www.ijaceeonline.com ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

Study of Techniques and Algorithms Involved in Unmanned Comment generation for Source Code

Prof. Prashant Govardhan
Assistant Professor, Department of Computer Science and Engineering,
Priyadarshini Institute of Engineering and Technology, Maharashtra, India
Email: er.phgovardhan@gmail.com

Shweta Patil
UG Scholar, Department of Computer Science and Engineering,
Priyadarshini Institute of Engineering and Technology, Maharashtra, India
Email: shwetaspr866@gmail.com

Nachiket Gaurihar
UG Scholar, Department of Computer Science and Engineering,
Priyadarshini Institute of Engineering and Technology, Nagpur
Email: nachiketgaurihar66@gmail.com

Sanket Adle
UG Scholar, Department of Computer Science and Engineering,
Priyadarshini Institute of Engineering and Technology, Maharashtra, India
Email: sanketadle6@gmail.com

Uzma Sheikh
UG Scholar, Department of Computer Science and Engineering,
Priyadarshini Institute of Engineering and Technology, Maharashtra, India
Email: uzmasheikh486@gmail.com

Abstract: *Code commenting is a good practice which has been suggested to be done by all programmers while they develop a code. However, to preserve the effort and the time overhead required to append the comments in a code usually programmers skip this essential step of code commenting. Comments in a code play a very crucial role in improving the readability and the understandability of the code when the code is referred by some other developer or even when the author of the code himself refers his own code after a long period of time. It basically provides the skeletal structure of the code and makes the other developers understand the logic of the programmer behind the code. Comments covey the intentions of the developer and provides a brief idea about the working of the code. But many developers skip this important step of commenting the code due to lack of knowledge about the importance of code commenting. Thus, there is a need of auto commenting the source code to enhance its readability and to convey the developer's logic efficiently and quickly. Hence, researchers and developers have focused their attention towards developing the source code comments automatically and have been working on this area and trying out different techniques from several years. The use of Information Retrieval techniques and Neural Network based techniques have gained immense popularity in the past decade due to their impressive results. Thus, in this paper we present the survey of the different approaches used so far and the existing systems to generate comments of source code automatically and we also give their drawbacks.*

Keyword: *Auto-commenting; Algorithms; Neural Network; Information Retrieval.*

Cite this paper:

Prashant Govardhan, Shweta Patil, Nachiket Gaurihar, Sanket Adle, Uzma Sheikh "Study of Techniques and Algorithms Involved in Unmanned Comment Generation for Source Code ", International Journal of Advances in Computer and Electronics Engineering, Vol. 4, No. 10, pp. 1-9 October 2019.

1. INTRODUCTION

Comments in a code enhance the readability of the code by making it easier to understand by providing brief information about the working of the code, logic of the code and the purpose of the different segments

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

used in the code. It helps to convey the developer's idea behind the creation of the code. It is very helpful when we work in a team to make the other team mates understand the code. The paper [1] illustrates an experiment on the PL/1 program which considers different parameters such as presence of comments, absence of comments, procedure formats such as none, internal and external. The program was modified by taking combinations of these parameters and six different variations of the same program was created which was presented in front of the students. The experiment concluded that in procedure less programming it was difficult to comprehend the code without comments rather than the one with comments thus, explaining the importance of code commenting. Many papers [2], [3], [4] also remark the importance of comments in a code. Absence of comments in a source code is an issue which has been carried since the advent of programming languages. In the past decade researches have been curious in dealing this issue hence have focused their attention on auto commenting the source code. Many techniques have been used to deal with this issue [5]. However, generating comments of the source code which are accurate and provide high quality is a complex task. There are many challenges which have to be faced during this process. Many papers have given description of these challenges which include collection of datasets, construction of source code models, text generation and quality assessment of the generated comments. In this paper we will learn about the basic steps to be followed in the process of automatic comment generation and also illustrate the techniques and systems used in the past.

