The Use of Text Retrieval and Natural Language Processing in Software Engineering

Sonia Haiduc
Florida State University
Tallahassee, FL, USA
shaiduc@cs.fsu.edu

Venera Arnaoudova
Washington State University
Pullman, WA, USA
varnaoud@eecs.wsu.edu

Andrian Marcus
Univ. of Texas at Dallas
Richardson, TX, USA
amarcus@utdallas.edu

Giuliano Antoniol
Polytechnique Montréal
Montreal, Canada
antoniol@ieee.org

ABSTRACT
This technical briefing presents the state of the art Text Retrieval and Natural Language Processing techniques used in Software Engineering and discusses their applications in the field.

CCS Concepts
• Information systems~Information retrieval • Computing methodologies~Natural language processing • Software and its engineering~Software creation and management.

Keywords
Text retrieval; natural language processing.

1. OVERVIEW
During software evolution many related artifacts are created or modified. Some of these are composed of structured data (e.g., analysis data), some contain semi-structured information (e.g., source code), and many include unstructured information (e.g., natural language text). In many software projects the amount of unstructured information exceeds that of the structured information by one order of magnitude. Software artifacts written in natural language (e.g., requirements, design documents, user manuals, use case scenarios, bug reports, developers' communication, etc.), together with the source code comments and identifiers encode important information related to the problem and application domains, developers' knowledge and decisions, software design, requirements, and the overall software advancement. Therefore, retrieving and analyzing the textual information present in the software are extremely important for supporting program comprehension and a variety of software evolution tasks. Lastly, many researchers are focusing these days on mining and analyzing textual information from internet-based sources, such as, Stack Overflow, app markets, etc. and use this information to gain new insights, build recommendation systems or simply mine knowledge. This gathered information is then used to support processes and development activities.

Text retrieval (TR) is a branch of information retrieval (IR) that leverages information stored primarily in the form of text. TR methods have been proved as suitable candidates for the retrieval and the analysis of textual data embedded in software or present in other sources. This technical briefing presents some of the most popular TR methods and focuses on their applications in software engineering (SE). In most SE applications, TR techniques are used in conjunction with natural language processing (NLP) tools. The main NLP techniques used by software engineering researchers and applied to software engineering tasks will also be presented.

The technical briefing aims at providing sufficient information about software engineering applications and also tool details to allow/enable/empower software engineering researchers and practitioners to start using TR and NLP methods in their day-to-day work.

ICSE'16 Companion, May 14–22, 2016, Austin, TX, USA
ACM 978-1-4503-4205-6/16/05.
http://dx.doi.org/10.1145/2889160.2891053

2. IMPORTANCE TO THE SE COMMUNITY
We have informally surveyed the literature over the past decades and found that more than 20 different SE tasks are being addressed through TR and NLP approaches applied to software artifacts. Example tasks include traceability link recovery, concern/concept/feature/bug location, software search, change impact analysis, requirements analysis, bug triage, refactoring, defect prediction, software redocumentation, etc. Our survey also shows a steep increase in the use of TR and NLP in SE over the last decade. We argue that the use of TR and NLP in software is one of the fastest growing areas of research in SE. Furthermore, we argue that no other technology (except probably the text editor) is currently used to support so many different SE tasks. Despite the large number of applications of TR and NLP in SE and the large number of recent research papers, most SE students, practitioners, and researchers lack any training in TR or NLP. Exposing the SE community to these techniques and their applications in SE would help to fill a gap in their current background and allow them to immediately use TR and NLP to advance their research.

Recently, previous versions of this briefing have been presented at ICSE 2015, FSE 2015, and SPLASH 2015, and have been met with great interest by the audience. We believe that the interest in TR and NLP is still growing in our field, and presenting this technical briefing at ICSE 2016 will allow more SE researchers, practitioners and students to be exposed to these topics.

3. CONTENT
The technical briefing will first give an introduction to the fields of TR and NLP, followed by an overview of the main TR and NLP techniques used in SE. The general principles behind these techniques will be presented, focusing on the most popular approaches. In particular, for TR, approaches such as Vector Space Model, Latent Semantic Analysis, and Latent Dirichlet Association will be covered. NLP techniques covered will include Language Models, part-of-speech tagging, semantics analysis, sentiment analysis, etc. Preprocessing approaches used on software data before TR and NLP approaches are applied will also be presented, including: stemming, stopword elimination, identifier splitting and expansion, etc. Adaptations of TR and

NLP techniques to better suit the nature of SE artifacts will be discussed. Additional material will be provided online, including descriptions of available tools, which implement many of these techniques, and a comparison of them based on their advantages and limitations.

