文章编号：1003-0077（2017）00-0000-00

**基于大语言模型的BCC语料库自然语言检索**

刘廷超1 鲁鹿鸣1 荀恩东2 靳泽莹1 杨兆勇1

（1.北京语言大学 信息科学学院，北京市 100083；

2.北京语言大学 语言资源高精尖创新中心，北京市 100083）

**摘要：** 语料库在语言学和自然语言处理领域至关重要。北京语言大学的BCC语料库，资源丰富且检索高效，备受推崇，然而，其BCC检索式的复杂性限制了普及。为此，本文提出TextToBCC模型，目标是实现自然语言对BCC语料库的检索。首先构建了一个均衡的BCC检索式数据集，利用大语言模型为BCC检索式生成了自然语言描述。随后，微调大语言模型进使其能够支持自然语言到BCC检索式的转换。实验结果证明了TextToBCC模型的优异性能。这一成果不仅降低了BCC语料库的使用难度，而且有助于促进其在更广泛领域的传播和应用，为语言学研究和自然语言处理实践带来便利。

**关键词：**语料库；检索式；大语言模型；微调

**中图分类号：**TP391  **文献标识码：**A

#### Natural Language Retrieval of BCC Corpus Based on Large Language Models

Tingchao Liu1, Luming Lu1, Endong Xun2, Zeying Jin1, Zhaoyong Yang1

(1. School of Information Science, Beijing Language and Culture University, Beijing, 100083, China;

2. Advanced Research Center of Language Resources, Beijing Language and Culture University,

Beijing, 100083, China)

**Abstract:** Corpora are crucial in the fields of linguistics and natural language processing. The BCC corpus of Beijing Language and Culture University is highly respected for its rich resources and efficient retrieval. However, the complexity of its BCC search query limits its popularity. To this end, this paper proposes the TextToBCC model, which aims to achieve natural language retrieval of the BCC corpus. First, a balanced BCC search queries dataset was constructed, and a natural language description was generated for the BCC search queries using a large language model. Subsequently, the large language model was fine-tuned to enable it to support the conversion from natural language to BCC search queries. Experimental results demonstrate the excellent performance of the TextToBCC model. This achievement reduces the difficulty of using the BCC corpus, and helps promote its dissemination and application in a wider range of fields, bringing convenience to linguistic research and natural language processing practice.

**Key words:** corpus; search query; large language models; fine-tuning

**0 引言**

在信息时代的浪潮下，语言数据的海量增长为语言学研究和自然语言处理带来了前所未有的机遇与挑战。其中，语料库作为语言研究中的一种重要工具，其价值和重要性日益凸显。

目前，中文语料库的建设已取得了显著成果，如北京语言大学的BCC语料库[1][2][3]（Beijing Language and Culture University Corpus Center，BCC）、北京大学的CCL现代汉语语料库[4]（Center for Chinese Linguistics Peking University，CCL）以及国家语言资源动态流通语料库[5]（Dynamic Circulation Corpus，DCC）等。这些语料库各具特色，涵盖了丰富的语言资源，为不同领域的研究和应用提供了有力支持。在使用语料库进行语料检索时，用户必须熟悉并掌握相应的检索式规则和使用方法。例如，在BCC语料库中查找以“跑”为首的双音节动词时，需编写检索式“跑./v”。而要查找结构为“名词+山+名词+海”的词组，并且要求两个名词相同，检索式则为“(n)山(n)海{$1=$2}”。掌握这些检索式的语法规则需要投入一定的学习成本，这无疑增加了用户在使用语料库时的负担。

在自然语言处理领域，自然语言查询正逐渐成为一种重要的查询方式，并获得了广泛的关注。它使用户能够以日常语言表达查询需求，显著降低了使用门槛。近年来，文本到SQL的转换任务取得了显著进展，可以将自然语言查询转换为SQL语句，实现对数据库的高效检索。然而，由于语料库没有统一的检索语言，自然语言查询到语料库检索式的转换研究仍然十分有限。

鉴于此，本文致力于探究自然语言到语料库检索式转换的研究任务，旨在打造一个大语言模型，以实现从自然语言到语料库检索式的转换，降低语料库使用的难度。为实现这一目标，本文选取BCC语料库作为研究对象，开发了TextToBCC模型，为BCC语料库引入自然语言查询功能。

首先，构建了一个庞大的数据集，其中包含了从BCC语料库查询日志中筛选出的用户查询实例。通过对这些数据的深入分析和整理，揭示了用户查询行为的多样性和复杂性，并针对数据稀缺的挑战，运用检索式生成技术进行了有效的数据增强。

接着，设计了一种基于大语言模型的检索式到自然语言的转换方法，通过编写提示语（Prompt），实现了将BCC检索式转化为自然语言的功能。在模型训练阶段，采用了微调现有大语言模型的策略，将BCC检索式数据集转换为适合特定模型的微调数据集，以适应自然语言查询到BCC检索式的转换任务。为了进一步优化模型的性能和提升其能力，还将错误日志中的错误BCC检索式及其对应的纠错信息纳入到大语言模型的微调过程中。这一策略使得大语言模型具备了自动纠正BCC检索式错误的功能。

最后，经过一系列实验验证，本文提出的TextToBCC模型在自然语言到BCC检索式转换任务上展现出了卓越的性能。

本文不仅为语料库的普及和应用开辟了新的途径，同时也为语言学研究、自然语言处理等领域的用户提供了一个高效、便捷的检索工具。通过降低语料库的使用门槛和提高检索效率，本文的方法有望促进相关领域的研究和应用实现更大的突破。

