检索技术，Johnson 等人[15]提出一种倒排索引加点积量化的方法，利用聚类的方法，对分段的高维向量进行分类后，对待检索的向量进行分段检索，再利用倒排索引，快速检索，针对高维空间中的海量稠密数据，提供了高效可靠的相似性聚类和检索方法。Manku 等人[16]将 simHash 算法利用到网页快速去重的方式，证明了 simHash 算法的有效性，利用“分桶而治”的思想，对文本内容进行 hash 编码，并进行分块匹配，从而快速检索相似的编码。孙宇等人[17]利用分词和倒排索引的方式，实现了对短文本的快速检索。

1.3 本文主要工作

针对上述的研究现状，本文对文本检索和文本生成存在的一些问题进行改进，并基于改进后的模型设计并实现了一个辅助写作的 AI 写作系统。本文的研究内容主要如下：

1. 基于倒排索引与 Hash 编码的文本检索。对于海量的句，段级别文本数据，如何在

保证检索速度的情况下，提高检索内容的准确性，是文本检索的一个关键问题。本文结合倒排索引的高效性以及 Hash 编码检索的准确性，提出一种基于倒排索引与 Hash 编码的快速检索方式。该模型使用倒排索引对输入的关键词信息进行快速地检索，得到包含该词的一些文本，实现初筛。再将待检索的标题与倒排索引得到的文本进行 Hash 编码，利用 SimHash 的思想实现精筛，进而实现初筛-精筛双层模型快速精准地检索。

2. 基于大纲规划的文本生成。通过对已有的智能写作系统的调查以及文本生成大模型的分析，本文发现在生成模型对内容进行生成之前，如果对生成模型进行一定的内容提示控制，会使得生成的内容更符合预期。本文基于以上可能的问题，提出基于大纲规划的文本生成方式，对于得到的用户输入关键信息，如标题与关键词等，不是直接将其作用于文本生成模型，而是类似现实生活中的写作思路，利用关键信息进行大纲的规划，再利用规划的大纲具体段落的提示，生成各段落的具体内容。在进行大纲规划的过程中，采用多种评估方式对大纲进行评估，从而得到优质的大纲以让生成模型的效果更优。

3. 基于大纲规划的 AI 写作系统设计与实现。本文根据实际调研的需求，将实现的文本检索以及文本生成模型应用到写作系统中，将其作为写作系统的核心模块，使用常用的分布式微服务架构，满足日常使用的高并发、高可用需求，设计并实现了一个基于大纲规划的 AI 写作系统。

1.4 论文章节安排

第一章为绪论部分，主要介绍基于大纲规划的 AI 写作系统的一些相关背景、意义以及当前的研究现状，然后概述了本文的主要工作，最后对本文的各章节结构进行了介绍。

第二章介绍本文涉及的一些相关理论和技术，首先介绍了常用的文本检索技术，如 SimHash 和倒排索引，然后介绍了当前流行的 GPT-2 文本生成模型。

第三章为基于文本检索的大纲规划，首先介绍了大纲规划任务以及整体流程，然后介绍了其中的核心模块，基于倒排索引和 Hash 编码的文本检索，最后展示实现模型的实验过程及结果分析。

第四章为系统设计与实现部分。结合介绍及实现的算法，进行整个 AI 写作系统的搭建，详细介绍了系统的需求、设计以及实现效果。

第五章为本文的总结与展望。主要对本文所作的工作以及得到的成果进行总结概括，对于不完善的地方进行对未来的展望。

2 相关理论与技术

2.1 文本检索

文本检索(Information Retrieval, IR)是一种从大量文本数据中自动获取相关信息的技术，是自然语言处理中的核心任务。其目的是在一个大型的文本集合中查找与用户查询相关的文本文档，并将查询到的内容按照相关性排序后返回给用户。

倒排索引（Inverted Index）是文本检索系统中最常用的索引结构之一，它是一种将文档中的词语映射为文档列表的数据结构。具体来说，倒排索引将文档中每个重要的语素设置成索引，以此检索包含该语素的文档。构建的倒排索引大致工作流程如图 2-1 所示。

图 2-1 倒排索引流程图

SimHash 是一种常用的文本相似度计算方法，它能快速计算出两个文本是否相似，最初 google 将该方法应用于解决亿万级别的网页去重任务。SimHash 通过提取出文本中的特征关键词，将关键词映射成特定长度的 hash 编码后，进行加权合并，得到每个关键词的编码，再将关键词的编码累加合并，最后进行降维，得到整个文本的二进制编码格式。在计算文本之间的相似度时，采用汉明距离，计算两个编码中不同的位数，即为两个编码的距离。大致流程如图 2-2 所示。

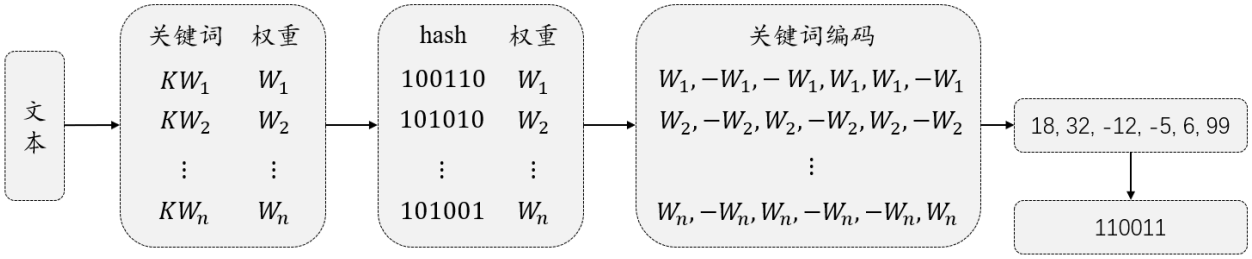

图 2-2 SimHash 编码流程

在以上编码过程中，每个关键词的权重可以由 IDF 或其他词权重方法进行计算。将

整个文本表示成二进制编码后，SimHash 采用“分桶而治”的思想（如图 2-3 所示），将一个多位的二进制编码进行分段，通过计算两个编码中每一段的匹配性，来计算两个编码的相关性。如将每一位看作一段，则此时计算得到的相关性为两个编码之间的海明距离。

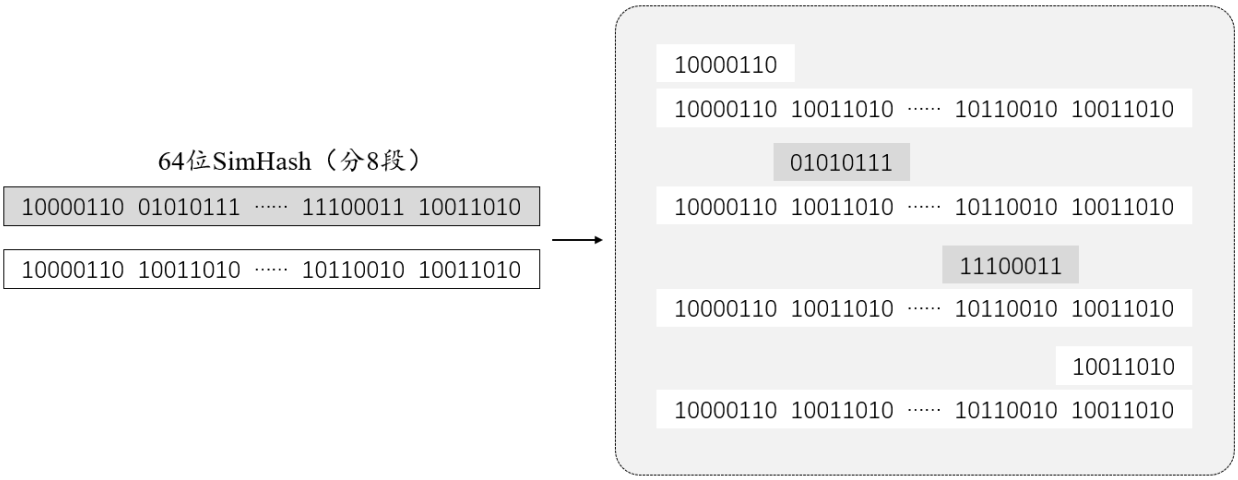

图 2-3 SimHash 分桶匹配检索方式

BM25[18]是信息检索领域的经典算法，它通过计算文本单元与文本的相关性，对相关性进行评分，从而找出代表文本的文本单元，在倒排索引的构建过程中常被用于提取文本中的关键词。其评分公式如下：

