


Automatically Labelled Software Topic Model

Youcef Bouziane, Université Oran1, Oran, Algeria

 <https://orcid.org/0000-0001-7072-2576>

Mustapha Kamel Abdi, Université Oran 1, Oran, Algeria

Salah Sadou, IRISA, Université Bretagne Sud, Vannes, France

ABSTRACT

Public software repositories (SR) maintain a massive amount of valuable data offering opportunities to support software engineering (SE) tasks. Researchers have applied information retrieval techniques in mining software repositories. Topic models are one of these techniques. However, this technique does not give an interpretation nor labels to the extracted topics and it requires manual analysis to identify them. Some approaches were proposed to automatically label the topics using tags in SR, but they do not consider the existence of spam-tags and they have difficulties to scale to large tag space. This article introduces a novel approach called automatically labelled software topic model (AL-STM) that labels the topics based on observed tags in SR. It mitigates the shortcomings of manual and automatic labelling of topics in SE. AL-STM is implemented using 22K GitHub projects and evaluated in a SE task (tag recommending) against the currently used techniques. The empirical results suggest that AL-STM is more robust in terms of MAP and nDCG, and more scalable to large tag space.

KEYWORDS

Mining Software Repositories, Open Source Software, Software Engineering, Software Tag, Topic Model

INTRODUCTION

Software repositories (SR) offer a real opportunity to understand software aspects, enhance software quality, and promote code reuse. The textual data in public SR are mostly unstructured data (Agrawal, Fu, & Menzies, 2018) (Chen, Thomas, & Hassan, A survey on the use of topic models when mining software repositories, 2016) that can be found in many software artefacts, such as source code, email archives, bugs report, etc. To exploit the latent information in these data, the software engineering (SE) community conducted several studies on mining software repositories (MSR) using the information retrieval (IR) technique. Topic models are one of the widely used IR techniques. They are statistical models that discover latent semantic structures in unstructured textual data and cluster them into topics. Where each topic is a set of co-occurring words, and a document is a mixture of topics. Several approaches based on topic models like Latent Dirichlet Allocation (LDA) (Blei, Ng, & Jordan, 2003) and Labelled LDA (LLDA) (Ramage, Hall, Nallapati, & Manning, 2009) were proposed to support SE tasks such as feature location and extraction (Binkley, Lawrie, Uehlinger, & Heinz, 2015) (Sun, Li, Leung, Li, & Li, 2015), traceability link recovery (Hindle, Bird, Zimmermann, & Nagappan, 2015) (Panichella, et al., 2013), software quality and metrics (Chen, Shang, Nagappan, Hassan, &

DOI: 10.4018/IJOSSP.2020010104

Thomas, 2017) (Hu & Wong, 2013), and software organisation and clustering, (Markovtsev & Kant, 2017) (Sharma, Thung, Kochhar, Sulistya, & Lo, 2017).

Despite their advantages, topic models have some shortcomings such as the dependency of their performance on the selected parameters, uninterpreted topics, and poor performance on short texts. Topic models do not give an interpretation of the generated topics and require an extra step to label them. This step in SE is costly and depends on experts' knowledge if done manually, especially when dealing with a big number of topics. Some approaches use the words with the highest marginal probability $p(w_i|t_j)$ for labelling. However, previous experiences (Kawaguchi, Garg, Matsushita, & Inoue, 2006), (Markovtsev & Kant, 2017), (Sharma, Thung, Kochhar, Sulistya, & Lo, 2017), (Tian, Revelle, & Poshyvanyk, 2009) showed that even with a handful number of topics, they could take several days to be completely analysed. In a recent survey (Chen, Thomas, & Hassan, 2016), the authors reported: "Labelling and interpreting topics can be difficult and subjective and may require much human effort. Future studies should explore ways to apply different approaches to automatically label the topics" (p. 34).