**1 研究背景**

随着信息技术的飞速发展，语料库的规模持续扩大，其重要价值也随之日益凸显。但语料库检索式的多样性在一定程度上限制了它们的普及和应用。因此，人们渴望借助自然语言查询技术来简化语料库的使用流程。随着大语言模型技术的不断进步，这一愿望正在逐步成为可能。

**1. 1 语料库**

语料库是按照一定采样标准采集，能够代表一种语言或语言的一种变体或文类的电子文本集[6]。语料库作为提取语言信息的枢纽，在语言数据挖掘过程中发挥着重要作用[7]。借助语料库，研究者能够对语言的规律性、变异性及历史演变进行深入剖析。此外，语料库为语言教学提供辅助手段[8]。在自然语言处理领域，语料库是训练语言模型和开发智能应用不可或缺的基础[9]。

目前，常用的中文语料库,各自在资源和功能上都有独到之处，但它们都依赖于检索式来进行信息检索。语料库检索式是用户查询语料库并获得所需信息的语言工具。掌握语料库检索式需要一定的语言学知识，包括检索式的语法结构、词性标注以及特殊符号的正确使用等，这对于非专业用户而言可能构成挑战。

语料库的检索方法存在显著差异。CCL语料库和BCC语料库都使用了自有的检索式体系，它们在语法规则和使用方法上有较大不同。DCC语料库则采用了基于语料库查询语言（Corpus Query Language，CQL）扩展的检索语言[10]，许多其他语料库也采用CQL或其扩展作为查询语言。尽管语料库检索式的多样性体现了它们强大的功能，但这也暴露了语料库检索语言非标准化的问题。对用户来说，这意味着在使用不同的数据库时，他们需要投入时间和精力去学习和适应每个语料库特有的检索语法和使用规则。

**1. 2 自然语言查询**

自然语言查询（Natural Language Query, NLQ）是以自然语言表达为基础的查询，是自然语言处理的内容之一。在人工智能中，通过用户输入自然语言表达，使计算机能接受和处理自然语言，还能对自然语言进行信息加工，以及能理解自然语言的过程，从而对信息进行反馈，这样的过程称为自然语言查询。

典型的NLQ任务是文本到SQL[11]的转换，实现自然语言查询到SQL语句的转换，使得开发者和最终用户都能够轻松地使用自然语言方式完成数据库查询操作。

本文提出了第一个文本到CQL任务，旨在将自然语言自动翻译成CQL。我们为这项任务提出了一个全面的框架，包括一个专门策划的大规模数据集和利用大型语言模型(llm)进行有效的文本到CQL任务的方法。此外，建立了先进的评估指标，以评估生成的查询的语法和语义准确性。我们创建了创新的基于llm的转换方法和详细的实验。结果证明了我们的方法的有效性，并提供了对文本到CQL任务复杂性的见解。

1 介绍

本文提出了第一个文本到CQL任务，旨在将自然语言自动翻译成CQL。我们为这项任务提出了一个全面的框架，包括一个专门策划的大规模数据集和利用大型语言模型(llm)进行有效的文本到CQL任务的方法。此外，建立了先进的评估指标，以评估生成的查询的语法和语义准确性。我们创建了创新的基于llm的转换方法和详细的实验。结果证明了我们的方法的有效性，并提供了对文本到CQL任务复杂性的见解。

1 介绍

在使用自然语言实现语料库查询方面，文献[12] 提出将自然语言转换成CQL的方法。构建了一个CQL数据集，利用大型语言模型将文本转换为CQL语句。还建立了一组指标，评估生成的CQL的准确性。

**1. 3 大语言模型及微调**

大语言模型(Large Language Models, LLMs)是近年来自然语言处理领域取得重大突破的关键技术之一[13]。LLMs通过深度学习的方法，在海量文本数据上进行训练，从而能够生成和理解复杂的自然语言文本。这些模型通常包含数十亿甚至数万亿的参数，使得它们能够捕捉语言中的细微模式和长距离依赖关系，从而在多种自然语言处理任务上展现出前所未有的性能。

随着计算能力的提升和大规模数据集的出现，LLMs不仅在语言理解（如问答系统、情感分析）方面取得了显著成效，还在语言生成（如文本创作、摘要生成）领域展现出了强大的能力。LLMs的核心在于其规模，巨大的参数空间允许模型捕捉更丰富的语言特征和上下文信息。模型首先在大量未标注文本上进行预训练，学习语言的通用表示；随后，通过针对特定任务的微调，快速适应不同场景的需求。LLMs展现出了零样本或少样本[14]学习的能力，即无需标注数据或仅需少量标注数据即可完成新任务，极大地降低了自然语言处理应用的门槛。

模型微调（Fine-tuning）作为LLMs应用中的关键技术，通过利用特定领域的数据对预训练模型进行针对性优化，显著提升模型在特定任务上的表现。然而，对于庞大的LLMs而言，直接微调往往伴随着高昂的计算成本和过拟合风险。为此，研究者们不断探索新的微调策略，产生了很多微调方法，比较流行的有：Prefix-Tuning[15]、Prompt Tuning[16]、P-Tuning[17]、P-Tuning v2[18]等方法，其中LoRA（Low-Rank Adaptation）微调[19]尤为引人注目。

LoRA微调通过在模型中引入可训练的低秩矩阵，有效减少了微调所需的参数量，这样不仅保持了性能的提升，还大幅度降低了计算复杂度和显存需求。这种方法不仅提高了微调过程的效率和稳定性，还使得LLMs能够在更多资源受限的环境中得到应用。目前，许多开源的LLMs已经提供了LoRA微调的示例代码，为研究者和开发者提供了便利的实践途径。

