**REFERENCES**

[1] G. Antoniol and Y.-G. Gueheneuc, “Feature identification: A novel approach and a case study,” in Proc. 21st IEEE Int. Conf. Softw. Maintenance,Washington, DC, USA, 2005, pp. 357–366.

[2] G. Antoniol and Y.-G. Gueheneuc, “Feature identification: An epidemiological metaphor,” IEEE Trans. Softw. Eng., vol. 32, no. 9, pp. 627–641, Sep. 2006.

[3] B. Ashok, J. Joy, H. Liang, S. K. Rajamani, G. Srinivasa, and V. Vangala, “Debugadvisor: A recommender system for debugging,” in Proc. 7th Joint Meeting Eur. Softw. Eng. Conf. ACM SIGSOFT Symp. Found. Softw. Eng., New York, NY, USA, 2009, pp. 373–382.

[4] A. Bacchelli and C. Bird, “Expectations, outcomes, and challenges of modern code review,” in Proc. Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2013, pp. 712–721.

[5] S. K. Bajracharya, J. Ossher, and C. V. Lopes, “Leveraging usage similarity for effective retrieval of examples in code repositories,” in Proc. 18th ACM SIGSOFT Int. Symp. Found. Softw. Eng., New York, NY, USA, 2010 pp. 157–166.

[6] R. M. Bell, T. J. Ostrand, and E. J. Weyuker, “Looking for bugs in all the right places,” in Proc. Int. Symp. Softw. Testing Anal., New York, NY, USA, 2006, pp. 61–72.

[7] N. Bettenburg, S. Just, A. Schr€oter, C. Weiss, R. Premraj, and T. Zimmermann, “What makes a good bug report?” in Proc. 16th ACM SIGSOFT Int. Symp. Found. Softw. Eng., New York, NY, USA, 2008, pp. 308–318.

[8] T. J. Biggerstaff, B. G. Mitbander, and D. Webster, “The concept assignment problem in program understanding,” in Proc. 15th Int. Conf. Softw. Eng., Los Alamitos, CA, USA, 1993, pp. 482–498.

[9] D. Binkley and D. Lawrie, “Learning to rank improves IR in SE,” in Proc. IEEE Int. Conf. Softw. Maintenance Evol., Washington, DC, USA, 2014, pp. 441–445.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” J. Mach. Learn. Res., vol. 3, pp. 993–1022 Mar. 2003.

[11] S. Breu, R. Premraj, J. Sillito, and T. Zimmermann, “Information needs in bug reports: Improving cooperation between developers and users,” in Proc. ACM Conf. Comput. Supported Cooperative Work, New York, NY, USA, 2010, pp. 301–310.

[12] B. Bruegge and A. H. Dutoit, Object-Oriented Software Engineering Using UML, Patterns, and Java, 3rd ed. Upper Saddle River, NJ, USA, Prentice-Hall, 2009.

[13] Y. Brun and M. D. Ernst, “Finding latent code errors via machine learning over program executions,” in Proc. 26th Int. Conf. Softw. Eng.,Washington, DC, USA, 2004, pp. 480–490.

[14] M. Burger and A. Zeller, “Minimizing reproduction of software failures,” in Proc. Int. Symp. Softw. Testing Anal., New York, NY, USA, 2011 pp. 221–231.

[15] R. P. L. Buse and T. Zimmermann, “Information needs for software development analytics,” in Proc. Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2012, pp. 987–996.

[16] H. Cleve and A. Zeller, “Locating causes of program failures,” in Proc. 27th Int. Conf. Softw. Eng., New York, NY, USA, 2005, pp. 342–351.

[17] V. Dallmeier and T. Zimmermann, “Extraction of bug localization benchmarks from history,” in Proc. 22nd IEEE/ACM Int. Conf. Autom. Softw. Eng., New York, NY, USA, 2007, pp. 433–436.

[18] T. Dasgupta, M. Grechanik, E. Moritz, B. Dit, and D. Poshyvanyk, “Enhancing software traceability by automatically expanding corpora with relevant documentation,” in Proc. IEEE Int. Conf. Softw. Maintenance, Washington, DC, USA, 2013, pp. 320–329.