The second part of the briefing will give an overview of the application of these TR and NLP techniques to specific SE tasks (such as, concept location, traceability link recovery, redocumentation recommendation, bug triage, software quality assessment, clone detection, etc.) and sources of information to which these techniques are applied (e.g., source code, requirements, design documents, bug descriptions, emails, StackOverflow discussions, software repositories, etc.). Basic evaluation measures for TR and NLP techniques in these particular SE applications will also be discussed.

The technical briefing will conclude with a discussion on the challenges, advantages, and limitations of the use of TR and NLP in software engineering and lay out future directions for TR and NLP in software engineering from the practice and research points of view.

The materials covered in the technical briefing, as well as additional materials for applying the TR and NLP techniques discussed in practice will be made available online.

4. TARGET AUDIENCE

The tutorial is suitable for both industry participants and academic researchers (students or faculty). No prior knowledge of TR or NLP is necessary. While the presenters will cover some technical aspects of various TR and NLP methods, the overall technical level of the presentations is rather low, and the emphasis will be on the state of the art application of these techniques in SE. Basic understanding of algebra and probabilities (at undergraduate level) will help to better understand the technical part of the tutorial. The audience will be pointed to further readings, suitable for those who want to know more in depth details about various TR and NLP methods.

5. PRESENTERS

The material to be included in the technical briefing is authored by Sonia Haiduc, Venera Arnaoudova, Andrian Marcus, and Giuliano Antoniol.

5.1 Sonia Haiduc

Sonia Haiduc is an Assistant Professor at Florida State University, in Tallahassee, FL, USA. Her research interests are in software maintenance, software evolution, and program comprehension. The topic of her Ph.D. dissertation (2013) focused on the use of NLP and machine learning techniques to improve applications of TR in software engineering, especially with query reformulations. Her papers have been published in several highly selective software engineering venues. She is one of the organizers of the past three editions of the Workshop on Mining Unstructured Data in Software Engineering (MUD) and currently a member of the organizing committee for SANER 2016 and ICSME 2016. She has also been involved in the organizing committee of several previous conferences in the field. Haiduc has also served as a program committee member for several conferences, including MSR, ICSME, ICPC, SANER, WCRE, CSMR, etc. She has also reviewed for various SE top journals, including TSE, TOSEM, EMSE, JSME, etc. More information is available at: http://www.cs.fsu.edu/~shaiduc/.

5.2 Venera Arnaoudova

Venera Arnaoudova is an Assistant Professor at Washington State University. She received her Ph.D. degree in 2014 from Polytechnique Montréal under the supervision of Dr. Giuliano Antoniol and Dr. Yann-Gaël Guéhéneuc. Her research interest is in the domain of software evolution and particularly, the analysis of source code lexicon and documentation. Her dissertation focused on the improvement of the code lexicon and its consistency using NLP, fault prediction models, and empirical studies. Arnaoudova has published in several international SE conferences and journals. She is currently serving as program committee member for ICPC 2016, ICSME 2016, SCAM 2016, SANER 2016 Tool Track. She has also served as an external reviewer for various previous conferences, including ICSE 2014, MSR 2014, CSMR 2013, and others. More information available at: http://www.veneraarnaoudova.ca.

5.3 Andrian Marcus

Andrian Marcus is an Associate Professor at The University of Texas at Dallas, in Richardson, TX, USA. His current research interests are focused on software evolution and program comprehension. He is best known for his more than decade-long work on using information retrieval and text mining techniques for software analysis to support comprehension tasks during software evolution, such as: concept location, impact analysis, error prediction, traceability link recovery, etc. Marcus received several Best Paper Awards and he is the recipient of the NSF CAREER award in 2009. Marcus gave more than 20 invited seminars of tutorials on the use of text retrieval techniques to support SE tasks at various universities, companies, and summer schools. He was the Chair of the steering committee of ICSME and served on many conferences as chair and program committee member and also serves on the editorial board of three SE journals (TSE, EMSE, JSEP). Together with Antoniol, they presented tutorials or technical briefings on related topics at: ASE 2010, ASE 2011, ESEC/FSE 2011, ICSE 2012. More information available at: http://www.utdallas.edu/~amarcus/.

5.4 Giuliano Antoniol

Giuliano Antoniol is a Professor at Polytechnique Montréal, where he works in the areas of software evolution, software traceability, search-based software engineering, and software maintenance. He worked for several software companies, research institutions and universities. In 2005 he was awarded the Canada Research Chair Tier I in Software Change and Evolution. He published more than 190 papers in journals and international conferences, several works on applying information retrieval approaches to software engineering. Some of his papers received Best Paper Awards. He served as program chair, industrial chair, tutorial, and general chair of many international conferences and workshops, on the editorial boards of five journals, and on the program committees of more than 30 IEEE and ACM conferences and workshops. More information available at: http://www.antoniol.net/.