



# 大模型 - 代码标注与训练

作者: Calvin

QQ: 179209347

Mail: 179209347@qq.com

# 介绍

## 笔记简介:

- 面向对象: 深度学习初学者
- 依赖课程: **线性代数**, **统计概率**, 优化理论, 图论, 离散数学, 微积分, 信息论

## 知乎专栏:

<https://zhuanlan.zhihu.com/p/693738275>

## Github & Gitee 地址:

[https://github.com/mymagicpower/AIAS/tree/main/deep\\_learning](https://github.com/mymagicpower/AIAS/tree/main/deep_learning)

[https://gitee.com/mymagicpower/AIAS/tree/main/deep\\_learning](https://gitee.com/mymagicpower/AIAS/tree/main/deep_learning)

## \* 版权声明:

- 仅限用于个人学习
- 禁止用于任何商业用途

| 数据集                    | 技术特点   | 语言   |   | 大小  | Github   |
|------------------------|--|--|---|---|--|
| code_search_net        | <ul style="list-style-type: none"><li>CodeSearchNet 语料库是来自 GitHub 上托管的开源库的 200 万对（注释、代码）的数据集。它包含多种编程语言的代码和文档。</li></ul>  | <ul style="list-style-type: none"><li>Go</li><li>Java</li><li>JavaScript</li></ul> | <ul style="list-style-type: none"><li>PHP</li><li>Python</li><li>Ruby</li></ul> | <ul style="list-style-type: none"><li>3.5GB</li></ul> | <a href="https://huggingface.co/datasets/code_search_net">https://huggingface.co/datasets/code_search_net</a><br><a href="https://github.com/github/CodeSearchNet">https://github.com/github/CodeSearchNet</a>     |
| codeparrot/github-code | <ul style="list-style-type: none"><li>GitHub 代码数据集由来自 GitHub 的 1.15 亿个代码文件组成，涉及 32 种编程语言和 60 个扩展，总计 1TB 数据。该数据集是根据 Google BigQuery 上的公共 GitHub 数据集创建的。</li></ul> | <ul style="list-style-type: none"><li>32 种编程语言和 60 个扩展</li></ul>                   |   | <ul style="list-style-type: none"><li>1TB</li></ul>   | <a href="https://huggingface.co/datasets/codeparrot/github-code">https://huggingface.co/datasets/codeparrot/github-code</a>  |
| bigcode/the-stack      | <ul style="list-style-type: none"><li>包含超过 6TB 的许可源代码文件，涵盖 358 种编程语言。该数据集是作为 BigCode 项目的一部分创建的，该项目是一个开放的科学合作项目，致力于负责任地开发大型代码语言模型（Code LLM）。</li></ul>            | <ul style="list-style-type: none"><li>358 种编程语言</li></ul>                          |   | <ul style="list-style-type: none"><li>6TB</li></ul>   | <a href="https://huggingface.co/datasets/bigcode/the-stack">https://huggingface.co/datasets/bigcode/the-stack</a>  |
| PolyCoder              | <ul style="list-style-type: none"><li>是 GitHub 上的公开代码，主要选取的是各种编程语言中比较受欢迎的库，每个库至少有 50 Stars，采用了多种编程语言代码集来训练，一共有 12 种语言。</li></ul>                                 | <ul style="list-style-type: none"><li>12 种编程语言</li></ul>                           |   | <ul style="list-style-type: none"><li>249GB</li></ul> | <a href="https://github.com/VHellendoorn/Code-LMs">https://github.com/VHellendoorn/Code-LMs</a>  |
| CodeNet                | <ul style="list-style-type: none"><li>项目的目标是为 AI-for-Code 研究社区提供大规模、多样化且高质量的精选数据集，以推动 AI 技术的创新。</li></ul>  | <ul style="list-style-type: none"><li>55 种编程语言</li></ul>                           |   | <ul style="list-style-type: none"><li>9GB</li></ul>   | <a href="https://github.com/IBM/Project_CodeNet">https://github.com/IBM/Project_CodeNet</a>  |
| CodeXGLUE              | <ul style="list-style-type: none"><li>microsoft 开源的，包含 10 个任务及 14 个数据集。代码到代码，文本代码，代码文本，文本-文本。</li></ul>  | <ul style="list-style-type: none"><li>具体看具体任务对应数据集</li></ul>                       |   |   | <a href="https://github.com/microsoft/CodeXGLUE">https://github.com/microsoft/CodeXGLUE</a><br><a href="https://huggingface.co/datasets?search=code_x_glue">https://huggingface.co/datasets?search=code_x_glue</a> |
| nl2code-dataset        | <ul style="list-style-type: none"><li>用于从自然语言到代码（text-code，自然语言输入，代码输出）生成模型的方法级测试数据，主要用于评估代码生成模型的能力。</li></ul>   | <ul style="list-style-type: none"><li>Java</li></ul>                               |   | <ul style="list-style-type: none"><li>200K</li></ul>  | <a href="https://github.com/aixcoder-plugin/nl2code-dataset">https://github.com/aixcoder-plugin/nl2code-dataset</a>  |



