南京航空航天大學
NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

基于大模型的智能软件研发

魏明强、宫丽娜
计算机科学与技术学院

目录

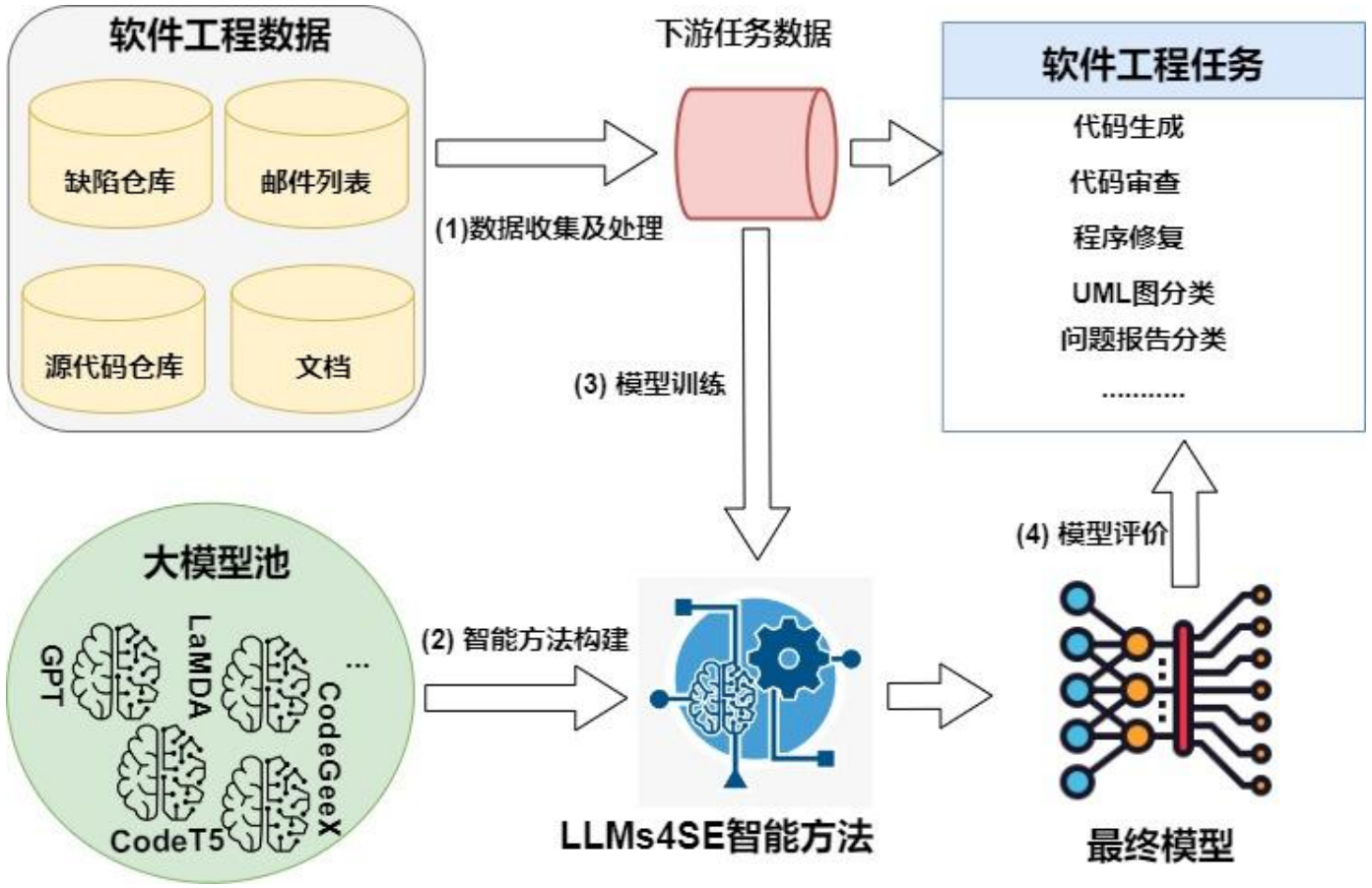
目录

基于大模型的智能软件研发框架

- **LLM4SE（Large Language Model for Software Engineering）**：使用具有少量标签的软件工程下游任务数据，通过微调已有大模型构建的针对软件研发中各种编程任务的智能模型
- 具体的构建过程主要包括：