2. THE MAIN CHALLENGES IN DESIGNING A SYSTEM FOR AUTOMATIC GENERATION FOR SOURCE CODE

Many works have been proposed in this domain of designing a system for the automatic generation of source code comments. The challenges faced by developers while initiating some of the necessary steps for the development of the system are:

2.1 Extracting the datasets

This is the most important step which forms a basis of the project. The collection of the correct data is essential for the overall development of the project. There are huge datasets available on the internet on various open source platforms such as Github and Stack Overflow. The humungous data available on these sites can be extracted and stored in the form of dataset for the purpose of generating automatic source code comments in natural languages.

Selection of high-quality data to develop a suitable dataset: It is necessary to pick the correct and high-quality data from all the sources available to produce more accurate and high-quality comments. There is a large amount of data available on the internet most of which is not useful for the project hence it is very necessary to analyze the data and pick the correct one. Thus, to initiate a project on automatic comment generation we will only be using source codes which already contain comments. The comments in the source code should be of high quality to obtain high quality results.

Labeling and storing the data in the database: Once the suitable data is selected for creating the dataset, the gathered source codes containing comments will be stored in a database in the form of labeled data that is the data will be tagged with some descriptive information about the data which can be used for training, generating comments in human readable languages and also for analyzing the quality of the comments generated. However, storing the collected data in the form of labeled data in the database is a tedious and a very essential task as depending on the source code the suitable comments will be fetched from the dataset to provide the required results.

2.1.1 Existing approaches

In the previous research papers on extracting the data from the resources available to generate a high quality, well suited dataset the following approaches have been proposed.

2.1.2 To extract the data from open source platforms

Tremendous data is available on open source platform like GitHub. It is one of the largest repositories which contain millions of codes. Therefore, we can directly download the required codes from these sites as per our needs. The greatest drawback of downloading code from GitHub is that it is such a large repository of code which contains large number of codes which makes it necessary to filter the appropriate codes containing comments from the entire free pool of the codes available. Filtering and selecting the high quality and appropriate source codes for the project from such a large repository is a tedious job.

2.1.3 Collection of data from Q&A sites

According to the approach proposed in [6], the data from Q&A sites such as Stack Overflow can be retrieved according to the questions asked by the users this means that when a user asks a question on a Q&A site it gives the description of the problem to which he demands an answer, for example, What is the code to count the even and odd numbers in a list? The answer to this question will be a code which counts the numbers and gives the required output. Thus, by using the problem statement specified by the user as a reference we can generate the comments for the source code, that is, the code which counts the numbers can be com-

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

mented as "to count even and odd numbers in a list" as specified by the user in the Q&A sites. This approach can be useful for creating more appropriate, convincing and high-quality comments. Figure [7] illustrate this approach. To count even and odd numbers in list. Figure 1 shows how the comments are used in the code.

```
# Python program to count Even
# and Odd numbers in a List

# list of numbers
list1 = [10, 21, 4, 45, 66, 93, 1]

even_count, odd_count = 0, 0

# iterating each number in list
for num in list1:

    # checking condition
    if num % 2 == 0:
        even_count += 1

    else:
        odd_count += 1

print("Even numbers in the list: ", even_count)
print("Odd numbers in the list: ", odd_count)
```

Figure 1 To count even and odd numbers in list.

The disadvantage of this approach is that the problem statement written by the user directly will be in the form of the question, hence we will have to remove some words to make it a statement which is suitable to be used as a comment. This can be done by using a technique similar to the Removal of Stop Words technique [8] and Stemming [9] which is used in data mining with the help of which we can also eliminate question words such as write, give, etcetera as well as eliminate the question mark. We can also use Natural Language Processing techniques to convert questions into high quality statements by constructing a parse tree and then removing the irrelevant words. So basically, we can mine the data [10], [11], [12], [13] available on the Q&A site to obtain the most accurate and desirable results.

2.2 Construction of the source code model

In automatic source commenting the task of constructing an appropriate source code model is very crucial. This source code model plays an important role in fetching the most suitable comments from the dataset by matching the source code models of the source code to the most similar model in the dataset thus retrieving the comments. There are various source code models which include:

2.2.1 Abstract syntax trees (AST's)

It represents the source code in the form of a hierarchal tree which gives information about the syntax of the source code on the basis of some rules.

