se2

저자 aviral goyal

제출일: 2021년 05월 24일 오후 12:18 (UTC+0900)
제출 아이디: 1591086586
파일명: SE_final_review_group_7_3.pdf (175.46K)
단어 수: 3132
글자 수: 16486

Project Title: Software project scheduling problem in the context of search-based software engineering

By:

Aviral Goyal (19BCE0883)

ArunKumar Mishra (19BCI0078)

Gokul Nath Reddy (19BCE0210)

Submitted to:

Prof. Deebak B D

Problem Addressed

Search Based Software Engineering (SBSE) is a technique that uses various metaheuristic techniques and applies them to search for the optimal solutions for problems that we may face during the course of a software engineering project. The solutions found using SBSE are optimized solutions, i.e., it will try to give the best possible solution to the problem that we may face. SBSE includes various algorithms like genetic algo., hill-climbing algo., simulated annealing algo. etc. which is applied to solve the scheduling problems and so we may often get confused and find it difficult to establish or decide which algorithm will be the best to use for the given scenario. As each algorithm has its own unique characteristics and benefits, it often becomes difficult for us to decide conclusively which algorithm is better and which is not. So, we have studied various research papers in this field, so we can determine the best algorithm amongst these.

Type of Research

We are doing qualitative research on the topic. Our main objective is to perform a comparative study of these algorithms and provide proof with definitive facts as well as prove which algorithm is best and also why it is the best. We are performing this research by going through various research papers in this field and doing a proper study of these papers.

Prior Research

The research paper that we are analysing is "**Hill-Climbing, Simulated Annealing and Genetic Algorithms: A Comparative Study and Application to the Mapping Problem, E-G.TALBI & T.MUNTEAN institut IMAG - Laboratoh de Genie Informatique (*) University of Grenoble** ". In this research paper, we have addressed the mapping problem and have executed

hil climbing algo., simulated anneling algo. and genetic algo. to resolve this problem and it also performs comparative study of these algorithms. The main motivation for this research was to perform a comparative study of these algorithms with respect to mapping problems, so that we can conclude the best algorithms to solve this problem as well as provide definitive proof of our conclusion. Also, another motivation is to provide an implementation of these algorithms so that they can be further used to solve other problems like shortest distance problem, salesman problem etc. Data visualization techniques and graphs are also used here so that a clear comparison can be performed of various factors affecting their performance so that a definitive conclusion can be reached. Various research papers have been cited in this paper that tell us about the prior research done by the authors of this paper. The research papers cited mainly discusses the mapping problem and explains it in detail, telling us what exactly the mapping problem is. It also includes papers that provide us with ways and methodology to solve the mapping problem theoretically. Some research papers cited provide us with practical knowledge of how these algorithms are used to solve this problem along with their practical implementation. The main shortcoming of prior research papers was that none of the papers performed a comparative study of these algorithms. Even if the best algorithm was mentioned that could be used to solve that problem, there was no proof or reason given as to why. This makes the research paper we are discussing crucial and important as it first implements these algorithms to solve the problem and then performs a proper comparative study to conclude the best algorithm to solve the problem and also give definitive proof as to why it is. "**Comparison of Genetic Algorithm and Hill Climbing for Shortest Path Optimization Mapping, Mona Fronita, Rahmat Gernowo, and Vincencius Gunawan, Master of Information System, School of Postgraduate Studies, Diponegoro University, Semarang – Indonesia 50242, Department of**

Physics, Faculty of Science and Mathematics, Diponegoro University, Semarang – Indonesia 50275", this is relevant research paper not cited here, this is relevant because it talks about the shortest distance mapping which is included in the mapping problem, as the shortest distance will only result in the least weighted path and hence, will be the least expensive path, making the route more suitable to use. Also, this paper compares two of the algorithms included in our base paper and reaches a similar conclusion as to what we have reached in the base paper.