  - **软件研发任务数据集收集及处理**
  - **基于大模型的智能方法构建**
  - **模型训练**
  - **模型评价**

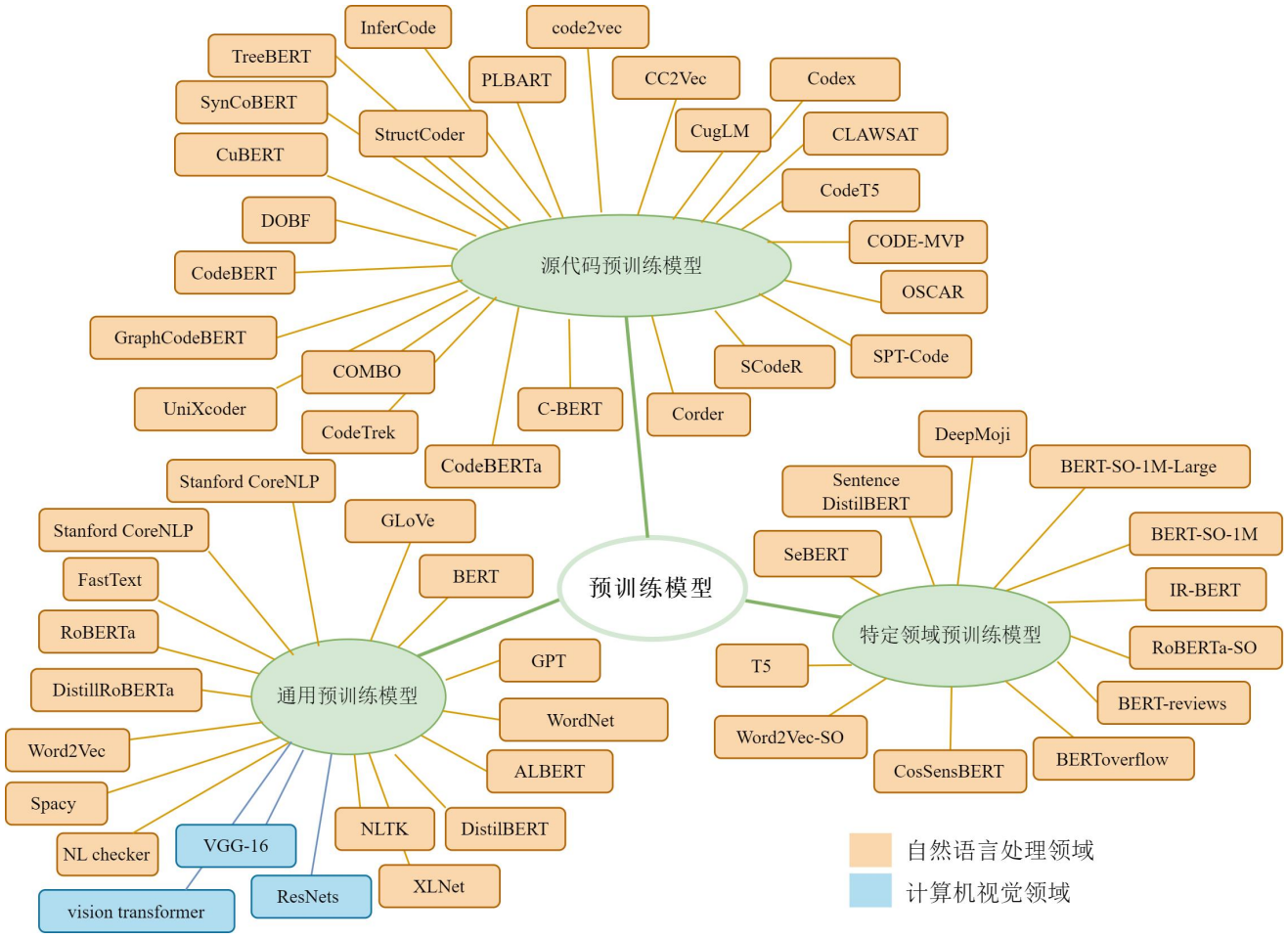
基于大模型的智能软件研发框架



- 数据收集及处理
  - 从软件工程仓库中收集所需的数据集，如源代码仓库中的代码数据、缺陷跟踪系统的缺陷报告等，对数据集进行标记及预处理，使其符合模型输入的要求。
- 基于大模型的智能方法构建
  - 根据软件工程具体任务，从大模型池中选取合适的大模型，如CodeGeeX，CodeBERT、GPT-C，基于该大模型构建符合需求的智能方法。
- 模型微调
  - 使用少量标签数据，选择合适的微调模式来微调大模型，从而构建高性能的LLM4SE模型。
- 模型评价
  - 通过性能评价指标，评价基于大模型的智能方法在软件研发具体任务中的性能。

