e-ISSN : 0976-5166
p-ISSN : 2231-3850

COMPARATIVE STUDY OF BUG REPORT SUMMARIZATION TECHNIQUES

Som Gupta
Research Scholar, AKTU Lucknow, Vistar Yojna, Sector 11, Naya Khera,
Jankipuram, Lucknow, UP 226031, India
somi.11ce@gmail.com

S.K Gupta
Associate Professor, BIET Jhansi, Kanpur Road NH-25,
Jhansi, UP 284128, India
guptask_biet@rediffmail.com

Abstract - Bug Reports are one of the very important artifacts during software development process and is one of the very popular artifacts of research among the researchers. Summarization is one application on bug reports which helps solve a lot of interesting issues of bug reports like bug triaging and bug duplicate detection. Many researchers have done research on bug report summarization using various techniques like supervised approaches, unsupervised approaches, deep learning approach, feature-based approach. In this paper, we have systematically evaluated the works and presented them in the comparative form. For our comparison work, we have selected five research papers among all. The papers are chosen with the thing in mind that all the important concepts which are getting used for bug report summarization gets covered.

The paper discusses the approach, concept, strengths, limitations, tools if used, dataset used, the evaluation techniques and the performance results that are used or obtained in the chosen research works. Our work will help other researchers have a clear overview of the very popular works in this field and thus will help improve and carry out further works in this field of research.

***Keywords*:** AUSUM;Feature-Based; Deep Learning; Semantic;Unsupervised

1. Introduction

With the emergence of world wide web, the data has increased enormously and is increasing exponentially. Thus maintenance of data and retrieval of information has become the major issues. For every web search, there is a big list of information which is displayed. Thus to get the desired information in less time, summarization is one solution. The research on text summarization started from 1958 but still achieving the summaries like human-generated summaries is a challenge. On the basis of type of approach used. the summarization techniques are classified into extractive and abstractive summarization. On the basis of number of documents considered for summary, the techniques are divided into single document summarization and multi-document summarization.

Not just now the summarization techniques are applied to normal text but now it is used for various domains like software engineering data, conversation-based data, etc. In this paper we discuss the comparative study on various works done in the field of software engineering data and in this field for the bug reports. During the software engineering process, various artifacts are produced like requirement analysis document, design documents, version control logs, bug reports, etc. Bug Reports among these artifacts are one very important document as it not only contains the information about the bug but also about the enhancements that can be done, about the resolution process and sometimes the critics to the software. Thus analyzing the bug reports is very important. The study addresses the following research questions:

- How the summarization techniques which work on text summarization well behaved when work with bug reports.
- What are the various approaches which are used for bug report summarization.
- What different datasets the researchers have used for their different approaches.
- How the various approaches performing when applied to a particular dataset.

Not just the usage of a proper model for summarizing the documents is a challenge but the proper framework for evaluation of summaries is a challenge. Mostly precision, recall, F-Score, ROUGE-Scores and Pyramid Scores are used for the evaluation of summaries but along with these readability, relevance, non-redundancy, conciseness and coverage are also important to be taken care of while evaluating the summaries. For the text summarization, feature-based approach, latent semantic analysis, graph-based methods, collaborative ranking based ,neural networks based techniques are getting used. But these text summarization techniques do not work as they work with normal data to the bug reports.

The major contributions of the paper are:

- We analyze the 5 different approaches used for bug report summarization.
- We compare the approaches on a dataset to see the impact of approaches on the dataset.
- We have found the strengths and limitations of the selected approaches which will help combine the approaches to get better results.
- We also discuss the evaluation measures being used for evaluating the quality of bug report summary.

We have organized our paper as follows: Section 2 discusses the related work in the field of bug report summarization. Section 3 discusses the summarization approaches and five techniques which we have selected from the view of various parameters. Section 4 discusses the datasets and evaluation measures being used for the bug report summarization. And finally the conclusion and future directions.

