

北京交通大学

本科毕业设计（论文）

毕业论文（设计）题目

Title of Project (Thesis)

学 院： 计算机科学与技术学院

专 业： 计算机科学与技术

学生姓名： 张三

学 号： 100044

指导教师： 李四

北京交通大学

2025 年 5 月

学士论文版权使用授权书

本学士论文作者完全了解北京交通大学有关保留、使用学士论文的规定。特授权北京交通大学可以将学士论文的全部或部分內容编入有关数据库进行检索，提供阅览服务，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。

（保密的学位论文在解密后适用本授权说明）

学位论文作者签名：

指导教师签名：

签字日期： 年 月 日

签字日期： 年 月 日

学士论文诚信声明

本人声明所呈交的毕业论文（设计），题目 毕业论文（设计）题目 是本人在指导教师的指导下，独立进行研究工作所取得的成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得北京交通大学或其他教育机构的学位或证书而使用过的材料。

申请学位论文与资料若有不实之处，本人承担一切相关责任。

本人签名：_____ 日期：_____

中文摘要

摘要：中文摘要应将论文的内容要点简短明了地表达出来，约 400 字左右，字体为宋体小四号。内容应包括工作目的、研究方法、成果和结论。要突出本论文的创新点，语言力求精炼。为了便于文献检索，应在本页下方另起一行注明论文的关键词（3-5 个），如有可能，尽量采用《汉语主题词表》等词表提供的规范词。图 X 幅，表 X 个，参考文献 X 篇。

关键词：关键词 1；关键词 2；关键词 3

ABSTRACT

ABSTRACT: 与中文摘要内容要相对应

KEYWORDS: keyword 1 ; keyword 2 ;keyword 3

目 录

中文摘要	I
ABSTRACT	II
目 录	III
1 绪论	1
1.1 编译方式说明	1
1.2 统计字数脚本	2
2 相关技术	4
2.1 插入说明	4
2.1.1 文献引用	5
2.1.2 关于代码	5
3 基本方法	8
3.1 2 级标题	8
3.1.1 3 级标题	8
4 实验	9
4.1 2 级标题	9
4.1.1 3 级标题	9
5 结论	10
5.1 总结	10
5.2 展望	10
参考文献	11
致 谢	12
附 录	13

1 绪论

引言（或绪论）简要说明研究工作的目的、范围、相关领域的前人工作和知识空白、理论基础和分析、研究设想、研究方法和实验设计、预期结果和意义等。应言简意赅，不要与摘要雷同，不要成为摘要的注释。一般教科书中有的知识，在引言中不必赘述。

1.1 编译方式说明

在 Overleaf 上编译

请注意以下事项：

- 请上传整个 `.zip` 压缩包。
- 不要删除 `.latexmkrc` 文件，Overleaf 会自动识别并使用 XeLaTeX 编译（你也可以手动设置为 XeLaTeX）。
- 在 `chapter` 文件夹下编辑你的各章节内容。

在本地编译

请按照以下步骤操作：

- 打开主 `.tex` 文件，例如：`bjtuthesis.tex`。
- 进入：选项 → 设置 TexStudio → 构建，设置如下：
 - 默认编译器选择：XeLaTeX
 - 默认文献工具选择：Biber
- 点击工具栏绿色箭头“构建 & 查看”（或按 F5）。
- 如果文献未显示，点击顶部菜单“工具 (T)” → “参考文献 (B)”运行 Biber（或按 F8）。
- 然后运行 XeLaTeX 两次，即点击两次“构建 & 查看”（F5）。

快捷方式：设置完成后，按顺序使用快捷键 F5 - F8 - F5 - F5 即可完整编译。

在服务器（Linux）编译

建议使用 Makefile，集成了编译、清理中间文件、打开 PDF 三个步骤。

注意：使用 Makefile 时请勿修改主 .tex 文件名。

可用命令如下：

- make: 等同于 make all，执行清理后编译 PDF。
- make bjtuthesis: 仅编译 PDF。
- make clean: 仅清理中间文件。
- make view: 仅打开生成的 PDF。