$$Score(Q,d)=\sum_{i}^{n}W_iR(q_i,d) \tag{2.1}$$

其中Q表示一条 query，q_i表示Q解析后的一个语素（如分词后的词），d表示某个相关文档，W_i表示q_i的权重，$R(q_i,d)$表示语素q_i与文档d的相关性得分。对于权重W_i，通常使用 IDF(Inverse Document Frequency)来进行计算，计算公式如下：

$$IDF(q_i)=log\frac{N-n(q_i)+0.5}{n(q_i)+0.5} \tag{2.2}$$

其中N表示索引中全部的文档数目，$n(q_i)$表示包含了语素q_i的文档数目，根据 IDF 的计算公式可以看出，包含语素q_i的文档越多，则该语素的重要性便越低，因为此时该语素的区分程度变低。对于相关性得分R，一般形式如下：

$$R(q_i,d)=\frac{f_i(k_1+1)}{f_i+K} \tag{2.3}$$

$$K=k_1\left(1-b+b\frac{dl}{avgdl}\right) \tag{2.4}$$

其中f_i为q_i在文档d中出现的频率，k_1, b为调节因子，dl表示d的长度，$avgdl$表示所有文档的平均长度。从K的计算公式中可以看出，b可以调整文档长度对相关性影响的大小，b越大，文档长度对相关性的影响程度越大，反之则反。以此可以控制由于长文带来的f_i过大的情况。

2.2 文本生成

当今自然语言处理领域中最重要的技术便是 Transformer 模型[19]，它是一种基于注意力机制的神经网络模型，由 Google 在 2017 年被提出，并被应用于自然语言任务中，如机器翻译、文本生成等。与传统的序列模型如循环神经网络(RNN)、长短时记忆网络(LSTM)不同，Transformer 模型采用了编码器-解码器的双塔结构，并且加入了注意力机制，将输入序列中的每一个元素都视为一个向量，并使用自注意力机制和多头注意力机制，来实现并行化等一些优化。

GPT(Generative Pre-Training)模型[20]由 OpenAI 公司提出，GPT-2[21]是其升级版，该模生成的文本在上下文连贯性和情感表达方面均取得了显著的进展。GPT 模型（如图 2-4 所示）采用 Transformer 的解码器替代传统的神经网络作为特征提取器进行堆叠，具有强大的特征抽取能力，每一层都包含带掩码的自注意力(Masked Multi-Head Attention)和前馈网络(Feed Forward Network)，并通过残差网络和正则化进行连接。

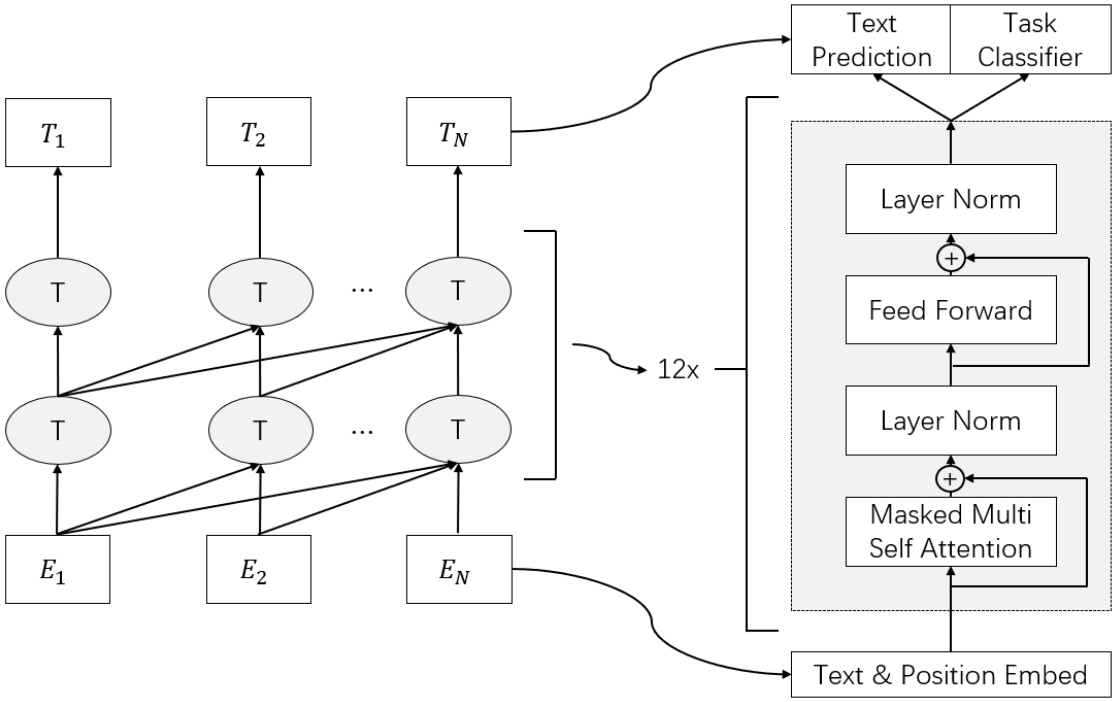

图 2-1 GPT 模型结构

GPT 是由多层的 Transformer 解码器进行堆叠而成，每一层被成为一个 Transformer block，模型的初始输入为词的嵌入向量加上与其对应的位置编码，每个 Transformer block 的输入是上一个 Transformer block 的输出，形式化表示为：

$$h_0 = UW_e + W_p \tag{2.5}$$

$$h_1 = Transformerblock(h_{l-1}) \forall i \in [1, n] \tag{2.6}$$

其中W_e和W_p分别为输入的词嵌入矩阵与位置嵌入矩阵，U为模型的输入。

从模型的结构上看，GPT-2 模型相较于 GPT 来说结构改动并不明显，仍使用单向的 Transformer 解码块进行堆叠而成，但是其参数规模增加了很多。主要的改进如下所示：

1. 更深的网络架构，GPT-2 有 12、24、36 和 48 个 Transformer 层的四种不同版本，而 GPT 只有 12 个 Transformer 层。

2. 更广泛的训练数据，GPT-2 包括了互联网上数百万个网页、书籍及其他的文本数据，而 GPT 采用的仅为一些小型的语料库，如维基百科等。

3. GPT-2 在训练过程中使用了随机掩码，即在输入文本中随机屏蔽一些单词，以鼓励模型学习如何预测丢失的单词，这种技术有利于提高模型的泛化能力。

2.3 本章小结

本章主要介绍了文本检索与文本生成的相关理论和具体技术。首先介绍了文本检索中的 SimHash 和倒排索引技术，然后介绍了构建倒排索引的过程中常用的 BM25 算法。文本生成技术主要介绍了当前主流的文本生成模型 GPT-2。本章介绍的内容将在本文的后续章节中具体展开描述与实现，最终应用到 AI 写作系统中。

3　基于文本检索的大纲规划

3.1 任务描述

针对文本生成模型对长文生成时会出现主题不连贯、可解释性较差两个常见的问题，设计大纲规划模型，利用大纲对文本生成模型的输入进行控制，以短文本拼接的形式进行长文的生成。大纲规划模型将大量的文本数据清洗处理后存入数据库中备用，利用文本检索技术进行大纲的检索规划，将关键信息转换成结构化的大纲信息，从而将各个部分的大纲内容作用于文本生成模型，生成各个文章各个段落的具体内容，形成一篇完整的长文，可以一定程度避免主题不连贯和可解释性差等问题。

3.2 基于文本检索的大纲规划

大纲规划模型需要根据用户输入的关键信息如标题、关键词等规划出一个连贯合理的大纲。本文所使用的大纲规划方法是检索式大纲，即从已有的大纲库中检索匹配出与输入信息最相关的大纲。模型的整体流程图如图 3-1 所示。