Some shortcomings were addressed in SE, particularly tuning the parameter (Agrawal, Fu, & Menzies, 2018) (Panichella et al., 2013). Others were treated in NLP and applied by SE community, particularly topics interpretation and labelling (Lau, Grieser, Newman, & Baldwin, 2011) (Ramage, Hall, Nallapati, & Manning, 2009) and adaptation for short texts (Yan, Guo, Lan, & Cheng, 2013) (Jin, Liu, Zhao, Yu, & Yang, 2011). In SE, the automatic labelling approaches use tags extracted from tagged SR to label the topics. These tags are the outcome of the tagging mechanism adopted by many Open-Source SR to organise software and facilitate project search. Some repositories such as SourceForge¹, allow only a restricted list of tags. Others such as GitHub², Freecode³ and Openhub⁴ allow the project owner and users to add any tag. This absence of controlled vocabulary in some repositories drove the number of tags to increase significantly, which led to the appearance of noisy tags among them. The authors studied 120K GitHub projects and found that there a high risk of spam-tags presence in SR.

LLDA (Ramage, Hall, Nallapati, & Manning, 2009) is a labelled topic model that was widely reused by NLP and SE community. It removes the need for defining one of the learning parameters (number of topics) and labelling the topics. In LLDA, the training data can be labelled with external labels. LLDA computes the probability distribution of topics based on the distribution of the labels, where each label is considered as a topic.

Nevertheless, LLDA has two shortcomings. The first is that it assumes that all the training data are correctly tagged with the document real domains (topics), and do not assume the presence of spam-tags. The second, according to the same authors (Ramage, Manning, & Dumais, 2011): "Labeled LDA does not assume the existence of any latent topics (neither global nor within a label)". They added: "Labeled Latent Dirichlet Allocation" is not so latent: every output dimension is in one-to-one correspondence with the input label space" (p. 458). This drawback was partially resolved by proposing PLDA and PLDP that assumes the existence of latent topic within labels and allow a one-to-many relationship between the labels and the topics, but this relation does not assume the existence of latent global topics that include labels. Furthermore, these models force the number of latent topics within labels to a fixed number for all the labels regardless of the observed data, which does not necessarily reflect the real topic distribution in the data and may result in redundant or irrelevant topics.

The purpose of the approach presented in this work is to address the challenge of automatically labelling the topics and overcome the drawback of tagging mechanisms in SR. Automatically Labelled Software Topic Model (AL-STM), is an unsupervised labelling mechanism for software topic models. For each topic, AL-STM generates a set of ranked label candidates based on tagged software. To get the most representative topic model of the software and their domains in the training corpus, AL-STM includes a parameter optimisation step. This helps in distinguishing between the correct tags and the spam-tags. It also uses a mechanism to reduce the impact of spam tags based on the concept of centroids. Besides, AL-STM defines a many-to-many correspondence between the topics

20 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/automatically-labelled-software-topic-model/251195

Related Content

Software Fault Prediction Using Deep Learning Algorithms

Osama Al Qasem and Mohammed Akour (2019). *International Journal of Open Source Software and Processes* (pp. 1-19).

www.irma-international.org/article/software-fault-prediction-using-deep-learning-algorithms/242945

Tools and Datasets for Mining Libre Software Repositories

Gregorio Robles, Jesús M. González-Barahona, Daniel Izquierdo-Cortazar and Israel Herraiz (2011). *Multi-Disciplinary Advancement in Open Source Software and Processes* (pp. 24-42).

www.irma-international.org/chapter/tools-datasets-mining-libre-software/52243

A Perspective on Software Engineering Education with Open Source Software

Pankaj Kamthan (2012). *International Journal of Open Source Software and Processes* (pp. 13-25).

www.irma-international.org/article/a-perspective-on-software-engineering-education-with-open-source-software/101203

A Comparative Analysis of Open Source Network Monitoring Tools

Ali Al Shidhani, Khalil Al Maawali, Dawood Al Abri and Hadj Bourdouden (2016). *International Journal of Open Source Software and Processes* (pp. 1-19).

www.irma-international.org/article/a-comparative-analysis-of-open-source-network-monitoring-tools/181324

Need of the Research Community: Open Source Solution for Research Knowledge Management

Dhananjay S. Deshpande, Pradeep R. Kulkarni and Pravin S. Metkewar (2017). *Open Source Solutions for Knowledge Management and Technological Ecosystems* (pp. 146-174).

www.irma-international.org/chapter/need-of-the-research-community/168982