大規模言語モデルによるソフトウェア脆弱性の検出

高知大学 理工学部情報科学科 横川武典

1. 研究背景

言語モデルがコード生成やデバッグ等に活用

→ 生成されるコードの**安全は担保されていない**

図1. コーディングに使われる言語モデルが含まれるツールの例

2. 研究目的

言語モデルで脆弱性を**自動で検知**

以下2つの問題に対応可能

1. 言語モデルが生成したコードは**安全と限らない**
   → コードに自動で脆弱性の検知を行い安全性を確保
2. 脆弱性を発見には**テスト**や**コードレビュー**が必要
   → 手間の削減が可能

3. 関連研究 [Sheng+2025]

- **コードの断片**から脆弱性を検出できる言語モデルが存在
- **リポジトリ**単位では限定的な脆弱性の検出のみ可能
- メモリ関連の脆弱性は検出精度が高い
- C/C++に関する研究が多い

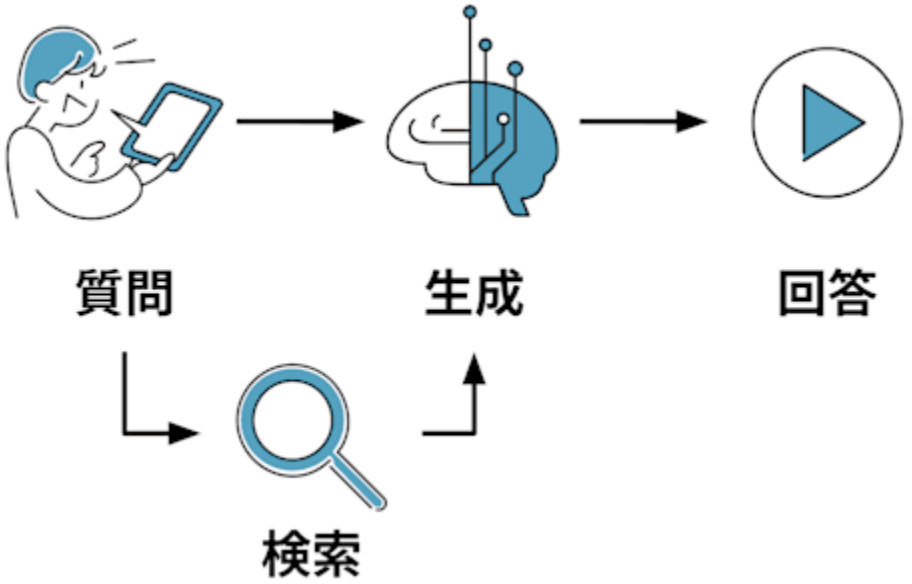
4. 研究手法

1. 脆弱性のデータベースを取得/作成
2. データベースを元に**RAG**の作成/**FineTuning**
    - データベースを参照/学習に使う
3. ソースコードから脆弱性を探す
    - 既存のソースコード
    - 言語モデルの生成したソースコード

RAG(Retrieval-Augmented Generation)とは

- RAGは事前学習していない**外部知識を参照**しその情報に基づいて文章を生成する手法

- 以下の3段階で機能する
  1. クエリと**類似する情報**をデータベースから参照
  2. **得られた情報**を入力に追加
  3. LLMで入力から推論



FineTuningとは

- 既存の学習済みモデルに，**追加の学習**を行い**特定のタスク用に調整**
- メリットは以下の3つ
  i. 一から学習しないため**コストが安い**
  ii. 最新の情報への対応
  iii. 特定の用途に**特化したモデル**を作成可

5. 現状の進捗

5-1. データの取得

- 脆弱性情報を管理するJVN/CVEから収集した脆弱性情報を元にデータベースを構築
- WordPressに関する脆弱性を取得しjsonで保管

```
"JVNDB-2025-009951": {
  "title": "AntoineH の WordPress 用 Football Pool における…",
  "description": "The Football Pool plugin for WordPress is …",
  "technologies": "AntoineH Football Pool 2.12.5 未満"
  },
```

図2. 脆弱性データベースの一例

5-2. 脆弱性の発見

- **調整していない言語モデル**に脆弱性を探させてみる
- 検証に使ったソースコードは"chatGPT-4o mini"で生成

1. オンラインのモデルとして"chatGPT-4o mini"を利用
   生成したphpのコード全文から脆弱性を探す
   → 簡単なコードの脆弱性は**検出/修正可**
2. ローカルのモデルとして"gpt-oss-20B"を利用
   生成したphpのコード全文から脆弱性を探す
   → 結果は**出力が不安定に**なることが大半