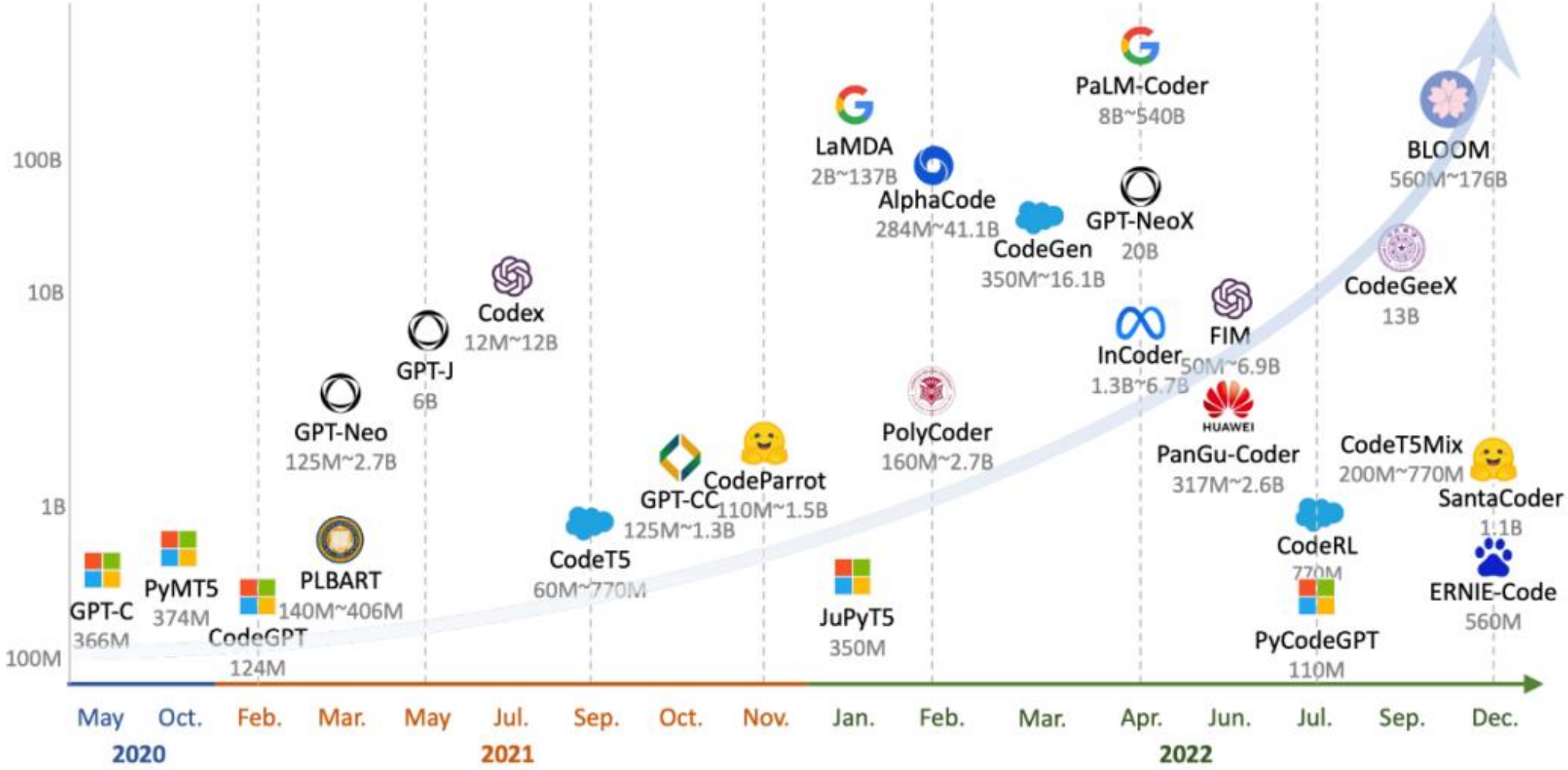
目录

常用大模型



- 通用大模型
  - **一般领域数据集上训练**出的大模型，如自然语言处理领域中使用英文维基百科或普通新闻数据集训练出的GPT模型等，以及计算机视觉中使用ImageNet等数据集预训练的ResNet和VGG模型等。
- 特定领域大模型
  - **软件工程领域特定数据集上训练**出的大模型。一般领域数据集训练出的大模型不能很好地适应软件工程领域的文本特性。因此，研究者收集大量的软件工程领域数据集从零训练深度学习模型形成软件工程特定领域大模型，如SeBERT、T5，进而解决软工领域任务。
- 代码大模型
  - **在源代码数据集上训练**出的大模型。为了能够更好地捕获源代码数据中的语法和语义信息，研究者们收集大量的源代码数据集，重新训练深度学信息模型形成软工领域的代码大模型，如CodeGPT、Starcoder、CodeT5等。