2.2.2 Parse tree

The parse tree represents the source code in the form of a tree which is constructed on the basis of the derivation of the grammar used in the source code. . The parse tree creates a skeletal structure of the source code which can also be matched with the similar parse tree in the dataset.

2.2.3 Token based models

It constructs a source code model which is a representation of all the tokens used in the source code. The names of the keywords, methods, operators, symbols are extracted as tokens from the source code. This model is majorly used in IR i.e. Information Retrieval based systems in which the tokens are matched with the existing code segment tokens in the dataset and the appropriate comments related to the code are fetched from the dataset.

2.2.4 Syntax based models

The syntax-based source code [14] [15] model constructs a syntax tree of the source code. It also deals with the semantics of the code and is majorly used in neural network-based algorithms.

2.2.5 Software Word Usage Model (SWUM)

It properly captures the variable names and the names of other identifiers from the signature of the methods.

There are other models as well which are used in [15]. However, researches are still going on the development of a source code model which is complete such that it correctly defines the tokens, lexical information, syntax and semantics of the source code to generate appropriate algorithms.

2.3 Matching the source code segments with the suitable data in the database

This is another crucial task which involves matching of the source code segments with the appropriate data available in the dataset and fetching the data to obtain desirable results. This can be achieved by information retrieval system and vector space models.

2.3.1 Information Retrieval system

Which can check for the required data from the large dataset by searching the data directly by the full-text or by the indexes provided for the text in the dataset. An information retrieval system retrieves the most appropriate information and collaborates well with the labeled data in this process.

2.3.2 Vector Space Models

Another technique which can be used for comparing the identifiers in a query with the identifiers of the matching code in the dataset is the Vector Space

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

Model. The vector space model creates a vector of tokens or identifiers and then compares and presents the similarities between two documents. This technique is very useful to retrieve appropriate Information from the IR system.

2.4 Generation of comments in natural languages

The generation of comments in natural languages from the source code model in the form of trees or tokens is a complicated task. The natural language is not structured which makes the generation of comments a more critical task. Many approaches have been used for the purpose of text generation. This is a critical task and could be performed as follows-

2.4.1 Approaches

2.4.1.1 Using a documentation generator

The authors [16] present a documentation generator which provides comments by analyzing the statements and the comments present in the source code. This tool actually works on the metadata provided by the programmer in the code. [17] explains the generation of the code summaries from the software artifacts or the code corpus.

2.4.1.2 Search oriented text generation method

It works by extracting the existing comment or the code summaries from the code corpus. The existing comments provided by the programmer are taken as the basis to formulate new suitable comments for the source code.

2.4.2 Information Retrieval based comment generation algorithm

There are basically three types of IR based algorithms used:

LSI-based algorithm: which is popular as Latent Semantic analysis is a technique which is used to increase the accuracy of the IR system as it finds out the underlying semantic of the source code by establishing relationships between the words. It results in to the proper understanding of the code by using Single Value Decomposition [18]. *Drawback:* Indexing based on individual attribute is quite difficult because the resultant representation is dense.

Code Clone Detection Technique: is a technique in which the clones of the codes are created by copying the code to some other location and then comparing the codes to find out the similarities between two codes and comparing whether they area like. *Drawback:* The drawback of this technique is that if multiple clones of the same code are made and there is a bug in the main code then that bug has to be corrected in all the clones of that code [19].

LDA based algorithm: which [20] is also known as Latent Dirichlet Allocation is a generative algorithm as it is capable of generating new words on the basis of the topics. The LDA algorithm bifurcates a sentence into different topics. For example, Cat drinks milk is a sentence which contains topics such as Cat and milk. After identifying the topics, it then generates the attributes for the topics for instance, cat can have attributes or related words like kitten, meow, etc. So likewise, new observations can be generated for existing observations.

2.4.3 Neural Network based comment generation algorithms

There are two algorithms for generating text in natural languages

Single encoder-based algorithm: This algorithm uses a single structure of encoder-decoder to generate the text in natural language. They fetch the information from the source code only via single encoder [21]. *Drawback:* The text generated is of moderate quality and less accurate

Multiple encoder-based algorithm: This algorithm uses multiple structures of encoder-decoder to generate the text in natural language. Due to the multiple structures this algorithm is capable of fetching or extracting more information from the source code due to which the results produced are more accurate and of high quality [22].