| 类别        | 任务     | 任务描述   |
|-----------|--------|--|
| Code-Code | • 克隆检测 | 预测代码对的语义相似性。   |
|           | • 缺陷检测 | 识别代码是否有缺陷。   |
|           | • 代码补全 | 能够根据正在编辑的上下文自动完成后续标记，帮助开发人员解决写作困惑。                     |
|           | • 代码修复 | 自动修复代码bug。   |
|           | • 代码翻译 | 能够帮助开发人员将一种编程语言（如Python）的代码转换为另一种编程语言（如Java），以实现相同的功能。 |
| Text-Code | • 代码搜索 | 能够帮助开发人员根据自然语言查询自动检索与查询意图相符的代码。                        |
|           | • 代码生成 | 给定自然语言的注释，自动生成代码。                                      |
| Code-Text | • 注释生成 | 根据代码，自动生成代码注释。   |

# 模型训练任务及对应数据集 (CodeXGLUE为例)

| 任务     | 数据集   | 描述  |
|--------|---|---|
| • 克隆检测 | • BigCloneBench<br>• POJ-104                                | 模型的任务是衡量代码之间的语义相似性。包括两个现有的数据集。一个用于对代码进行二元分类，另一个用于在给定代码作为查询时检索语义相似的代码。   |
| • 缺陷检测 | • Devign  | 模型的任务是确定一段源代码是否包含可能用于攻击软件系统的缺陷，例如资源泄漏、使用后释放漏洞和拒绝服务攻击。   |
| • 代码补全 | • PY150<br>• GitHub Java Corpus                             | 模型的任务是在给定代码上下文的情况下预测接下来的标记。涵盖了标记级和行级补全。标记级任务类似于语言建模，我们在这里包括了两个有影响力的数据集。行级数据集是新创建的，用于测试模型自动补全一行代码的能力。                  |
| • 代码修复 | • Bugs2Fix  | 模型的任务是尝试自动修复可能有错误或复杂的代码。  |
| • 代码翻译 | • CodeTrans   | 模型的任务是将一种编程语言的代码翻译成另一种编程语言的代码。  |
| • 代码搜索 | • CodeSearchNet, AdvTest;<br>CodeSearchNet,<br>WebQueryTest | 模型的任务是衡量文本和代码之间的语义相似性。在检索场景中，新创建了一个测试集，其中测试集中的函数名和变量被替换以测试模型的泛化能力。在文本-代码分类场景中，创建了一个测试集，其中自然语言查询来自Bing查询日志，用于测试真实用户查询。 |
| • 代码生成 | • CONCODE   | 模型的任务是根据自然语言描述生成代码。   |
| • 注释生成 | • CodeSearchNet   | 模型的任务是为代码生成自然语言注释。  |

➤ 给定两个代码作为输入，任务是进行二进制分类（0/1），其中1表示语义等价，0表示其他情况。

| 数据集举例           | 数据下载  |
|-----------------|---|
| • BigCloneBench | <a href="https://huggingface.co/datasets/code_x_glue_cc_clone_detection_big_clone_bench">https://huggingface.co/datasets/code_x_glue_cc_clone_detection_big_clone_bench</a> |

