ZHE YU

Department of Software Engineering, RIT, Rochester, NY, 14623
(919)-949-1382 ◇ https://zhe-yu.github.io ◇ zxyvse@rit.edu

I believe the future of AI is not replacing humans, but, rather, better supporting humans with automated intelligences. Hence, my research focuses on the creation of "human in the loop" machine learning environments.

RESEARCH INTEREST

Software Engineering, Machine Learning, Information Retrieval, Human-Computer Interaction

EDUCATION

PhD in Computer Science Aug 2015 - May 2020
North Carolina State University

MS in Control Science and Engineering Sep 2011 - Mar 2014
Shanghai Jiao Tong University

BS in Automation Sep 2007 - May 2011
Shanghai Jiao Tong University

EMPLOYMENT

Assistant Professor Aug 2020 - Present
Department of Software Engineering Rochester Institute of Technology, Rochester, NY

Graduate Research Assistant Aug 2015 - May 2020
Department of Computer Science North Carolina State University, Raleigh, NC

Summer Intern May 2019 - Aug 2019
Traffic Estimation for Ads Google, Los Angeles, CA

Summer Intern May 2018 - Aug 2018
Data Engine Machine Learning Google, Mountain View, CA

Summer Intern May 2017 - Aug 2017
LexisNexis, Raleigh, NC

Summer Intern May 2016 - Aug 2016
LexisNexis, Raleigh, NC

SERVICE

Conference Committee Member: MSR 2021, SCAM 2020 NIER track

Conference Review: CSCW 2020

Journal Review: IEEE Transactions on Software Engineering (TSE), ACM Transactions on Software Engineering and Methodology (TOSEM), Empirical Software Engineering (EMSE), Information and Software Technology (IST), IEEE Access

TEACHING

DSCI-633: Foundation of Data Science

RECENT RESEARCH PROJECTS

Few-shot Active Learning for Better Information Retrieval Aug 2020 - Present
Rochester Institute of Technology

- This project aims to improve efficiency of information retrieval with a combination of few-shot learning and active learning.
- When a deep neural network model learns to retrieve different relevant information (of the same type, e.g. literature reviews of different topics), it optimizes its inner structure so that it will learn faster and better to retrieve new, unseen information.
- Active learning is also import in retrieving new, unseen information by (1) continuously learning from human decisions on which is relevant and (2) suggesting what should be reviewed next based on the learned model.

Human Ethical Bias Detection With Machine Learning Software Aug 2020 - Present
Rochester Institute of Technology

- Instead of trying to reduce bias in machine learning software, this project aims to reduce bias from its source—the human decisions.
- That is, this work proposes to utilize the machine learning bias inherited from the training data (human decisions) as an indicator for detecting human bias.
- While it is difficult to directly test whether a human has bias, with current research on machine learning bias, it is now easy to test, in large scale at low cost, whether a machine learning software has bias.

Total Recall and Software Engineering Aug 2015 - May 2020
North Carolina State University *Partially funded by an NSF Grant*

- Apply machine learning algorithms to support human retrieve all desired information from big data with less effort, a class of information retrieval problem called total recall.
- Developed an active learning based framework—FASTREAD—to support fast selection of primary studies in systematic reviews and all the total recall problems.
  - Validated in simulations, FASTREAD was usually able to find 95% of relevant studies by asking humans to review 10% of the candidates, which outperformed the prior state of the art total recall solutions.
  - FASTREAD accurately estimated the total number of relevant studies in the candidates and provided a reliable stopping rule for high target recalls, e.g. 90%, 95%, or 99%.
  - FASTREAD suggested which labels should be double checked to correct human errors. By double checking 50% of the labeled studies, 96% of the human errors could be covered.
- A tool has been developed to implement FASTREAD at `https://github.com/fastread/src`.
- Same idea applied to solve other software engineering problems such as software security vulnerability prediction and test case prioritization.

Test Case Prioritization for Automated UI Testing Sep 2018 - Apr 2019
Cooperation project with LexisNexis Legal & Professional

- Conducted a systematic literature review on test case prioritization researches, using the FASTREAD tool.
  - Validated by 6 graduate students, 90% of the relevant studies were found by reviewing 6% of the candidates with FASTREAD targeting at 90% recall.
- Proposed a novel test case prioritization framework by adapting FASTREAD to the automated UI testing problem.
- Improved performance by 9% (measured in APFDc) using the proposed framework.

Social Network of US Public Companies Feb 2018 - Dec 2018
Cooperation project with LexisNexis Legal & Professional

- Extracted board of directors from 10-K filings by rule-based named entity recognition.
- Connected companies with mutual board of directors (find connected components in the graph).
- Found that 40% US public companies were fully connected with each other while the rest were isolated ones.
- Validated that 70% of the US top 500 companies were fully connected. This suggested that companies connected with others are more likely to succeed.

Scalable FASTREAD on HPCC Systems Feb 2017 - Dec 2017
Cooperation project with LexisNexis Legal & Professional

- Implemented FASTREAD tool on HPCC Systems for high scalability.
- Enabled multi-users to work on the same project in parallel.

Youtube eCPM Seasonality in TEA May 2019 - Aug 2019
Internship at Google (TEA: Traffic Estimation for Ads)

- Analyzed which features are significantly correlated to outliers (when forecasts were way off from actuals).
  - Enabled null hypothesis tests on scalar features.
  - Built a new feature for the internal validation tool to analyze outlier features in drilldown pages.
- Added seasonality predictions to the current TEA forecasts
  - Improved seasonality predictions—time series analysis on previous years to predict the curve in next year.
  - Validated the overall TEA forecasting performance improvement with seasonality predictions via A/B testing.