常用大模型



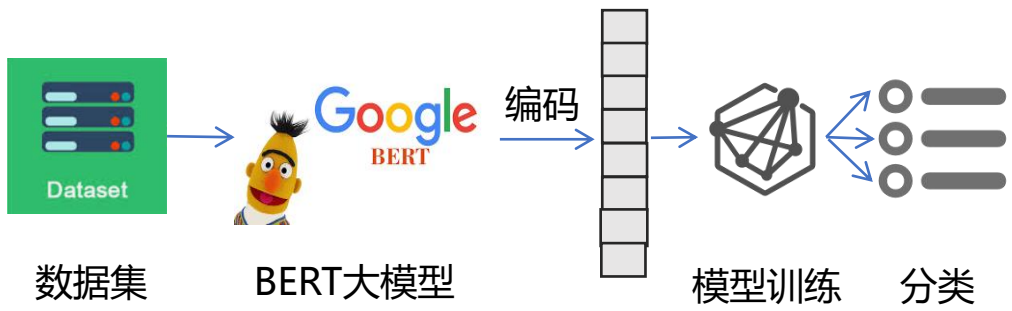
目前流行的27个源代码大模型

目录

预训练范式

预训练—特征表示

- 软件工程领域研究者直接使用在大规模数据集上训练的大模型，对软工领域的任务数据集进行编码特征表征，然后通过编码的特征表征，构建较好的分类器等模型，以实现软件工程领域任务。



预训练—微调

- 软件工程领域研究者以在大规模数据集上训练的大模型为基础，构建包含下游任务的智能模型，然后通过少量下游任务数据集对构建的智能模型进行微调训练，最终构建适应下游任务的智能模型。这种模式避免了针对不同任务需要大量数据集从头训练模型。
- 预训练–微调范式中根据微调的范围又可分为只微调任务层和微调整个智能模型两类。
  - 微调任务层是**冻结大模型的特征表示层**，通过少量的下游任务数据，**只微调模型的任务层参数**。
  - 微调整个智能模型是使用下游任务数据集**微调基于大模型的智能模型**，包括大模型的参数，得到最终的智能模型。