2.4.4 Others algorithms

There are various other methods and algorithms used they are:

Rule based text generation method: The rule-based text generating approach works on a set of predefined rules. These rule help to generate the text in natural languages. It also considers natural language models and templates to define the rules.

Text generation based on heuristic values: This approach works on the heuristic values which generates new text comments based on the experience with similar code segments. It works on the facts which are the heuristic values, for example, if there are clouds it could rain. This is an example of heuristics which deal with facts [23], [24] and [25]. The other comment generation algorithms include.

Comment generation algorithms based on stereotypes: Another approach used in [26] is creating stereotypes of the method in C++ programs [27],[28]. This paper deals by creating a skeletal structure of the methods

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

leaving some places like the name of the method, the parameters, and the operation performed by the method blank. These blank spaces are then filled up by the tokens which are generated from the methods of the source code. For instance, if there is a method add which takes two parameters as input and performs addition then the description will be "The method add of class operation takes parameters a and b and performs addition operation". Table I shows the different systems model and algorithm.

TABLE I: SYSTEMS, MODELS AND ALGORITHM

Systems	Models	Algorithms
Information Retrieval	Token based models	• LSI algorithm • Code clone detection-based algorithm • LDA based algorithm
Neural Network	Syntax based models	• Single encoder-based algorithm • Multiple encoder-based algorithm
Others	Various source code models	• Rule based text generation method • Text generation based on heuristics values • Comment generation algorithms based on stereotypes • Software word usage model-based algorithm

So basically, in this paper a generalized structure is given and then the values are inserted according to the source code. This approach can also be used for loops in C and C++. *Drawback:* This work deals with the signature of the methods which themselves give inadequate information [29], [30]. Therefore, the comments generated are of poor quality. The drawbacks of this technique can be removed by using Software Word Usage Model (SWUM).

Software Word Usage Model based algorithms (SWUM): This algorithm works on SWUM model to remove the drawbacks of stereotyping algorithms. The SWUM is capable of representing the lexical and structural information of the code much better. It properly captures the variable names and the names of other identifiers from the signature of the methods. Then on the basis of this information it calculates the score by comparing values and then on the basis of the score it generates text.

2.5 Appending the comments at appropriate locations in the source code

Another important issue which is very important in making the comments more effective is the location in the source code where the comments should be appended. The location is very important as it is very important to make the meaning of the code clearer. Hence, it is very essential to provide the comments at correct locations in the source code. Table II shows the comparison of the merits and demerits of the systems.

TABLE II: COMPARISONS OF MERIT AND DEMERIT OF SYSTEMS

Systems	Merits	Demerits
Information Retrieval	• Similarity computation is easier. • It provides some basic metric which enables to extract most relevant information.	• The result produced by this system does not represent a proper structure but only contains the tokens. • IR systems does not output the comments when the tokens do not match any code snippets present in the dataset.
Neural Network	• It is an adaptive system. • It gives output in any situation.	• It is a complex concept.

2.5.1Existing System

We will now deal with the existing systems and techniques which are being used to produce the computer-generated comments. Here we summarize the existing work in this domain of auto commenting. Due to the increased popularity of software development among the masses and the industries billions of codes as being generated every day. The software industry has gained momentum in past decade with the increasing demand of software products. Figure 2 shows the two types of existing systems.

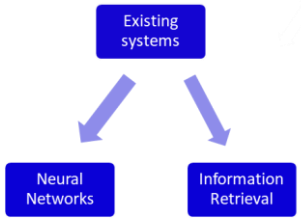

Figure 2 Two types of existing systems

Researches have shown more than 60% of the entire cost of software development is expended on the maintenance of the software which includes debugging, updating, understanding the software [20], [31]. As understanding the code is the prerequisite for making any changes in the source code the time consumed in comprehending the code is very high [32], [33], [34],

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

[35].