**2 构建BCC检索式数据集**

通过对BCC语料库的检索日志和错误日志的分析，构建了一个包含正确检索式和错误检索式两大类别的检索式数据集。首先，对这两类日志数据进行分类整理，随后依据预设比例从各分类中提取样本。为了确保数据集的均衡性，还设计了检索式生成策略，用以有效补充那些数据量不足的分类。最终，建立了一个平衡且全面的BCC检索式数据集。BCC检索式数据集的生成流程如图1所示。

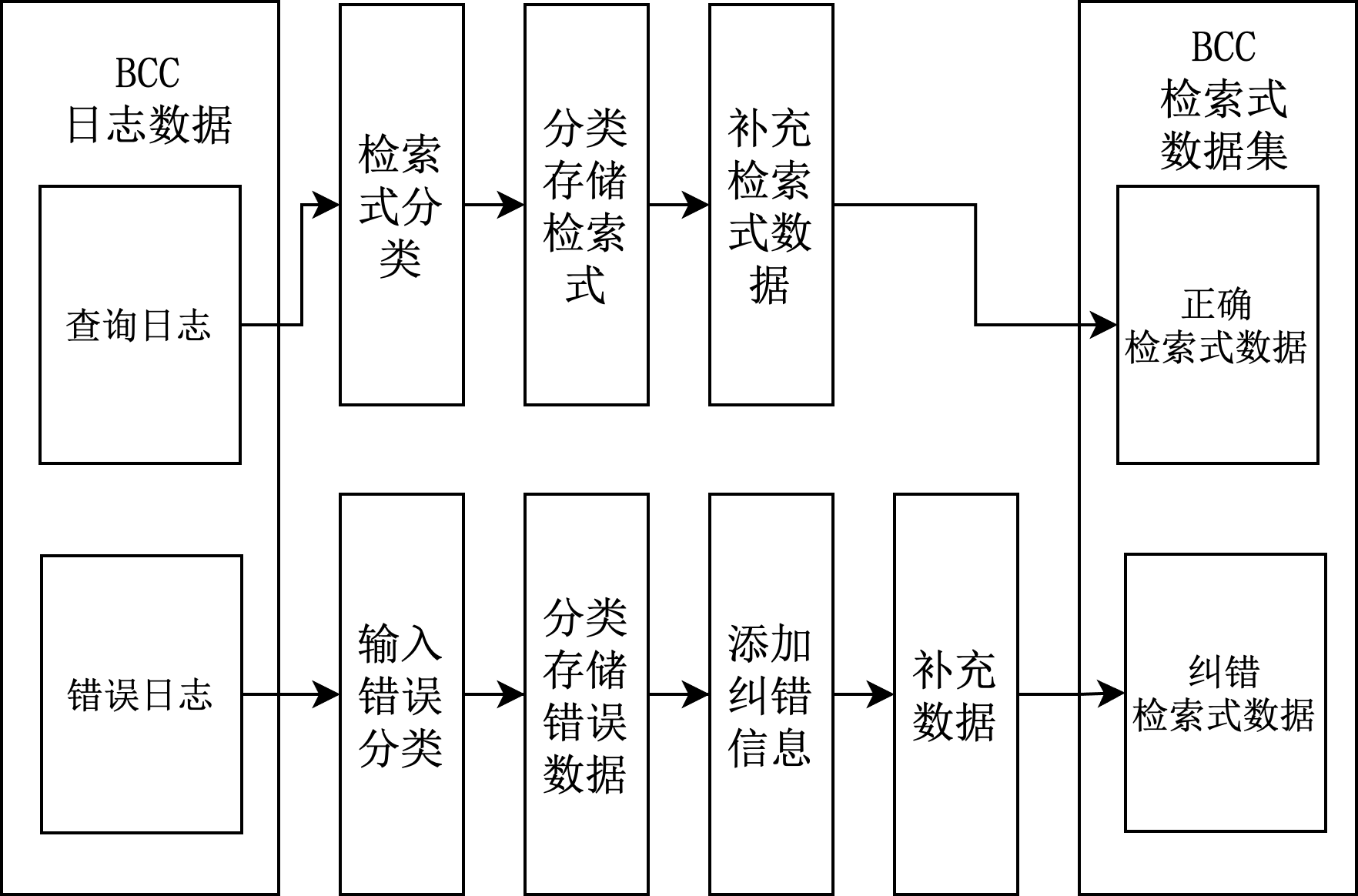


图1 BCC检索式数据集生成流程图

**2. 1 BCC语料库**

BCC语料库是以汉语为主、兼有其他语种的在线语料库。该语料库以其庞大的规模和广泛的语料选材，全面反映了当代社会的语言生活，具备数据量大、覆盖领域广和检索便捷等显著优点。自问世以来，BCC语料库为语言本体研究、语言应用研究以及语言教学提供了坚实的数据和技术支持。

BCC语料库使用检索式进行语料检索，检索式是一种通过组合字符、词串、词性等属性符号来执行语料库查询的方式。BCC检索式的构成包括查询对象、限制条件和功能操作三个部分。其语法如下：Query{Condition1; Condition2; …} Operation。其中，Query代表查询对象；{}内的内容为限制语句，Condition指代检索内容的条件限定；Operation表示功能操作。

查询对象由汉字串、词性符号以及一些特殊符号构成。汉字串代表检索的文本内容，词性符号则是词语的词性标签，如“n”表示名词，“v”表示动词。一个BCC检索式可以包含多个词性标签。特殊含义符号用于对检索内容进行抽象或限定，包括通配符、集合符、限定符、离合符、属性约束符号、分隔符等。