目录

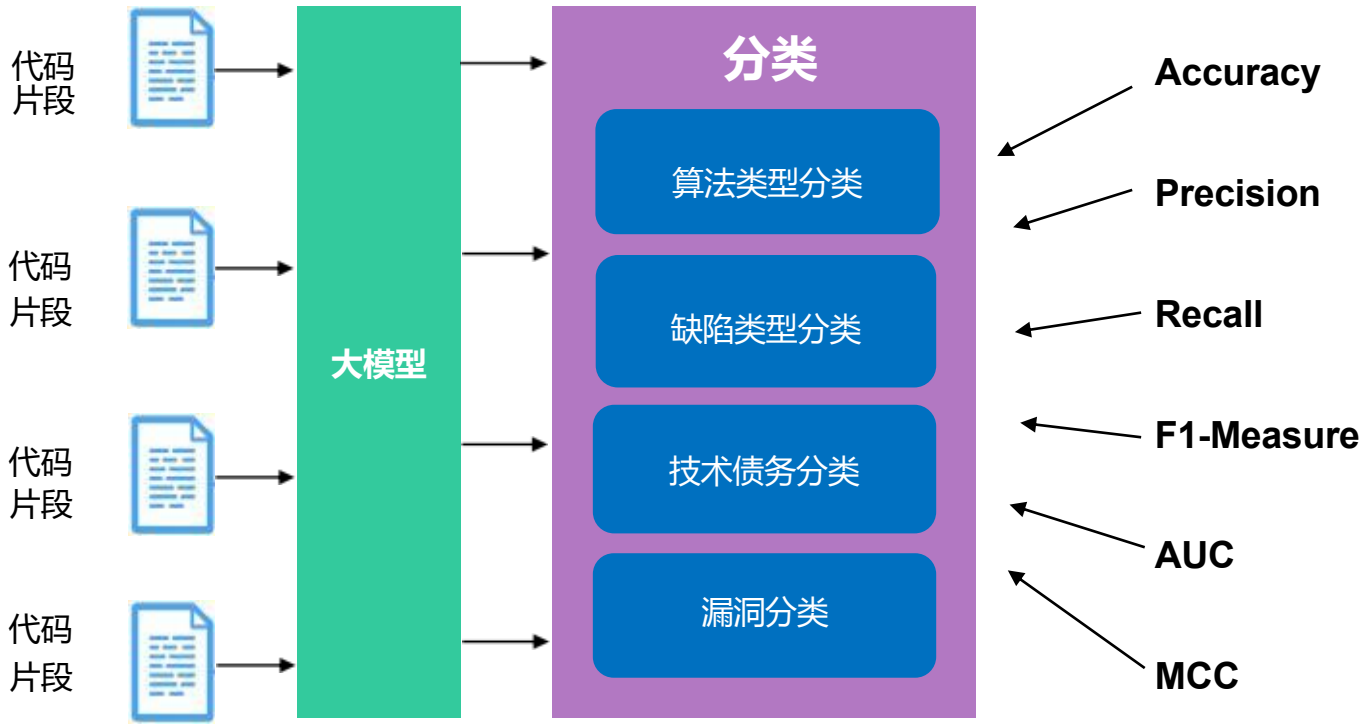
预训练模型任务分类

- 根据输入数据类型，智能软件研发领域的下游任务可以分为**程序语言 (Program Lan_x0002_guage：PL) 相关任务**、软件工程领域**自然语言 (Natural Language：NL) 相关任务**、**程序语言与自然语言交互任务**、**软件工程领域图像相关**任务。

程序语言PL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **代码片段分类(Code Snippets Classification):** 通过基于大模型的智能方法捕获代码片段中丰富的语法和语义信息，以预测代码片段的类型，为开发者提供有用的分类信息，帮助开发者更好地理解代码。
- 该任务常用的性能评价指标有：Accuracy、Precision、Recall、F1-Measure、AUC、MCC(Matthews Correlation Coefficient score)



程序语言PL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

□ **程序修复(Program Repair):** 采用不同的技术自动生成修复缺陷代码片段。为了提高程序修复的性能，软件工程领域部分研究者利用大模型来学习成对的代码片段的语法与语义信息，并将该信息应用到程序修复任务中。

□ 该任务常用的性能评价指标有：Exact Match(EM)、Number of fixed bugs、BLEU(Bilingual Evaluation Understudy)等。

- EM用来衡量生成的修复代码是否与开发人员实际实现的修复代码完全相同。
- Number of fixed bugs 是通过运行测试用例来查看生成的补丁是否通过测试, 进而得到修复的 bugs 的个数。
- BLEU 用来计算预测和正确答案之间的 n-gram 相似度，为 n-gram 匹配精度分数的几何平均值。
- EM、BLEU 和Number of fixed bugs 越大，模型越好。

程序语言PL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **代码补全(Code Completion):** 基于上下文代码信息实时建议下一个可能的符号，例如类名、方法名等，用以补全代码片段，加速软件的开发。大模型技术可以通过上下文代码规律解决代码补全任务。
- 该任务常用的性能评价指标有：EditSIM(Edit Similarity)、Exact Match(EM)、Perplexity等。

  - Perplexity 是 token-level 代码补全的评估指标，用来度量模型预测样本的好坏程度，即下一个词时的平均可选择数量 。
  - EM 和 EditSIM 是 Line-level代码补全的评估指标。
  - EditSIM 是两个单词之间 Levenshtein 距离，即将一个单词更改为另一个单词所需的最小单字符编辑次数，包括插入、删除或替换。
  - EM 用来评估模型预测中匹配到正确答案的百分比。
  - 通常是 Perplexity 和 EditSIM 越小，模型越好。Accuracy 和 EM越大，模型越好。

程序语言PL相关任务

南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **程序语言迁移(Program Translation):** 即程序语言翻译任务，是利用模型将源语言编写的代码作为输入翻译为等效的目标语言编写的代码，翻译后的代码语义应该与输入的代码完全匹配。因此，将大模型引入到程序语言迁移任务中，提升程序语言迁移任务的性能。
- 该任务常用的性能评价指标有：BLEU、Exact Match(EM)、CodeBLEU等。
  - CodeBLEU 是除了 n-gram匹配之外，还考虑了基于代码结构的句法和语义匹配的评价指标。
  - BLEU、EM 和 CodeBLEU越大，模型越好。

- **API推荐(API Recommendation):** 是根据开发人员实际需要来自动化地推荐给开发人员合适的 API 序列，满足开发人员实际需求。
- 该任务常用的性能评价指标有：Precision、Recall等。

目录

自然语言NL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

□ **软件工程相关文本分类(Software Engineering-Related Text Classification):** 采是通过基于大模型的智能方法，捕获软件工程相关文本（如问题报告、代码提交及评论等）的编码表示来预测文本的类型，如**问题报告的分类**、**APP 评论分类**和**文本情感分类**等，为开发者提供有用的分类信息，帮助开发者更好的理解软件工程领域的文本信息。