1.2 统计字数脚本

模板提供了统计字数脚本，可在服务器或本地运行：

使用 Bash 运行 word_count.sh

1. 打开终端（Windows 推荐使用 Git Bash 或 WSL + VS Code），切换到项目目录：

```
cd path\to\your\project
```

2. 执行脚本，传入主 .tex 文件名：

```
./scripts/word_count.sh bjtuthesis.tex
```

3. 输出说明：

- 控制台显示简要统计信息；
- 完整统计信息输出至：outputs/bjtuthesis.wordcnt。

使用 PowerShell 运行 word_count.ps1

1. 打开终端，切换到项目目录：

```
cd path\to\your\project
```


2. 执行脚本，传入主 `.tex` 文件名：

```
./scripts/word_count.ps1 bjtuthesis.tex
```

3. 输出说明：

- 控制台显示简要统计信息；
- 完整统计信息输出至： `outputs/bjtuthesis.wordcnt`。

2 相关技术

2.1 插入说明

我们可以用 `includegraphics` 来插入现有的不同格式的图片，如图1。论文中尽量使用矢量图，如.pdf 格式的图片

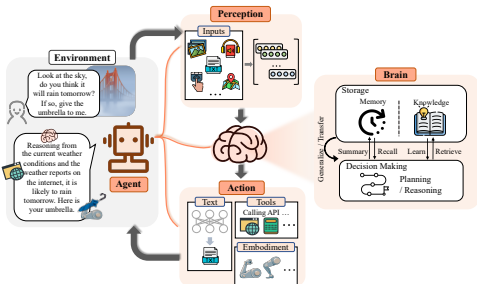


图 1 AI agent 的组成.pdf

如1所示，这是一张自动调节列宽的表格。推荐两个常用的在线表格生成工具：Advanced Table Editor(<https://www.latex-tables.com/>)和 Tables Generator(<https://www.tablesgenerator.com/>)。

表 1 代码智能体相比 Text/JSON 在 LLM 动作上的优势

代码智能体用于 LLM 动作		JSON 或 Text 用于 LLM 动作
预训练数据兼容性	有大量代码可用于预训练	需要针对特定格式进行数据整理
复杂操作支持	通过控制流和数据流原生支持	即便可行，也需要精心设计（例如定义新工具以模拟 if 语句）
工具可用性	可以直接使用现有的软件包	需手动适配工具
自动反馈能力	反馈机制（如 <code>traceback</code> ）作为基础机制已经在大多数编程语言中实现	需额外设计反馈流程

如1，这是一个公式

$$\text{Logits}(t_n) = \text{GenModel}(t_n \mid \{\text{input}, t_1, \dots, t_{n-1}\})$$

(1)

如算法1，这是一个算法

这是一个无序列表

- aaa
- bbb
- ccc

这是一个有序列表 (默认编号格式)

1. aaa

Algorithm 1 CodeAgent-based Workflow

```

memory  $\leftarrow$  [user_defined_task]
while llm_should_continue(memory) do
    action  $\leftarrow$  llm_get_next_action(memory)
    observations  $\leftarrow$  execute_action(action)
    memory  $\leftarrow$  memory + [action, observations]
end while

```

2. bbb

3. ccc

下面是两个有序列表 (自定义编号格式)，你可以按照你的偏好设置

1) aaa

2) bbb

3) ccc

(a) athesiaa

(b) bbb

(c) ccc

2.1.1 文献引用

文献引用可以使用两种方式，下面两种方式任选一个即可，但是需要全文统一

引用文献方式一：文献^[1]、文献^[2]

引用文献方式二：文献[3]、文献[4]

2.1.2 关于代码

目前学院没有明确给出代码格式，你可以自己定义，也可以参照我的设置来使用

%复制以下代码到导言区

```
\definecolor{dkgreen}{rgb}{0,0.6,0}
```

```
\definecolor{gray}{rgb}{0.5,0.5,0.5}
```

```
\definecolor{mauve}{rgb}{0.58,0,0.82}
```

```
\lstset{
    language=python,
    basicstyle=\footnotesize,
    numbers=left,
    numberstyle=\tiny\color{gray},
    stepnumber=1,
    numbersep=5pt,
    backgroundcolor=\color{white},
    showspaces=false,
    showstringspaces=false,
    showtabs=false,
    frame=single,
    rulecolor=\color{black},
    tabsize=2,
    captionpos=b,
    breaklines=true,
    breakatwhitespace=false,
    title=\lstname,
    keywordstyle=\color{blue},
    commentstyle=\color{dkgreen},
    stringstyle=\color{mauve},
    escapeinside={\%*}{*}},
    morekeywords={*,...}
}
```