## 数据标注结构：

| 字段名   | 类型     | 描述              |
|-------|--------|-----------------|
| id    | int32  | 样本索引编号          |
| id1   | int32  | 第一个方法id         |
| id2   | int32  | 第二个方法id         |
| func1 | string | 第一个方法代码         |
| func2 | string | 第二个方法代码         |
| label | bool   | 1 代表语义等价, 0 则相反 |

```
{
  "func1": "    @Test(expected =
              GadgetException.class)\n .....",
  "func2": "    public InputStream
              getInputStream() {.....",
  "id": 0,
  "id1": 2381663,
  "id2": 4458076,
  "label": false
}
```

| id<br>int32 | id1<br>int32 | id2<br>int32 | func1<br>string  | func2<br>string   | label<br>bool |
|-------------|--------------|--------------|--|---|---------------|
|             |              |              | trace, int bytes_read = 0, int last_contentLength...   | a_d11) throws IOException { URL u_d11 = new...  |               |
| 2           | 21,354,223   | 7,421,563    | public String kodetu(String testusoila) {<br>MessageDigest md = null; try { md =...                    | private StringBuffer encoder(String arg) { if (arg<br>== null) { arg = ""; } MessageDigest md5 = null;... | true          |
| 3           | 15,826,299   | 19,728,871   | public static void printResponseHeaders(String<br>address) { logger.info("Address: " + address); tr... | public static String getEncodedPassword(String<br>buff) { if (buff == null) return null; String t =...    | false         |
| 4           | 9,938,081    | 11,517,213   | public void load(String fileName) { BufferedReader<br>bufReader; loaded = false;...                    | private static void copyFile(File sourceFile, File<br>destFile) { try { if (!destFile.exists()) {...      | false         |
| 5           | 18,220,543   | 17,366,812   | private MapProperties readProperties(URL url) {<br>@SuppressWarnings("unchecked") MapProperties...     | public String<br>transportRemoteUnitToLocalTempFile(String urlStr)...                                     | false         |
| 6           | 22,328,849   | 17,334,846   | protected void doRestoreOrganize() throws<br>Exception { Connection con = null;...                     | static String encodeEmailAsUserId(String email) {<br>try { MessageDigest md5 =...                         | false         |
| 7           | 19,130,322   | 15,710,690   | private String sha1(String s) { String encrypt =<br>s; try { MessageDigest sha =...                    | @SuppressWarnings("unused") private String<br>getMD5(String value) { MessageDigest md5; try {...          | true          |

- 给定一段源代码，任务是确定它是否是一段可能攻击软件系统的不安全代码，例如资源泄露、使用后释放漏洞和拒绝服务（DoS）攻击。我们将这个任务视为二元分类（0/1），其中1表示不安全代码，0表示安全代码。

| 数据集举例                             | 数据下载  |
|-----------------------------------|---|
| • code_x_glue_cc_defect_detection | <a href="https://huggingface.co/datasets/code_x_glue_cc_defect_detection">https://huggingface.co/datasets/code_x_glue_cc_defect_detection</a> |

## 数据标注结构：

| 字段名       | 类型     | 描述        |
|-----------|--------|-----------|
| id        | int32  | 索引编号      |
| func      | string | 源代码       |
| target    | bool   | 0 or 1    |
| project   | string | 包含该代码的项目  |
| commit_id | string | 项目代码提交者id |

```
{
  "commit_id":
  "aa1530dec499f7525d2ccaa0e3a876dc8089ed1e",
  "func": "static void filter_mirror_setup...",
  "id": 8,
  "project": "qemu",
  "target": true
}
```



| id<br>int32   | func<br>string · lengths   | target<br>bool | project<br>string · classes   | commit_id<br>string · lengths   |
|---|--|----------------|---|---|
| <br>0 27.3k  | <br>26 142k                           |                | <br>2 values | <br>40 |
| 0   | <code>static av_cold int vdadec_init(AVCodecContext *avctx) { VDADecoderContext *ctx = avctx-...</code>                | false          | FFmpeg  | 973b1a6b907   |
| 1   | <code>static int transcode(AVFormatContext **output_files, int nb_output_files, InputFile...</code>                    | false          | FFmpeg  | 321b2a9ded6   |
| 2   | <code>static void v4l2_free_buffer(void *opaque, uint8_t *unused) { V4L2Buffer* avbuf = opaque;...</code>              | false          | FFmpeg  | 5d5de3eba4c   |
| 4   | <code>int av_openc1_buffer_write(cl_mem dst_cl_buf, uint8_t *src_buf, size_t buf_size) { cl_int...</code>              | false          | FFmpeg  | 57d77b3963c   |
| 5   | <code>static int r3d_read_rdvo(AVFormatContext *s, Atom *atom) { R3DContext *r3d = s-&gt;priv_data; AVStream...</code> | true           | FFmpeg  | aba232cfa9k   |
| 6   | <code>static int dds_decode(AVCodecContext *avctx, void *data, int *got_frame, AVPacket *avpkt) {...</code>            | true           | FFmpeg  | afb4632cc36   |
| <div> <span>&lt; Previous</span> <span>1</span> <span>2</span> <span>3</span> <span>...</span> <span>219</span> <span>Next &gt;</span> </div> |  |                |   |   |