Significance

In this project, our approach is more towards the analysis and mathematical computation, we are trying to quantify things down into numbers, we are trying to plot graphs and are trying to give the approximate time frame that the mentioned algorithms take when some generic algorithms are executed. Most of the papers that we came across gave a theoretical and vague approach into classifying the best algorithm. We are trying to make it align more on the computational side. Mainly, three SBSE algorithms have been used here to analyse some generic problems in SBSE like the mapping problem, The algorithms are hill-climbing, simulated annealing, and genetic algorithms. The significance also covers the fact that most of the papers that we came across compared either of the two algorithms out of the three stacked but in our project we are trying to compare them all. It is acknowledged that all of them have some used cases and based on that each algorithm has been measured on sets of test cases. Every algorithm present in SBSE is evaluated independently and is optimized according to the algorithm's parameters. The comparative study that has been done in this paper considers mainly two factors to check the quality of the algorithm, these factors are the qality of upcoming answers obtained and time taken, that is time complexity to reach that solution.

Introduction

Meta-heuristic optimization algorithms are of a higher importance for reaching solutions to various actual problems in Computer Science, Engineering and as in General Science and Medicine. These algorithms have a good range of applications in a variety of arenas like cost reduction, Artificial Intelligence , and medicine. By the term cost, one could simply say that this cost is related to the worth of a function of several independent variables. Most of the times, when handling engineering/science problems, we would like to level up the worth of a function so as to realize an optimum, efficient, or to maximise another parameter which increases with a decrease within the cost (the value of this function). These heuristic cost reduction algorithms work by finding the optimal values of the variables that the worth of the function (the "cost") is that the minimum.

The algorithms used in SBSE uses two techniques to find the globally optimised solution, these include to search for the solutions in unknown areas of search space and second is to use the knowledge of previously visited solutions to reach better solutions.

빠뜨렸거나 불필요한 관사

연결상의 오류가 있는 문장

Hill Climbing

This algorithm after randomly choosing a point within the search dimension. It examines this particular chosen solution that is in the vicinity of the original set; i.e. answers present in the search vicinity that are the same as the solution we have found or are as close as possible to it. If it is the case that a neighbouring set solution is found to be having better fitness, the search 'proceeds' to that new answer. After that, it begins searching in the vicinity of that new possible answer for better solutions, and this process continues, till the neighbour of current possible answer gives no new development. This kind of answer is known as locally optimal, and may or may not be the global optimal answer, and so the search is often done repeatedly in order to find even better solutions.

빠뜨렸거나 불필요한 관사

Simulated Annealing

Simulated annealing is a random-search technique. It uses an analogy between the way in which a metal cools and freezes. After that, it searches for a least optimal solution in many regular systems; thus it leads us to the basis of optimization techniques to analyse combinatorial problems. The analogy can be understood by employing a ball which will bounce over hills from one gap to another. The process initialises at high “temperature”, this enables the ball to perform lofty bounces allowing it to move across any hill and enter any gap it wants to, after it has gone through enough bounces. As the temperature reduces, the ball is not able to bounce as high as earlier, and so, it settles down in a relatively small range of gaps. The generating distribution from this algorithm creates all possible gaps or points that can be inspected. An acceptance distribution is additionally defined, this depends on the difference between the funct. value of created gap to be inspected and therefore the last saved lowest gap. The term acceptance distribution in Genetic Algorithm concludes probabilistically if we can stay in a new lower gap or loft out of the gap in a certain moment. The subsequent generating and acceptance distributions function are highly associated with temperature. SA is able to find the global optimum as and when the rate of cooling is controlled. However, this needs infinite time.

Genetic algorithm

GA is a meta-heuristic way that got its inspiration from the method of survival that belongs to the larger class of evolutionary algo.. GA allows mutation and crossover of two parent chromosomes of the population and places the new offspring in the advanced population, from which it gives the best answer. It repeats the process till it finds the best answer of a given func.. It randomly explores the population and finds answers that local search algorithms cannot, and it doesn't get to know the fitness func. derivative. A Genetic Algorithm withholds within itself, a

population of possible answers, and that at each step, chooses pairs of feasible solutions, it then merges them (crossover), and applies few irregular changes (analogous to mutation in biology). The algo. is predicated on thought of "survival of the fittest" in which the choice procedure is completely consistent with a fitness standard. This crossover or hybridisation is done with the certainty that two good answers, combined at certain ratios and within certain parameters, might give an even better solution to the problem.