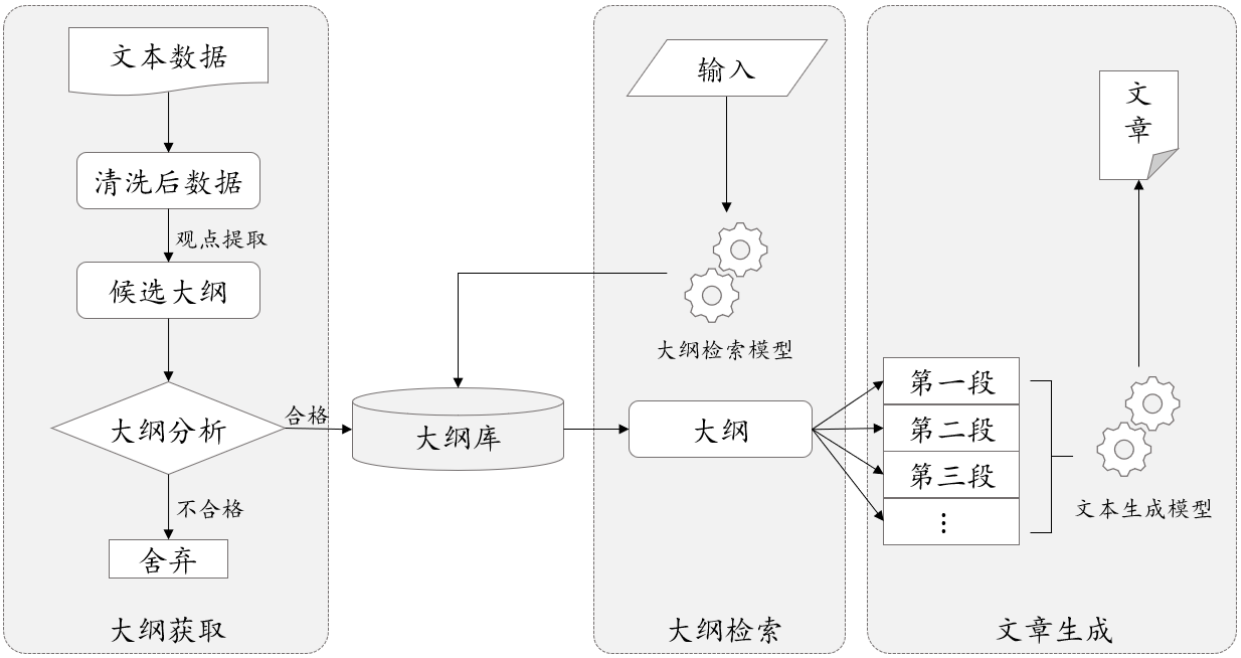

图 3-1 大纲规划模型流程

大纲获取。首先对文本数据进行清洗，去除文本中的噪声和冗余信息，以便能够得到规范的段落文本数据。清洗后的文本数据更容易进行后续的处理和分析。然后使用观点提取和文本摘要等方法，对每个段落的内容进行提取。观点提取是指从文本中提取出

作者或者其他人的观点或者看法，而文本摘要则是通过对文本进行压缩，提取出其主要信息。以此得到每段话的概括内容，作为候选大纲。针对得到的候选大纲，根据段落结构的合理性和大纲对文本内容的概括性对候选大纲进行分析和评估。对内容合理的大纲进行入库存储，不合理的大纲进行舍弃。

大纲检索。是大纲规划模型的核心部分，采用改进后的 SimHash 与倒排索引相结合的文本检索技术，将在 3.3 节中详细介绍。

文章生成。根据大纲检索得到的各段摘要内容，利用文本生成模型进行每段具体内容的生成，最后拼接成一篇完整的文章。

3.3 基于倒排索引与 Hash 编码的文本检索

针对文本检索任务中所面临的一些问题，如检索速度受限，检索内容不匹配等，基于本文 2.1 节的相关内容，本文发现在网页去重的任务中，使用 SimHash 原有的分桶的方式，并以字段的匹配程度来计算编码之间相似性的方式，具有很高的效率。但是将 SimHash 原有的基于匹配程度的相似性计算方式应用于文本相似性检索的任务时，会导致大量文本无法检索到与其相似的文本，从而导致检索方法失效。

基于这一问题，本文在计算 Hash 编码之间的距离时，在对文本编码进行分段后，不是以每一段编码的匹配与否的方式来计算文本之间的相似性，而是具体计算每一段对应编码的海明距离，将各段的海明距离进行求和，以此来计算编码之间的相似程度。此时，若所有字段的海明距离求和得到的结果越小，则其相似性越高。具体计算公式如下所示：

$$dist(d, d_i) = dist(d^1, d_i^1) + dist(d^2, d_i^2) + \cdots + dist(d^n, d_i^n) \tag{3.1}$$

其中，$dist(d, d_i)$为计算d与d_i两个文本编码之间的海明距离，$dist(d^1, d_i^1)$为计算d与d_i两个文本编码在分桶后的第一个桶内编码之间的距离。

基于以上改进 Hash 编码相似性计算方式，本章提出了一种结合倒排索引与 Hash 编码的文本检索方式，通过倒排索引实现快速初筛，再采取改进后 Hash 编码分桶检索的方式，对初筛得到的内容进行精筛提取。改进后的检索模型如图 3-2 所示，主要包括以下五个步骤：

1. 将文本进行 Hash 编码，转换成二进制的编码形式；
2. 将文本进行分词、去停用词后，利用倒排索引初步检索出相关的文本；
3. 将倒排索引检索出的文本进行 Hash 编码；

