AUTOMATIC KEYWORD AND SENTENCE-BASED TEXT SUMMARIZATION FOR SOFTWARE BUG REPORTS

[1]KASARLA SPANDANA, [2]MURALI PONAGANTI
[1]MCA Student, [2]Associate Professor
DEPARTMENT OF MCA
SREE CHAITANYA COLLEGE OF ENGINEERING, KARIMNAGAR

ABSTRACT:

Text Summarization is a process which efficiently retrieves the relevant information from documents. The objective of the proposed, unsupervised approach is to summarize bug reports (software artefacts) with complete content and diversified information. The proposed approach utilizes Rapid Automatic Keyword Extraction and term frequency-inverse document frequency method to extract meaningful keywords and key-phrases with a relevant score. For sentence extraction, fuzzy C-means clustering is used to extracts sentences having high degree of membership from each cluster above a set threshold value. A rule engine is used for sentence selection. The rules are generated with the domain knowledge and based on the extracted information by the keywords and sentences selected by the clustering method. Cohesive and coherent summary is generated by the proposed method on apache bug reports. For redundancy removal and to re-rank generated summary, hierarchical clustering is presented to enrich the extracted summary. The proposed approach is evaluated on newly constructed Apache project Bug Report Corpus (APBRC) and existing Bug Report Corpus (BRC). The results are compared on the basis of performance metrics such as precision, recall, pyramid precision and F-score. The experimental results depict that our proposed approach attains significant improvement over other baseline approaches such as BRC and LRCA. It also attains significant improvement over existing state-of-art unsupervised approaches such as Hurried, centroid and others. It extracts significant keyword phrases and sentences from each cluster to achieve full coverage and coherent summary. The results evaluated on APBRC corpus attains an average value of 78.22%, 82.18%, 80.10% and 81.66% for precision, recall, f-score and pyramid precision respectively.

I. INTRODUCTION

In recent years, plenty of information is available on the internet from several domains. With huge amount of available data, it is an arduous and time-consuming task to read entire text documents and retrieve relevant information. To automatically attain relevant information in brief, text summarization is used. Generating accurate summary of a text document is a complex task and requires human intelligence to extract meaningful information from the text. Automatic text summarization has been used in several domains such as document summary [1], [2], essay or news summary [3], [4] and e-mail summarization [5], [6]. Apart from these, several software repositories such as Jira, Bugzilla manages numerous bug reports [7]–[9] with several open source projects. To oversee diverse number of bug reports, several automation tasks have been conducted such as detection of duplicate bug reports [10], [11], fixation of bugs [12], bug report triaging [13]–[15] and others. To accomplish these tasks, software testers and developers need to wade through entire bug reports having hundreds of sentences. A tester or developer needs to

understand history of bug reports with specific domain knowledge as it is not a generic text summarization. In this research work, authors have focused on generation of a summary of bug reports which are the most valued artefacts of software project. It encloses several attributes such as title, one-line description, BugID, detailed description, comments made by several contributors and others. This information is useful for locating and fixing the bug in software project. To read and understand an entire bug report is a tedious task and therefore, bug summarization is an emerging field of research to help several developers to improve bug resolution process.

In literature, two categories of algorithms exist: abstractive summarization and extractive summarization. In abstractive summarization, semantic-representation, word-order and natural language of a text is modified with same contextual meaning. In this area, a major breakthrough is achieved through deep learning techniques. Several researchers have worked on Convolution neural network (CNN) [16], Recurrent neural network (RNN) [17], Reinforcement learning and Generative Adversarial networks (GAN) [18] with high accuracy as compared to other approaches used in literature. The major shortcoming of applying deep learning methods is the un-availability of training data as it is a supervised approach and standard golden summaries are not available in every domain. Whereas, in extractive summarization, sentences are extracted from the text document with same order and language to produce a condensed summary. In literature, several supervised [19]–[21] and unsupervised [22], [23] approaches have been proposed to summarize bug reports automatically. In supervised approach, one such research is carried out by Rastkar et al., [19] in which a corpus of 36 bug reports from various open source projects was created named Bug Report Corpus (BRC). For each bug report, 24 features were calculated which comprises of four categories: Lexical, participant, length and structure. Logistic regression classifier was used which was trained on corpus of manually generated golden summaries by annotators. The results illustrate a low precision of 57%, recall of 35%, 40% f-score and 66% pyramid precision. To improve upon the results of [19], Jiang et al. proposed another approach PRST [20]. The authors consider 36 bug reports of BRC corpus [19] and its duplicate bug reports and constructed a new corpus named Modified Bug Report Corpus (MBRC). Page-rank algorithm is used to compute the textual similarity among sentences and then probability of each sentence was computed using logistic regression classifier. The results were merged and then sentences with high probability value were selected to form a summary. There was a slight improvement in precision and pyramid precision with same values of recall and f-score.