2. Related Work

Text summarization is not a new area of research. It started with the work of [Luhn (1958)] in 1958 where he studies the impact of frequency words to the important sentence extraction. The work was then carry forwarded by [Edmundson (1969)] who added the sentence location, cue-phrases and the similarity to the title to calculate the importance of the sentence. After the popularity of feature-based summarization approach, the unsupervised approaches started coming to the picture with the work of [Radev(2001)] where they used the cluster based approach utilizing the concept of centroid score to extract the important sentences from the text. After this the importance of semantic analysis started coming to the picture with the popularity of natural language processing. [Jagadeesh J (2005)], calculated the verbs , part of speech, named entities and similarity with headings to analyze the text semantically and obtained the summaries. In 2009, fuzzy-logic concept along with the feature-based approach started coming to the picture and it got a lot of popularity among the researchers. In 2013, again the use of semantic analysis along with the feature-extraction started taking the popularity and the work of [Suanmali et al.(2009)] proved how the consideration of semantic features like morphological transformation, synonyms and co-references helps improve the sentence ranking process during the summarization process. In 2014 with the work of [S.A.Babar & D.Patil (2014)], latent semantic analysis technique also started coming to the picture. In the same year 2014, the creation of summaries at the paragraph level started coming to the picture. The unsupervised approaches for extractive summarization again taking the popularity with the use of MMR technique by [Kurmi & Jain (2014)] which help reduce the redundancy in the summary. In 2016, [Jafari et al. (2016)] used the combination of semantic analysis, feature-based approach and the fuzzy logic to improve the summaries. Extending the concept of unsupervised approaches, [Liu et al. (2017) used the PageRank to improve and create the personalized summaries.

The above mentioned techniques were creating the extractive summaries. The abstractive summaries also started becoming popular with the increasing interest in natural language processing among the researchers. Abstractive techniques are classified into structure-based and semantic-based. Few of the famous works in structure-based summarization are Opiniosis [Ganesan et al. (2010)] where the graph based structure was used to create the abstractive summaries. Extending the Opiniosis graph-like structure, AMR graphs were introduced by [Lyu & Titov (2018)], [Barzilay & McKeown (2005)],[Yousfi-Monod & Prince (2008)] used the fusion and linearization techniques in the tree structures to find the abstractive summaries. Template-Based summaries are the very common structure-based abstractive summaries among the researchers which got its popularity from the works of [Harabagiu et al. (2001)] GISTEXTER, [Carenini et al. (2012)] SEA. [Tanaka et al (2009)] used the lead and body phrase, again a low-cost abstractive summarization technique to create the summaries. [Kasture1 et al. (2014)] used the rules to create the abstractive summaries. [Zhang et al. (2016)] used the predicate-argument structure to create the cross-platform abstractive summaries. [Tanaka et al. (2009)] used the neural-based AMR graphs to create the abstractive summaries. [Jobson & Gutirrez (2016)], [Nallapati et al. (2016)], [Rush et al.(2015)], [Chopra et al. (2016)] used the Deep-Learning based encoder-decoder model to create the abstractive summaries.

In the above mentioned two paragraphs, we have discussed the extractive and abstractive summarization techniques which are used for the generic text. But [Kumarasamy Mani et al. (2012)] and many researchers who are working for the bug reports proved that the above mentioned techniques do not work well for the bug reports. Bug Reports are the conversational-artifacts and resemble the meeting minutes. Bug Report contains the comments which the developers or users write when a bug occurs. Most of the researchers working on bug report summarization use the techniques which the researchers have used for the conversation artifacts like email threads [Murray & Carenini (2008)] and meeting minutes. Most popular approach used by bug report researchers is Feature-Based Approach where the researchers like [Rastkar et al.

(2010)],[Rastkar et al. (2014)],[Taware & Shinde (2015)] used the domain-based features to extract the important sentences from the bug report data. Apart from feature-based approach, unsupervised approach is also one of the other famous approach used by researchers like [Kumarasamy Mani et al. (2012)], [Lotufo et al. (2015)], [YANG et al. (2018)]. Along with the feature-based and unsupervised approaches the researchers have also used machine-based classifier approach and the support vector machines [Lotufo et al.(2015)],[Rastkar et al. (2010)], [Rastkar et al. (2014)],[YANG et al. (2018)], [He et al. (2017)].

Topic Modelling techniques like Latent Dirichlet Allocation(LDA) is also used in bug reports to extract the important topics [Nagwani & Verma (2016)]. [Li et al. (2018)] used the Deep Learning Based Approach to create the bug report summaries using deep learning concept.