4. 分桶计算原输入文本与倒排得到的文本 Hash 编码之间的海明距离；
5. 得出海明距离最小的文本即为最佳检索结果。

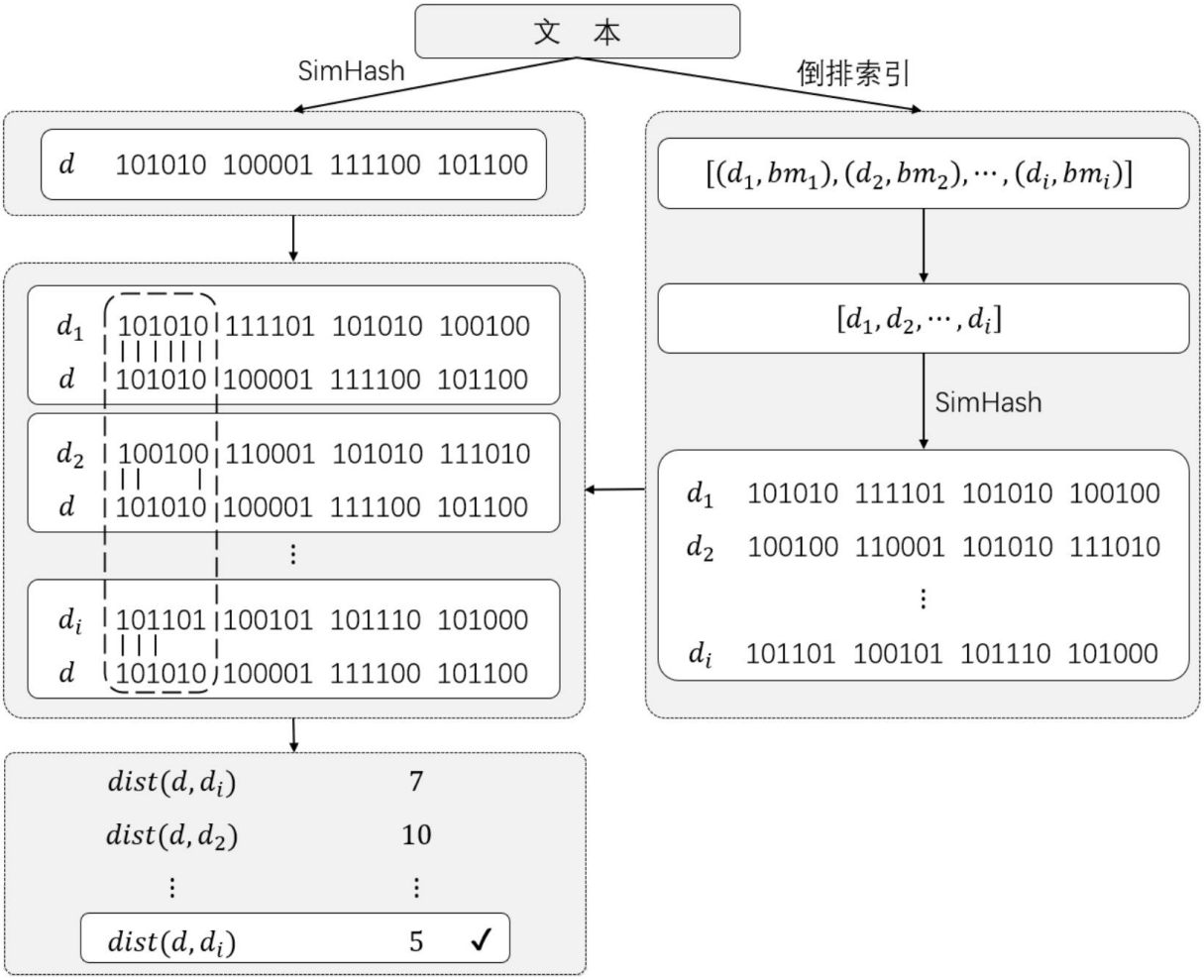

图 3-2 倒排索引与 Hash 编码文本检索模型流程图

3.4 实验

3.4.1 实验环境

具体的实验环境如表 3.1 所示。

表 3.1 实验环境及配置

CPU	内存	硬盘	系统	MySQL
Intel(R) Xeon(R) W-3223 @3.50GHz	64GB	1TB SSD	Centos7	5.7.39

3.4.2 数据预处理及数据库设计

本文所采用的数据集是网络推文数据，数据集中存在大量的“脏”数据，为了避免在构建倒排索引以及 Hash 编码时，对模型构成不必要的含义偏移，导致模型效果受干扰，需要对语料数据进行特殊化的清洗处理。

在对数据进行观察后，发现数据中存在外语、繁体、序号、特殊字符等不符合需求的内容，并且这些内容会导致语义的偏移与文本的乱码。文本检索与生成的任务，一般要求语料为纯段落或句子语料，即纯中文或英文的规范语料，不包含其他的附加结构如图表等。所以首先对语料进行初步的清洗，具体待清洗数据及实例如表 3.2 所示。

表 3.2 需要清洗的数据类型及实例

数据类型	实例
特殊字符及空格等	< > \ / | ● 《 》 á ◌ □ ◆等
网页链接、表格、图片等	<title></title>、图<数字>、表<数字>等
非正文的其他部分	猎云网（微信：ilieyun）北京】7 月 13 日报道等
无关的外语	日语（テクノロジー）、韩文（기술）、繁体等

基于以上清洗后的数据，进一步对语料进行规范化处理，得到 2651128 段大纲数据以及 482720 条摘要数据。得到语料数据后，利用数据库对数据进行存储，本文涉及的数据表主要有两个部分：内容数据库与索引数据库，前者用于存储大纲、摘要的内容，后者用于存储倒排索引与 SimHash 索引。针对内容数据库，本文主要设计大纲数据库与摘要数据库，如表 3.3 和 3.4 所示。其中 outlineID 与 abstractID 用于与索引表对应。

表 3.3 大纲内容数据表

字段名	字段类型	可否为空	键	注释
outlineID	bigint	NO	PRIMARY	每个大纲的 ID
Outline	varchar(255)	NO		大纲内容

表 3.4 摘要内容数据表

字段名	字段类型	可否为空	键	注释
abstractID	bigint	NO	PRIMARY	每个摘要的 ID
abstract	varchar(255)	NO		摘要内容

针对倒排索引与 SimHash 索引，本文主要设计倒排索引表与 Hash 索引表，如表 3.5 和表 3.6 所示。倒排索引表包括词的 ID、该词的 idf 值、以及该词对应的一个倒排列表，倒排列表中的每个文档都包含该词。倒排索引表用于初筛阶段，根据所给的关键词快速获取包含该词的所有文档。Hash 索引表包括一段文本的 ID，与大纲内容表的大纲 ID 进行对应，同时包含该文本的 Hash 编码，用于实现精筛阶段的 Hash 检索。

表 3.5 倒排索引表

字段名	字段类型	可否为空	键	注释
wordID	bigint	NO	PRIMARY	每个词的 ID
word	varchar(255)	NO		词
idf	double	NO		idf 值
docList	longtext	NO		倒排列表

表 3.6 Hash 索引表

字段名	字段类型	可否为空	键	注释
docID	bigint	NO	PRIMARY	文档的 ID
HashID	varchar(100)	NO	PRIMARY	文档对应的 Hash 编码

3.4.3 评价方法

本章的实验需要验证模型的三个效果，分别是文本检索的速度、文本检索的内容以及大纲规划的内容对文本生成模型的效果。

文本检索速度。采用接口测试工具 ApiFox 对接口进行并发请求，可以得到每一次请求结果的反馈时间，以这个时间来对文本检索的速度进行评价，系统的反馈时间越短，则说明文本检索的效果越好。

文本检索内容。将并发请求得到的反馈内容进行记录，使用人工评估的方式对内容进行评价。首先是大纲内容的连贯性，观察大纲是否前后逻辑连贯，每一段的内容是否能衔接；其次是内容的丰富性，即不同段落大纲的内容是否不同，在保证总体主题性统一的情况下做到内容丰富多样；最后是主题的统一性，即大纲与用户输入内容是否主题一致，若出现主题偏移，则大纲的质量较差。

大纲规划对文本生成模型的效果。利用规划出的大纲，作用于文本生成模型，得到

完整的文章输出，以文章的主题连贯性、内容丰富性来反馈评估大纲内容的连贯性与丰富性。同时，还需要对生成每一段的内容与每一段的大纲进行评估，以检测大纲规划后是否对文章生成的内容有效。

3.4.4 实验结果与分析

本文采用模拟真实应用场景的方式来对模型进行实验，即利用接口测试工具对大纲检索以及摘要检索的两个接口进行测试，检索时间的实验结果如表 3.7 所示。

表 3.7 对两种内容检索的时间

检索内容	检索方式	检索时间(100 次测试平均)	
		Top-3	Top-10
大纲	倒排索引	**79ms**	**85ms**
	本章模型	479ms	493ms
摘要	倒排索引	**85ms**	**90ms**
	本章模型	475ms	481ms

从表 3.7 中可以看出，在使用本章提出的改进模型时，检索时间大致在 500ms 以内，基本与使用 SimHash 方法进行检索的时间持平，而仅使用倒排索引的检索时间可以达到 100ms 以内。由此本文推断在本章改进模型中，检索速度主要受 Hash 检索阶段的速度限制。在检索时间比较的基础上，利用倒排索引和本章模型分别检索内容的比较，比较示例如图 3-3 所示。

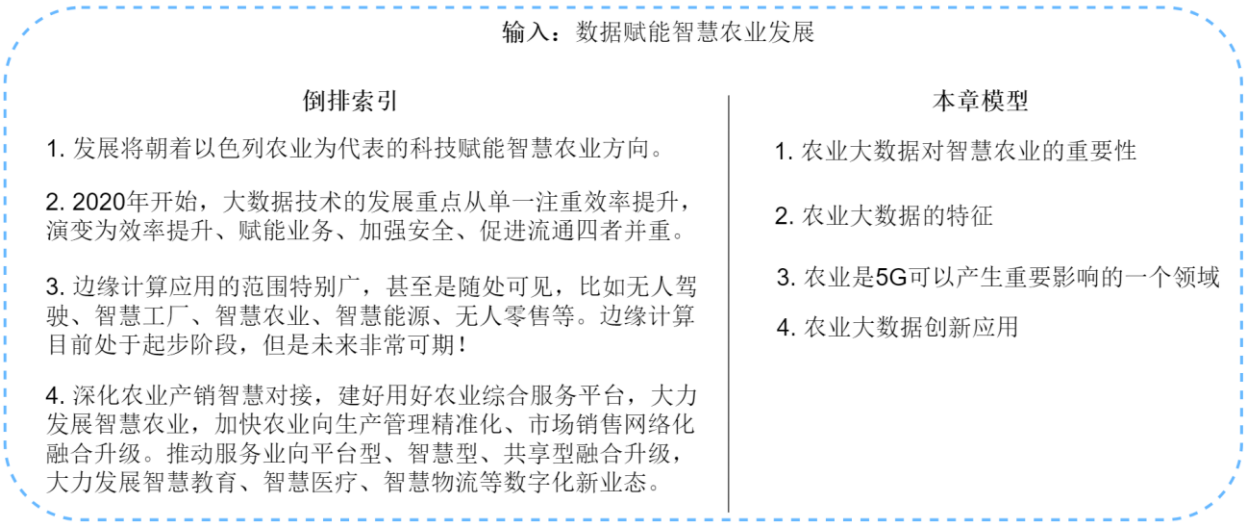

图 3-3 倒排索引与本文模型检索内容对比

从图 3-3 可以看出，仅使用倒排索引进行检索，难以排除文本长度(关键词数量)对检索结果的影响，导致检索出来的内容普遍偏长且包含关键词信息较多。而本章模型所检索出的四条大纲内容与输入主题性较倒排索引更为匹配，且语句精炼，内容性更强。