限制条件通过使用限制函数对内容进行限定，多个限制条件之间用“;”分隔，每个Query最多可包含两处限定条件，受限部分用()括起来，并根据括号的顺序，用$加序号进行指代。限制条件支持的函数分为内容限制、频次限制和长度限制三个类别。

功能操作用于指定检索式的检索目的，可以是实例检索（Context）、频次统计（Freq）或历时检索（Count）。如果没有指定功能操作，默认为实例检索。

在使用BCC语料库进行检索时，用户需编写BCC检索式来执行查询操作。BCC检索式的示例详见表1。

表1 BCC检索式示例

|  |  |
| --- | --- |
| BCC检索式 | 中文表达 |
| 高大的n | 高大的+名词 |
| v了一v | 动词+了一+动词 |
| 见\*面 | 见后面离合出现面 |
| 洗.澡 | 洗后面隔一个字后接澡 |
| ../v | 二字动词 |
| 爱(v)不(v){$1=$2} | 爱+动词+爱+动词，并且两个动词相同 |
| (nr) 说 m q {len($1)>1;  begin($1)=[老]} | 人名加说加数词再加量词，人名长度大于1并且以“老”字开头 |

**2.2 BCC检索日志**

本文从BCC语料库的检索日志中抽取了五万条用户查询实例，构建了研究的核心数据集，这一数据集反映了用户的检索策略和偏好。经过详尽的分类分析，将检索式细分为六个类别，并对各类别的数量进行了详尽的统计，这为后续的数据挖掘和模型训练打下了坚实的基础。为了解决数据稀缺的问题，运用数据生成技术进行了数据增强，以此确保了数据集的完备性与均衡性。另外，还从检索错误日志中提取了一万条数据，并在此基础上设计了错误检测与纠正机制，旨在提升用户的检索体验和系统的互动性。经过这一系列工作，打造了一个高质量的BCC检索式数据集。

2.1.1 BCC检索式分类

本文对BCC检索式进行了分类，依据其构成元素的不同，将检索式划分为六个类别：

1）纯字符串检索式：仅包含待搜索的字符，无特殊符号，体现了最直接的检索需求。

2）词性标签检索式：结合了检索字符与词性标签，体现了用户对词性信息的关注。

3）通配符检索式：包含如“.”“~”等通配符号，用于模糊匹配，增加检索灵活性。

4）集合符检索式：利用“[]”表示检索单元的取值范围，实现特定集合内的精确检索。

5）属性约束检索式：通过“/”对检索单元施加属性约束，如词语、词性等，增强检索的精确性。

6）限制条件检索式：利用“{}”引入内容、长度、频率等限制条件，对检索结果进行进一步筛选。

值得注意的是，除纯字符串检索式外，其余类别均可能融入词性标签，这反映了用户检索时高度依赖词性标签。

基于上述分类，对检索日志数据进行了详尽的统计，分析各分类中不重复检索式的数量及占比，结果如表2所示。

表2 BCC检索式使用分布

|  |  |  |
| --- | --- | --- |
| 类型 | 数量 | 比例 |
| 字符串 | 14767 | 41.18% |
| 词性标签 | 8716 | 24.30% |
| 通配符 | 836 | 2.33% |
| 集合符 | 8065 | 22.49% |
| 属性约束 | 561 | 1.56% |
| 限制条件 | 2918 | 8.14% |

数据揭示了一个明显的趋势：用户在使用BCC检索式的不同类别时表现出显著的不平衡性，尤其是属性约束和限制条件检索式的数据量远远少于其他类别。考虑到这种数据分布的不均衡性，采取了人工合成数据的策略，针对性地扩充了属性约束和限制条件等类别的数据量。这一措施确保了模型在训练过程中能够接触到全面而均衡的数据资源，进而提升模型的整体性能。

2.1.2 建立纠错数据集

在检索错误日志中，用户输入的检索式存在语法错误或不当使用符号的情况。这些错误不仅削弱了检索效率，还有可能误导用户进行错误的查询。为了解决这一问题，对这些错误的检索式进行了系统的整理和分类，并尝试对其进行修复或给出错误提示。

表3 BCC检索式错误示例

|  |  |  |
| --- | --- | --- |
| 类型 | 检索式 | 修复及提示 |
| 函数遗漏 | (v)着{($1)>  1} | (v)着{len  ($1)>1} |
| 符号重复 | 诞辰\*\*\*\*周年 | 诞辰\*周年 |
| 符号缺失 | 没n就v  {$1=$2} | 没(n)就(v){$1=$2} |
| 语法错误 | 要是{len(30} | 语法错误，缺少被限定的单元 |
| 语法错误 | 一n一个--人 | 语法错误，存在无效符号“-” |
| 语法错误 | 撞见^^被 | 语法错误，存在无效符号“^” |

对于那些可以被修复的检索式，直接提供了修正后的正确形式；而对于那些无法立即修复的检索式，则提供了清晰的错误提示，以指导用户进行必要的修正。最终，将这些修复后的检索式连同相应的错误提示以及原始的错误检索式一并存储，形成了BCC检索式纠错数据集。这一数据集为后续的错误自动修正工作提供支持。

**2. 2 BCC检索式生成**

2.2.1 生成例句标注数据集

为了确保训练数据的全面性和代表性，从BCC语料库中精心挑选了2万条句子，这些句子的长度介于5至15个汉字之间，并涵盖了文学、报刊和对话等多个领域，从而保证了数据的多样性和广泛性。