Therefore, the generations of automatic source code comments have gained a lot of attention and is a trending topic for researchers. The comments themselves can be classified into two broad classes, first are the comments which are written manually by the programmers which are termed as native comments and second are the comments which are automatically generated by the computer based on the source code and these comments are termed as analytical comments. The comments can further be classified into different types such as: Full Line Comments, Summary comments, End-of-line Comments, Block Comments, Nested Comments, Mega Comments, and Comments for hiding code. The major and the most popular systems which are used to generate the source code comments are the Information Retrieval based system and the neural network-based system.

2.5.1.1 Information Retrieval based system

In information retrieval-based system the dataset is manipulated to produce the most desirable comments [36],[37]. The information retrieval-based system focuses on finding out the similarities between the query demanded and the dataset by matching the tokens and depending on the similarities it gives scores. It then checks whether the scores are good enough and closer in relativity to the tokens present in the query and on this basis, it generates the results.

Merits:

- Similarity computation is easier.
- It provides some basic metric which enables to extract most relevant information.

Drawbacks:

- The limitation of this system is that the IR system deals with only the tokens which are commonly termed as "bag of words" and it does not focus on the structure of the statement. Therefore, the results produced by this system does not represent a proper structure but only contains the tokens. Hence, it does not produce comments which can be directly used for the source code. A proper statement needs to be constructed from the tokens.

- The IR system fails to generate accurate tokens from the source code when the keywords, the variable names and the methods are poorly named or cited [38], [39]. This limitation can affect the quality of the generated comments.

- The other limitation is that the IR system does not output the comments when the token does not match any code snippets present in the dataset. Thus, it is not a completely reliable method. The different information retrieval-based algorithms.

2.5.1.2 Neural Network based System

Another popular system used is Neural Network (NN) based system which has a unique ability of learning new things with the help of the existing input [40],[41]. It actually works like the neurons in the brain. The Figure 3 shows how the neurons are connected in the neural network.

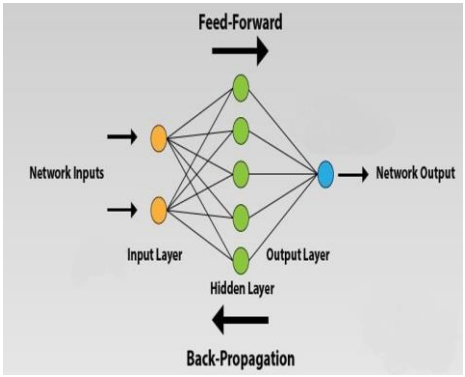

Figure 3 Neurons connected in a Neural Network

As shown in Figure [42], a neural network is actually a network of interconnected nodes which have different layers as shown in the figure. The starting layer is for the input which is provided to the network while the last layer generates the results based on the inputs received. The layers in between are responsible for the actually processing of the data which modifies the result at every increasing level. The intermediate nodes take input from the nodes connected in the previous level and based on the heuristic approach and reasoning develops a result which is again passed on to the node in the next layer. In generating automatic source comments as well the input layer takes the source code as input, then processes the source code by building up the comment and outputs the final comment.

It works on the syntax-based model. It uses algorithms such as Single encoder algorithm and multiple encoder-based algorithm which generate code comments

Merits:

- After the comments are generated the quality of the resultant comment is evaluated and if it is not satisfactory the comment is again sent as an input to the first layer and the process continues. Due to this adaptive nature of the neural networks it has gained immense popularity in the generation of comments automatically.
- Unlike, information retrieval systems it gives the results in any case as it learns from the previous outputs and keeps on generating the desired outputs.

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

Demerit:

- Neural networks are a complex concept with lot of intricacies involved in it as it also learns new things while it is processing the output. Therefore, working with it requires good proficiency about it.