最后，对比无大纲作用的文本生成所得到的文章（图 3-4）与利用本章规划大纲后所得到的文章（图 3-5），评价其差异。使用统一标题为“数据赋能智慧农业发展”，大纲为图 3-3 中检索得到的大纲。

"大数据时代已经到来，数据赋能智慧农业发展，智慧农业是数字乡村建设的必由之路。在数字乡村发展的大背景下，数据已经成为智慧农业发展的重要支撑。随着数字化转型进入深水区，数据的价值将得到进一步释放，智慧农业作为数字经济的重要组成部分，也将成为数字中国建设的重要内容。在数据赋能智慧农业发展过程中，大数据将扮演重要角色。随着数据的价值日益凸显，数据在赋能农业发展的过程中将起到关键作用。"，
"大数据时代已经到来，数据的价值将进一步得到体现。数据的价值不仅仅体现在数据的价值，而且将会更加体现在数据的价值。数据是数字经济时代的石油，数据赋能智慧农业发展，数据赋能产业升级，数据赋能农业发展，数据赋能农业发展。数据的价值不仅体现在数据的价值，而且在于数据在智慧农业中发挥着重要作用。数据的价值在于对数据进行价值的挖掘。数据的价值是通过对数据进行分析，得出有价值的数据。但现有数据价值并不能转化为数据价值。在智慧农业中，数据的价值主要来源于对数据进行价值的挖掘。智慧农业中数据量大且价值高，数据量也远远高于智慧农业中的数据量。数据的价值主要来源于数据的应用价值。"，
"大数据时代已经到来，在数据赋能智慧农业发展中，数据的价值不断释放，数据的价值也在不断提高，数据赋能智慧农业发展，数据的价值将得到充分挖掘，数据的价值也将更加体现，数据赋能智慧农业发展的前景也将更加广阔。"

图 3-4 无大纲生成文章内容

"农业大数据对智慧农业的重要性毋庸置疑，数据是智慧农业的核心要素，是智慧农业发展的基础。数据赋能智慧农业发展，离不开大数据技术的支撑，大数据技术的加持，大数据的价值将更加凸显。在数字技术的加持下，农业大数据应用范围将不断拓展、深化，在精准营销、精准农业、精准农业、精准扶贫、精准农业、精准农产品、精准农业、精准农民、精准农业等领域，大数据技术将为智慧农业的精准施策提供有力的技术支撑，在农业生产、经营和管理中起到关键作用，为农产品精准营销、精准农业管理提供有力支撑。数据赋能智慧农业发展，离不开智慧农业的支撑。大数据技术能够为智慧农业的发展提供数据支持。大数据可以帮助农业企业实现对全面、动态的数据采集、分析与应用，为决策提供依据。"

"农业大数据的特征是在数据源头进行采集、清洗、加工和分析，数据赋能智慧农业发展。随着智慧农业的不断发展，对于农业大数据的需求不断攀升。在农业数据的基础上，对于大数据的应用有着更高的要求，数据采集、分析、加工和分析等技术也成为了农业智慧化发展的必要条件。在农业大数据的应用中，基于数据的产品、技术和服务也是农业大数据应用的主要方向。在农业大数据应用过程中，通过对大数据技术应用的深度分析，形成可用于农业生产经营管理、农业服务、农产品销售、农业生产管理、农产品销售等方面的数据分析应用。在农业大数据应用过程中，通过对大数据技术应用的深度分析，形成可用于农业管理、农业决策、农业生产、农户服务、农村消费、农业生产、农村经济、农民服务等方面的数据分析应用。"

"农业是5G可以产生重要影响的一个领域，5g技术将推动农业发展，数据赋能智慧农业发展，5g智慧农业的发展将为智慧农业的发展提供更多可能。中国移动通信集团有限公司董事长杨杰表示。中国移动将在农业农村建设中，将通过乡村地区5g网络建设，缩小城乡数字鸿沟，推动智慧农业的发展。同时，将在加快数智乡村建设的基础上，加快5g、物联网等信息基础设施的部署，提升乡村数字化服务的能力。中国移动通信集团有限公司董事长杨杰表示。5g、ai和数据是新型基础设施的核心要素，也是驱动产业发展的新引擎。数据作为新的生产要素，已经成为经济社会发展的新型生产资料。据中国信息通信研究院预测，到2025年，5g可拉动电信运营商网络投资约2万亿元；带来的信息消费规模累计将超过3万亿元。数据赋能智慧农业发展，需要的不仅仅是5g网络的铺设，还需要的是5g与行业应用的融合。"

"农业大数据创新应用，助力农业产业转型升级。一是加快数据融通。推进农业大数据中心、云服务平台建设，建成国家级农业数据中心，建成农业大数据云平台，汇集农业数据，实现农业生产数据的智能化采集分析、应用、服务。二是加快数据融通。建立数据资产管理体系，加快数据汇聚共享和开放。建设国家级大数据中心，汇聚政府数据、企业数据，建立政企数据库，打造大数据应用场景，支持企业开展数据资产化管理。建设全国一体化政务大数据平台，推进数据跨部门、跨区域共享，打造数据开放、共享、流通、加工的全生命周期运营体系，实现政务服务一个入口、社会治理一个通、政府服务一个平台、社会服务一个平台。三是深化数据赋能。深化大数据应用创新，支持各地区各部门开展大数据创新应用，培育大数据服务新业态，建设面向重点行业的大数据应用创新中心。"

图 3-5 本章大纲规划后生成文章内容

对比无大纲和有大纲所生成的文章内容，可以发现无大纲作用时，生成的文章每一段的内容欠佳，每一段的主题性不明显，整篇文章的篇幅较小。而在本章规划大纲的支持下，所生成的文章中，每一段的主题紧贴所规划的大纲，并且整篇文章的篇幅较大，内容丰富。因此，大纲规划对于文本生成模型的作用是显著的。

综合以上三个方面的比较，在时间维度上，100ms 与 500ms 在用户实际体验过程中的感受并无明显差别，但是在内容维度上，本章所构建的模型所检索出的内容，在 Hash 检索的作用下，检索出的结果明显优于倒排索引，且作用于文本生成模型后，所生成的内容更丰富，主题性更强。

3.5 本章小结

本章主要介绍了大纲规划模型，并详细阐述本章使用的大纲规划方法，即改进的基于倒排索引于 Hash 编码的本文检索。首先介绍了模型的整体设计思路，然后介绍了其中文本检索模块的实现方法，最后对数据集的处理方法进行描述后，利用实验证明本章提出的文本检索模型的高效性以及大纲规划对于文本生成的有效性。

4　AI 写作系统设计与实现

在当前社交媒体和互联网高度普及的时代，内容创作的产业发展潜力极大，AI 写作系统所带来的智能化辅助，能够大大提高内容创作者的生产效率和文本质量。本章设计并实现的 AI 写作系统旨在为用户提供一站式的辅助写作平台，从灵感发现，到大纲规划，再到整篇文章的生成，只需要用户输入少量的提示信息，即可生成一篇完整的文章。