BCC语料库采用的是北京大学的词性标签体系[20]。本文采用了北京语言大学开发的基于网格的自然语言结构分析框架（Grid-based Parsing Framework, GPF）[21]对这些句子进行了分词和词性标注。值得注意的是，GPF同样采用了北京大学的词性标签体系，这确保了句子的标注结果与BCC语料库中的词性标注保持一致。本文使用GPF工具[[1]](#footnote-2)对所有的例句进行分词和词性标注，并将结果保存为句子标注数据集。

2.2.2 BCC检索式生成策略

为了增强训练数据的丰富度和确保分类的均衡性，设计了一套BCC检索式生成规则。具体操作如下，随机从例句中选取两个位置（标记为a和b），然后应用多种生成规则来构造多样化的检索式：

1）纯字符串检索式：直接采用位置a和b处的词语作为检索式，生成基础的字符串检索式。

2）词性标签检索式：将位置a和b的词语替换为词性标签，生成带有词性信息的检索式。

3）通配符检索式：在位置a处用通配符替代具体词语，生成包含通配符的检索式。

4）集合符检索式：引入具有相似词性和相似词语的备选项，生成带有集合限制的检索式。

5）属性约束检索式：为a和b位置的词语添加词性信息，生成带有属性限制的检索式。

6）限制条件检索式：将位置a和b替换为词性标签，并增加内容、长度、频次等限制，生成带有限制性条件的检索式。

2.2.3构建检索式数据集

以索日志数据为基础，采用BCC检索式生成策略构建数据集。为确保各类检索式在数据集中均匀分布，依照2:2:1:2:1:1的比例对它们进行了分配和补充，最终打造出一个包含18000条记录的检索式数据集。

这些数据以JSON格式进行存储，详尽记录了每个检索式及其对应的原始例句。这种存储方式不仅便于数据的快速检索与处理，而且确保了数据的可追溯性和可验证性。这个均衡的检索式数据集将为后续的模型训练与评估提供强有力的数据支撑。

**3 TextToBCC模型训练**

为了实现自然语言到BCC检索式的转换，本文提出了TextToBCC模型，该模型是通过对开源大模型进行微调实现的。在训练TextToBCC模型的过程中，需要使用带有自然语言描述的BCC检索式数据。为此，首先使用大语言模型为BCC检索式数据成自然语言描述。接下来，将纠错检索式数据与添加自然语言描述的BCC检索式数据一起组合生成微调数据集。然后使用该数据集对大语言模型进行微调。最终，将微调完成的TextToBCC模型部署为在线服务，实现了从自然语言到BCC检索式的转换过程。整个TextToBCC模型的训练流程，如图2所示。