3. Approaches Used:

From the survey of papers, we have found that these number of techniques have been used by researchers for the summarization purpose:

Semantic Analysis Based: In these methods, the semantics of document are taken into consideration for the selection of sentences to generate the summaries. Including the semantics, help achieve the cohesion. For extractive summaries, latent semantic analysis and topic models are among the most popular approaches while information-item, predicate-argument,rich semantic graphs, AMR Graphs, Aspect Hierarchy Trees are among the most popular approaches for generation of abstractive summaries [Gupta & Gupta(2019)].

For the bug report summarization, the researchers have used the latent semantic analysis, topic models and extraction of features like classification of sentence on some criteria like question, investigation, anthropogenic, procedural, suggestions, etc to determine the relevance of the sentence.

Graph Based: These methods help identify the structure of the sentences for determining its relevance. Paraphrasing, Word Graphs, LexRank, PageRank are among the popular techniques for creating graph-based structure for evaluation of sentence. Graph-Based approaches are also very commonly used techniques for the generation of bug report summaries. PageRank has been used by [He et al. (2017)], [Lotufo et al. (2015)] for bug report summaries.

Machine-Learning Based: These are machine-learning based approaches where the importance of sentence on the basis of training data is taken into consideration. SVM, Naive Bayesian classifiers [Gupta & S.K (2017)], and corpora-related classifiers have been used for the training of data purpose. This is also one of the widely used approaches for bug report summarization. [Rastkar et al. (2010)] , [Rastkar et al. (2014)],[YANG et al. (2018)] used the BRC classifier to train the data.

Neural-Based Deep Learning Models: These models are based upon the neural-network architecture. For the text summarization purpose, Restricted Boltzmann machine, encoder-decoder models are most widely used. In the encoder-decoder architecture, the input is fed to the encoder-part, then the encoded part goes through the various hidden layers involving some function for transformation, this transformed form goes to the decoder-part to generate the final processed form. Convolutional Neural Networks(CNN), Recurrent Neural Networks(RNN), Long Term Short Term Memory(LSTM) are again among the popular techniques in encoder-decoder model [Gupta & S.K (2018)]. Even though a lot of works have been done in the field of text summarization but these approaches have not been used much in the field of bug report summarization. [Li et al. (2018)] have used the deep-learning approaches for creating the bug report summaries.

Even though there are a number of techniques which have been applied successfully to the bug report summarization. We here in our paper, select only 5 papers for the thorough analysis. We have chosen the papers from the high-impact journals and all the papers use different approach. The following approaches are the most widely used approaches for the bug report summarization. The thorough analysis will open up the opportunity to exploit the strength of one approach to overcome the other approach limitations.

For the comparative study purpose, we have used the following works for the comparison:

- Automatic Summarization of Bug Reports [Rastkar et al. (2014)]
- AUSUM: approach for unsupervised bug report summarization [Kumarasamy Mani et al. (2012)]
- Modelling the 'Hurried' Bug Report Reading Process to Summarize Bug Reports [Lotufo et al (2015)
- Unsupervised Deep Bug Report Summarization [Li et al. (2018)]
- Towards an Improvement of Bug Report Summarization Using Two-Layer Semantic Information [YANG et al. (2018)]

Table 1 lists down the papers along with the main concept and approach followed by them. Table 4 shows the summary of the approaches chosen by us for the comparison purpose. Strengths and Limitations of the approaches have also been listed in tabular form in Table [6] to understand the advantages and the drawbacks of the methods.

TABLE 1: Approaches: Basic Information

Paper	Author	Type of Summarization	Concept and Techniques used
Automatic Summarization of Bug Reports	Rastkar et al. [Rastkar et al. (2014)]	Extractive Summarization	Feature Based + BRC
AUSUM: approach for unsupervised bug report summarization	Mani et al. [Kumarasamy Mani et al. (2012)]	Extractive Summarization	Unsupervised Approaches
Modelling the 'Hurried' Bug Report Reading Process to Summarize Bug Reports	Lotufo et al. [Lotufo et al(2015)]	Extractive Summarization	Markov-Based PageRank
Unsupervised Deep Bug Report Summarization	Xiaochen Li et al. [Li et al. (2018)]	Extractive Summarization	Stepped Auto-Encoder Deep Learning Model(Latent Semantic Analysis for Term- Frequency Vector, Cosine Similarity)
Towards an Improvement of Bug Report Summarization Using Two-Layer Semantic Information	[YANG et al. (2018)]	Extractive Summarization	Feature Based+ BRC+ Noise Removal