- 根据先前的上下文完成未完成的句子。当软件开发人员完成当前行的一个或多个标记时，期望行级完成模型能够生成完整的语法正确的代码行。行级代码完成任务与标记级完成共享训练/开发数据集。在CodeCompletion-token上训练模型后，您可以直接将其用于行级完成的测试。

| 数据集举例                                  | 数据下载  |
|--|---|
| • code_x_glue_cc_code_completion_token | <a href="https://huggingface.co/datasets/code_x_glue_cc_code_completion_token">https://huggingface.co/datasets/code_x_glue_cc_code_completion_token</a> |

## 数据标注结构：

| 字段名  | 类型               | 描述        |
|------|------------------|-----------|
| id   | int32            | 样本索引编号    |
| code | Sequence[string] | 代码 Tokens |
|      |                  |           |

```
{
  "code": ["<s>", "from", "bootstrap",
"import", "Bootstrap", "<EOL>", "from", ...,
"</s>"],
  "id": 0,
  "path": "00/wikihouse/urls.py\n"
}
```

| id<br>int32<br><br>0 12.9k   | code<br>sequence  |
|---|---|
| 0   | [ "<s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ";", "import", "org", ".", "eclipse", ".", "xtext", ".", "ui", ".", "DefaultUiModule", ";", ... |
| 1   | [ "<br><s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ".", "contentassist", ".", "..."  |
| 2   | [ "<br><s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ".", "contentassist", ".", "..."  |
| 3   | [ "<br><s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ".", "contentassist", ".", "..."  |
| 4   | [ "<br><s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ".", "contentassist", ".", "..."  |
| 5   | [ "<br><s>", "package", "org", ".", "sqlproc", ".", "dsl", ".", "ui", ".", "contentassist", ";", "..."  |
| <div> <a href="#">&lt; Previous</a> <span>1</span> <span>2</span> <span>3</span> <span>...</span> <span>130</span> <a href="#">Next &gt;</a> </div> |   |

- 数据包含带有错误的Java函数，以及经过修正的函数。所有的函数和变量名都已经标准化。他们的数据集包含两个子集（即小型和中型），基于函数的长度进行划分。

| 数据集举例                            | 数据下载  |
|----------------------------------|---|
| • code_x_glue_cc_code_refinement | <a href="https://huggingface.co/datasets/code_x_glue_cc_code_refinement">https://huggingface.co/datasets/code_x_glue_cc_code_refinement</a> |

数据标注结构：

| 字段名   | 类型     | 描述     |
|-------|--------|--------|
| id    | int32  | 样本索引编号 |
| buggy | string | Bug代码  |
| fixed | string | 修复的代码  |
|       |        |        |
|       |        |        |
|       |        |        |

```
{
  "buggy": "public java.util.List < TYPE_1 > METHOD_1 ()
{ java.util.ArrayList < TYPE_1 > VAR_1 = new java.util.ArrayList
< TYPE_1 > (); for ( TYPE_2 VAR_2 : VAR_3 ) { VAR_1 .
METHOD_2 ( VAR_2 . METHOD_1 () ); } return VAR_1 ; } \n",
  "fixed": "public java.util.List < TYPE_1 > METHOD_1 ()
{ return VAR_1 ; } \n",
  "id": 0
}
```



