**First-Step filtrate：**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Maintenance Standards | Maintenance Tasks | ACM DL Number | | Springer Number | | IEEE Xplore Number | |
| former | result | former | resut | former | result |
| Adaptive | Code generation | 3947 | 5 | 6096 | 3 | 353 | 2 |
| Comment generation | 3274 | 5 | 1794 | 1 | 760 | 7 |
| Documentation generation | 3462 | 3 | 1523 | 1 | 207 | 6 |
| Perfective | Pull requests | 2286 | 7 | 548 | 2 | 139 | 11 |
| Intention mining | 0 | 0 | 0 | 0 | 199 | 7 |
| API summary | 2811 | 10 | 0 | 0 | 30 | 5 |
| Corrective | Bug report | 643 | 3 | 740 | 8 | 654 | 24 |
| Error detection | 2969 | 4 | 1159 | 2 | 0 | 0 |

**Paper List：**

Code Generation:

**[Coverage-driven test code generation for concurrent classes](https://dl.acm.org/doi/abs/10.1145/2884781.2884876)**

**[Online model editing, simulation and code generation for web and mobile applications](https://dl.acm.org/doi/abs/10.5555/3104068.3104079)**

**[WL++: code generation of multi-platform mobile clients to RESTful back-ends](https://dl.acm.org/doi/abs/10.5555/2825041.2825072)**

[**The Gamma statechart composition framework: Design, verification and code generation for component-based reactive systems**](https://dl.acm.org/doi/abs/10.1145/3183440.3183489)

[**On the naturalness of auto-generated code: can we identify auto-generated code automatically?**](https://dl.acm.org/doi/abs/10.1145/3196321.3196356)

[**Developing a software system for automata-based code generation**](https://link.springer.com/article/10.1134/S0361768816030075)

[**A Framework for UML-Based Component-Based Design and Code Generation for Reactive Systems**](https://link.springer.com/chapter/10.1007/978-3-319-94764-8_13)

[**Automated Test Case Generation for the CTRL Programming Language Using Pex: Lessons Learned**](https://link.springer.com/chapter/10.1007/978-3-319-45892-2_9)

[**Pseudogen: A Tool to Automatically Generate Pseudo-Code from Source Code**](https://ieeexplore.ieee.org/document/7372074/)

[**A Reflexive and Automated Approach to Syntactic Pattern Matching in Code Transformations**](https://ieeexplore.ieee.org/document/8530049/)

Comment Generation:

[**Deep code comment generation**](https://dl.acm.org/doi/abs/10.1145/3196321.3196334)

[**MCT: a tool for commenting programs by multimedia comments**](https://dl.acm.org/doi/abs/10.5555/2486788.2487000)

[**A large-scale empirical study on code-comment inconsistencies**](https://dl.acm.org/doi/abs/10.1109/ICPC.2019.00019)

[**Classifying code comments in Java open-source software systems**](https://dl.acm.org/doi/abs/10.1109/MSR.2017.63)

[**Generating commit messages from diffs using pointer-generator network**](https://dl.acm.org/doi/abs/10.1109/MSR.2019.00056)

[**Learning to Generate Comments for API-Based Code Snippets**](https://link.springer.com/chapter/10.1007/978-981-15-0310-8_1)

**[Retrieve and Refine: Exemplar-Based Neural Comment Generation](https://ieeexplore.ieee.org/document/8952536/)**

**[Learning Comment Generation by Leveraging User-generated Data](https://ieeexplore.ieee.org/document/8682945/)**

[**CloCom: Mining existing source code for automatic comment generation**](https://ieeexplore.ieee.org/document/7081848/)

[**AutoComment: Mining question and answer sites for automatic comment generation**](https://ieeexplore.ieee.org/document/6693113/)

[**A Survey of Automatic Generation of Source Code Comments: Algorithms and Techniques**](https://ieeexplore.ieee.org/document/8778714/)

[**Automatic comment generation using only source code**](https://ieeexplore.ieee.org/document/7960702/)

[**Neural Comment Generation for Source Code with Auxiliary Code Classification Task**](https://ieeexplore.ieee.org/document/8945708/)

Documentation Generation

[**Automatic documentation generation via source code summarization**](https://dl.acm.org/doi/abs/10.5555/2819009.2819210)