[19] H. Daum\_e, III and D. Marcu, “A large-scale exploration of effective global features for a joint entity detection and tracking model,” in Proc. Conf. Human Lang. Technol. Empirical Methods Natural Lang. Process., Stroudsburg, PA, USA, 2005, pp. 97–104.

[20] B. Dit, A. Holtzhauer, D. Poshyvanyk, and H. Kagdi, “A dataset from change history to support evaluation of software maintenance tasks,” in Proc. 10th Working Conf. Mining Softw. Repositories, Piscataway, NJ, USA, 2013 pp. 131–134.

[21] B. Dit, M. Revelle, M. Gethers, and D. Poshyvanyk, “Feature location in source code: A taxonomy and survey,” J. Softw.: Evol. Process, vol. 25, no. 1, pp. 53–95, 2013.

[22] B. Dit, M. Revelle, and D. Poshyvanyk, “Integrating information retrieval, execution and link analysis algorithms to improve feature location in software,” Empirical Softw. Eng., vol. 18, no. 2, pp. 277–309, 2013.

[23] E. Enslen, E. Hill, L. Pollock, and K. Vijay-Shanker, “Mining source code to automatically split identifiers for software analysis,” in Proc. 6th IEEE Int. Working Conf. Mining Softw. Repositories, Washington, DC, USA, 2009, pp. 71–80.

[24] G. Gay, S. Haiduc, A. Marcus, and T. Menzies, “On the use of relevance feedback in IR-based concept location,” in Proc. IEEE Int. Conf. Software Maintenance, 2009, pp. 351–360.

[25] M. Gethers, B. Dit, H. Kagdi, and D. Poshyvanyk, “Integrated impact analysis for managing software changes,” in Proc. 34th Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2012, pp. 430–440.

[26] A. E. Hassan, “Predicting faults using the complexity of code changes,” in Proc. 31st Int. Conf. Softw. Eng., Washington, DC, USA, 2009, pp. 78–88.

[27] W. Jin and A. Orso, “F3: Fault localization for field failures,” in Proc. Int. Symp. Softw. Testing Anal., New York, NY, USA, 2013 pp. 213–223.

[28] T. Joachims, “Optimizing search engines using clickthrough data,” in Proc. 8th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA, 2002 pp. 133–142.

[29] T. Joachims, “Training linear SVMs in linear time,” in Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, New York, NY, USA, 2006 pp. 217–226.

[30] D. Kim, Y. Tao, S. Kim and A. Zeller, “Where should we fix this bug? A two-phase recommendation model,” IEEE Trans. Softw. Eng., vol. 39, no. 11, pp. 1597–1610, Nov. 2013.

[31] S. Kim, E. J. Whitehead, Jr., and Y. Zhang, “Classifying software changes: Clean or Buggy?” IEEE Trans. Softw. Eng., vol. 34, no. 2, pp. 181–196, Mar. 2008.

[32] S. Kim, T. Zimmermann, E. J. Whitehead Jr, and A. Zeller, “Predicting faults from cached history,” in Proc. 29th Int. Conf. Softw. Eng., Washington, DC, USA, 2007, pp. 489–498.

[33] J. M. Kleinberg, “Authoritative sources in a hyperlinked environment,” J. ACM, vol. 46, no. 5, pp. 604–632, Sep. 1999.

[34] R. Kohavi and G. H. John, “Wrappers for feature subset selection,” Artif. Intell., vol. 97, nos. 1-2, pp. 273–324, Dec. 1997.

[35] T. D. LaToza and B. A. Myers, “Hard-to-answer questions about code,” in Proc. Eval. Usability Programm. Lang. Tools, New York, NY, USA, 2010, pp. 8:1–8:6.

[36] T. Lee, J. Nam, D. Han, S. Kim, and H. P. In, “Micro interaction metrics for defect prediction,” in Proc. 19th ACM SIGSOFT Symp. 13th Eur. Conf. Found. Softw. Eng., New York, NY, USA, 2011, pp. 311–321.

[37] C. Liu, X. Yan, L. Fei, J. Han, and S. P. Midkiff, “Sober: Statistical model-based bug localization,” in Proc. 10th Eur. Softw. Eng. Conf. Held Jointly 13th ACM SIGSOFT Int. Symp. Found. Softw. Eng., New York, NY, USA, 2005 pp. 286–295.