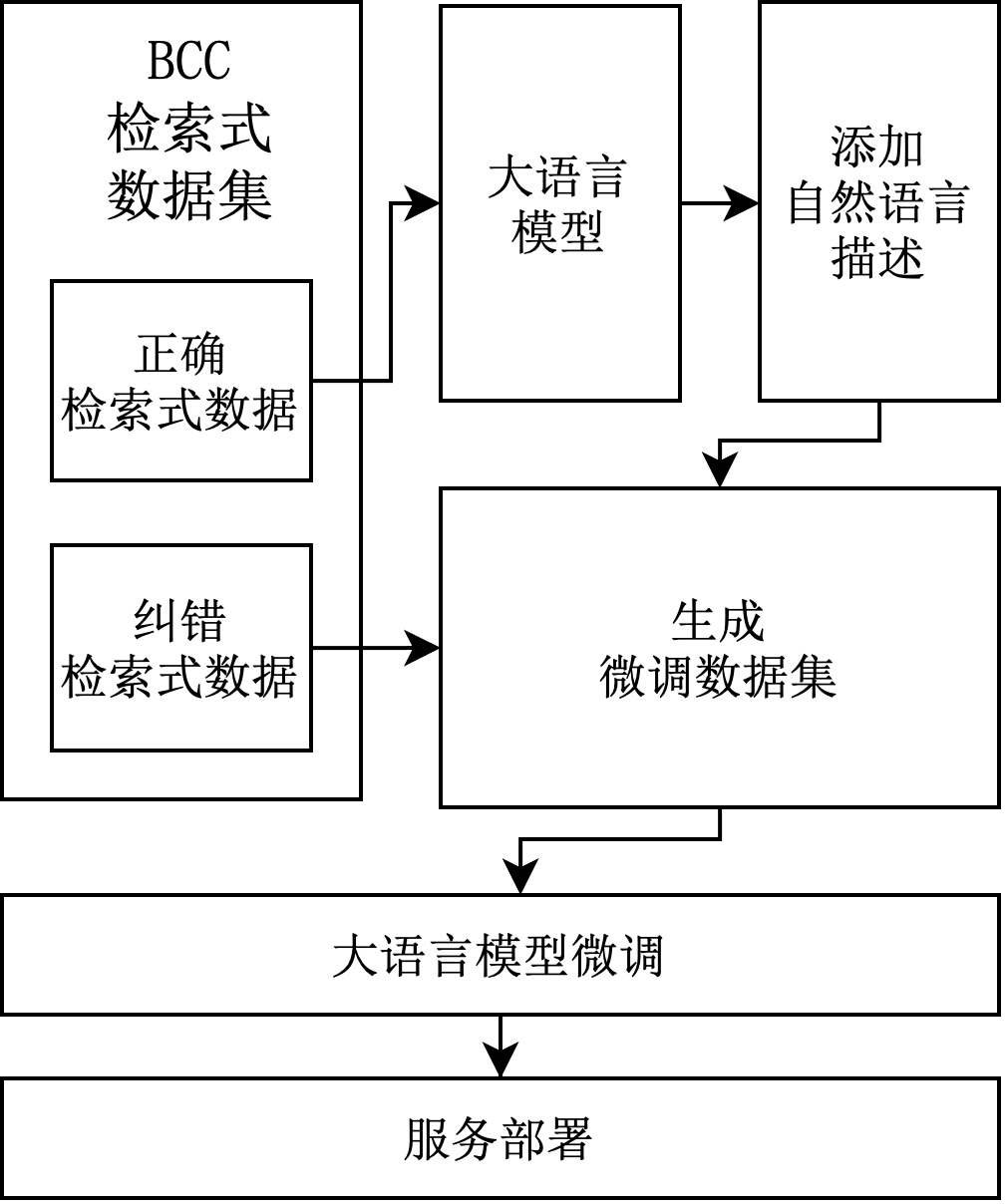


图2 TextToBCC训练流程示意图

**3. 1 生成自然语言描述**

为了获取BCC检索式的自然语言描述，本文探讨了利用在线大型语言模型服务来生成这些描述的方法。这一过程的关键在于设计有效的提示，以指导大型语言模型准确地生成BCC检索式的自然语言描述。提示的设计应包含以下几个关键要素：

1）格式与结构解析：明确定义BCC检索式的格式，阐明其结构组成和元素间的逻辑关系，确保提示能够帮助模型理解并准确反映检索式的语法结构。

2）特殊符号与规则阐述：详尽阐释检索式中使用的特殊符号及其应用规则，协助模型构建必要的上下文理解，以便生成更为精确的自然语言描述。

3）示例引导：提供一系列具体的BCC检索式及其对应的自然语言描述作为参考示例，为模型生成过程提供直观的学习样本。

4）输出规范：明确规定模型输出内容的JSON格式要求，包括字段名称、数据类型等，以确保生成的数据在后续处理中保持一致性。

完成提示设计后，采用通义千问[[2]](#footnote-3)以及ChatGLM[[3]](#footnote-4)这两款在线大语言模型来生成BCC检索式对应的自然语言描述。

为了高效生成自然语言描述，通过API编程接口与大语言模型进行交互，将设计好的提示和待转换的BCC检索式一同发送至模型。在接收到模型返回的数据后，提取其中的JSON部分，将其保存至微调数据集中。大语言模型生成的自然语言描述示例,如表4所示。

表4 自然语言描述示例

|  |  |  |
| --- | --- | --- |
| 检索式 | 通义千问 | ChatGLM |
| ./d高 ，支持率 ./d | 单音节副词后接高，逗号，支持率，再接单音节副词 | 单音节副词后接高，并且高后面是逗号和空格，接着是支持率，支持率后面再接一个单音节副词 |
| 投票率 ./[d v] 高 | 投票率后接单音节副词或动词，紧接着是高 | 投票率后面隔一个字接一个副词或动词，再接高 |
| ，~ 灿烂 | 句子中以逗号分隔，其后紧跟任何词性的一个词，最后是灿烂 | 逗号后面接一个词，再接灿烂 |
| (v)当代 (vn){len($1)=2; end($2)=[高 妈]} | 长度为2的动词后接当代，再接以高或妈结尾的名动词 | 动词后面接当代，当代后面接名动词，该动词是双音节，并且名动词以高、妈结尾 |
| 正确数量 | 4 | 3 |

在查看这两个大语言模型生成的自然语言描述时，发现通义千问模型在描述BCC检索式方面展现出了更为自然流畅的语言生成能力，并且具有较高的正确率。鉴于此，本文决定采用通义千问的生成结果。经过人工审核和必要的修正后，构建出一个适用于大语言模型微调的BCC检索式微调数据集。

**3. 2 生成TextToBCC模型**

本文期望TextBCC模型能够兼容BCC检索式和自然语言输入，并且能够自动纠正错误的检索式，对于无法自动纠正的情况，则返回恰当的错误提示。因此，在构建训练数据集的过程中，采取了以下步骤：

1）将BCC微调数据集中的BCC检索式同时作为模型的输入和期望输出，目的是让模型在接收到BCC检索式时能够直接输出相同的BCC检索式。

2）将BCC微调数据集中的自然语言描述作为输入，而将对应的BCC检索式作为期望输出，这样模型在面对自然语言输入时能够转换为相应的BCC检索式。

3）选取BCC微调数据集中的错误BCC检索式作为输入，将纠错后的检索式及错误提示作为输出，旨在训练模型识别并修正错误的BCC检索式。

在微调阶段，微调了ChatGLM3[22]与通义千问[23]两个开源模型。为了强化任务指向性，在每条数据前统一添加了指令前缀“请将下文解析成BCC检索式：”，以帮助模型准确把握转换的目标。最终，将数据集划分为训练（80%）、验证（10%）、测试（10%）三个子集，以便于模型的评估和优化工作。

3.2.1ChatGLM3微调

ChatGLM模型是由智谱AI公司开发的一款高性能大语言模型，它基于GLM130B千亿基础模型，拥有卓越的自然语言理解和生成能力。

在本文中，采用了ChatGLM3-6B模型，为了减小训练参数的数量和降低计算成本，运用了LoRA微调技术来进行模型的微调。首先，通过脚本程序对数据格式进行了转换，以确保训练数据集符合ChatGLM3的格式要求。接着，调整了微调配置文件（lora.yaml），设定了模型参数、优化器参数以及训练参数，其中最大迭代步数被设置为200,000步。最后，通过命令行运行微调脚本（finetune\_hf.py），启动了微调过程。在微调结束后，选择了表现最佳的微调权重，并将其整合到原始模型中，以此来增强模型在自然语言到BCC检索式转换任务上的性能。