| id<br>int32  | buggy<br>string  | fixed<br>string  |
|--|--|--|
| 0  | public static TYPE_1 init ( java.lang.String name ,<br>java.util.Date date ) { TYPE_1 VAR_1 = new TYPE_1 ... | public static TYPE_1 init ( java.lang.String name ,<br>java.util.Date date ) { TYPE_1 VAR_1 = new TYPE_1 ... |
| 1  | public TYPE_1 METHOD_1 ( java.lang.String name ) {<br>if ( name . equals ( STRING_1 ) ) return new TYPE_...  | public TYPE_1 METHOD_1 ( java.lang.String name ) {<br>if ( name . equals ( STRING_3 ) ) return new TYPE_...  |
| 2  | private boolean METHOD_1 ( TYPE_1 VAR_1 ) { boolean<br>VAR_2 = false ; VAR_2 = VAR_2    ( ( VAR_3 ...        | private boolean METHOD_1 ( TYPE_1 VAR_1 ) { boolean<br>VAR_2 = ( VAR_3 . compareTo ( VAR_1 . METHOD_2 ( )... |
| 3  | public void METHOD_1 ( TYPE_1 VAR_1 , boolean VAR_2<br>) { if ( VAR_2 ) { VAR_3 . METHOD_2 ( 1 , CHAR_1 )... | public void METHOD_1 ( TYPE_1 VAR_1 , boolean VAR_2<br>) { if ( VAR_2 ) { VAR_3 . METHOD_2 ( 1 , CHAR_1 )... |
| 4  | public boolean METHOD_1 ( ) { if ( ( VAR_1 ) == ( ( VAR_2 .<br>METHOD_2 ( VAR_1 ) ) - 1 ) ) { return fals... | public boolean METHOD_1 ( ) { if ( ( VAR_1 ) >= ( ( VAR_2 .<br>METHOD_2 ( VAR_3 ) ) - 1 ) ) { return fals... |
| 5  | public boolean METHOD_1 ( java.util.Collection < ?<br>extends java.lang.Integer > c ) { if ( VAR_1 class...  | public boolean METHOD_1 ( java.util.Collection < ?<br>extends java.lang.Integer > c ) { if ( ! ( VAR_1...    |
| 6  | public void METHOD_1 ( TYPE_1 VAR_1 ) { VAR_2 = new  | public void METHOD_1 ( TYPE_1 VAR_1 ) { VAR_2 = new  |
| <div> <a href="#">&lt; Previous</a> <a href="#">1</a> <a href="#">2</a> <a href="#">3</a> <a href="#">...</a> <a href="#">524</a> <a href="#">Next &gt;</a> </div> |  |  |

- 数据集收集了Java和C#版本的代码，并找到了并行函数。在去除重复和空函数后，将整个数据集分成训练集、验证集和测试集。

| 数据集举例                               | 数据下载  |
|-------------------------------------|---|
| • code_x_glue_cc_code_to_code_trans | • <a href="https://huggingface.co/datasets/code_x_glue_cc_code_to_code_trans">https://huggingface.co/datasets/code_x_glue_cc_code_to_code_trans</a> |

## 数据标注结构：

| 字段名  | 类型     | 描述     |
|------|--------|--------|
| id   | int32  | 样本索引编号 |
| java | string | Java代码 |
| cs   | string | C# 代码  |
|      |        |        |

```
{
  "CS": "public DVRecord(RecordInputStream
in1){_option_flags = in1.ReadInt();_promptTitle =
ReadUnicodeString(in1);_errorTitle =
ReadUnicodeString(in1);_promptText = ...",
  "id": 0,
  "java": "public DVRecord(RecordInputStream in)
{_option_flags = in.readInt();_promptTitle =
readUnicodeString(in);_errorTitle =
readUnicodeString(in);_promptText ..."}
}
```

| id<br>int32   | java<br>string · lengths   | cs<br>string · lengths  |
|---|--|---|
|    |                         |                    |
| 0   | public ListSpeechSynthesisTasksResult<br>listSpeechSynthesisTasks(ListSpeechSynthesisTasks...            | public virtual ListSpeechSynthesisTasksResult<br>ListSpeechSynthesisTasks(ListSpeechSynthesisTasks... |
| 1   | public UpdateJourneyStateResult<br>updateJourneyState(UpdateJourneyStateRequest...                       | public virtual UpdateJourneyStateResponse<br>UpdateJourneyState(UpdateJourneyStateRequest...          |
| 2   | public void removePresentationFormat()<br>{remove1stProperty(PropertyIDMap.PID_PRESFORMAT);}             | public void RemovePresentationFormat()<br>{MutableSection s =...                                      |
| 3   | public CellRangeAddressList(int firstRow, int<br>lastRow, int firstCol, int lastCol)...                  | public CellRangeAddressList(int firstRow,<br>lastRow, int firstCol, int lastCol): th...               |
| 4   | public void delete(int key) {int i =<br>binarySearch(mKeys, 0, mSize, key);if (i >= 0) {i...             | public virtual void delete(int key){int<br>binarySearch(mKeys, 0, mSize, key);if (...                 |
| 5   | public CreateBranchCommand setStartPoint(RevCommit<br>startPoint) {checkCallable();this.startCommit =... | public virtual NGit.Api.CreateBranchComm<br>SetStartPoint(RevCommit startPoint)...                    |
| <div>             &lt; Previous             <span>1</span>             2             3             ...             103             Next &gt; </div> |  |   |

