Software Engineering II - Study Notes

Administrative Details

- **Lecturer:** Shaowei Wang (shaowei@cs.umanitoba.ca)
- Sprint 4's deadline: Postponed to Dec 1st (Friday)
- Final project deliverable: Due on Dec 15th

Final Exam Details

- **Schedule:** Dec 13, 6 pm
- **Scope:** Lecture 1 - 12 (including fuzz testing, excluding data-driven SE)
- **Format:** In-person, digital download from UMlearn, submit to UMlearn
- **Type:** Multi-choice + problem-solving
- **Guideline:** Open book (Class notes, textbooks, slides allowed; no online search or generative AI)

Data-Driven Software Engineering

- Sources to mine from: Source Code, Issue Reports, QA Pairs (Stack Overflow), Pull Requests, Traces/Logs, Mobile Apps Marketplace

Artifact: Issue Reports

- Content in issue reports: Description, Steps to reproduce, Severity level, System parts affected, Failure traces
- Issue repositories: GitHub, Jira, Bugzilla, etc.

Applications

- Bug localization (identify bugs in source code from reports)
- Recommending developers for bug fixes
- Predicting bug severity

Bug Localization: Techniques

- Modeling similarity between reports and files/methods
    - Information retrieval techniques (e.g., Vector space model - tfidf, frequency-based, machine learning models like Word2vec, Doc2vec, BERT)
    - Observations on bug proneness (a file that recently had a bug is more likely to have bugs again; similar bugs often happen in bursts)
- Time-window capture for frequency of file commits
- Observations on previous bug reports (similar bugs likely in similar files)
- Hybrid techniques involving multiple data sources

Information Retrieval Techniques

- Use Vector Space Model (VSM):
    - Represent documents and queries as vectors

  - Compute similarity (e.g., cosine similarity)
  - Return top-k most similar documents

Stack Overflow Mining

- Investigate text descriptions, code snippets, votes, tags
- Utilize big data from SE crowd knowledge
- Empirical studies on developers' usage of code from Stack Overflow

Source Code Reuse from Stack Overflow

- Developers often modify reused code for different needs
- Suggestions for Stack Overflow improvements mostly revolve around code quality
- Advanced tagging systems and integration with validators could improve code reuse

Topic Modeling

- Black-Box View of Topic Modeling (e.g., LDA)
- Dimensionality reduction with topics
- Linking documents and queries that share related words of the same topics

Developer Recommendations for Bug Fixes

- Leverage activity profiles and expertise modeling (e.g., ownership, review history, topic model)
- Match developers and bug reports based on expertise
- Topic matching using similarity calculations

Challenges and Problems Faced by Developers

- Publications leveraging Stack Exchange data to understand developers' issues

Thesaurus in SE

- Software-specific terms differ from general use (e.g., Apple as a cell phone vs. fruit)
- Need for software-specific thesaurus to contain morphological forms

SEthesaurus and Term Semantics

- Collect software-specific corpus (Stack Overflow) and domain agnostic corpus (Wikipedia)
- Pre-processing for text cleaning, tokenization, phrase detection
- Build software-specific vocabulary by contrasting frequencies in specific and general corpora
- Extract semantically related terms using models and similarity measures

Embedding and Similarities

- Embed words by means of neighbors
- Continuous skip-gram model and FastText model for learning term semantics

Distinctions Between Synonyms and Abbreviations

- Use Levenshtein distance and heuristics-based lexical rules to discriminate between synonyms and abbreviations

Coverage

- Remember topics such as the ones mentioned above that are specified in the class notes provided for the exam preparation.

These notes are structured in a way to reflect the key topics mentioned in the lecture slides, organized sequentially for ease of review during the open-book exam. The markdown format allows for the straightforward addition of code or further notes if necessary during the study process or the exam itself.