3.2.2 通义千问微调

通义千问是由阿里云研发的一款超大规模语言模型，基于Transformer架构，由阿里巴巴自然语言处理实验室倾力打造，是国内规模领先的中文预训练模型之一。

本文选用Qwen-7B模型进行微调。首先，在数据准备阶段，依照Qwen微调数据的格式要求，为每条记录指定了唯一的ID，并将其转换为Qwen所需的格式。接着，采用了Qwen官方提供的LoRA微调单GPU脚本，对模型和数据路径参数进行了调整，并将最大训练轮数设置为50轮。最后，通过运行脚本启动了微调流程。微调完成后，将优化后的模型与原始模型合并，打造出了一个集成了强大基础能力和针对BCC检索式转换任务调整的模型。

**4 实验**

为了检验微调模型的BCC检索式生成效果，对ChatGLM3-6B微调模型和Qwen-7B微调模型进行了一系列测试，包括自然语言转换测试、BCC检索式纠错测试以及检索性能测试。

**4. 1 评价指标**

受单轮文本转SQL评估方法的启发，本文采用了精确匹配率（exact-set-match accuracy, EM）和执行正确率（execution accuracy, EX）作为评价指标[24]。

精确匹配率是指预测生成的BCC检索式与标准BCC检索式在字符串层面上完全一致的比率；而执行正确率则是指使用预测得到的BCC检索式执行查询能够得到正确结果的概率。

**4. 2 检索式转换测试**

利用测试集数据对微调后的ChatGLM3-6B和Qwen-7B模型在六类BCC检索式转换任务上的精确匹配率和执行正确率进行了评估。

经过测试，结果表明Qwen-7B微调模型在六种类型的BCC检索式转换任务上均表现出更高的性能，其平均精确匹配率和平均执行正确率均比ChatGLM3-6B微调模型高出大约三个百分点。这一结果清晰地表明，在BCC检索式转换任务的整体正确率上，Qwen-7B微调模型优于ChatGLM3-6B微调模型。各个分类的转换结果，如表5所示。

表5检索式转换结果

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Qwen-7B | | ChatGLM3-6B | |
|  | EM | EX | EM | EX |
| 字符串 | 88.20% | 89.47% | 86.04% | 86.42% |
| 词性标签 | 72.17% | 84.92% | 67.63% | 80.49% |
| 通配符 | 73.94% | 79.34% | 71.83% | 78.40% |
| 限制条件 | 60.96% | 74.07% | 57.25% | 70.68% |
| 集合符 | 65.85% | 72.70% | 63.41% | 69.69% |
| 属性约束 | 56.29% | 66.33% | 55.27% | 66.16% |
| 平均 | 69.61% | 77.74% | 66.85% | 74.96% |

**4. 3 检索式纠错测试**

在纠错测试环节，分别统计了两种错误修复的情况。提示正确率指的是当BCC检索式存在错误且无法自动修复时，模型能够给出正确提示的频率。纠错正确率则是指BCC检索式存在可自动修正的错误时，模型成功修正这些错误的比率。检索式纠错结果如表6所示。

表6 模型纠错结果

|  |  |  |
| --- | --- | --- |
|  | Qwen-7B | ChatGLM3-6B |
| 提示正确率 | 99.22% | 100% |
| 纠错正确率 | 100% | 99.00% |
| 平均 | 99.56% | 99.56% |

在纠错测试中，无论是Qwen-7B微调模型还是ChatGLM3-6B微调模型，都显示出了较高的错误识别和纠正能力。这两个模型的平均纠错能力均高达99.56%。

**4. 4 性能测试**

在性能测试方面，重点考察了将自然语言转换为BCC检索式的单条查询所需的平均响应时间，具体数据详见表7。

表7 模型响应时间

|  |  |  |
| --- | --- | --- |
|  | Qwen-7B | ChatGLM3-6B |
| 耗时（秒） | 1800.34 | 3348.32 |
| 样本（条） | 5012 | 5012 |
| 平均（秒） | 0.36 | 0.67 |

性能测试结果显示，Qwen-7B微调模型处理单条查询的平均响应时间为0.36秒，相比之下，ChatGLM3-6B微调模型的平均响应时间为0.67秒。在处理速度方面，Qwen-7B微调模型具有较为明显的优势。

**4. 5 结果分析**

在本实验中，对ChatGLM3-6B和Qwen-7B两个微调模型在检索式转换、检索式纠错以及性能表现方面进行了全面的评估。

在检索式转换测试中，Qwen-7B微调模型在准确性上超越了ChatGLM3-6B微调模型，特别是在字符串、词性标签和限制条件等类型的检索式转换任务上表现得尤为出色。

在纠错能力测试中，两个模型均显示出了极高的错误识别和修正能力，能够有效地处理并纠正BCC检索式中的错误。

在性能评估方面，Qwen-7B微调模型展现出了明显的优势，其平均响应时间显著低于ChatGLM3-6B微调模型，这表明Qwen-7B微调模型在处理速度和效率方面更为卓越。

综合以上分析，Qwen-7B微调模型在检索式转换的准确率和性能上均优于ChatGLM3-6B微调模型。因此，最终部署Qwen-7B微调模型作为TextToBCC模型，以提供自然语言到BCC检索式的转换服务。