**[Documentation generation from annotated source code of scientific software: position paper](https://dl.acm.org/doi/abs/10.1145/2897676.2897679)**

**[Automatically generating documentation for lambda expressions in Java](https://dl.acm.org/doi/abs/10.1109/MSR.2019.00057)**

[**Generating Software Documentation in Use Case Maps from Filtered Execution Traces**](https://link.springer.com/chapter/10.1007/978-3-319-24912-4_13)

**[Automatic Generation of API Documentations for Open-Source Projects](https://ieeexplore.ieee.org/document/8530111/)**

**[A Systematic Mapping Study on API Documentation Generation Approaches](https://ieeexplore.ieee.org/document/8498248/)**

[**Behavior-Informed Algorithms for Automatic Documentation Generation**](https://ieeexplore.ieee.org/document/8094477/)

[**ARENA: An Approach for the Automated Generation of Release Notes**](https://ieeexplore.ieee.org/document/7513412/)

[**DevDocOps: Towards Automated Documentation for DevOps**](https://ieeexplore.ieee.org/document/8804428/)

Pull Request:

[**Studying pull request merges: a case study of shopify's active merchant**](https://dl.acm.org/doi/abs/10.1145/3183519.3183542)

[**Wait for it: determinants of pull request evaluation latency on GitHub**](https://dl.acm.org/doi/abs/10.5555/2820518.2820564)

[**EARec: leveraging expertise and authority for pull-request reviewer recommendation in GitHub**](https://dl.acm.org/doi/abs/10.1145/2897659.2897660)

[**Replication can improve prior results: a GitHub study of pull request acceptance**](https://dl.acm.org/doi/abs/10.1109/ICPC.2019.00037)

[**How developers document pull requests with external references**](https://dl.acm.org/doi/abs/10.1109/ICPC.2017.30)

[**Automatically prioritizing pull requests**](https://dl.acm.org/doi/abs/10.5555/2820518.2820562)

**[Beyond the code itself: how programmers](https://dl.acm.org/doi/abs/10.1109/ICSE-SEIS.2019.17) *[really](https://dl.acm.org/doi/abs/10.1109/ICSE-SEIS.2019.17)* [look at pull requests](https://dl.acm.org/doi/abs/10.1109/ICSE-SEIS.2019.17)**

**[Changes Are Similar: Measuring Similarity of Pull Requests That Change the Same Code in GitHub](https://link.springer.com/chapter/10.1007/978-981-15-0310-8_8)**

[**What Are They Talking About? Analyzing Code Reviews in Pull-Based Development Model**](https://link.springer.com/article/10.1007/s11390-017-1783-2)

**Automatic Generation of Pull Request Descriptions**

**Locating latent design information in developer discussionsA study on pull requests**

[**How do Multiple Pull Requests Change the Same Code: A Study of Competing Pull Requests in GitHub**](https://ieeexplore.ieee.org/document/8530032/)

[**A Study on the Interplay between Pull Request Review and Continuous Integration Builds**](https://ieeexplore.ieee.org/document/8667996/)

[**Studying the Impact of Adopting Continuous Integration on the Delivery Time of Pull Requests**](https://ieeexplore.ieee.org/document/8595196/)

**[Effects of Personality Traits on Pull Request Acceptance](https://ieeexplore.ieee.org/document/8935389/)**

**[Understanding the Effect of Developer Sentiment on Fix-Inducing Changes: An Exploratory Study on GitHub Pull Requests](https://ieeexplore.ieee.org/document/8945613/)**

[**Programming Style Analysis with Recurrent Neural Network to Automatic Pull Request Approval**](https://ieeexplore.ieee.org/document/8852255/)

[**Can automated pull requests encourage software developers to upgrade out-of-date dependencies?**](https://ieeexplore.ieee.org/document/8115621/)

[**NBSL: A Supervised Classification Model of Pull Request in Github**](https://ieeexplore.ieee.org/document/8422103/)

[**EARec: Leveraging Expertise and Authority for Pull-Request Reviewer Recommendation in GitHub**](https://ieeexplore.ieee.org/document/7809395/)

Intention Mining: (Relevant advanced technology and few papers, most papers are selected from IEEE Xplore)

[**User Intention Mining in Bussiness Reviews: A Review**](https://ieeexplore.ieee.org/document/8697303/)