Literature Review & Methodology:

Comparative Study of these algorithms as under

Analysis 1: The mapping problem

The mapping problem in SBSE is analysed using a function

n : Vp(no. of processes that are mapped) → Vt(no. of processors used in target architecture)

Each process is assigned to a processor. To associate values to each mapping and to make it easier for us to compare all the possible solutions, we define cost function.

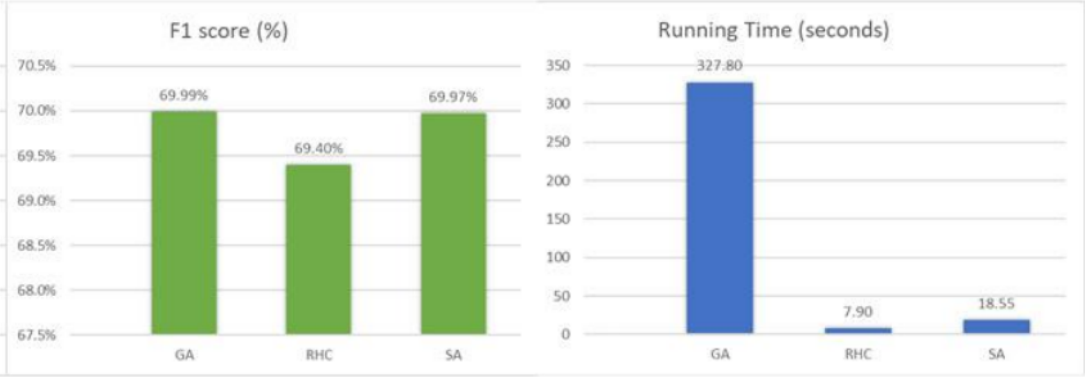
Two mapping criteria have been taken into account here:

- Firstly , in between processes, total communication cost is minimized.
- Secondly to minimize the imbalance in load across the entire system.The variance of the charge of diff. processors is a quantitative measure that is used to deal with such criteria.

Algorithm	**Solution**				CPU time(sec)
	Min	Max	Mean	Variance	
Hill Climbing	28	40	34.19	10.36	61
Simulated Annealing	16	25	18.89	10.09	1313
Genetic Algorithm	16	21	18.1	4.48	57

Analysis 2 : Replacing Backpropagation with these three algorithms in neural networks.

The first problem that we have applied the algorithms are the Neural Networks in Deep Learning on a graph based data structure**.** According to our observation, the finest backpropagation for the analysed system was 2 hidden layers, each with 10 different neural junctions, having a value of alpha as 0.064. Here we have used this neural network setup, but instead of backpropagation we have used the randomized optimization algorithms for the weights. We used fitness score, or the F1 score to measure the model performance with respect to each algorithm. The algorithms were tested with various parameters both tuned and untuned and the results are plotted as under.



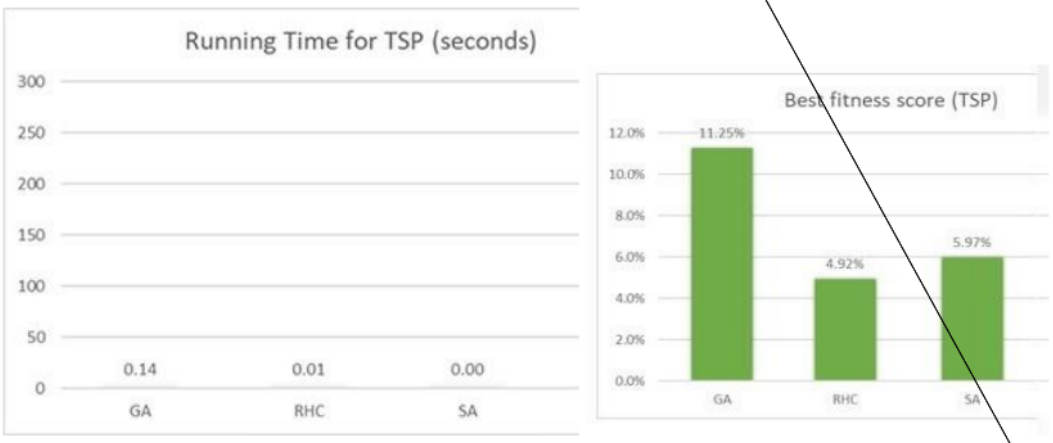
Analysis 3: Travelling Salesman Problem (TSP)