6. 今後の課題

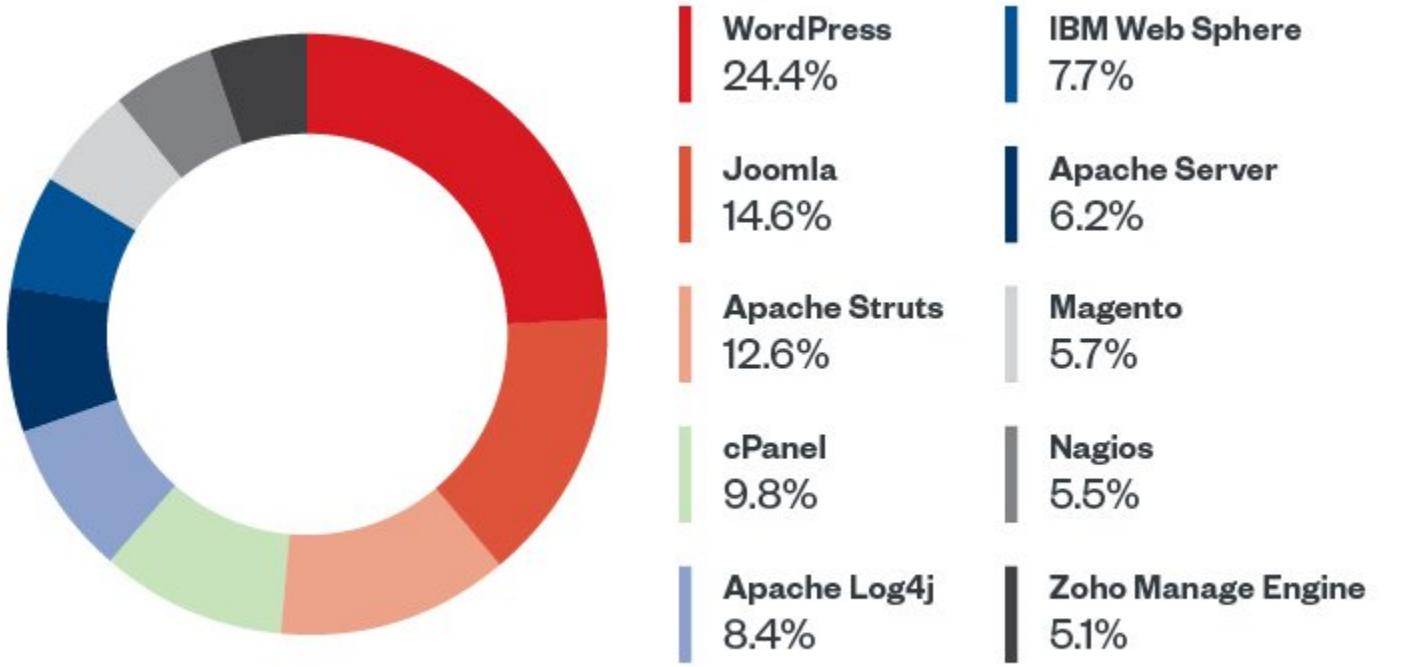
1. 入力長が長くても安定した出力を得られる言語モデルを探す
2. 脆弱性情報を元に言語モデルを**FineTuning**
   または**RAG**等を作成し言語モデルが情報を参照可能に
3. どの程度の脆弱性を発見できるか調査

参考文献

- LLMs in Software Security: A Survey of Vulnerability Detection Techniques and Insights [Ze Sheng+2025] https://arxiv.org/html/2502.07049v2
- JVN iPedia - 脆弱性対策情報データベース https://jvndb.jvn.jp/
- CVE: Common Vulnerabilities and Exposures https://www.cve.org/CVERecord

Appendix A. WEB言語の脆弱性の傾向

2022年に脆弱性が悪用された技術としてWordPressがトップ
全体の1/4を占めた



Appendix B. データベースからFineTuning/RAGを作成する手法

- RAGを作成する場合
  自然言語で書かれた脆弱性のデータベースを**ベクトル**に変換
  この処理はPythonの"LangChain"等のライブラリを使用
- FineTuningの場合
  オープンソースで公開されている言語モデルに脆弱性の
  データベースで**追加学習**
  この処理はPythonの"unsloth", "trl"等のライブラリを使用

Appendix C. ローカルで動作する言語モデルの不安定な出力

- gpt-oss-20b

  10回応答を生成して、問題を検出できたのは1度のみ

  クロスサイトリクエストフォージェリx1

```
<|end|><|start|>assistant!!!! for.
 You`` might. be Let's your..   the0..
 a let's
 theLet's..   be1.
the. . to!. Let's e this.. e .?
```

- Qwen3-14B
  10回応答を生成して、問題を検出できたのは7回
  セッションハイジャックx1
  クロスサイトリクエストフォージェリx7

```
But the results section is only displayed if $results is not empty.
 ⋮
the code doesn't display the results section again because $re
$
```
↑最終的な出力に移らず終了する