3. CONCLUSION

In this paper we have presented the different techniques and algorithms which have been used to generate source code comments automatically. We have also discussed about the drawbacks of the techniques used so far which makes us understand the scope of development in this area. So far 100% accuracy has not been achieved with the generated comments and this leaves a scope for further improvements in the techniques used as also discussed in papers [43], [44]. The discussion about the detailed study of the different steps involved in making a system which automatically generates comments for the source code have also been provided and the comparative study of the most famous systems for automatic comment generation, which are the Information Retrieval System and the neural networks have also been achieved in this paper. We have discussed about these two systems in depth and analyzed that the accuracy of the generated comments can be improved by the use of neural networks as it is adaptive and has the ability to learn continuously and keeps on synthesizing new results based on the previous results. Therefore, we believe the working with neural networks can remove the drawbacks of the previous techniques and provide more desirable outputs.

4. ACKNOWLEDEMENT

We would like to express our sincere thanks to Board of Management and Principal, Priyadarshini Institute of Engineering and Technology, Nagpur, Maharashtra, India for providing opportunity to carried out our work in very loving and peaceful environment. We are also thankful to Dr. P. S. Prasad, Head of Department, Computer Science and Engineering Department at Priyadarshini Institute of Engineering and Technology, Nagpur, Maharashtra, India for his constant support and guidance. Finally, we would like to thank all those who extended their direct or indirect support for carrying out our research work.

REFERENCES

[1] Ted Tenny, (1998), "Program readability: Procedures versus comments", *IEEE Transactions on Software Engineering*, 14(9): pp.1271–1279.

[2] Ted Tenny (1985), "Procedures and comments vs the banker's algorithm", ACM *SIGCSE Bulletin*, 17(3), pp.44–53.

[3] Giriprasad Sridhara, Emily Hill, Divya Muppaneni, Lori Pol lock, and K. Vijay- Shanker (2010), "Towards Automatically Generating Summary Comments for Java Methods," *in Automated Software Engineering*, pp.43–52.

[4] Giriprasad Sridhara, Lori Pollock, and K. Vijay-Shanker, (2011), "Automatically Detecting and Describing High Level Actions within Methods*", in International Conference on Software Engineering*, pp. 101– 110

[5] Bai Yang, Zhang Liping and Zhao Fengrong, (2019), "A survey on research code comment*", In Proceedings of the 2019 3rd International Conference on Management Engineering, Software Engineering and Service Sciences*, pp. 45–51.

[6] E. Wong, J. Yang, and L. Tan, (2013), "Auto Comment: Mining question and answer sites for automatic comment generation*", in Automated Software Engineering*, pp. 562–567

[7] Retrieved date: [15, September,2019], online available at: https://www.geeksforgeeks.org/python-program-to-count-even-and-odd-numbers-in-a-list/

[8] Prabhu Kavin B, Ganapathy S, (2017), "Data Mining Techniques for Providing Network Security through Intrusion Detection Systems: A Survey", *International Journal of Advances in Computer and Electronics Engineering*, Vol 2, Issue. 10, pp. 1–6

[9] Retrieved date: [15, September,2019], online available at: https://wikipedia.org>stop_words

[10] Retrieved date: [15, September,2019], online available at: https://www.geeksforgeeks.org ›introduction-to-stemming

[11] Prashant Govardhan, K. P. Wagh , P. N. Chatur (2014), "Web Document Clustering using Proposed Similarity Measure", *International Journal of Computer Applications*,pp.09758887.

[12] Prashant Govardhan, Prakash Prasad (2017), "A Survey on Approaches to Feature Selection", *International Journal of Advanced Engineering, Management and Science (IJAEMS)*, Issue.2, pp.2454- 1311.

[13] Prashant Govardhan, Prof. K. P. Wagh, Dr. P. N. Chatur (2013), "Survey on Similarity Measure for Clustering", *International Journal of Advanced Research in Computer Science and Software Engineering*, Vol 3, Issue.2, pp.2277-128X

[14] Prashant Govardhan, Rashi Lanjewar, Rasika Korde, (2018), "Survey on Crop Yield Prediction Using Data Mining Techniques", *International Journal of Advances in Computer and Electronics Engineering*, Vol 3, Issue.12, pp.1–6.

[15] Tung Thanh Nguyen, Anh Tuan Nguyen, Hoan Anh Nguyen, and Tien N Nguyen, (2013), "A statistical semantic language model for source code." *In Proceedings of the 2013 9th Joint Meeting on Foundations of Software Engineering*," pp. 532–542.