[38] D. Liu, A. Marcus, D. Poshyvanyk, and V. Rajlich, “Feature location via information retrieval based filtering of a single scenario execution trace,” in Proc. 22nd IEEE/ACM Int. Conf. Autom. Softw. Eng., New York, NY, USA, 2007 pp. 234–243.

[39] S. K. Lukins, N. A. Kraft, and L. H. Etzkorn, “Bug localization using latent Dirichlet allocation,” Inf. Softw. Technol., vol. 52, no. 9, pp. 972–990, Sep. 2010.

[40] H. B. Mann and D. R. Whitney, “On a test of whether one of two random variables is stochastically larger than the other,” Ann. Math. Statist., vol. 18, no. 1, pp. 50–60, 03 1947.

[41] C. D. Manning, P. Raghavan, and H. Sch€utze, Introduction to Information Retrieval. New York, NY, USA: Cambridge Univ. Press, 2008.

[42] A. Marcus, A. Sergeyev, V. Rajlich, and J. I. Maletic, “An information retrieval approach to concept location in source code,” in Proc. 11th Working Conf. Reverse Eng., Washington, DC, USA, 2004, pp. 214–223.

[43] C. McMillan, M. Grechanik, D. Poshyvanyk, C. Fu, and Q. Xie, “Exemplar: A source code search engine for finding highly relevant applications,” IEEE Trans. Softw. Eng., vol. 38, no. 5, pp. 1069–1087, Sep./Oct. 2012.

[44] C. McMillan, M. Grechanik, D. Poshyvanyk, Q. Xie, and C. Fu, “Portfolio: Finding relevant functions and their usage,” in Proc. 33rd Int. Conf. Softw. Eng., New York, NY, USA, 2011 pp. 111–120.

[45] C. Mcmillan, D. Poshyvanyk, M. Grechanik, Q. Xie, and C. Fu, “Portfolio: Searching for relevant functions and their usages in millions of lines of code,” ACM Trans. Softw. Eng. Methodol., vol. 22, no. 4, pp. 37:1–37:30, Oct. 2013.

[46] P. Melville and V. Sindhwani, “Recommender systems,” in Encyclopedia of Machine Learning, C. Sammut and G. Webb, Eds., New York, NY, USA: Springer, 2010, pp. 829–838.

[47] T. Menzies, J. Greenwald, and A. Frank, “Data mining static code attributes to learn defect predictors,” IEEE Trans. Softw. Eng., vol. 33, no. 1, pp. 2–13, Jan. 2007.

[48] R. Moser, W. Pedrycz, and G. Succi, “A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction,” in Proc. 30th Int. Conf. Softw. Eng., New York, NY, USA, 2008 pp. 181–190.

[49] E. Murphy-Hill, T. Zimmermann, C. Bird, and N. Nagappan, “The design of bug fixes,” in Proc. Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2013, pp. 332–341.

[50] N. Nagappan, T. Ball, and A. Zeller, “Mining metrics to predict component failures,” in Proc. 28th Int. Conf. Softw. Eng., New York, NY, USA, 2006, pp. 452–461.

[51] N. Nagappan, A. Zeller, T. Zimmermann, K. Herzig, and B. Murphy, “Change bursts as defect predictors,” in Proc. IEEE 21st Int. Symp. Softw. Rel. Eng., Washington, DC, USA, 2010, pp. 309–318.

[52] A. T. Nguyen, T. T. Nguyen, J. Al-Kofahi, H. V. Nguyen, and T. N. Nguyen, “A topic-based approach for narrowing the search space of buggy files from a bug report,” in Proc. 26th IEEE/ACM Int. Conf. Autom. Softw. Eng., Washington, DC, USA, 2011 pp. 263–272.

[53] T. J. Ostrand, E. J. Weyuker, and R. M. Bell, “Predicting the location and number of faults in large software systems,” IEEE Trans. Softw. Eng., vol. 31, no. 4, pp. 340–355, Apr. 2005.

[54] L. Page, S. Brin, R. Motwani, and T. Winograd, “The PageRank citation ranking: Bringing order to the web,” Stanford InfoLab, Stanford University, Tech. Rep. 1999-66, Nov. 1999.

[55] D. Poshyvanyk, M. Gethers, and A. Marcus, “Concept location using formal concept analysis and information retrieval,” ACM Trans. Softw. Eng. Methodol., vol. 21, no. 4, pp. 23:1–23:34, Feb. 2013.