TABLE 2: DataSet Information

Paper	Dataset	No of Bug Reports	Bug Report Statistics
Rastkar et al. [Rastkar et al. (2014)]	BRC	36 bug reports	5-25 comments per bug report, 2361 total sentences
AUSUM: approach for unsupervised bug report summarization [Kumarasamy Mani et al. (2012)]	SDS and IBM DB2	36 bug reports(SDS), 19(DB2)	SDS(2361 sentences total, 25-15 comments per bug report), IBM DB2(2-114 comments per bug re- port, 6304 sentences total)
Modelling the Hurried Bug Report Reading Process to Summarize Bug Reports [Lotufo et al(2015)]	BRC	36 bug reports	5-25 comments per bug report, 2361 total sentences
Unsupervised Deep Bug Report Summarization [Li et al. (2018)]	SDS and ACS	36 bug reports in SDS and 96 bug reports in ADS	Average 10.83 comments per report
Towards an Improvement of Bug Report Summarization Using Two- Layer Semantic Information [YANG et al. (2018)]	BRC	36 bug reports	2361 sentences total,1906 sentences not included in any summary

4. DataSet and Evaluation Techniques:

Even though different researchers have used different datasets for the evaluation of their approach. From the survey, we observed that few datasets have been frequently used for the summarization purpose. We have listed the dataset information in Table 2 which lists down the dataset used, no of bug reports available in the corpora and their statistics.

For the evaluation purpose, following are the commonly used parameters through out the summarization works:

Precision: It refers to the number of sentences which are generated in the summary obtained from automatic summarization process, which are there in the goldenset summary. Precision helps find the accuracy and thus the usefulness of the summary.

Precision=(No of sentences common between golden-set and system generated summary)

/(No of sentences in generated summary)

Recall: It refers to the fraction of the number of sentences which are there in the golden-set summary, which belongs to the generated summary.

Recall=(No of sentences selected from Golden-set summary)

/(No of sentences in Golden-Set summary)

F-Score: It is the harmonic mean of Precision and Recall.

F-Score=(2* Precision * Recall)/(Precision+Recall). It balances the use of both Precision and Recall.

Pyramid Score: Many times annotators are used for the evaluation purpose. They help find the content which is of high-weightage [Nenkova & Passonneau (2004)]. It is based on the idea that there is no single best summary. It helps reduce short comings of human-based evaluation. The basic unit of the approach is Summary Content Unit(SCU).

ROUGE Scores in terms of Precision, Recall and F-Score: Mainly ROUGE-1, ROUGE-2 and ROUGE-L have been used for the evaluation purpose. It is a recall-based metric and depending upon the overlapping units considered for the evaluation they are classified into ROUGE-1, ROUGE-2 and ROUGE-L.

Qualitative Parameters for evaluating summaries qualitatively: Many parameters have been chosen for finding the usefulness of summary from different context. Few of the parameters chosen by researchers to evaluate the summary qualitatively in bug report summarization are:

– Accuracy [Rastkar et al. (2014)][He et al.(2017)] [Ferreira et al. (2013)]

– Time To Completion [Rastkar et al. (2014):]: It represents the difference of time that the participant took for performing a particular task without summary and with summary.

– Participant Satisfaction [Rastkar et al. (2014)]: How much the participants were satisfied with the generated summaries.

Table 2 states the approaches along with the results obtained by them in terms of Precision, Recall, F-Score, ROUGE-1, ROUGE-2 and Pyramid Precision in tabular form.

TABLE 3: Summary of Approaches

Paper	Summary of Approach
Rastkar et al. [Rastkar et al. (2014)]	They first classified the sentences of the bug reports into vector forms by extracting the features from the sentences. They used 24 features to create the vector and classifier these 24 features into four categories namely structural, participant, length and lexical. Structural features include position of sentence in the bug report, the position of the sentence in the comment, etc. Participant features include author of comment, participant dominance in the sentence. Lexical features include no of clue words, cosine similarity of sentence and the sentence,Mean Turn Probability. Length features include word count globally normalized, word count locally normalized. After creating the vectors, they trained the Bug Report Classifier to extract the important sentences and arranged them to create the extractive summaries.