4.1 总体设计

总体设计的目标是回答系统如何实现的问题，旨在从功能框架，技术框架等方面给出完善的方案。

4.1.1 功能框架

功能框架指系统需要实现的各个具体的功能点，用户将会利用这些功能点来完成对应的任务。在系统的开发过程中，功能点通常是从用户实际的业务需求出发进行设计的。本章设计的 AI 写作系统功能框架如图 4-1 所示。

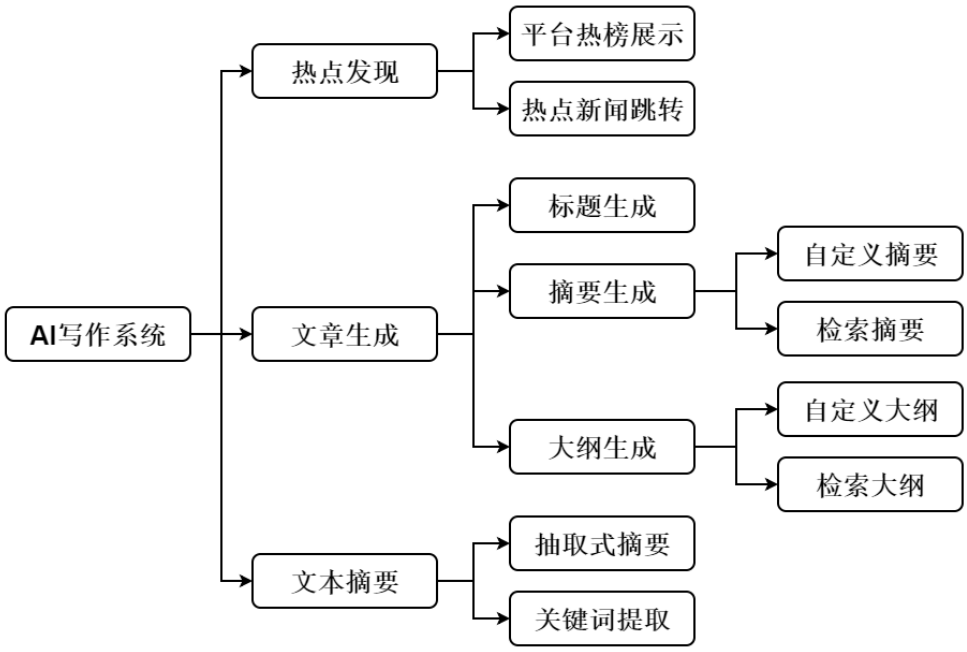

图 4-1 AI 写作系统功能框架图

（1）热点发现模块：用户在此模块内查看各个平台当前的热点新闻，并在此发现写作的灵感。热点发现模块主要为用户提供一个信息来源，用户可以通过点击跳转到每个具体的新闻热点中去查看自己感兴趣的信息。

（2）文章生成模块：本文写作系统的主要功能模块，根据用户输入的提示信息，进行摘要的检索、大纲的规划，或接收用户自定义的摘要与大纲，进行整篇文章的写作。

（3）文本摘要模块：利用文本摘要的相关技术，对用户输入的长文本进行摘要与关键词的提取，从而帮助用户快速获取长文本中的重要信息。

4.1.2 技术框架

整体技术框架的目标是回答系统如何实现的问题，旨在从架构，技术选型等方面给出完善的方案。基于以上功能框架的各个功能点需求，本章搭建整个系统的技术框架如图 4-2 所示。系统建立在深度学习环境以及数据库等支持基础之上，通过服务后台进行数据及模型的整合，最后由客户端进行功能的展示。其中客户端与服务后台通过 HTTP 进行交互。客户端主要供用户进行操作以及向用户进行结果的反馈，服务后台主要包括数据处理，模型支撑、逻辑处理三个模块。其中数据支撑提供各种语料数据，主要包含生成模型训练所需的推文语料、热点新闻展示所需的新闻数据以及大纲规划所需的大纲语料等，为模型的实现和调取提供支撑；模型支撑由检索模型，大纲规划模型，生成模型等组成，主要负责对功能接口的一些实现；逻辑处理针对客户端发送的请求，进行对应的接口调用，以实现各个功能。

图 4-2 AI 写作系统整体架构

针对服务后台与客户端中所涉及的技术，在此做一个简要的技术选型介绍。

1. 后端。在 Web 平台开发中，后端通常是指应用程序或软件系统背后的实际逻辑。

在使用 Python 进行系统的开发时，后端常用的技术是 Django 或 Flask 框架。本文设计的 AI 写作系统是基于 Django 进行开发的，其遵循 MVT(Model, View, Template)设计模式，在其 Model 层自带数据库 ORM 组件，方便用户进行数据库的访问，而其高效灵活的 URL 映射也为开发带来极大的便利。

2. 工具中间件。在前端和后端的交互过程中，采用适当的工具中间件，可以提高后端的功能支持能力，也能加强前端展示的效果。本文设计系统时所采用的中间件主要有 Nginx 与 UWSGI，其中 Nginx 的目的是接收客户端的大量请求，并将请求分为动态请求与静态请求，从而快速针对请求返回静态资源以及调用 UWSGI。UWSGI 作为 Nginx 与 Django 的中间件，负责 Nginx 与 Django 之间的通信。

3. 前端。前端即网页的前台部分，本文设计的 AI 写作系统所使用的前端技术主要基于 Bootstrap 框架以及 HTML、JavaScript、Ajax 等基础的网页开发语言。

4.1.3 非功能性设计

非功能性设计，是指在系统应用中，为满足用户的实际业务需求而必须具有的除功能性设计以外的方案，与功能性设计不同的是，非功能性设计关注的不是具体的功能点，而是整个系统在运行、交互、维护等环节中需要注意的点。本文实现的 AI 写作系统所包含的非功能性设计主要包括易用性、可靠性、高性能性、可扩展性。

（1）易用性：易用性的重点在于如何让产品的设计能够更贴合用户的使用习惯与需求。具体来说，系统的设计对用户而言应当是简单好用，能快速上手的。本文实现的 AI 写作系统，页面简洁精炼，功能操作方便，能让用户快速地实现写作及其他需求。

（2）可靠性：当系统的功能越完善、结构越复杂，可靠性也随之越来越重要，此时，应用对系统运行的对可靠性的要求也就越高。对于本文的 AI 写作系统，在运行时需要保证出现异常时能及时进行响应和恢复，针对服务器宕机等可能出现的情况，要有对应的措施来保证系统的稳定运行。

（3）高性能性：用户在使用系统时，如果等待时间过长，则体验会变差，所以需要系统的高性能来满足用户具体的需求。在用户进行摘要、大纲等检索时，在保证结果准确性的前提下，响应时间要控制在毫秒级别；在用户使用文本生成的功能时，一篇文章的生成时间不能超过 5 秒。

（4）可扩展性：考虑到现有的功能只是基础性的一些接口，当业务场景不断丰富时，

便需要系统能快速进行功能的扩展；同时当用户访问量增加后，对系统的运行压力会变大，此时便需要对系统的硬件设施进行升级扩展。因此，系统的可扩展性能为系统的升级改进提供良好的基础。

4.2 详细设计

详细设计是在总体设计的基础上，从逻辑上实现每一个模块的功能，本章主要介绍热点发现模块、AI 写作模块以及文章摘要模块的详细设计与具体实现。

4.2.1 热点发现模块

从本文 AI 写作系统的设计初衷出发，用户需要从本系统中获取写作的灵感，热点发现模块从常见的新闻平台（如今日头条、微博等）中爬取每日的热点新闻，并每隔一小时进行定时更行，帮助用户实时把握不同的写作主题与观点。

在网络爬虫过程中，使用 Requests 库对新闻平台进行信息的请求，得到响应后，针对不同的新闻平台建立对应的 XPath 解析路径，并制定对应的信息提取策略，本文对于不同新闻平台所制定的爬取策略如下所示：