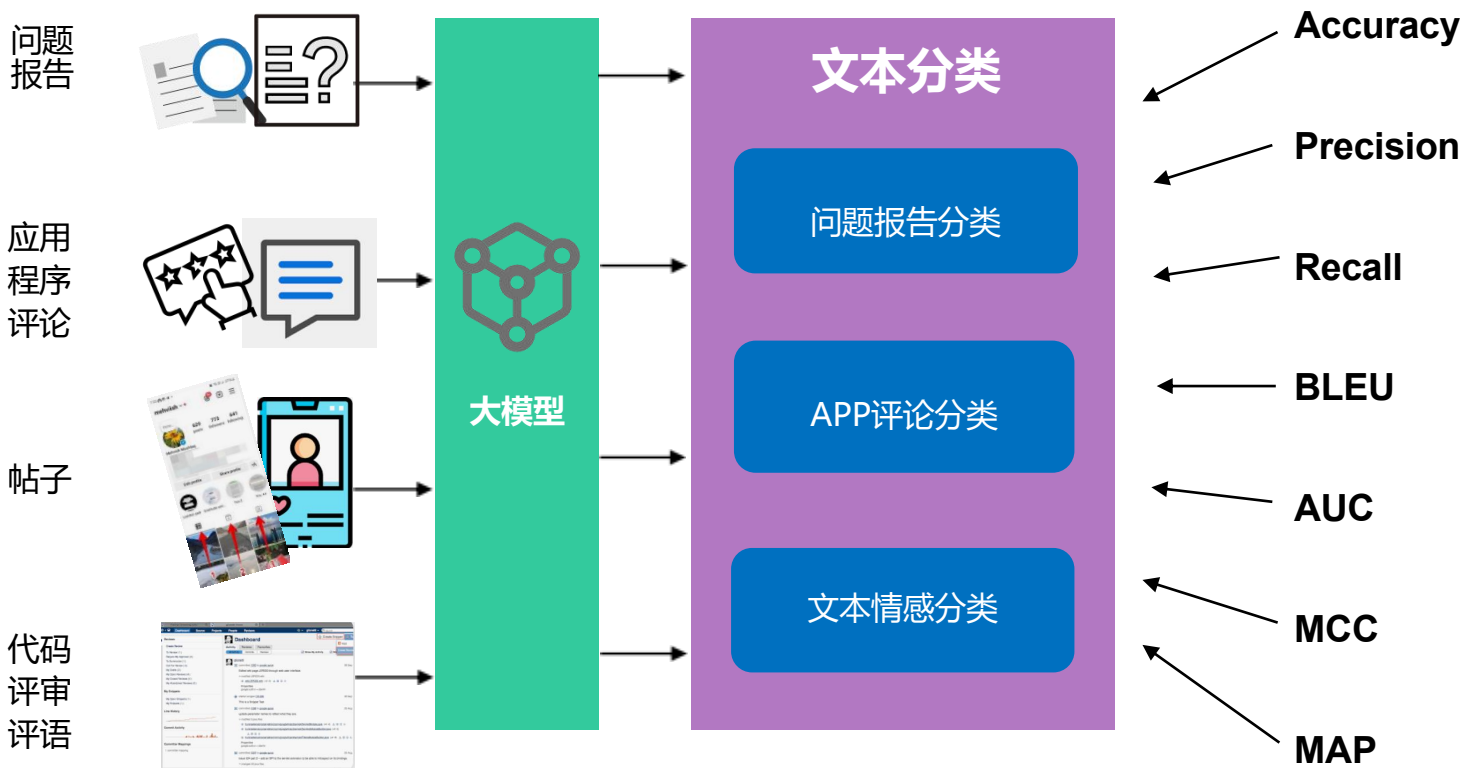
- **问题报告分类任务**是利用不同技术对问题跟踪系统问题报告的类型和优先级进行分类。问题报告是用户在使用软件系统过程中发现问题的自然语言描述。
- **APP 评论分类任务**是采用不同技术对应用程序 (如 Google Play Store、Apple App Store、Twitter data) 中的评论进行分类 (如问题报告、功能需求等)，这些信息可以应用在软件工程诸多领域 (如需求工程、测试等)。
- **文本情感分类任务**是对软件工程领域相关文本 (e.g., Stack Overflow 中的帖子，代码评审评语等) 的情感进行分类，是开发者或用户对软件系统工件的观点、态度或情绪。

自然语言NL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **软件工程相关文本分类(Software Engineering-Related Text Classification):** 采是通过基于大模型的智能方法，捕获软件工程相关文本（如问题报告、代码提交及评论等）的编码表示来预测文本的类型，如**问题报告的分类**、**APP 评论分类**和**文本情感分类**等，为开发者提供有用的分类信息，帮助开发者更好的理解软件工程领域的文本信息。

- 该任务常用的性能评价指标主要有 Recall、F1-score、Accuracy、precision、MCC、AUC、BLEU、MAP(Mean Average Precision) 等。



自然语言NL相关任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **评论回复自动生成(Review Response Generation):** 是针对用户对应用程序的评论准确自动化地给出相应回复文本的任务。准确的回复应用程序评论是缓解用户的担忧，改善用户体验的方法之一。
- 该任务常用的性能评价指标有：BLEU等。