AUSUM: approach for unsupervised bug report summa- rization [Kumarasamy Mani et al. (2012)]	Instead of directly applying the approach to extract the important sentences. First they classified the sentences into four categories: Question, Code, Investigation, Others. For finding out the categories, they used the parsing and keyword dictionary based approach. They removed all sentence types except Others, and then used four unsupervised approaches namely Centroid, MMR, DivRank and Grasshopper. They observed how the unsupervised approaches perform with the bug reports and found that only MMR and DivRank worked well with the bug reports. They also observed that the unsupervised approaches gave results similar to [26].
Modelling the Hurried Bug Report Reading Process to Summarize Bug Reports [Lotufo et al(2015)]	They used the unsupervised approaches observing the pattern of skimming through the bug report when a person is in a hurry. They used three hypotheses to create the summaries: use of frequently discussed topics, similarity to the title and description and sentences involving evaluation or assessment.
Unsupervised Deep Bug Report Summarization [Li et al. (2018)]	They used the deep learning based approach to create the bug report summary. First they preprocessed the document by removing the stop words, performing stemming and removing the sentences with less than three words. Then they fed the bug reports into the stepped auto-encoder. Before adding a bug report, first it finds the cosine similarity between the bug reports to find the k- most similar bug reports and then feed these similar bug reports to the trainer. Sentences of the bug reports are extracted and converted into the term-frequency vectors and then the evaluation enhancement is used to re-initialize the vectors. They categorized the sentences into software language sentences, natural language sentences by participants, natural language sentences by reporters. They detected the software sentences using Infozilla and regular expressions. Encoding and Decoding of the sentences is done according to the sentence type. To the stepped encoder-decoder model, three vectors are fed to the network. The objective is to minimize the difference between the input and output vectors. They used the five hidden layers with layer1 and layer5 having 1000 hidden layers, layer2 and layer4 having 250 hidden layers and layer3 having 10 hidden layers. RMSProp Optimizer is used to optimize the network parameters. Initial learning rate of 0.01 is used. DropOut strategy is used to prevent overfitting.
Towards an Improvement of Bug Report Summarization Using Two-Layer Semantic Information [YANG et al. (2018)]	They proposed a 2 layer model where first they have identified the semantic filtering model to filter out the sentences and then they used BRC model to train the model to find the relevant sentences. In the first step they classified the sentences into six categories namely Question, Code, Investigation, Anthropogenic, Procedural and Others. They filtered the Other sentences out. Then on the basis of the calculated 5 classes, they trained the summarizer using supervised logistic regression model. For the classification of sentences, they used the regular expressions and the keyword dictionary.

TABLE 4: Future Directions Discussed in the Above Mentioned Approaches

Paper	**Future Directions Suggested**
Automatic Summarization of Bug Reports [Rastkar et al. (2014)]	They discussed about incorporating the domain-specific features to improve the quality of generated summaries like inclusion of active authors with the comments and including steps to reproduce in the summary. They also emphasized on task-based evaluation of bug report summaries. They have evaluated their summaries for the duplicate bug report detection. But it can be done for other tasks also like relevance from the topic of interest, help in change-task during evolution process.
AUSUM: approach for unsupervised bug report summarization [Kumarasamy Mani et al. (2012)]	They emphasized on the need of improving the precision of approaches so that they can be used to carry out other related activities like extracting the frequently-asked questions. They also wish to use this approach for carrying out the code summarization by consider- ing the comments natural language text to generate class level and package level summaries.
Modelling the Hurried Bug Report Reading Process to Summarize Bug Reports [Lotufo et al(2015)]	They emphasized on analysis of LDA and other topic models for the calculation of similarity between sentences to improve the sentence relevance. They also said that the training of the corpus in- volving the characteristics of communication to annotate the sentence sentiment-wise can be done to find the relevant sentences. They also suggested the need of navigation based bug report summaries.
Unsupervised Deep Bug Report Summa-rization [Li et al. (2018)]	They emphasized on conducting the case studies for various tasks to find the effectiveness of their model. They also discussed the use of cloud computing to reduce the time to summarize for neural-based networks.
Towards an Improvement of Bug Report Summarization Using Two-Layer Semantic Information [YANG et al. (2018)]	As in the bug reports, the diversity and natural language is there and the work mainly relies on the effectiveness of classification of sentences. It is important to analyze and improve the classification system to improve the summaries.