[16] Sonia Haiduc, Jairo Aponte, Laura Moreno, and Andrian Marcus, (2010), "The use of automated text summarization techniques for summarizing source code." *In Reverse Engineering (WCRE), 2010 17th Working Conference* IEEE, pp.35–44.

[17] Paul W Burney and Collin McMillan., (2014), "Automatic documentation generation via source code summarization of method context", *In Proceedings of the 22nd International Conference on Program Comprehension ACM, pp.*279–290.

[18] Girish Maskeri, Santonu Sarkar, and Kenneth Heafield, (2008), "Mining business topics in source code using latent dirichlet allocation", *In Proceedings of the 1st India Software Engineering Conference (ISEC) ACM*, pp.113–120.

[19] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, (2014), "Learning phrase representations using rnn encoder-decoder for statistical machine translation", arXiv preprint arXiv:1406.1078.

[20] Edmund Wong, Taiyue Liu, and Lin Tan. Clocom, (2015), "Mining existing source code for automatic comment generation" *In Software Analysis, Evolutionand Reengineering (SANER), 2015 IEEE 22nd International Conference*, pp.380–389.

[21] David M Blei, Andrew Y Ng, and Michael I Jordan, (2003), "Latent dirichlet allocation. Journal of machine Learning research", pp.993– 1022.

[22] Yuding Liang and Kenny Qili Zhu, (2018), "Automatic generation of text descriptive comments for code blocks", *In Thirty-Second AAAI Conference on Artificial Intelligence*.

[23] Yao Wan, Zhou Zhao, Min Yang, Guandong Xu, Haochao Ying, Jian Wu, and Philip S Yu,(2011), "Improving automatic source code summarization via deep reinforcement learning.", *In Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, pp.397– 407.

[24] Yusuke Ode, Hiroyuki Hudoba, Graham Nesbit, Hideaki Hate, Saurian Sakti, Tomoki Toda, and Satoshi Nakamura, (2015), ". Learning to generate pseudo code from source code using statistical machine translation (t)", *In Automated Software Engineering (ASE), 2015 30th IEEE/ACM International Conference*, pp.574–584.

[25] Laura Moreno, Jairo Aponte, Giriprasad Sridhara, Andrian Marcus, Lori Pollock, and K Vijay-Shanker, (2013), "Automatic generation of natural language summaries for java classes", *In Program Comprehension (ICPC), 2013 IEEE 21st International Conference*, pp.23–32.

[26] Sarah Rastkar, Gail Murphy, and Alexander WJ Bradley, (2011), "Generating natural language summaries for crosscutting source code concerns", *In Software Maintenance (ICSM), 2011 27th IEEE International Conference*, pp.103–112.

[27] Nahla Abid, Natalia Dragan, Michael Collard, and Jonathan Maletic. (2015), "Using stereotypes in the automatic generation of natural language summaries for c++ methods", *In Software Maintenance and Evolution (ICSME), 2015 IEEE International Conference*, pp.561–565.

[28] Giriprasad Sridhara, Emily Hill, D. Muppaneni, Lori Pollock, and K. Vijay- Shanker, (2010), "Towards Automatically Generating Summary Comments for Java Methods," *in Automated Software Engineering*, pp. 43–52.

[29] Giriprasad Sridhara, Lori Pollock, and K. Vijay-Shanker, (2011), "Generating Parameter Comments and Integrating with Method Summaries,", *In International Conference on Program Comprehension*, pp. 71–80.

[30] Retrieved date: [15, September,2019], online available at: Pseudocode.https://en.wikipedia.org/wiki/Pseudocode

[31] Emily Hill, (2010), "Integrating natural language and program structure information to improve software search and exploration", *University of Delaware*.

[32] Håkan Burden and Rogardt Heldal, (2011), "Natural language generation from class diagrams", *In Proceedings of the 8th International Workshop on Model-Driven Engineering, Verification and Validation, pp.8 ACM.*

[33] Preetha Chatterjee, Benjamin Gause, Hunter Hedinger, and Lori Pollock, (2017), "Extracting code segments and their descriptions from research articles", *In Mining Software Repositories (MSR), 2017IEEE/ACM14thInternational Conference* on, pp.91–101.