[56] D. Poshyvanyk, Y.-G. Gueheneuc, A. Marcus, G. Antoniol, and V. Rajlich, “Feature location using probabilistic ranking of methods based on execution scenarios and information retrieval,” IEEE Trans. Softw. Eng., vol. 33, no. 6, pp. 420–432, Jun. 2007.

[57] D. Poshyvanyk and A. Marcus, “Combining formal concept analysis with information retrieval for concept location in source code,” in Proc. 15th IEEE Int. Conf. Program Comprehension, Washington, DC, USA, 2007, pp. 37–48.

[58] D. Poshyvanyk, A. Marcus, V. Rajlich, Y.-G. Gueheneuc, and G. Antoniol, “Combining probabilistic ranking and latent semantic indexing for feature identification,” in Proc. 14th IEEE Int. Conf. Program Comprehension,Washington, DC, USA, 2006 pp. 137–148.

[59] F. Rahman and P. Devanbu, “How, and why, process metrics are better,” in Proc. Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2013, pp. 432–441.

[60] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub, “Multiclass cancer diagnosis using tumor gene expression signatures,” Proc. Nat. Acad. Sci., vol. 98, no. 26, pp. 15149–15154, 2001.

[61] S. Rao and A. Kak, “Retrieval from software libraries for bug localization: A comparative study of generic and composite text models,” in Proc. 8th Working Conf. Mining Softw. Repositories, New York, NY, USA, 2011, pp. 43–52.

[62] R. Saha, M. Lease, S. Khurshid, and D. Perry, “Improving bug localization using structured information retrieval,” in Proc. IEEE/ ACM 28th Int. Conf. Autom. Softw. Eng., Nov. 2013, pp. 345–355.

[63] D. Shepherd, Z. P. Fry, E. Hill, L. Pollock, and K. Vijay-Shanker, “Using natural language program analysis to locate and understand action-oriented concerns,” in Proc. 6th Int. Conf. Aspect- Oriented Softw. Develop., New York, NY, USA, 2007 pp. 212–224.

[64] S. Shivaji, E. J. Whitehead, Jr., R. Akella, and S. Kim, “Reducing features to improve code change-based bug prediction,” IEEE Trans. Softw. Eng., vol. 39, no. 4, pp. 552–569, Apr. 2013.

[65] M. Stoerzer, B. G. Ryder, X. Ren, and F. Tip, “Finding failure inducing changes in Java programs using change classification,” in Proc. 14th ACM SIGSOFT Int. Symp. Found. Softw. Eng., New York, NY, USA, 2006, pp. 57–68.

[66] E. M. Voorhees, “The TREC-8 question answering track report,” in Proc. TREC-8, 1999, pp. 77–82.

[67] A. W. Whitney, “A direct method of nonparametric measurement selection,” IEEE Trans. Comput., vol. 20, no. 9, pp. 1100–1103, Sep. 1971.

[68] N. Wilde, J. Gomez, T. Gust, and D. Strasburg, “Locating user functionality in old code,” in Proc. Conf. Softw. Maintenance, Nov. 1992, pp. 200–205.

[69] C.-P. Wong, Y. Xiong, H. Zhang, D. Hao, L. Zhang, and H. Mei, “Boosting bug-report-oriented fault localization with segmentation and stack-trace analysis,” in Proc. IEEE Int. Conf. Softw. Maintenance Evol., 2014, pp. 181–190.

[70] X. Ye, R. Bunescu, and C. Liu, “Learning to rank relevant files for bug reports using domain knowledge,” in Proc. 22nd ACM SIGSOFT Int. Symp. Found. Softw. Eng., New York, NY, USA, 2014, pp. 689–699.

[71] J. Zhou, H. Zhang, and D. Lo, “Where should the bugs be fixed? - more accurate information retrieval-based bug localization based on bug reports,” in Proc. Int. Conf. Softw. Eng., Piscataway, NJ, USA, 2012 pp. 14–24.

[72] T. Zimmermann and N. Nagappan, “Predicting defects using network analysis on dependency graphs,” in Proc. 30th Int. Conf. Softw. Eng., New York, NY, USA, 2008, pp. 531–540.