上述设置最后生成的代码样式如下：

Java 代码示例：

```
1 private void parseJSONWithJSONObject(String jsonData, SQLiteDatabase db){
2     try {
3         JSONObject jsonObject = new JSONObject(jsonData);
4         JSONArray results;
5         if(jsonObject.has("result")) results = jsonObject.getJSONArray("result");
6         // 获取名为 "result" 的 JSON 数组
7         else return;;
8         for (int i = 0; i < results.length(); i++) {
9             JSONObject item = results.getJSONObject(i);
```

```
9      String name = item.optString("name"); // 使用 optString 防止 null 引起
      程序崩溃
10      String img = item.optString("img");
11      String heat = item.optString("heat");
12      String pinyin = getFullPinyin(name);
13      if(name.length() > 10 || name == "null") continue;
14      int calor = extractCalories(heat);
15      try {
16          String sql = "INSERT INTO food (foodname, img, calories, unit,
              pinyin) VALUES ('" + name + "', '" + img + "', '" + calor + "',
              '100', '" + pinyin + "')";
17          db.execSQL(sql);
18          Log.d("We", "Insert successful: foodname " + name + ", img " + img
              + ", heat " + calor + ", pinyin " + pinyin);
19      } catch (SQLException e) {
20          Log.e("We", "Insert failed: " + e.getMessage()); // 处理插入失败的情
              况
21      }
22  }
23  }
24  catch (Exception e) {
25      Log.d("Fooddatabase failed", "failed", e);
26      e.printStackTrace();
27  }
28 }
```

Python 代码示例:

```
1 def generate_with_cue_words(self, background: str):
2     problem, message_input = self.api_helper.generate_problem_with_cue_words(
3         background, self.paper_list, self.cue_words
4     )
5     idea = self.api_helper.generate_idea_with_cue_words(
6         problem, self.paper_list, self.cue_words
7     )
8     idea_filtered = self.api_helper.filter_idea(idea, background)
9     return message_input, problem, idea, idea_filtered
```

3 基本方法

1 级标题内容

3.1 2 级标题

2 级标题内容

3.1.1 3 级标题

3 级标题内容

4 实验

1 级标题内容

4.1 2 级标题

2 级标题内容

4.1.1 3 级标题

3 级标题内容

5 结论

论文的结论是最终的、总体的结论，不是正文中各段的小结的简单重复。结论应该准确、完整、明确、精练。如果不可能导出应有的结论，也可以没有结论而进行必要的讨论。可以在结论或讨论中提出建议、研究设想、仪器设备改进意见以及尚待解决的问题等。

5.1 总结

5.2 展望

参考文献

- [1] CHASE H. LangChain[CP/OL]. 2022. <https://github.com/langchain-ai/langchain>.
- [2] WANG X, CHEN Y, YUAN L, et al. Executable Code Actions Elicit Better LLM Agents[C/OL]// ICML. 2024. eprint: 2402.01030.
- [3] ELOVIC A. gpt-researcher[CP/OL]. 0.5.4. 2023. <https://github.com/assafelovic/gpt-researcher>.
- [4] ROUCHER A, del MORAL A V, WOLF T, et al. ‘smolagents’: a smol library to build great agentic systems.[Z]. <https://github.com/huggingface/smolagents>. 2025.

致 谢

放置在参考文献页后，对象包括：1）国家科学基金，资助研究工作的奖学金基金，合同单位，资助或支持的企业、组织或个人。2）协助完成研究工作和提供便利条件的组织或个人。3）在研究工作中提出建议和提供帮助的人。4）给予转载和引用权的资料、图片、文献、研究思想和设想的所有者。5）其他应感谢的组织和个人。

附 录

附录 A 程序代码

附录是作为论文主体的补充项目，并不是必须的。

附录 B 工程图纸