**Automating Intention Mining**

[**Development Emails Content Analyzer: Intention Mining in Developer Discussions (T)**](https://ieeexplore.ieee.org/document/7371991/)

[**Communicative Intention in Code Review Questions**](https://ieeexplore.ieee.org/document/8530057/)

**[SMARTLOG: Place error log statement by deep understanding of log intention](https://ieeexplore.ieee.org/document/8330197/)**

**[Contributor's Performance, Participation Intentions, Its Influencers and Project Performance](https://ieeexplore.ieee.org/document/7203114/)**

API Summary/Summarize sth from source code:

[**A learning-based approach for automatic construction of domain glossary from source code and documentation**](https://dl.acm.org/doi/abs/10.1145/3338906.3338963)

[**Automatic model generation from documentation for Java API functions**](https://dl.acm.org/doi/abs/10.1145/2884781.2884881)

**[Using stereotypes in the automatic generation of natural language summaries for C++ methods](https://ieeexplore.ieee.org/document/7332514/)**

**[Leveraging informal documentation to summarize classes and methods in context](https://dl.acm.org/doi/abs/10.5555/2819009.2819125)**

**[Developer reading behavior while summarizing Java methods: size and context matters](https://dl.acm.org/doi/abs/10.1109/ICSE.2019.00052)**

[**Exploring API embedding for API usages and applications**](https://dl.acm.org/doi/abs/10.1109/ICSE.2017.47)

[**Towards crowd-sourced API documentation**](https://dl.acm.org/doi/abs/10.1109/ICSE-Companion.2019.00129)

[**Statistical migration of API usages**](https://dl.acm.org/doi/abs/10.1109/ICSE-C.2017.17)

[**Inferring API elements relevant to an english query**](https://dl.acm.org/doi/abs/10.1145/3183440.3195079)

[**Learning API usages from bytecode: a statistical approach**](https://dl.acm.org/doi/abs/10.1145/2884781.2884873)

[**Predicate Callback Summaries**](https://ieeexplore.ieee.org/document/7965334/)

[**Automatic summarization of API reviews**](https://ieeexplore.ieee.org/document/8115629/)

[**Generating Predicate Callback Summaries for the Android Framework**](https://ieeexplore.ieee.org/document/7972720/)

[**Opiner: An opinion search and summarization engine for APIs**](https://ieeexplore.ieee.org/document/8115715/)

[**Understanding How and Why Developers Seek and Analyze API-related Opinions**](https://ieeexplore.ieee.org/document/8658125/)

Bug Report: (bug-detection is mostly included while developing, and bug-report driven maintenance is conducted after the developing step)

**iFixR: bug report driven program repair**

[**Improving IR-based bug localization with context-aware query reformulation**](https://dl.acm.org/doi/abs/10.1145/3236024.3236065)

[**ReCDroid: automatically reproducing Android application crashes from bug reports**](https://dl.acm.org/doi/abs/10.1109/ICSE.2019.00030)

[**Automated prediction of bug report priority using multi-factor analysis**](https://link.springer.com/article/10.1007/s10664-014-9331-y)

[**A contextual approach towards more accurate duplicate bug report detection and ranking**](https://link.springer.com/article/10.1007/s10664-015-9387-3)

[**Modelling the ‘hurried’ bug report reading process to summarize bug reports**](https://link.springer.com/article/10.1007/s10664-014-9311-2)

[**Structured information in bug report descriptions—influence on IR-based bug localization and developers**](https://link.springer.com/article/10.1007/s11219-019-09445-6)

[**Preventing duplicate bug reports by continuously querying bug reports**](https://link.springer.com/article/10.1007/s10664-018-9643-4)

**[An Analysis of Software Bug Reports Using Machine Learning Techniques](https://link.springer.com/article/10.1007/s42979-019-0004-1)**

**[Invalid bug reports complicate the software aging situation](https://link.springer.com/article/10.1007/s11219-019-09481-2)**

[**On the Effectiveness of Labeled Latent Dirichlet Allocation in Automatic Bug-Report Categorization**](https://ieeexplore.ieee.org/document/7883381/)

[**Can We Detect Bug Report Duplication with Unfinished Bug Reports?**](https://ieeexplore.ieee.org/document/7467295/)