TSP is famous one NP-hard combinatorial optimisation prob.. It is imp. in operations research, thoretical engineering and CS. This problem answers the question, "**From collection of cities with some distance bet. each pair of them, which path is the shortest that lets us visit each city and at the end, return to the original city?**" . It is possible to represent TSP a graph problem by making the cities as the vertices of the graph, the paths as the edges, and edges weight as the path's distance. It is a shortest path problem that starts from one particular vertex and traverses back at the same peak after visiting all other points only once. Assuming 100 cities

(N=100), we test the three algorithms with various parameters and plot them according to our findings in papers. Each run had 3,000 iterations. For SA, the best parameter was Temperature=1e10 and CE=0.55. For Genetic Algorithm, the hypertuned values were Mate=50, Mutate=10, Pop=100.

연결상의 오류가 있는 문장

Results in graphical representation



Contributions

The idea for the study was conceived by all authors, and the research was planned by all of them. We were in charge of creating and evaluating the checklist. The findings were considered by all authors, and they all contributed to the final publication.

수동태

수동태

Here we got to know that these three algorithms are good only in some aspects such as software testing, reliability, cost, effort estimation, resource allocations, speed and computation power etc.. The researchers compared the search approaches Genetic algorithm, simulated annealing, and hill climbing in the study. Because it focuses on solutions and fitness, the Genetic Algorithm has been proven to be the best algorithm. In comparison to hill climbing and simulated annealing, they are simple to build and understand. Combinatorial problems are best solved with genetic algorithms.

쉼표 빠뜨림

쉼표 빠뜨림

Key Issues:

Hill Climbing

Hill climbing algorithm is very useful in situations when we want fast outputs. So, in such case hill climbing is preferred over simulated annealing and genetic algorithm as it may provide us with only local solutions but the computation power and the time needed to obtain that solution is way less than needed by other two algorithms. For convex issues, hill climbing will find optimal solutions; for other issues, it will find only local optima (solutions that cannot be improved by any neighbouring configurations), which aren't always the best possible solution (the global optimum) out of all feasible solutions.

Simulated Annealing

Simulated Annealing only follows one solution in a space of many possible solutions, and at each iteration, it decides whether to move to a neighbouring solution or stay in the current one based on probability. The accuracy of the statistics utilised in SA implementation can have a big impact on the quality of the final product.

Genetic Algorithm

Genetic algorithms are solution optimizing methods that are based on population genetics. In this method, the individual solutions are represented, evaluated, and combined together at each step. They are able to escape local optima better than other algorithms because of the recombination steps. They're often used in engineering to optimize part design and other design aspects. However, they take a long time to reach the optimal solution, and also its computational aspects can often limit its application to problems as an optimization strategy.

Further Research

As we said earlier, we worked and collected some important data for comparing the three algorithms in different aspects. SBSE has been successfully used in programme optimization, which is the process of changing a piece of software to make it more efficient in terms of speed

and resource usage. Successful SBSE applications in industry are largely found in software testing, where the ability to automatically create random test inputs for detecting problems on a large-scale appeal to businesses. Hence, as SBSE algorithms are now becoming more and more common, it has become necessary to improve this field even further and find even better algorithms.

The main area in which further research can be done is in the field of developing new algorithms and also analysis of the new algorithms that are being developed. For instance, nowadays many research papers are focussing on hybrid algorithms in which they combine two of the algorithms and form a new algorithm from that. The resultant hybrid algorithms have similar properties as the combining algorithms and inherits best features of both, making them better than the existing algorithms. Another field of further research can be developing new and better ways of performing the comparative analysis of these algorithms including the further new algorithms that are currently under development. No, I don't think the article has called for research in any specific issue, but, seeing the trends of SBSE we can infer that new and better algorithms will be developed. Also, research can be done in improving the existing algorithms instead of developing new ones. Some limitations of the article is that it considers only one situation and presents the analysis according to that but not every problem will be similar and each problem can have different conditions to fulfil and so, the preferred algorithm will also change like if we want quick results instead of best results then we will opt for hill climbing algorithms. But, this is not considered in the article. The article also does not consider the possibility of development of any new and better algorithms than the existing ones.