TABLE 5: Performances of the Above Mentioned Approaches in BRC DataSet

Approach	**Precision**	**Recall**	**F-Score**	**Rouge-1**	**Rouge-2**	**Pyramid Precision**
[Rastkar et al. (2014)]	0.57	0.35	0.40	0.521	0.140	0.630
[Li et al. (2018)]	0.621	0.388	0.462	0.563	0.177	0.621
Centroid: [Kumarasamy Mani et al. (2012)]	0.636	0.269	0.3433	0.471	0.126	0.460
MMR: [Kumarasamy Mani et al. (2012)]	0.617	0.353	0.429	0.498	0.145	0.551
Hurried: [Lotufo et al(2015)]	0.710	0.300	0.410	0.525	0.153	0.710
[YANG et al. (2018)]	0.520	0.541	0.530	-	-	-

5. CONCLUSION AND FUTURE DIRECTIONS:

Bug report is the valuable artifact produced during the software development process. It is the first document which is referred when the similar problem comes. Searching for an appropriate bug report is a challenging task. Automatic summarization of bug reports help search the relevant bug report quickly. But the informal nature of bug reports in terms of conversation, domain-specific nature, noise due to the usage of abbreviation elevates the problem of automatic bug report summarization. General Text summarization approaches do not work well with the bug reports. Thus to create bug report specific summarization approach to make the summaries more efficient is the need of time.

In this paper, we have chosen the five approaches and have compared them in terms of the results. We have explained their approaches in tabular form so that their differences and similarities can be easily determined. We have also identified these approaches from the corpus point of view. We have also listed the future directions in the tabular form in Table 4 so that the readers can have a full list of research works that can be performed for improving the summarization approach. TABLE 6: Strengths and Limitations of the Above Mentioned Approaches

Paper	**Strengths**	**Limitations**
Automatic Summarization of Bug Reports [Rastkar et al. (2014)]	• Their model helps in the duplicate bug report detection with no degradation in accuracy. The model used the already existing approach for the bug reports with incorpora- tion of bug report specific features. With simple model, they are able to achieve very good results for bug re- port summarization. Along with just summarizing the bug report, they evaluate their summaries on the basis of how useful summaries are for the software developers.	• The model requires the training data which adds cost to their approach. The model uses almost all the features which were used for email sum- marization. As the nature of bug re- port is very specific to the project, the model needs the project-specific information to find the sentence rele- vance. This limits its performance.
AUSUM: approach for unsuper- vised bug report summarization [Kumarasamy Mani et al. (2012)]	• The approach does not require train-ing data. As they are not dependent on any data, the model is domain- independent.	• As clear from the results obtained, if the unsupervised approaches are directly applied to the application, for few reports convergence was not happening. But once the filtration of sentences was done, they were able to converge. Thus we need to de- vise better filtering mechanisms to help retain more useful information for better summaries is required. In this paper, the filtration of sentences is based on the keyword-based dictionary and regular expressions. For these they rely on the Stanford NLP Parser. The limitations of Stanford NLP Parser affects the performance of summaries obtained from this approach.
Modelling the Hurried Bug Report Reading Process to Summarize Bug Reports [28]	• The model uses unsupervised approaches for summarization. Thus makes the approach domain-independent and helps get rid of tedious task of preparation of training data.	• As the model relies on the hypothesis of how a developer will skim through the bug reports when in hurry.
Unsupervised Deep Bug Report Summarization [Li et al. (2018)]	• The model is unsupervised so with it, it is possible to perform the deep neural network processing without the need of big training data.	• The model is very time-consuming. It takes on an average 5.6 minutes to summarize one bug report. • The model with too-much complex- ity is giving the results similar to other approaches.
Towards an Improvement of Bug Report Summarization Using Two- Layer Semantic Information [YANG et al. (2018)]	• Along with the features which are specific to the domain, they also con- sidered the sentence type for filter- ing the sentences. For classifying the sentences, they took the semantic- information in consideration which resulted in obtaining the better sum- maries in terms of F-Score.	• The model relies upon the classifi- cation of sentences for the noise re- moval and consideration for the sum- mary generation.