KIWI: Knowledge In Web Images May 2018 - Aug 2018
Internship at Google (DEML: Data Engine Machine Learning)

- Mined image-entity pairs in web images with alt text, image url, etc. Trained a model to measure the image-entity pair quality and filter out low-quality pairs.
  - Trained a dual encode model, between entity and image starburst.

- Designed and tried different metrics to evaluate the model performance.
- Added a feature to dual encoder framework to support dense feature.

Legal Document Headnotes Generation and Classification May 2017 - Aug 2017
Internship at LexisNexis Legal & Professional

- Developed a text summarization framework for generating"headnote" of more than 1 million legal documents.
- Designed a scalable classification scheme with doc2vec to categorize documents into specific legal topics.
- Demonstrated that the above framework can reduce document review time by $\geq 50\%$ according to user surveys.

Improve Legal Document Retrieval Efficiency of DiscoveryIQ May 2016 - Aug 2016
Internship at LexisNexis Legal & Professional

- Created a sandbox for prototyping new DiscoveryIQ features.
- Developed new features to "open the black box" of DiscoveryIQ.
- Incorporate new features into the current DiscoveryIQ product.

SELECTED PUBLICATIONS

[1] Zhe Yu, Fahmid Morshed Fahid, Huy Tu, and Tim Menzies. "Identifying Self-Admitted Technical Debts with Jitterbug: A Two-step Approach." IEEE Transactions on Software Engineering.

[2] Zhe Yu, Jeffrey C. Carver, Gregg Rothermel, Tim Menzies. 2020. "Searching for Better Test Case Prioritization Schemes: a Case Study of AI-assisted Systematic Literature Review." **Under Review.**

[3] Yang, Xueqi, Zhe Yu, Junjie Wang, and Tim Menzies. "An Expert System for Learning Software Engineering Knowledge (with Case Studies in Understanding Static Code Warning)." Expert Systems with Applications.

[4] Yang, Xueqi, Jianfeng Chen, Rahul Yedida, Zhe Yu, and Tim Menzies. "How to Recognize Actionable Static Code Warnings (Using Linear SVMs)." **Under Review.**

[5] Chakraborty, Joymallya, Suvodeep Majumder, Zhe Yu, and Tim Menzies. "Fairway: A Way to Build Fair ML Software." In Proceedings of ESEC/FSE 2020.

[6] Agrawal, Amritanshu, Tim Menzies, Leandro L. Minku, Markus Wagner, and Zhe Yu. "Better software analytics via DUO: Data mining algorithms using/used-by optimizers." Empirical Software Engineering 25, no. 3 (2020): 2099-2136.

[7] Huy Tu, Zhe Yu, Tim Menzies. 2020. "Better Data Labelling with EMBLEM (and how that Impacts Defect Prediction)." IEEE Transactions on Software Engineering.

[8] Zhe Yu, Christopher Theisen, Laurie Williams, Tim Menzies. 2019. "Improving Vulnerability Inspection Efficiency Using Active Learning." IEEE Transactions on Software Engineering.

[9] Zhe Yu, Fahmid M. Fahid, Tim Menzies, Gregg Rothermel, Kyle Patrick, Snehit Cherian. 2019. "TERMINATOR: Better Automated UI Test Case Prioritization." In Proceedings of ESEC/FSE'19, Software Engineering in Practice, 883-894. http://doi.acm.org/10.1145/3338906.3340448

[10] Zhe Yu and Tim Menzies. 2019. "FAST2: An intelligent assistant for finding relevant papers." Expert Systems with Applications. 120: 57-71. https://www.sciencedirect.com/science/article/pii/S0957417418307413

[11] Zhe Yu, Nicholas A. Kraft, and Tim Menzies. 2018. "Finding Better Active Learners for Faster Literature Reviews." Empirical Software Engineering. 23(6): 3161-3186. https://link.springer.com/article/10.1007/s10664-017-9587-0

[12] Zhe Yu and Tim Menzies. 2018. "Total recall, language processing, and software engineering." In Proceedings of NL4SE Workshop 2018, 10-13. https://dl.acm.org/citation.cfm?id=3283818

[13] Vivek Nair, Zhe Yu, Tim Menzies, Norbert Siegmund, and Sven Apel. 2018. "Finding faster configurations using flash." IEEE Transactions on Software Engineering.

[14] Vivek Nair, Amrit Agrawal, Jianfeng Chen, Wei Fu, George Mathew, Tim Menzies, Leandro Minku, Markus Wagner, and Zhe Yu. 2018. "Data-Driven Search-based Software Engineering." The Mining Software Repositories (MSR).

[15] Agrawal, Amritanshu, Tim Menzies, Leandro L. Minku, Markus Wagner, and Zhe Yu. "Better software analytics via" DUO": Data mining algorithms using/used-by optimizers." Empirical Software Engineering 25, no. 3 (2020): 2099-2136.

[16] Zhe Yu and Tim Menzies. 2017. "Data Balancing for Technologically Assisted Reviews: Undersampling or Reweighting." In CLEF (Working Notes). http://ceur-ws.org/Vol-1866/paper_120.pdf