In contrast, an unsupervised approach, assigns a value based on some measure to each sentence and top ranked sentences are selected to form a summary. Lotufo et al. [23] proposed an unsupervised approach to generate summary by investigating how a long report is scanned by developers. In another work, Mani et al. [22] employed four approaches viz., Maximal Marginal Relevance (MMR), centroid, diverse rank and grasshopper. In previous researches [19], [20], keywords, lexical similarity and sentence weight features were used as feature set.

In this work, we focus on unsupervised approach and new method is constructed based on keyword-based features and sentence-based features to facilitate bug report summarization. For keyword-based feature extraction, two methods term frequency-inverse document frequency(tf-idf) and Rapid Automatic Keyword Extraction (RAKE) are used. In previous work,

approaches for keyword extraction focus on corpus-oriented statistics [24], [25]. To avoid this drawback, documented oriented methods were used [26] which utilizes natural language processing to identify parts of speech in combination with other statistical, machine learning or supervised approaches. To avoid these drawbacks, RAKE is used which is language independent, domain independent and unsupervised in nature. It gives reasonable precision along with simplicity and computational efficiency [27]. For sentence-based feature extraction, Fuzzy C-means clustering has been used as compared to other features such as length, position, title word, thematic word and others as used in past work [19], [28], [29].

II. SYSTEM ANALYSIS

EXISTING SYSTEM:

The bug reports were extracted from the duration of by a tool named bug report collection system (BRCS). A pool of bug reports was extracted which has immense data such as title, description, comments and other attributes. From this, 21 bug reports were fetched to form APBRC. Bug reports of varied lengths were fetched which consists of minimum 10 number of comments. For construction of corpus, links of patches were removed. The statistical information related to bug reports is presented. The proposed approach is also evaluated on existing bug report corpus (BRC) which consists of bug reports form four open source projects: mozilla, eclipse, gnome and Kde. All the bug reports and source code is available at link.

DISADVANTAGES:

- To address the issue of summarization of multiple documents, Rautray and Chandra proposed a new approach based on Cat Swarm Optimization (CSO).
- This is due to the fact that fuzzy c-means algorithm reduces the issues of ambiguity, vagueness and incompleteness.
- Based on word and sentence features, fuzzy rules were created to generate summary of documents. To remove redundancy among sentences of generated summary, cosine similarity measures is used.

PROPOSED SYSTEM:

In literature, several supervised and unsupervised approaches have been proposed to summarize bug reports automatically. In supervised approach, one such research is carried out in which a corpus of 36 bug reports from various open source projects was created named Bug Report Corpus (BRC). For each bug report, features were calculated which comprises of four categories: Lexical, participant, length and structure. Logistic regression classifier was used which was trained on corpus of manually generated golden summaries by annotators. In contrast, an unsupervised approach, assigns a value based on some measure to each sentence and top ranked sentences are selected to form a summary. Proposed an unsupervised approach to generate summary by investigating how a long report is scanned by developers. In another work, employed four approaches viz., Maximal Marginal Relevance (MMR), centroid, diverse rank and grasshopper. In previous researches, keywords, lexical similarity and sentence weight features were used as feature set.

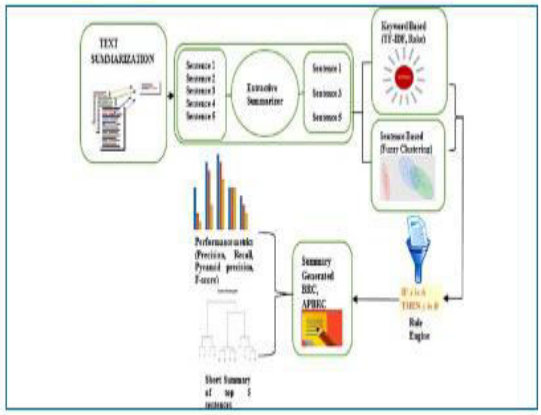
ADVANTAGES:

- In previous researches , keywords, lexical similarity and sentence weight features were used as feature set.
- In this work, we focus on unsupervised approach and new method is constructed

based on keyword-based features and sentence-based features to facilitate bug report summarization.

- It gives reasonable precision along with simplicity and computational efficiency. For sentence-based feature extraction, Fuzzy C-means clustering has been used as compared to other features such as length, position, title word, thematic word and others as used in past work.

ARCHITECTURE



III. IMPLEMENTATION MODULES DESCRIPTION:

Text summarization:

In recent years, plenty of information is available on the internet from several domains. With huge amount of available data, it is an arduous and time-consuming task to read entire text documents and retrieve relevant information. To automatically attain relevant information in brief, text summarization is used. Generating accurate summary of a text document is a complex task and requires human intelligence to extract meaningful information from the text. Automatic text summarization has been used in several domains such as document summary, essay or news summary and e-mail summarization. Apart from these, several software repositories such as manages numerous bug reports with several open source projects. To oversee diverse number of bug reports, several automation tasks have been conducted such as detection of duplicate bug reports.