1. 人民网。首先，在爬取人民网新闻过程中，需要对得到的 html 页面内容进行解码再编码，将其转换成 GB2312 的编码格式，以防出现乱码的情况。其次，对于人民网每日的“头条一览”部分的新闻内容，会出现以图片的格式进行新闻标题的展示，故要对类似的新闻进行剔除，以防出现新闻内容为空的情况。

2. 今日头条。在分析返回信息的 Json 数据时发现，在今日头条的置顶新闻中，所给的信息不全，针对这一情况，本文在解析信息时，将置顶新闻的数据进行剔除。

3. 微博、百度。微博和百度平台在新闻中会插入一些广告信息，对于新闻内容来说是一种“脏数据”，在分析后，本文对于广告数据进行剔除。

4. 知乎。在对知乎返回内容进行解析后，发现其中的 url 链接中进行了装饰，在对比真实的 url 链接与爬取得到的 url 链接后，采用将装饰信息进行定点替换的方法，对链接进行提取。

得到以上各新闻平台的新闻内容后，将新闻的标题与链接存储在内存中，以方便系统进行快速地访问。

4.2.2 AI 写作模块

AI 写作模块作为系统的核心部分，主要由数据和算法进行支撑。其中数据部分包括由大量的网络推文数据处理后得到的文本摘要数据库与文本大纲数据库，算法部分主要包括本文第三章和第四章所研究的算法及模型。针对不同的用户需求，从简洁的用户输入，到高自定义的用户输入需求，用户可以输入不同的提示信息与系统进行交互，主要包括三种交互方式，如表 4.1 所示。

表 4.1 AI 写作系统的三种交互方式

交互方式	输入信息
标题生成	标题、关键词
摘要生成	标题、关键词、摘要（检索/自定义）
大纲生成	标题、关键词、摘要（检索/自定义）、大纲（检索/自定义）

针对以上三种不同程度的需求，本文构建一个通用的生成模型(图 4-3)，由必要输入、自定义输入以及输出三个部分组成。

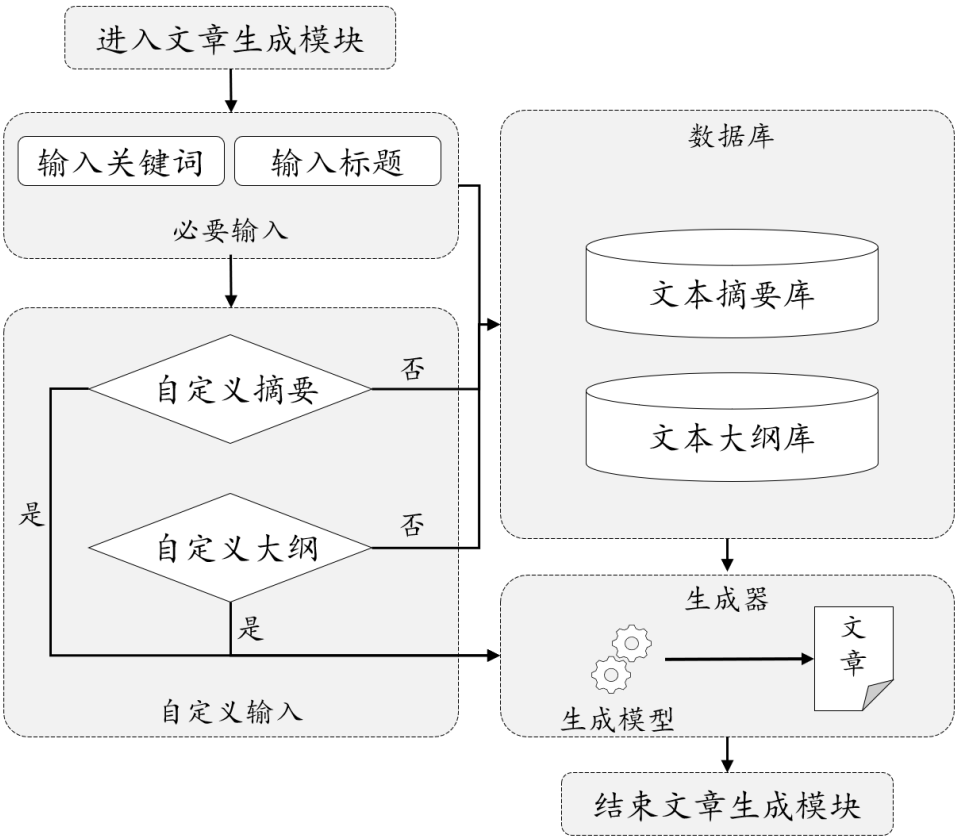

图 4-3 AI 写作模块架构图

其中，主要通过自定义输入部分，来实现对三种不同需求的调控。首先通过必要输入模块获取用户输入的标题与关键词，再由用户是否自定义摘要与大纲，判断是否需要进行摘要和大纲的抽取，最后将以上获取到的信息，输入到生成模型中，进行整篇文章的生成。

4.2.3 文本摘要模块

文本摘要模块根据用户输入的大段文字，进行摘要和关键词的提取，从而帮助快速了解长文本的主要内容。本章实现的文本摘要模块主要使用了 TextRank 算法，进行关键词和关键句的提取。TextRank 是一种常用的抽取式文本摘要技术，它基于 Google 最初提出的应用于网页检索的网络排名算法 PageRank，是一种基于图的排序技术，可以自动抽取一篇文章中的重要句子或词。

TextRank 算法的基本思想是将文本中的不同语义单元（如词、短语、句子等）作为图的顶点，将这些语义单元之间的关系作为顶点之间的边，当一个顶点连接到另一个顶点时，实际上是在做对另一个顶点的“投票”。某个顶点的的投票数越多，则这个顶点在整个文本中的重要程度越高。此外，该顶点本身的重要程度对于所投的票的重要性也有影响。因此，顶点的重要性分数是根据投票来决定的，这包括两方面，一方面是为它投票的顶点个数以及为他投票顶点本身的重要性。

传统的 PageRank 在构建图的过程中使用的是无权图，但是对于自然语言来说，指明两个顶点V_i和V_j之间所对应的权重w_{ij}是有必要的。因此，在 TextRank 计算图中顶点的得分时，需要在 PageRank 的基础上加入权重，计算某个顶点V_i的重要性公式如下所示：

$$WS(V_i) = (1-d) + d * \sum_{V_j \in In(V_i)} \frac{w_{ij}}{\sum_{V_k \in Out(V_j)} w_{jk}} WS(V_j) \tag{4.1}$$

其中d是一个介于 0 到 1 之间的阻尼系数，它控制从当前顶点跳转到图中另一个随机顶点的概率，一般设置为 0.75。w_{ij}为顶点V_i和V_j之间的权重，$WS(V_j)$是指向该顶点的重要性程度。$In(V_i)$表示所有指向v_i的顶点，$Out(V_j)$指v_j顶点所有指向的顶点。

由以上的方法计算出图中每个顶点的重要性后，对顶点进行排序，若某个顶点的重要性最高，则该顶点所代表的文本单元在文章中是最重要的，从而可以提取出文章中的重要文本单元作为文章的摘要。

4.3 系统实现

4.3.1 开发环境

本文主要采用基于 Python 的 Django 框架进行系统的开发，系统部署采用微服务架构，将客户端部署在云服务器，使用 Nginx 进行反向代理，后台服务部署在本地的 GPU 服务器，使用 UWSGI 提供接口服务。整个系统的开发环境如表 4.2 所示。

表 4.2 AI 写作系统开发环境

服务器	环境	说明
云服务器	CPU	Intel(R) Xeon(R) 8255C @2.50GHz
	内存	4GB
	系统	Centos7
	Nginx	1.14.0
本地服务器	CPU	Intel(R) Xeon(R) W-3223 @3.50GHz
	内存	64GB
	GPU	NVIDIA RTX A4000 16G
	系统	Centos7
	MySQL	5.7.39
	Python	3.7
	UWSGI	2.0.21
	Django	3.2.5

基于以上环境，将系统在本地开发完成后进行线上部署。

4.3.2 系统展示

1. 主页

系统的主页面如图 4-4 所示，界面上方是本系统的导航条，包括系统名称、主页、热点发现、文章生成以及文本摘要，右上方是用户个人中心。同时，主页中还包括一个“文本续写”的展示功能，以及对各个功能模块的大致介绍。

