Helpful Automatic Development Email Summarization

Alberto Bacchelli *and* Michele Lanza

REVEAL @ Faculty of Informatics - University of Lugano, Switzerland

Abstract

Software development projects, especially if large-scale, require a remarkable coordination effort amongst the people involved. Coordination challenges are tackled by communication. In co-located development teams, unplanned informal face-to-face meetings are the favorite form of communication when developers need to coordinate or face program comprehension problems [1]. Personal meetings, besides disrupting developers' attention and retaining knowledge by a few developers [2], are inapplicable to distributed development projects. Developers, thus, replace face-to-face meetings with electronic communication means. Instant messaging, wikis, forums are viable options, but *the decisive role is played by emails*, indeed: "Mailing lists are the bread and butter of project communications" [3]. Emails are asynchronous, thus evade time zone barriers and do not disrupt the attention of developers; mailing lists broadcast discussions, announcements, and decisions to all participants, thus maintaining the awareness of developers; emails are not bound to specific abstraction levels (as opposed to commit messages, design documents, or code comments), thus they can be used to discuss issues ranging from low-level decisions (*e.g.,* implementation, bug fixing) up to high-level considerations (*e.g.,* design rationales); mailing lists archive messages, thus offering a historical perspective.

Considering the many reasons why programmers start email conversations with a single colleague or the entire rest of the team, and that software projects may be carried out for several months or years, we can easily to picture the vast amount of resulting emails. For example, on the Apache developer mailing list, there were 4,996 messages in the year 2004 and 2,340 in 2005; for `gcc` these numbers were 19,173 and 15,082 [4]. To maintain team coordination and perform work on a software system, software developers must keep track, read, and understand voluminous amounts of electronic communication. This is a challenge, not only because of the large number of emails popping up day after day in a developer's mailbox, thus requiring time and effort to be read, but also because retrieving information from such emails can be time consuming and frustrating. Sometimes, the amount of information may be overwhelming, causing messages to be left unread and searches to be abandoned, which in turn lead to duplicate, uncoordinated, or non-optimized work to be performed [5].

One way to alleviate this information overload issue is to provide a summary of development emails. A correct and helpful summary (*i.e.,* the text reduced to a version without the parts unnecessary for the comprehension) enables developers to reduce the time spent reading new emails or perusing messages that have been returned from searches, found through browsing, or recommended by team members.

Given the growing amount of produced and recorded textual information, often referenced as *big data*, there is substantial interest and a large body of work in the automated generation of summaries for natural language documents, and software development related artifacts (*e.g.,* [6,5]). On the basis of the success of existing efforts on automatic summarization, we started investigating an approach to automatically generate helpful summaries of development emails [7,8]. In this presentation, we analyze and discuss the challenges that go hand in hand with such an endeavor, and present our ongoing results. We first clarify the challenges in devising a successful research methodology and in creating an effective summarization technique to be implemented in a tool that can be used in real world scenarios. Subsequently, we explain how we conducted our research so far, in particular:

1. We conducted a pilot study with two students with the same background to decide whether we should aim for summaries based on keywords or based on sentences;
2. we created a golden set for extractive summaries, by asking 18 participants (10 undergraduate and 6 graduate students in informatics, and 2 industrial programmers) to generate sentence-based extractive summaries for 12 email threads, taken from two open source software systems;
3. we studied the obtained results and devised features to characterize email sentences. We measured the importance of each feature by means of two machine learning methods (*i.e.,* classification trees and Naïve Bayes), thus defining an algorithm to classify the importance of each sentence in an email thread.

Finally, we present the current status of HADES (Helpful Automatic Summarizer of Development Emails), an interactive plugin for the Eclipse IDE, which implements our findings into a tool that can be integrated in programmers' workflow.

References

1. LaToza, T.D., Venolia, G., DeLine, R.: Maintaining mental models: a study of developer work habits. In: Proceedings of ICSE 2006 (28th ACM International Conference on Software Engineering), ACM (2006) 492–501
2. Ko, A.J., DeLine, R., Venolia, G.: Information needs in collocated software development teams. In: Proceedings of ICSE 2007 (29th ACM/IEEE International Conference on Software Engineering), IEEE Computer Society (2007) 344–353
3. Fogel, K.: Producing Open Source Software. First edn. O'Reilly Media (2005)
4. Bird, C., Gourley, A., Devanbu, P., Gertz, M., Swaminathan, A.: Mining email social networks. In: Proceedings of MSR 2006 (3th International Workshop on Mining Software Repositories), ACM (2006) 137–143
5. Rastkar, S., Murphy, G.C., Murray, G.: Summarizing software artifacts: a case study of bug reports. In: In Proceedings of ICSE 2010. (2012) 505–514
6. Haiduc, S., Aponte, J., Marcus, A.: Supporting program comprehension with source code summarization. In: Proceedings of ICSE 2010, ACM (2010) 223–226
7. Mastrodicasa, E.: Extractive summarization of development emails. Bachelor's thesis, University of Lugano (June 2012)
8. Bacchelli, A., Lanza, M., Mastrodicasa, E.: On the road to hades–helpful automatic development email summarization. In: In Proceedings of TAinSM 2012 (1st International Workshop on on the Next Five Years of Text Analysis in Software Maintenance). (2012) to be published