Convolutional neural network:

In this area, a major breakthrough is achieved through deep learning techniques. Several researchers have worked on Convolution neural network (CNN), Recurrent neural network (RNN), Reinforcement learning and Generative Adversarial networks (GAN) with high accuracy as compared to other approaches used in literature. The major shortcoming of applying deep learning methods is the un-availability of training data as it is a supervised approach and standard golden summaries are not available in every domain. Whereas, in extractive summarization, sentences are extracted from the text document with same order and language to produce a condensed summary. In literature, several supervised and unsupervised approaches have been proposed to summarize bug reports automatically.

Bug report corpus:

In literature, several supervised and unsupervised approaches have been proposed to summarize bug reports automatically. In supervised approach, one such research is carried out by in which a corpus of 36 bug reports from various open source projects was created named Bug Report Corpus (BRC). For each bug report, 24 features were calculated which comprises of four categories: Lexical, participant, length and structure. Logistic regression classifier was used which was trained on corpus of manually generated golden summaries by annotators. The results illustrate a

low precision of 57%, recall of 35%, 40% f-score and 66% pyramid precision.

Rapid automatic keyword extraction:

In this work, we focus on unsupervised approach and new method is constructed based on keyword-based features and sentence-based features to facilitate bug report summarization. For keyword-based feature extraction, two methods term frequency-inverse document frequency(tf-idf) and Rapid Automatic Keyword Extraction (RAKE) are used. In previous work, approaches for keyword extraction focus on corpus-oriented statistics. To avoid this drawback, documented oriented methods were used which utilizes natural language processing to identify parts of speech in combination with other statistical, machine learning or supervised approaches. To avoid these drawbacks, RAKE is used which is language independent, domain independent and unsupervised in nature.

IV. CONCLUSION

This paper proposes an unsupervised approach to automatically summarize software bug reports based on keywords and sentence-based features. To eliminate the drawbacks of corpus-oriented and document-oriented approaches as used in literature, two feature extraction methods are used: Term frequency-Inverse document frequency and Rapid automatic keyword extraction are used. RAKE is language and domain independent and is unsupervised in nature which does not require any domain knowledge. For sentence extraction, bug reports are divided into clusters. To compute optimum number of clusters, four methods: K-means, GSS, Silhouette and WSS are used. For optimum number of clusters obtained, fuzzy c-means clustering is utilized to deal with uncertain information in bug reports and sentences in each cluster with high degree of membership value is selected.

FUTURE WORK:

In future, different clustering algorithms will be employed and their impact will be analyzed. Also, the performance of the proposed approach is evaluated through various performance metrics. However, in literature, ROUGE (Recall- Oriented Understudy for Gisting Evaluation) is used for text documents. It has not been used in evaluating the performance of bug reports. In future, it will be considered to employ on bug reports.

REFERENCES

1. User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
2. Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
3. Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.
4. Data Communications and Networking, by Behrouz A Forouzan.
5. Computer Networking: A Top-Down Approach, by James F. Kurose.
6. Operating System Concepts, by Abraham Silberschatz.
7. M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009.
8. “The apache cassandra project,” http://cassandra.apache.org/.
9. L. Lamport, “The part-time parliament,” ACM Transactions on Computer Systems, vol. 16, pp. 133–169, 1998.

10. N. Bonvin, T. G. Papaioannou, and K. Aberer, "Cost-efficient and differentiated data availability guarantees in data clouds," in Proc. of the ICDE, Long Beach, CA, USA, 2010.
11. O. Regev and N. Nisan, "The popcorn market. online markets for computational resources," Decision Support Systems, vol. 28, no. 1-2, pp. 177 – 189, 2000.
12. A. Helsinger and T. Wright, "Cougaar: A robust configurable multi agent platform," in Proc. of the IEEE Aerospace Conference, 2005.
13. J. Brunelle, P. Hurst, J. Huth, L. Kang, C. Ng, D. C. Parkes, M. Seltzer, J. Shank, and S. Youssef, "Egg: an extensible and economics-inspired open grid computing platform," in Proc. of the GECON, Singapore, May 2006.
14. J. Norris, K. Coleman, A. Fox, and G. Candea, "Oncall: Defeating spikes with a free-market application cluster," in Proc. of the International Conference on Autonomic Computing, New York, NY, USA, May 2004.
15. C. Pautasso, T. Heinis, and G. Alonso, "Autonomic resource provisioning for software business processes," Information and Software Technology, vol. 49, pp. 65–80, 2007.