- 该数据集可用于训练一个模型，用于从给定的英文自然语言查询中检索前k个代码。

| 数据集举例                             | 数据下载  |
|-----------------------------------|---|
| code_x_glue_tc_nl_code_search_adv | <a href="https://huggingface.co/datasets/code_x_glue_tc_nl_code_search_adv">https://huggingface.co/datasets/code_x_glue_tc_nl_code_search_adv</a> |

数据标注结构：

| 字段名       | 类型     | 描述     |
|-----------|--------|--------|
| id        | int32  | 样本索引编号 |
| code      | string | 方法代码   |
| docstring | string | 方法注释   |
| language  | string | 编程语言   |
| url       | string | 代码地址链接 |
| ...       | ...    | ...    |

```
{
  "argument_list": "",
  "code": "def Func(arg_0, arg_1='.', a..."arg_2", ")"]",
  "docstring": "Downloads Dailymotion videos by URL.",
  "docstring_summary": "Downloads Dailymotion videos by URL.",
  "func_name": "",
  "id": 0,
  "language": "python",
  "path": "src/you_get/extractors/dailymotion.py",
  "repo": "",
  "return_statement": "",
  "score": 0.9997601509094238,
  "sha": "b746ac01c9f39de94cac2d56f665285b0523b974",
  "url": "https://github.com/soimort....",
  ...
}
```



➤ 一个包含超过100,000个示例的大型数据集，其中包括来自在线代码仓库的Java类。用于根据自然语言描述和类环境生成Java类成员函数的源代码。

| 数据集举例               | 数据下载  |
|---------------------|---|
| • CodeXGLUE-CONCODE | <a href="https://huggingface.co/datasets/AhmedSSoliman/CodeXGLUE-CONCODE">https://huggingface.co/datasets/AhmedSSoliman/CodeXGLUE-CONCODE</a> |

数据标注结构：

| 字段名  | 类型     | 描述      |
|------|--------|---------|
| nl   | string | 自然语言描述  |
| code | string | 对应的实现代码 |
|      |        |         |

```
{
  "nl": "Increment this vector in this place.
con_elem_sep double[] vecElement
con_elem_sep double[] weights con_func_sep
void add(double)",
  "code": "public void inc ( ) { this . add ( 1 ) ; }"
}
```

| <div>code</div> <div>string</div>  | <div>n1</div> <div>string</div>   |
|--|---|
| <div>boolean function ( ) { return isParsed ; }</div>  | <div>check if details are parsed . concode_field_sep<br/>Container parent concode_elem_sep boolean isParsed...</div>          |
| <div>File function ( ) { return libraryFile ; }</div>  | <div>answer the library file defining the library<br/>containing the compilation unit to be indexed or null...</div>          |
| <div>void function ( Directory arg0 , Collection &lt;<br/>SnapshotMetaData &gt; arg1 , long arg2 ) { List &lt;...</div>      | <div>this method deletes index files of the @linkplain<br/>indexcommit for the specified generation number ...</div>          |
| <div>byte [ ] function ( Class &lt; ? &gt; arg0 , Configuration<br/>arg1 ) { return AuthenticationTokenSerializer ....</div> | <div>do n't use this . no , really , do n't use this . you<br/>already have an authenticationtoken with...</div>              |
| <div>void function ( Binder arg0 ) { EventBus loc0 = new<br/>EventBus ( ) ; AmbariEventPublisher loc1 = new...</div>         | <div>force the eventbus from ambarieventpublisher to be<br/>serialand synchronous . concode_field_sep Placeholder...</div>    |
| <div>boolean function ( ) { return false ; }</div>   | <div>todo summary sentence for isform ... concode_field_sep<br/>String name concode_elem_sep URI ns concode_elem_sep...</div> |
| <div>String function ( ) { return clientId ; }</div>   | <div>returns the id used by the client to authenticate</div>  |
| <div><div>&lt; Previous</div><div>1</div><div>2</div><div>3</div><div>...</div><div>1,000</div><div>Next &gt;</div></div>    |   |