References

[1] Barzilay, R., & McKeown, K. R. (2005). Sentence fusion for mul tidocument news summarization. Computational Linguistics, 31, 297–327.
[2] Carenini, G., Ng, R. T., & Pauls, A. (2012). Multi - document summarization of evaluative text, doi:http://sci-hub.tw/10.1111/j.1467-8640.2012.00417.x.
[3] Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive sentence summarization with attentive recurrent neural networks. In Conference: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies.
[4] Edmundson, H. P. (1969). New methods in automatic ex- tracting. Journal of the ACM (JACM), 16, 264–285. doi:10.1145/321510.321519.
[5] Ferreira, R., de Souza Cabral, L., Lins, R. D., de Frana Silva, G., & Freitas, F. (2013). Assessing sentence scoring techniques for extractive text summarization. Elsevier Expert Systems with Appli- cations, .
[6] Ganesan, K., Zhai, C., & Han, J. (2010). Opinosis: A graph-based approach to abstractive summarization of highly redundant opinions. In 23rd International Conference on Computational Linguistics (pp. 340–348).
[7] Gupta, S., & Gupta, S. K. (2019). Abstractive summarization: An overview of the state of the art. Expert Syst. Appl., 121, 49– 65. URL: https://doi.org/10.1016/j.eswa.2018.12.011. doi:10.1016/j.eswa.2018.12.011.
[8] Gupta, S., & S.K, G. (2017). Summarization of software artifacts : A review. International Journal of Computer Science and Informa- tion Technology, 9, 165–187. doi:10.5121/ijcsit.2017.9512.
[9] Gupta, S., & S.K, G. (2018). Deep learning in automatic text summa- rization. International Journal of Computer Science and Information Security (IJCSIS), 16, 150–155.
[10] Harabagiu, S., Moldovan, D., Morarescu, P., Lacatusu, F., Mihalcea, R., Rus, V., & Girju, R. (2001). Gistexter: A system for summarizing text documents, .
[11] He, J., Nazar, N., Zhang, J., Zhang, T., & Ren, Z. (2017). Prst: A pagerank-based summarization technique for summarizing bug reports with duplicates. International Journal of Soft- ware Engineering and Knowledge Engineering, 27, 869–896. doi:10.1142/S0218194017500322.
[12] Jafari, M., Shahabi, A. S., Jing Wang, Y. Q., Tao, X., & Gheisari, M. (2016). Automatic text summarization using fuzzy inference. In 22nd International Conference on Automation and Computing. doi:10.1109/IConAC.2016.7604928.
[13] Jagadeesh J, V. V., Prasad Pingali (2005). Sentence extraction based single document summarization. Workshop on Document Summa- rization, .
[14] Jobson, E., & Gutirrez, A. (2016). Abstractive text summarization using attentive sequence-to-sequence rnns. Stanford Reports,. Kasture1, N. R., Yargal, N., Singh, N. N., Kulkarni, N., & Mathur,
[15] V. (2014). A survey on methods of abstractive summarization. INTERNATIONAL JOURNAL FOR RESEARCH IN EMERGING SCIENCE AND TECHNOLOGY, 1, 53–57.
[16] Kumarasamy Mani, S. K., Catherine, R., Sinha, V., & Dubey, A. (2012). Ausum: Approach for unsupervised bug report summarization. (p. 11). doi:10.1145/2393596.2393607.
[17] Kurmi, R., & Jain, P. (2014). Text summarization using enhanced mmr technique. In International Conference on Computer Communication and Informatics. doi:978-1-4799 2352-6/14.
[18] Li, X., Jiang, H., Liu, D., Ren, Z., & Li, G. (2018). Unsupervised deep bug report summarization. In Proceedings of the 26th Conference on Program Comprehension ICPC '18 (pp. 144–155). New York, NY, USA: ACM. URL: http://doi.acm.org/10.1145/ 3196321.3196326. doi:10.1145/3196321.3196326.
[19] Liu, Y., Wang, X., Zhang, J., & Hu, H. (2017). Personalized pagerank based multi-document summarization. In IEEE Workshop on Semantic Computing and Systems (pp. 169–173). doi:10.1109/WSCS.2008.32.