- **跟踪链接发现(Traceability Links Discovery):** 是通过不同的技术发现软件工件之间的关联关系，如问题报告与修复问题报告的提交之间的链接。准确地发现软件工件间的关联为后续挖掘软件开发过程有效信息、挖掘漏洞等提供了有利保障。
- 该任务常用的性能评价指标有：F1-scores、Mean Average Precision (MAP) 等

目录

程序语言与自然语言交互任务

南京航空航天大学 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

- **代码摘要(Code Summarization):** 即是为了帮助开发者更便捷地理解成程序语言代码，依据上下文代码，采用不同技术**对代码生成自然语言的总结文本**，即 PL-To-NL 任务，如代码注释生成、代码提交摘要生成等。可以通过引入不同的大模型来深入挖掘程序语言与自然语言之间的关系，进而产生高质量代码摘要。
- 该任务常用的性能评价指标有：BLEU、ROUGE、METEOR等。
  - ROUGE 通过将模型生成的摘要与正确答案按 N-gram 拆分后，计算召回率来衡量生成摘要与正确答案的匹配程度。
  - METEOR 用来测量基于单精度的加权调和平均数和单字召回率，其可以解决 BLEU 标准中单纯基于精度的问题。同时与 BLEU 比较，基于召回率的 ROUGE和 METEOR 和人工判断的结果更相关。
  - BLEU、ROUGE 和 METEOR 越大，模型越好。

- **代码生成 (Code Generation):** 是与代码摘要完全相反的任务，是**从其自然语言描述中生成一个程序源代码** (在目标程序语言中) 的任务，即 NL-To-PL 任务。
- 该任务常用的性能评价指标有：BLEU、CodeBLEU、EM 和 Pass@k(k=1,10,100)等。
  - Pass@k 是给定生成的n($n \geq k$) 个样本，计算通过单元测试的样本的数量，并计算无偏估计量。

程序语言与自然语言交互任务

南京航空航天大學 NANJING UNIVERSITY OF AERONAUTICS AND ASTRONAUTICS

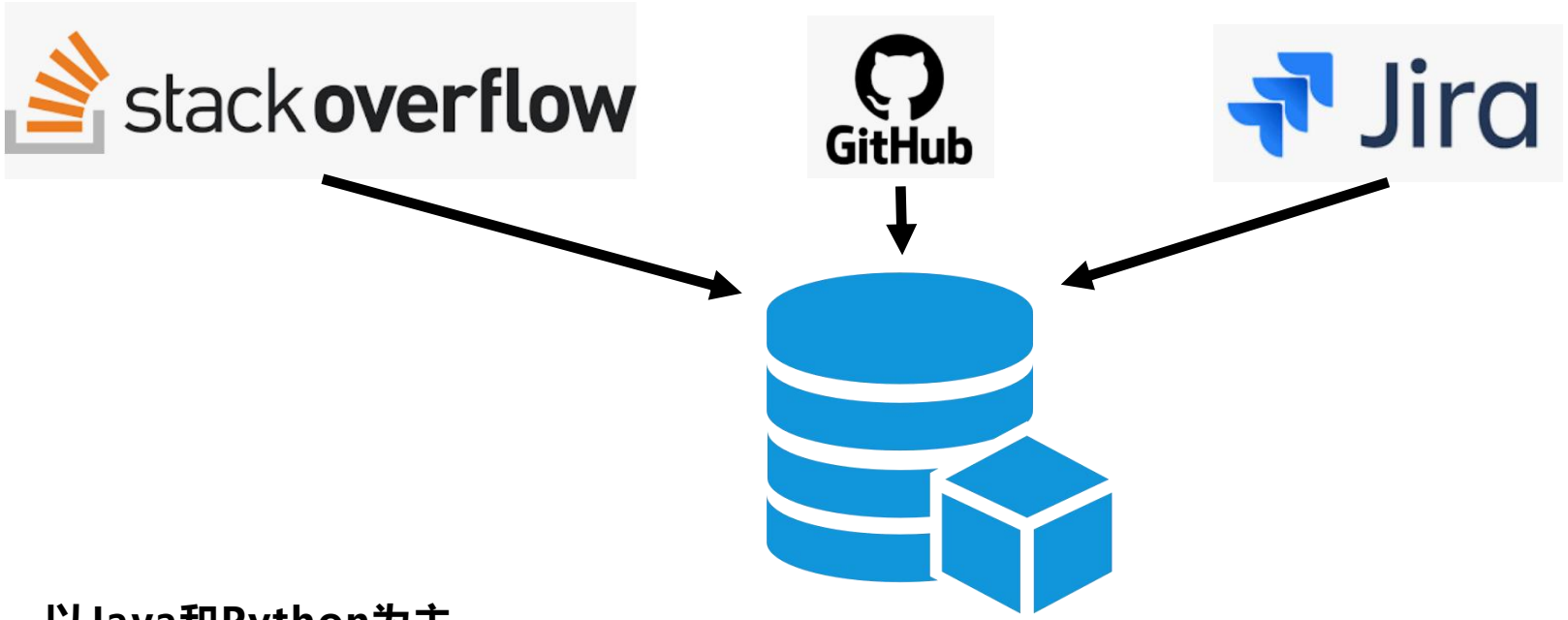
- **代码审查 (Code Review):** 即是开源和工业项目中广泛采用保障软件质量的一种实践过程。考虑到这一过程不可忽视的成本，研究者已研究通过大模型技术自动化特定代码审查任务。可以分为**代码更改质量评估**，根据提交审查的代码**生成评审人评审** (即 PL-To-NL 任务) 和根据提交审查代码审查人评论**生成评审人所要求的更改** (PL+NL-To-PL)。
- 该任务常用的性能评价指标有：Accuracy、precision、recall、F1、BLEU、EM等。
  - Accuracy、precision、recall、F1 用于评估代码更改质量，BLEU 和 EM 用于评价审查评论生成。

