Introduction

Software maintenance is a major part of the software development lifecycle [4]. It can often account for a large proportion of a projects total expenses, so most, if not all, businesses have a great interest in ensuring that no time is wasted in the maintenance process. One maintenance area that commonly lacks efficiency is Bug Reporting, where a software bug is defined as an error, flaw or [fault](https://en.wikipedia.org/wiki/Fault_(technology)) in a [computer program](https://en.wikipedia.org/wiki/Computer_program) or [system](https://en.wikipedia.org/wiki/Software_system) that causes it to produce an incorrect or unexpected result, or to behave in unintended ways [5]. Developers usually employ a bug tracking system (e.g., Jira, Bugzilla) to file bugs found by testers and issues encountered by users[3], however it is often the case that many of these reports describe the same issues while using different phrasing. Some studies have found that upwards of 20% of bug reports are duplicates of this kind[2]. Bug reports can often be quite cryptic and verbose, thus causing the tester to use a fair amount of time digesting the information. A large amount of duplicate bug reports cause developers and testers to spend much time looking over redundant data, and thus time and money are wasted in the process. There is indeed a need for tools that can quickly and accurately identify duplicate bugs so that software maintenance resources are not wasted.

Over the years, many solutions to automatically detect duplicate bugs have been proposed. Some of the methods employed include the use of methods such as Latent Drinchlett Allocation [6, 7], Learning to Rank (L2R) [8], and Support Vector Machines [9], yet in spite of the substantial volume of work, the accuracy of existing duplicate detection systems are comparatively low [2]. Past methods do not appear to sufficiently extract and make use of latent features nor adequately capture the semantic relationships between text documents [2], therefore this has left much room for improvement.

Sources

[1] Jianjun He, Ling Xu, Meng Yan, Xin Xia, Yan Lei. 2020. Duplicate Bug Report Detection Using Dual-Channel Convolutional Neural Networks. ICPC ’20, October 5–6, 2020.

[2] Ashima Kukkar, Rajni Mohana, Yugal Kumar, Anand Nayyar, Muhammad Bilal, Kyung-Sup Kwak. 2020. Duplicate Bug Report Detection and Classification System Based on Deep Learning Technique. IEEE Access V.8, 2020.

[3] Qi Xie∗ , Zhiyuan Wen§ , Jieming Zhu‡ , Cuiyun Gao¶ , Zibin Zheng. 2018. Detecting Duplicate Bug Reports with Convolutional Neural Networks. 25th Asia-Pacific Software Engineering Conference (APSEC).

[4] Jayati Deshmukh, Annervaz K M, Sanjay Podder, Shubhashis Sengupta, Neville Dubash. 2017. Towards Accurate Duplicate Bug Retrieval using Deep Learning Techniques. IEEE International Conference on Software Maintenance and Evolution, 2017.

[5] <https://en.wikipedia.org/wiki/Software_bug>

[6] DavidMBlei, AndrewYNg, MichaelIJordan. 2003. Latentdirichletallocation. Journal of machine Learning research 3, Jan (2003), 993–1022.

[7] YingFu, MengYan, XiaohongZhang, LingXu, DanYang, and JeffreyDKymer. 2015. Automated classification of software change messages by semi-supervised Latent Dirichlet Allocation. Information and Software Technology 57 (2015), 369–377.

[8] Tie-Yan Liu et al. 2009. Learning to rank for information retrieval. Foundations and Trends® in Information Retrieval 3, 3 (2009), 225–331.

[9] WenZhang,TaketoshiYoshida,andXijinTang.2008.Textclassificationbasedon multi-word with support vector machine. Knowledge-Based Systems 21, 8 (2008), 879–886.