➤ 数据源自CodeSearchNet，反过来使用。并对其数据集进行了细分。

| 数据集举例           | 数据下载   |
|-----------------|--|
| • CodeSearchNet | <a href="https://huggingface.co/datasets/CM/codexglue_code2text_java">https://huggingface.co/datasets/CM/codexglue_code2text_java</a><br><a href="https://huggingface.co/datasets/CM/codexglue_code2text_python">https://huggingface.co/datasets/CM/codexglue_code2text_python</a> |

数据标注结构：

| 字段名       | 类型     | 描述     |
|-----------|--------|--------|
| id        | int32  | 样本索引编号 |
| code      | string | 方法代码   |
| docstring | string | 方法注释   |
| language  | string | 编程语言   |
| url       | string | 代码地址链接 |
| ...       | ...    | ...    |


```
{
  "argument_list": "",
  "code": "def Func(arg_0, arg_1='.', a..."arg_2", ")",
  "docstring": "Downloads Dailymotion videos by URL.",
  "docstring_summary": "Downloads Dailymotion videos by URL.",
  "func_name": "",
  "id": 0,
  "language": "python",
  "path": "src/you_get/extractors/dailymotion.py",
  "repo": "",
  "return_statement": "",
  "score": 0.9997601509094238,
  "sha": "b746ac01c9f39de94cac2d56f665285b0523b974",
  "url": "https://github.com/soimort.....",
  ...
}
```



- 因为大模型的参数量非常大，从头训练一个自己的大模型训练成本非常高；
- 提示词工程的效果达不到要求，通过自有数据微调，更好的提升大模型在特定领域的能力；
- 训练一个轻量级的微调模型，提升特定业务场景个性化服务能力；
- 数据安全的问题；

| 大模型微调技术        | 技术特点                           | 参数组合形式 | 缺点                            | 优点                                |
|----------------|--------------------------------|--------|-------------------------------|-----------------------------------|
| Adapter Tuning | 固定原参数，微调Adapter结构              | 相加式    | 增加了模型层数，引入了额外的推理延迟            | 需训练参数规模小                          |
| Prefix Tuning  | 利用前缀训练                         | 门控式    | 难于训练，挤占下游任务的输入序列空间，影响模型性能     | 具有学习能力的提示词                        |
| P-Tuning v2    | 加入更多token                      | 门控式    | 容易导致旧知识遗忘，微调后在之前的问题上表现明显变差    | 在小模型上也有较好效果                       |
| 🌟🌟🌟🌟🌟<br>LoRA  | 低秩自适应，计算代价低                    | 缩放式    | 使用低精度权重，准确率受影响                | 计算高效，显著降低计算资源代价                   |
| 🌟🌟🌟<br>Q-LoRA  | 针对量化优化，保留高精度权重特征               | 缩放式    | 微调过程中需要高低精度转换，可能导致精度损失，缺少加速优化 | 既减少模型的内存占用，同时保留训练的基本精度            |
| QA-LoRA        | 将权重矩阵分组为更小的段，并对每个组单独应用量化和低秩自适应 | 缩放式    | /                             | 融合量化和低秩适应的好处，同时保持过程高效和模型对所需任务的有效性 |

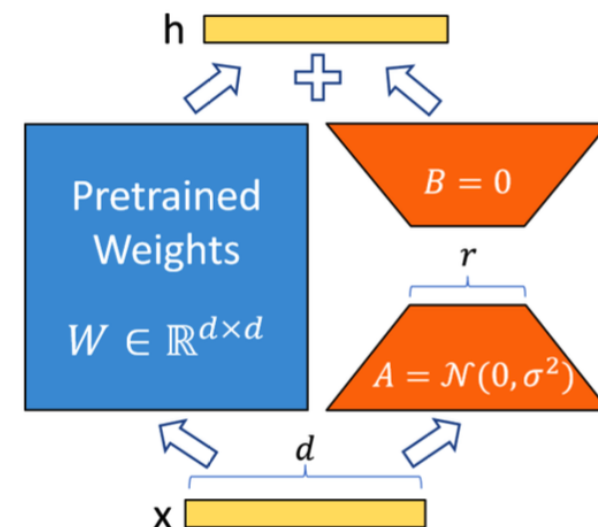


➤ 与多款开源大模型的无缝衔接，可执行增量预训练、指令微调等任务类型。开发者仅需使用8GB消费级显卡，就可以训练出适用于具体业务场景的“专属大模型”。这极大地降低了进行大模型训练的“真金白银”成本。