- **代码搜索 (Code Search):** 是给定一种自然语言作为输入，**从一组程序源代码中找到语义上最相关的代码**，即 NL-To-PL 任务。
- 该任务常用的性能评价指标有： MRR(Mean Reciprocal Rank)等。
  - MRR 用来评估搜索算法的统计量指标，为搜索 N 次代码结果倒数排名的平均值。
  - 倒数排名为搜索 N 次的第一正确答案排名的倒数乘积，MRR 表明了搜到的代码是否摆在用户更明显的位置。
  - MRR 越大，模型越好。

目录

预训练数据集

目前常用的大量软件工程领域数据集包括 SE 文本数据集、源代码数据集和 SE 文本和源代码的混合数据集。



- 以Java和Python为主
- 以英语为主
- 大多针对一种程序语言的代码进行训练
- 具训练集的大小跨度较大

目录

下游任务数据集

根据任务的不同，下游任务数据集可以分为 SE 文本数据集 (NL)、源代码数据集 (PL)、源代码和文本混合数据集以及图片数据集等。

- 针对 SE的各任务已有丰富的公开的数据集供研究者使用
- 在 NL 和 PL 交互任务中主要以 Python语言为主
- 在 PL 的代码片段分类任务中，特别是漏洞检测任务中主要是以 C/C++ 语言为主
- 在程序修复任务中，以 Java 和 JavaScript 程序语言为主

目录

思考

- **有效大模型技术在软件工程领域的应用值得关注**
  - 进一步整合深度学习领域最新的大模型研究成果，更好地编码表示 SE 领域的工件，涵盖软件工程领域自然语言以及不同程序语言的语法语义信息特征
  - 探索软件工程领域下游任务与不同大模型之间的关联，以便为不同 SE 下游任务选择合适的大模型技术提供指导。
- **组合多数据类型的预训练数据集值得关注**
  - 建立软件工程领域的特定大模型时，除了分析一些被充分代表的数据类型外，有机地组合多种数据类型（即多模态及跨语言数据集），从而训练出更具泛化性的预训练模型。
  - 对于 SE 任务中的源代码，考虑到源代码的文本描述外，还可以整合代码注释、源代码的抽象语法树（AST）以及数据流图（DFD）表示，以及相同语义的其他程序语言代码，形成多模态跨语言的源代码数据集。

思考

- **如何依据下游任务数据实现大模型微调值得关注**
  - 关注如何弥合预训练和特定任务微调之间的差距。固定 LLMs 的原始参数，通过为特定任务添加小型、可微调的自适应模块。这种方式可以使用一个共享的 PTM 来服务多个软件工程领域下游任务，更灵活地挖掘 LLMs 中的知识，避免在每个任务中都独立进行全模型微调的问题。
  - 考虑不从头开始微调面向任务的大模型，而是通过使用模型压缩技术，从现有的 PTMs 中提取部分特定任务的知识。

- **软件工程领域大模型的可解释性值得关注**
  - 尽管 LLMs 在软件工程领域任务中表现出令人印象深刻的性能，但其深层的非线性架构使得决策过程变得高度不透明。目前软件工程领域任务中绝大多数采用基于 Transformer 架构的 BERT 模型，这增加了解释 LLMs4SE 的难度。因此，作为解决软件工程领域下游任务的关键组件，大模型在软件工程领域的可解释性需要引起更多关注和深入探索。

谢谢！
Thanks!