References

Base paper: Hill-Climbing, Simulated Annealing and Genetic Algorithms: A Comparative Study and Application to the Mapping Problem E-G.TALBI & T.MUNTEAN institut IMAG - Laboratoh de Genie Informatique (*) University of Grenoble

Reference papers:

[1]S.H.Bokhari. "On the mapping problem", IEEE Trans. on Comp., Vol.C-30, pp.207-214, Mar 1981.

[2]D.R.Greening, "Parallel simulated annealing techniques", Physica D: Nonlinear phenomena, Vo1.42. pp.293- 306, 1990.

[3]J.H.Holland, "Adaptation in natural and artificial systems", AM Arbor: Univ. of Michigan Press, 1975.

[4]D.S .Johnson, C. H .Papadimitriou, M.Yannakakis, "How easy is local search ?", Proc. Annual Symp. of Foundation of Computer Science, pp.39- 41, 1985.

[5]Comparison of Genetic Algorithm and Hill Climbing for Shortest Path Optimization Mapping, Mona Fronita, Rahmat Gernowo, and Vincencius Gunawan, Master of Information System, School of Postgraduate Studies, Diponegoro University, Semarang – Indonesia 50242, Department of Physics, Faculty of Science and Mathematics, Diponegoro University, Semarang – Indonesia 50275

[6]Search Based Software Engineering: A Comprehensive Analysis and Review of Trends Techniques and Applications, Mark Harman, S. Afshin Mansouri and Yuanyuan Zhang, April 9, 2009, Technical Report TR-09-03

[7]Search Based Software Engineering: Techniques, Taxonomy, Tutorial Mark Harman1 , Phil McMinn2 , Jerffeson Teixeira de Souza3 , and Shin Yoo1 1 University College London, UK 2 University of Sheffield, UK 3 State University of Cear´a, Brazil

[8]Evolution of a salesman: A complete genetic algorithm tutorial for Python, Eric Stoltz, Jul 17, 2018

[9]Neural Networks using Genetic Algorithms Richa Mahajan Student, Guru Nanak Dev University, Amritsar Gaganpreet Kaur Student, Guru Nanak Dev University, Amritsar

se2

독창성 보고서

20%	16%	11%	14%
유사성 지표	인터넷 출처	출판물	학생 보고서

일차 출처

1	E.-G. Talbi, T. Muntean. "Hill-climbing, simulated annealing and genetic algorithms: a comparative study and application to the mapping problem", [1993] Proceedings of the Twenty-sixth Hawaii International Conference on System Sciences, 1993 출판물	4%
2	Mona Fronita, Rahmat Gernowo, Vincencius Gunawan. "Comparison of Genetic Algorithm and Hill Climbing for Shortest Path Optimization Mapping", E3S Web of Conferences, 2018 출판물	3%
3	digitalcommons.mtu.edu 인터넷 출처	3%
4	medium.com 인터넷 출처	2%
5	Submitted to University of London External System 학생 보고서	1%

6	Submitted to Georgia Institute of Technology Main Campus 학생 보고서	1 %
7	docplayer.net 인터넷 출처	1 %
8	Submitted to University of Cambridge 학생 보고서	1 %
9	Submitted to Birla Institute of Technology and Science Pilani 학생 보고서	1 %
10	Submitted to Universiti Teknologi MARA 학생 보고서	1 %
11	www.coursehero.com 인터넷 출처	1 %
12	McMullen, Shearer. "Prime Implicants, Minimum Covers, and the Complexity of Logic Simplification", IEEE Transactions on Computers, 1986 출판물	1 %
13	towardsdatascience.com 인터넷 출처	<1 %
14	coinse.kaist.ac.kr 인터넷 출처	<1 %
15	researchr.org 인터넷 출처	<1 %

16	www0.cs.ucl.ac.uk 인터넷 출처	<1 %
17	Andrea Arcuri. "It really does matter how you normalize the branch distance in search-based software testing", Software Testing, Verification and Reliability, 2013 출판물	<1 %
18	Meta-Heuristics, 1996. 출판물	<1 %

인용문 제외	꺼짐	일치 제외	꺼짐
참고 문헌 제외	꺼짐		

se2

페이지 1

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

페이지 2

빠뜨렸거나 불필요한 관사 여기에 이 관사가 불필요할 수도 있다.