| 微调框架  | 技术特点  | 技术支持         | Github  |
|---|---|--------------|---|
| <br>DeepSpeed | <ul style="list-style-type: none"><li>结合Ray、HuggingFace训练模型</li><li>被用于高效地微调大模型，并且能使用多节点获得最高性价比而不带来额外的复杂度；</li></ul>  | 微软           | <a href="https://github.com/microsoft/DeepSpeed">https://github.com/microsoft/DeepSpeed</a>       |
| XTuner  | <ul style="list-style-type: none"><li>轻量级: 支持在消费级显卡上微调大语言模型。</li><li>多样性: 支持多种大语言模型，数据集和微调算法（QLoRA、LoRA），支撑用户根据自身具体需求选择合适的解决方案。</li><li>兼容性: 兼容 DeepSpeed 和 HuggingFace 的训练流程，支撑用户无感式集成与使用。</li></ul>   | 上海AI实验室      | <a href="https://github.com/InternLM/xtuner">https://github.com/InternLM/xtuner</a>               |
| MFTCoder  | <ul style="list-style-type: none"><li>开源的多任务代码大语言模型项目，包含代码大模型的模型、数据、训练等；</li><li>高精度、高效率、多任务、多模型支持、多训练算法，大模型代码能力微调框架；</li></ul>   | 开源社区         | <a href="https://github.com/codefuse-ai/MFTCoder">https://github.com/codefuse-ai/MFTCoder</a>     |
| <br>PEFT      | <ul style="list-style-type: none"><li>旨在通过最小化微调参数的数量和计算复杂度，来提高预训练模型在新任务上的性能，从而缓解大型预训练模型的训练成本。</li><li>模型参数高效微调，目前支持Prefix Tuning、Prompt Tuning、PTuningV1、PTuningV2、Adapter、LoRA、AdaLoRA, LoRA</li></ul> | hugging face | <a href="https://github.com/huggingface/peft">https://github.com/huggingface/peft</a>             |
| lamini  | <ul style="list-style-type: none"><li>它是免费的，速度快，适用于小型 LLMs（语言模型）<br/>它就像使用无限提示大小一样，比最大提示的空间多出 1000 倍以上<br/>它能够学习新信息，而不仅仅是根据已学习的内容来理解信息（检索增强生成）</li></ul>  | 开源社区         | <a href="https://github.com/lamini-ai/lamini">https://github.com/lamini-ai/lamini</a>             |
| LLMTune   | <ul style="list-style-type: none"><li>多个LLM的模块化支持（目前支持MetaAI开源的2个模型，LLaMA和OPT）</li><li>支持广泛的消费级NVIDIA的GPU显卡（包括RTX系列、A系列、GTX系列等）</li><li>代码库微小且易于使用（整个源代码仅64k大小）</li></ul>                               | 康奈尔大学        | <a href="https://github.com/kuleshov-group/llmtune">https://github.com/kuleshov-group/llmtune</a> |

| 大模型微调技术   | 技术特点                           | 参数组合形式 | 缺点                            | 优点                                |
|---|--------------------------------|--------|-------------------------------|-----------------------------------|
| <br>LoRA   | 低秩自适应，计算代价低                    | 缩放式    | 使用低精度权重，准确率受影响                | 计算高效，显著降低计算资源代价                   |
| <br>Q-LoRA | 针对量化优化，保留高精度权重特征               | 缩放式    | 微调过程中需要高低精度转换，可能导致精度损失，缺少加速优化 | 既减少模型的内存占用，同时保留训练的基本精度            |
| QA-LoRA   | 将权重矩阵分组为更小的段，并对每个组单独应用量化和低秩自适应 | 缩放式    | /                             | 融合量化和低秩适应的好处，同时保持过程高效和模型对所需任务的有效性 |

以ChatGLM 为例：  
需要显存 15GB GPU







Thank

You