[20] Lotufo, R., Malik, Z., & Czarnecki, K. (2015). Modelling the 'hurried' bug report reading process to summarize bug reports. Empirical Software Engineering Journal, 20, 516– 548. doi:10.1007/s10664-014-9311-2.
[21] Luhn, H. (1958). The automatic creation of literature abstracts. IBM Journal of Research and Development, 2, 159–165.
[22] Lyu, C., & Titov, I. (2018). Amr parsing as graph prediction with la- tent alignment. ACM Transactions on Asian Language Information Processing, . doi:arXiv:1805.05286v1.
[23] Murray, G., & Carenini, G. (2008). Summarizing spoken and writ- ten conversations. In Proceedings of the Conference on Empirical Methods in Natural Language Processing EMNLP '08 (pp. 773–782). Stroudsburg, PA, USA: Association for Computational Linguistics. URL: http://dl.acm.org/citation.cfm?id= 1613715.1613813.
[24] Nagwani, N. K., & Verma, S. (2016). Generating intelli- gent summary terms for improving knowledge discovery in software bug repositories. International Journal of Soft- ware Engineering and Knowledge Engineering, 26, 827– 844. doi:https://doi.org/10.1142/S0218194016500273.
[25] Nallapati, R., Zhou, B., dos Santos, C., Gulehre, C., & Lapata, M. (2016). Abstractive text summarization using sequence-to-sequence rnns and beyond. In The SIGNLL Conference on Com- putational Natural Language Learning..
[26] Nenkova, A., & Passonneau, R. (2004). Evaluating content selection in summarization: The pyramid method. In Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT- NAACL 2004 (pp. 145–152). Boston, Massachusetts, USA: As- sociation for Computational Linguistics. URL: https://www. aclweb.org/anthology/N04-1019.
[27] Radev, D. R. (2001). Experiments in single and multi-document summarization using mead. First Document Understanding Conference,.
[28] Rastkar, S., Murphy, G. C., & Murray, G. (2010). Summarizing soft- ware artifacts:a case study of bug reports. In Proceedings of the 26th Conference on Program Comprehension ICSE 2010.
[29] Rastkar, S., Murphy, G. C., & Murray, G. (2014). Automatic summa- rization of bug reports. IEEE Transactions on Software Engineering, 40, 366–380.
[30] Rush, A. M., Chopra, S., & Weston, J. (2015). A neural attention model for abstractive sentence summarization.
[31] S.A.Babar, & D.Patil, P. (2014). Improving performance of text summarization. In International Conference on Information and Communication Technologies. volume 46. doi:10.1016/j.procs.2015.02.031.
[32] Suanmali, L., Salim, N., & Binwahlan, M. S. (2009). Fuzzy logic based method for improving text summarization. International Journal of Computer Science and Information Security, 2.
[33] Tanaka, H., Kinoshita, A., Kobayakawa, T., Kumano, T., & Kato, N. (2009). Syntax-driven sentence revision for broadcast news sum- marization. In Workshop on Language Generation and Summarisation (pp. 39–47).
[34] Taware, M. R. K., & Shinde, P. S. A. (2015). Complete bug report summarization using task-based evaluation. International Journal of Research, (pp. 418–422).

[35] YANG, C.-Z., Cheng-Min, & CHUNG, Y.-H. (2018). Towards an improvement of bug report summarization using two-layer semantic information. IEICE TRANS. INF. and SYST., 101, 1743– 1750.
[36] Yousfi-Monod, M., & Prince, V. (2008). Sentence compression as a step in summarization or an alternative path in text shortening. In Coling 2008 (pp. 139–142).
[37] Zhang, J., Zhou, Y., & Zong, C. (2016). Abstractive cross-language summarization via translation model enhanced predicate argu- ment structure fusing. IEEE/ACM TRANSACTIONS ON AU- DIO, SPEECH, AND LANGUAGE PROCESSING, (pp. 1–25). doi:10.1109/TASLP.2016.2586608.