**5 结论**

本文设计了TextToBCC模型为BCC语料库引入自然语言查询能力。通过构建一个全面且均衡的BCC检索式数据集，并对大型语言模型进行微调，成功地实现了自然语言向BCC检索式的转换功能。本文的主要贡献和结论包括：

1）构建了一个全面而均衡的BCC检索式数据集，为模型的训练提供了坚实的基础。

2）提出了BCC检索式与自然语言的联合训练方法，使得TextToBCC模型能够同时处理BCC检索式和自然语言查询，显著提升用户体验。

3）通过引入纠错数据，使模型具备了BCC检索式的纠错功能，提高了模型的实用性。

实验结果充分证明了TextToBCC模型的高性能，尤其是在纠错能力方面表现出众。这一研究有效地降低了BCC语料库的使用难度，并优化了用户体验。

综上所述，本文的研究为BCC语料库的普及和应用开辟了新的途径，为用户提供了高效、便捷的语料库检索工具。

**参考文献**

1. 荀恩东.自然语言结构计算BCC语料库[M].北京：人民邮电出版社，2023：52-104.
2. 荀恩东, 饶高琦, 肖晓悦, 等. 大数据背景下BCC语料库的研制[J]，语料库语言学，2016(1).
3. 荀恩东, 饶高琦, 谢佳莉,等. 现代汉语词汇历时检索系统的建设与应用[J]. 中文信息学报, 2015, 29(3):169-176.
4. 詹卫东,郭锐,常宝宝,等.北京大学CCL语料库的研制[J].语料库语言学,2019,6(01):71-86+116.
5. 朱君辉, 刘鑫, 杨麟儿, 等. 文心语料库检索平台的研制[C]. 第十二届全国语言文字应用学术研讨会, 2022.
6. 梁茂成, 李文中, 许家金. 语料库应用教程[M]. 北京：外语教学与研究出版社, 2010: 3-25.
7. 程学旗, 靳小龙, 王元卓, 等. 大数据系统和分析技术综述[J]. 软件学报, 2014, 25(9):1889–1908.
8. Alex Boulton. Corpora in language teaching and learning[J]. Language Teaching, 2017, 50(4): 483–506.
9. 奚雪峰, 周国栋. 面向自然语言处理的深度学习研究[J]. 自动化学报, 2016, 42(10):1445-1465.
10. 吴良平. CQP语法赋能语言研究及语言学习[J]. 语料库语言学, 2023, 10:98-114.
11. Bowen Qin, Binyuan Hui, Lihan Wang, et al. A Survey on Text-to-SQL Parsing: Concepts, Methods, and Future Directions. arViv, 2022, arXiv: 2208.13629v1.
12. Luming Lu, Jiyuan An, Yujie Wang, et al. From Text to CQL: Bridging Natural Language and Corpus Search Engine. ArXiv, 2024. abs/2402.13740
13. Shervin Minaee, Tomas Mikolov, Narjes Nikzad, et al. Large Language Models: A Survey. arViv, 2024, arXiv: 2402.06196v2.
14. Yisheng Song, Ting Wang, Subrota K Mondal, et al. A Comprehensive Survey of Few-shot Learning: Evolution, Applications, Challenges, and Opportunities. ArXiv, 2022. arXiv:2205.06743v2.
15. Xiang Lisa Li, Percy Liang. Prefix-Tuning: Optimizing Continuous Prompts for Generation[J]. 2021. DOI:10.48550/arXiv.2101.00190.
16. Brian Lester, Rami Al-Rfou, Noah Constant. The Power of Scale for Parameter-Efficient Prompt Tuning[J]. 2021. DOI:10.48550/arXiv.2104.08691.
17. Xiao Liu, Yanan Zheng, Zhengxiao Du, et al. GPT Understands, Too[J]. 2021. DOI:10.48550/ arXiv.2103.10385.
18. Xiao Liu, Kaixuan Ji, Yicheng Fu, et al. P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks[J]. 2021. DOI:10.48550/arXiv.2110.07602.
19. Edward J. Hu, Yelong Shen, Phillip Wallis,et al. LoRA: Low-Rank Adaptation of Large Language Models[J]. 2021. DOI:10.48550/arXiv.2106.09685.
20. 俞士汶,段慧明,朱学锋,等.北京大学现代汉语语料库基本加工规范[J].中文信息学报, 2002.
21. 荀恩东.自然语言结构计算——GPF结构分析框架[M].北京:人民邮电出版社.2022
22. Team GLM, Aohan Zeng, Bin Xu ,et al. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv, 2024. arXiv: 2406.12793.
23. Jinze Bai, Shuai Bai, Yunfei Chu, et al. Qwen Technical Report. arXiv, 2023, arXiv:2309.16609.
24. Dawei Gao, Haibin Wang, Yaliang Li, et al. Text-to-SQL Empowered by Large Language Models: A Benchmark Evaluation. arXiv, 2023. arXiv:2308. 15363v4



刘廷超（1978—），博士研究生，主要研究领域为信息检索、自然语言处理。

E-mail：liutingchao@hotmail.com



第二作者姓名（出生年—），学历，职称，主要研究领域为。

E-mail：\*\*\*@\*\*\*



第三作者姓名（出生年—），学历，职称，主要研究领域为。

E-mail：\*\*\*@\*\*\*

1. https://pypi.org/project/gpflib [↑](#footnote-ref-2)
2. <https://tongyi.aliyun.com/qianwen> [↑](#footnote-ref-3)
3. <https://chatglm.cn> [↑](#footnote-ref-4)