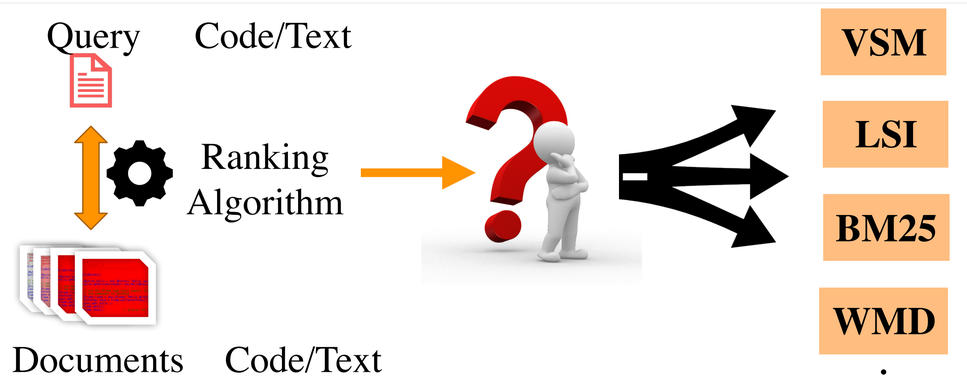
Which Similarity Metric to Use for Software Documents?

A study on Information Retrieval based Software Engineering Tasks

Md Masudur Rahman *masud@virginia.edu*
Saikat Chakraborty *saikatc@virginia.edu*
Baishakhi Ray *rayb@virginia.edu*

- Different IR-based Software Engineering (SE) tasks operate on different document artifacts.
- SE artifacts are heterogeneous in nature and different in characteristics than natural language.
- We investigate the impact of IR models on different SE artifacts and analyze how such informed choice could lead to an improved performance.



Representative Software Engineering Task

Project Recommendation
Given a project as a query, the task is to find functionally similar projects from GitHub. For example, given a query with a Video Recorder project, the system tries to return a list of Video Recorder projects.

Dataset: 1590 GitHub projects

Text: Description, Readme
Code: Method-Class name, Package name, API

Bug Localization
Given a bug report as query, this task ranks all the source files in the project repository based on their relevance with the query. The files that top the ranking are more likely to contain the cause of the bug.

Dataset: 200 bug reports of JDT

Mixture of text and code: Bug reports, source code

IR Model

- Vector Space Model (**VSM**)
- Latent Semantic Indexing (**LSI**)
- **BM25**
- Embedding based Word Mover's Distance (**WMD**)

Evaluation Metric

- Mean Average Precision (MAP)
- Mean Reciprocal Rank (MRR)

Performance of Different Models on Various SE Artifacts (MAP@10)

Model	Text - Text: Description	Text - Text: Readme	Code - Code: Method Class	Code - Code: Package Name	Code - Code: API	Text - Code: Bug Reports
VSM	0.51	0.37	**0.37**	**0.29**	**0.31**	0.06
LSI	**0.57**	**0.39**	0.36	0.24	0.25	0.02
BM25	0.51	0.26	0.16	0.007	0.22	**0.28**
WMD	0.51	0.29	0.25	0.20	0.25	0.001

☑ Context-aware models such as LSI and WMD perform better on textual artifacts.
☑ **LSI** performs the best for text-text documents.

☑ Keyword based bag-of-words(BOW) model **VSM** performs best for code only artifacts.
☑ Code artifacts lack context information.

☑ For mixture (text - code) documents, **BM25** performs the best.
☑ Surprisingly, BM25 is not that effective for text only and code only artifacts.

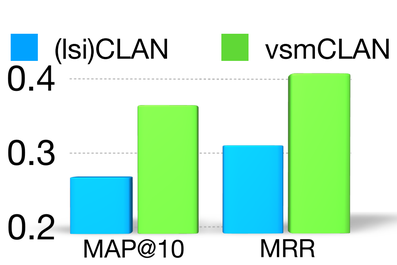
Can an informed choice of IR model improve the performance of SE tasks?

Project Recommendation

We replace the ranking algorithm from LSI to VSM of CLAN[1], a project recommendation tool.

- CLAN leverages code - code document artifacts.
- Our experiments show VSM performs best for such code - code artifacts.

☑ Modified model, vsmCLAN achieves up to 35% improved performance.

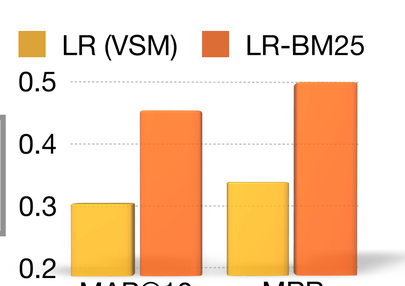


Bug Localization

We replace the ranking algorithm from VSM to BM25 of LR[2], a bug localization tool.

- LR leverages mixture (text - code) document artifacts.
- Our experiments show BM25 performs best for such mixture artifacts.

☑ Modified model, LR-BM25 achieves 43% improved performance.



Reference:
1. Collin McMillan, Mark Grechanik, and Denys Poshyvanyk. 2012. Detecting similar software applications. In 2012 34th International Conference on Software Engineering (ICSE). IEEE, 364–374
2. Xin Ye, Razvan Bunescu, and Chang Liu. 2014. Learning to rank relevant files for bug reports using domain knowledge. In Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. ACM, 689–699

Department of Computer Science
University of Virginia

ICSE 40TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING MAY 27 - JUNE 3 2018 GOTHENBURG, SWEDEN