[34] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bah danau, and Yoshua Bengio, (2014), "On the properties of neural machine translation: Encoder-decoder approaches.", *arXiv preprint arXiv*:1409.1259.

[35] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. (2014), "Learning phrase representations using rnn encoder-decoder for statistical machine translation." *arXiv preprint arXiv:*1406.1078.

[36] Michael Denkowski and Alon Lavie. Meteor (2014) universal: *Language specific translation evaluation for any target language. In Proceedings of the ninth workshop on statistical machine translation*, pp.376–380.

[37] Lehrstuhl Iv Der Technischen and Daniela Steidl, (2012), "Quality analysis and assessment of source code comments" Martin Fowler, (2018), "Refactoring: improving the design of existing code", *Addison-Wesley Professional*.

[38] Sonia Haiduc, Jairo Aponte, and Andrian Marcus (2010), "Supporting program comprehension with source code summarization."*In Proceedings of the 32Nd ACM/IEEE International Conference on Software Engineering, ACM*, Vol.2, pp.223–226.

[39] Sonia Haiduc, Jairo Aponte, Laura Moreno, and Andrian Marcus (2010) "On the use of automated text summarization techniques for summarizing source code."*In Reverse Engineering (WCRE), 2010 17th Working Conference IEEE*, pp.35–44.

[40] Vincent J Hellendoorn and Premkumar Devanbu. Are deep neural networks the best choice for modeling source code? In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering, pages 763–773. ACM,2017.

[41] Retrieved date: [15, September,2019], online available at: https://data-flair.training/blogs/artificial-neural-networks-for-machine-learning/

[42] Dorsaf Haouari, Houari Sahraoui, and Philippe Langlais, (2011), "How good is your comment? a study of comments in java programs", *In Empirical Software Engineering and Measurement (ESEM), 2011International Symposium*, pp.137–146.

[43] P.Showmiya, V. Priya, (2016), "Optimised Summary Generation Using Genetic Algorithm" *International Journal of Advances in Computer and Electronics Engineering*, Vol.1, Issue.1, pp. 35 – 39 .

Authors Biography

Mr. Prashant Govardhan, in 2014 he joined the Department of Computer Science and Engineering at Priyadarshini Institute of Engineering and Technology, Nagpur, Maharashtra, India as an Assistant professor. He received his B. E. and M. Tech.in Computer Science and Engineering from Sant Gadge Baba Amravati University, Amravati. His research work interest includes Data Mining, Data Science, Distributed Systems and Machine Learning. His work has been documented in many publications. Currently he is working on Unmanned Comment Generation for Source Code.

Miss. Shweta Patil, perceiving Under Graduate Course of Engineering computer science and engineering from Priyadarshini Institute of Engineering and Technology Nagpur Maharashtra. Ongoing she is active in Unmanned Comment Generation for Source Code.

ISSN: 2456 - 3935
International Journal of Advances in Computer and Electronics Engineering
Volume 4, Issue 10, October 2019, pp. 1–9

Master Nachiket Gaurihar, perceiving Under Graduate Course of Engineering computer science and engineering from Priyadarshini Institute of Engineering and Technology Nagpur Maharashtra. Ongoing he is active in Unmanned Comment Generation for Source Code.

Master Sanket Adle, perceiving Under Graduate Course of Engineering computer science and engineering from Priyadarshini Institute of Engineering and Technology Nagpur Maharashtra. Ongoing he is active in Unmanned Comment Generation for Source Code.

Miss. Uzma Sheikh, perceiving Under Engineering computer science and engineering from Priyadarshini Institute of Engineering and Technology Nagpur, Maharashtra. Ongoing she is active in Unmanned Comment Generation for Source Code.

Cite this paper:

Prashant Govardhan, Shweta Patil, Nachiket Gaurihar, Sanket Adle, Uzma Sheikh "Study of Techniques and Algorithms Involved in Unmanned Comment Generation for Source Code", International Journal of Advances in Computer and Electronics Engineering, Vol. 4, No. 10, pp. 1-9 October 2019.