图 4-4 AI 写作系统主页

2. 热点发现

热点发现模块的界面如图 4-5 所示，包括各个平台实时更新的新闻，用户点击对应新闻的链接后，可以跳转到对应的新闻界面，获取写作的素材以及灵感。

图 4-5 热点发现界面

3. 文本生成

文本生成模块是本文写作系统的主要功能模块，包括标题生成、摘要生成、大纲生成三个界面，每个界面中用户输入的内容各不相同。由于大纲生成界面包含标题生成与

大纲生成的相关内容，本文仅展示大纲生成的界面。如 4-6 所示，用户需要输入标题与关键词，可以选择检索或自定义摘要与大纲，点击一键生成文章按钮后，会生成一篇完整的文章展示在右侧的输出框内。具体的实现效果如图 4-7 所示。

图 4-6 大纲生成界面

图 4-7 大纲生成模块具体效果展示

4. 文本摘要

文本摘要模块的展示界面如图 4-8 所示，用户可以在文本输入框中，输入大段的长文本，点击按钮后，系统将自动从输入文本中抽取出摘要与关键词，并在输出框中进行内容的展示，该模块的具体效果展示如图 4-9 所示。

图 4-8 文本摘要界面展示

图 4-9 文本摘要模块具体实现效果

4.4 本章小结

本章从系统的总体设计出发，给出了功能及技术的完整框架，在此基础上，基于本文介绍及提出的模型，设计并实现了各个具体的功能模块，并结合 Django、UWSGI、Nginx 等技术完成了对系统的搭建。最后对系统的功能模块进行简略地展示，在验证算法和模型有效性的同时，也展示了本文提出的 AI 写作系统在日常生活中的实用性。

5 总结与展望

5.1 全文总结

基于当今自然语言处理尤其是自然语言生成技术快速发展的时代背景，本文研究了文本检索、文本生成、文本摘要等相关技术，介绍了本文提出的结合倒排索引与 Hash 编码的文本检索、基于大纲规划的文本生成两种模型，构建了一个旨在为用户提供写作辅助的 AI 写作系统。本文的主要工作如下：

1. 提出了一种结合倒排索引与 Hash 编码的文本检索模型。对于文本检索任务，本文对传统的 SimHash 检索方式进行改进，利用海明距离替代编码段匹配的相似性计算方式，并结合倒排索引的快速检索能力与 Hash 编码的语义匹配能力，实现由初筛到精筛的双层文本检索模型。该文本检索模型在保证速度的前提下，也实现了内容的精准匹配。实验结果证明了本文检索模型的有效性，在将检索时间控制在 500ms 以内的情况下，本文模型检索出的内容比仅用倒排索引检索的内容更精炼，丰富。

2. 设计并实现了一个基于大纲规划的 AI 写作系统。该系统使用本文提出的文本生成模型作为核心模块，结合网络爬虫、文本摘要等技术进行热点发现、文本摘要模块开发。利用分布式微服务的架构以及 Nginx、UWSGI 等常用的系统搭建手段进行系统的部署搭建，以满足高可用性、并发性、可扩展性等需求。通过接口测试等一些系统测试手段，验证了系统的可靠性、可用性等优势。通过对用户的需求调研以及页面的设计，搭建出友好的前端交互界面，展示了系统在日常生活中的实用性，可以实时、准确地满足辅助用户写作的需求。

5.2 未来工作展望

本文主要对文本检索、可控文本生成等技术进行了研究，提出了两个模型，用于搭建 AI 写作系统。在研究过程中，发现仍有一些不足之处，值得成为未来的研究方向，具体描述如下：

1. 本文所研究的文本检索方式，是利用 Hash 编码来进行文本语义性表示，在现有的文本语义表示方法中，Hash 编码的效果并不是很好，主流的基于高维向量如 Word2Vec、BERT 等向量的表示方法，能更好地表示文本的语义信息。所以如何利用高维的向量更快速、精准地实现文本的检索，还需要在未来的工作中进行深入研究。

2. 对于大纲规划，本文采用的是检索式的规划方法，这种规划方式对于语料的质量要求较高，而对数据的清洗与对大纲的提取是很繁琐的工作，所以在对大纲进行规划的阶段，有进一步的改进空间，我们是否可以采用生成式的大纲规划方式，如果可以，如何进行生成式的规划，是未来研究的重点方向。

3. 对于检索出的大纲，本文采用的评估方式主要是人工评估，然而人工评估存在工作量大、评价结果不客观等缺点。因此，我们希望能找到一种机器评估方法，以更加合理的方式对规划出的大纲进行分析和评测。

参考文献

[1] Prabhumoye S, Black A W, Salakhutdinov R. Exploring Controllable Text Generation Techniques, 10.18653/v1/2020.coling-main.1[P]. 2020.

[2] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in Neural Information Processing Systems, 2020, 33: 9459-9474.

[3] Xu P, Patwary M, Shoeybi M, et al. MEGATRON-CNTRL: Controllable story generation with external knowledge using large-scale language models[J]. arXiv preprint arXiv:2010.00840, 2020.

[4] Yang Y, Pan B, Cai D, et al. TopNet: Learning from Neural Topic Model to Generate Long Stories[C]. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 1997-2005.

[5] Yao L, Peng N, Weischedel R, et al. Plan-and-write: Towards better automatic storytelling[C]. Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33(01): 7378-7385.

[6] Zhai F, Demberg V, Koller A. Story generation with rich details[C]. Proceedings of the 28th International Conference on Computational Linguistics. 2020: 2346-2351.

[7] Rashkin H, Celikyilmaz A, Choi Y, et al. Plotmachines: Outline-conditioned generation with dynamic plot state tracking[J]. arXiv preprint arXiv:2004.14967, 2020.

[8] Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

[9] Mihalcea R, Tarau P. Textrank: Bringing order into text[C]//Proceedings of the 2004 conference on empirical methods in natural language processing. 2004: 404-411.

[10]XU Xintao, CHAI Xiaoli, XIE Bin，et al. Extraction of Chinese text summarization based on improved TextRank algorithm[J]. Computer Engineering，2019，45(3) : 273-277.

[11]李维, 闫晓东, 解晓庆. 基于改进 TextRank 的藏文抽取式摘要生成[J]. 中文信息学报, 2020, 34(9): 36-43.

[12]谷莹, 李贺, 祝琳琳. 融合主题聚类和语义图模型的产品评论自动摘要方法研究[J]. 图书

情报工作, 2022, 66(13): 118.

[13]Oliveira H, Ferreira R, Lima R, et al. Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization[J]. Expert Systems with Applications, 2016, 65: 68-86.

[14]Wang W M, Li Z, Wang J W, et al. How far we can go with extractive text summarization? Heuristic methods to obtain near upper bounds[J]. Expert Systems with Applications, 2017, 90(dec.30):439–463.

[15]Johnson J, Douze M, Jégou H. Billion-scale similarity search with gpus[J]. IEEE Transactions on Big Data, 2019, 7(3): 535-547.

[16]Manku G S, Jain A, Das Sarma A. Detecting near-duplicates for web crawling[C]. Proceedings of the 16th international conference on World Wide Web. 2007: 141-150.

[17]孙宇, 刘憬, 张宇, 等. 基于分词和倒排索引的短文本检索技术的研究与实现[J]. 黑龙江省计算机学会 2007 年学术交流年会论文集, 2007.

[18]Lu W, Robertson S, Macfarlane A. Field-weighted XML retrieval based on BM25[C]. Advances in XML Information Retrieval and Evaluation: 4th International Workshop of the Initiative for the Evaluation of XML Retrieval, INEX 2005, Dagstuhl Castle, Germany, November 28-30, 2005. Revised Selected Papers 4. Springer Berlin Heidelberg, 2006: 161-171..

[19]Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. Advances in neural information processing systems, 2017, 30.

[20]Radford A, Narasimhan K, Salimans T, et al. Improving language understanding by generative pre-training[J]. 2018.

[21]Radford A, Wu J, Child R, et al. Language models are unsupervised multitask learners[J]. OpenAI blog, 2019, 1(8): 9.