[**Poster: DWEN: Deep Word Embedding Network for Duplicate Bug Report Detection in Software Repositories**](https://ieeexplore.ieee.org/document/8449496/)

[**Poster: Improving Bug Localization with Report Quality Dynamics and Query Reformulation**](https://ieeexplore.ieee.org/document/8449572/)

[**Influence of Structured Information in Bug Report Descriptions on IR-Based Bug Localization**](https://ieeexplore.ieee.org/document/8498181/)

[**Key Features Recommendation to Improve Bug Reporting**](https://ieeexplore.ieee.org/document/8812831/)

[**A Method of Bug Report Quality Detection Based on Vector Space Model**](https://ieeexplore.ieee.org/document/8859459/)

[**Improving Bug Reporting, Duplicate Detection, and Localization**](https://ieeexplore.ieee.org/document/7965374/)

[**Poster: LWE: LDA Refined Word Embeddings for Duplicate Bug Report Detection**](https://ieeexplore.ieee.org/document/8449482/)

[**Bug or Not? Bug Report Classification Using N-Gram IDF**](https://ieeexplore.ieee.org/document/8094457/)

[**Recommending Similar Bug Reports: A Novel Approach Using Document Embedding Model**](https://ieeexplore.ieee.org/document/8719552/)

[**Bug Localization Based on Code Change Histories and Bug Reports**](https://ieeexplore.ieee.org/document/7467300/)

[**Source Code Retrieval for Bug Localization using Bug Report**](https://ieeexplore.ieee.org/document/8959535/)

[**Automated Bug Report Field Reassignment and Refinement Prediction**](https://ieeexplore.ieee.org/document/7307231/)

[**Exploring Metadata in Bug Reports for Bug Localization**](https://ieeexplore.ieee.org/document/8305955/)

[**Detecting Duplicate Bug Reports with Convolutional Neural Networks**](https://ieeexplore.ieee.org/document/8719497/)

[**BugIdentifier: An Approach to Identifying Bugs via Log Mining for Accelerating Bug Reporting Stage**](https://ieeexplore.ieee.org/document/8854736/)

[**Combining Deep Learning with Information Retrieval to Localize Buggy Files for Bug Reports (N)**](https://ieeexplore.ieee.org/document/7372035/)

[**Reorganizing and Optimizing Post-Inspection on Suspicious Bug Reports in Path-Sensitive Analysis**](https://ieeexplore.ieee.org/document/8854722/)

[**DURFEX: A Feature Extraction Technique for Efficient Detection of Duplicate Bug Reports**](https://ieeexplore.ieee.org/document/8009929/)

[**Automated Identification of High Impact Bug Reports Leveraging Imbalanced Learning Strategies**](https://ieeexplore.ieee.org/document/7552013/)

[**Combining Word Embedding with Information Retrieval to Recommend Similar Bug Reports**](https://ieeexplore.ieee.org/document/7774514/)

[**Mapping Bug Reports to Relevant Files: A Ranking Model, a Fine-Grained Benchmark, and Feature Evaluation**](https://ieeexplore.ieee.org/document/7270328/)

[**Bug report, feature request, or simply praise? On automatically classifying app reviews**](https://ieeexplore.ieee.org/document/7320414/)

[**Text Filtering and Ranking for Security Bug Report Prediction**](https://ieeexplore.ieee.org/document/8240740/)

Error Detection/Fault Detection/Code Smell Detection:

[**DoubleTake: fast and precise error detection via evidence-based dynamic analysis**](https://dl.acm.org/doi/abs/10.1145/2884781.2884784)

[**Modeling time in Java programs for automatic error detection**](https://dl.acm.org/doi/abs/10.1145/3193992.3193997)

[**Developer mistakes in writing Android manifests: an empirical study of configuration errors**](https://dl.acm.org/doi/abs/10.1109/MSR.2017.41)

[**CUSTODES: automatic spreadsheet cell clustering and smell detection using strong and weak features**](https://dl.acm.org/doi/abs/10.1145/2884781.2884796)

[**Generalized software fault detection and correction modeling framework through imperfect debugging, error generation and change point**](https://link.springer.com/article/10.1007/s41870-019-00321-x)

**[Early detection of evolving system failures and temporal conflicts using parameterized formal specifications and bounded constraint-solving](https://link.springer.com/article/10.1007/s11334-015-0244-8)**