빠뜨렸거나 불필요한 관사 여기에 이 관사가 불필요할 수도 있다.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.

페이지 3

불완전한 문장 혹은 쉼표 빠뜨림 이 문장은 불완전한 문장이거나 잘못된 구두법이 쓰였을 수 있다. 문장을 다시 읽고 올바른 구두점과 주어와 동사가 있는 독립절이 있는 지 확인하시오.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

주어 동사 일치 문장의 주어와 동사가 일치하지 않을 수 있다. 문장을 다시 읽고 주어와 동사를 주의해서 읽도록 하라.

빠뜨렸거나 불필요한 관사 여기에 이 관사가 불필요할 수도 있다.

수동태 이 문장에 수동태를 썼다. 능동태를 쓰는 것이 나을 수도 있다.

페이지 4

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

연결상의 오류가 있는 문장 이 문장은 연결상의 오류가 있는 문장일 수 있다. 연결사나, 구두점을 붙이거나 두 문장으로 만들 필요가 있다.

전치사 오류 잘못된 전치사를 사용하였을 수 있다.

빠뜨렸거나 불필요한 관사 여기에 이 관사가 불필요할 수도 있다.

연결상의 오류가 있는 문장 이 문장은 연결상의 오류가 있는 문장일 수 있다. 연결사나, 구두점을 붙이거나 두 문장으로 만들 필요가 있다.

주어 동사 일치 문장의 주어와 동사가 일치하지 않을 수 있다. 문장을 다시 읽고 주어와 동사를 주의해서 읽도록 하라.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.

잘못된 형태의 단어 이 단어의 잘못된 형태를 썼을 수 있다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.

와전된 문장 이 문장은 몇 가지 문법이나 철자법 상의 오류가 있어서 혼동스럽다.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

의문 부호 빠뜨림 의문문의 끝에 의문부호를 빠뜨리지 않게 주의한다.

전치사 오류 잘못된 전치사를 사용하였을 수 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

고유 명사에 대문자 쓰기 이 단어가 고유 명사이면 대문자로 시작해야 한다.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

불완전한 문장 혹은 쉼표 빠뜨림 이 문장은 불완전한 문장이거나 잘못된 구두법이 쓰였을 수 있다. 문장을 다시 읽고 올바른 구두점과 주어와 동사가 있는 독립절이 있는 지 확인하시오.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

종료 부호 빠뜨림 문장 끝에 구두점이 빠졌다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

불완전한 문장 혹은 쉼표 빠뜨림 이 문장은 불완전한 문장이거나 잘못된 구두법이 쓰였을 수 있다. 문장을 다시 읽고 올바른 구두점과 주어와 동사가 있는 독립절이 있는 지 확인하시오.

주어 동사 일치 문장의 주어와 동사가 일치하지 않을 수 있다. 문장을 다시 읽고 주어와 동사를 주의해서 읽도록 하라.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

불완전한 문장 혹은 쉼표 빠뜨림 이 문장은 불완전한 문장이거나 잘못된 구두법이 쓰였을 수 있다. 문장을 다시 읽고 올바른 구두점과 주어와 동사가 있는 독립절이 있는 지 확인하시오.

대문자로 시작되지 않은 문장 모든 문장을 대문자로 시작하도록 주의한다.

교정할 것! 문장의 이 부분에는 문장을 이해하기 힘들게 만드는 오류나 철자법의 오류가 있다

연결상의 오류가 있는 문장 이 문장은 연결상의 오류가 있는 문장일 수 있다. 연결사나, 구두점을 붙이거나 두 문장으로 만들 필요가 있다.

수동태 이 문장에 수동태를 썼다. 능동태를 쓰는 것이 나을 수도 있다.

수동태 이 문장에 수동태를 썼다. 능동태를 쓰는 것이 나을 수도 있다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

잘못된 형태의 단어 이 단어의 잘못된 형태를 썼을 수 있다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

수동태 이 문장에 수동태를 썼다. 능동태를 쓰는 것이 나을 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사가 필요할 수도 있다.

빠뜨렸거나 불필요한 관사 여기에 이 관사가 불필요할 수도 있다.

혼동되는 단어 이 문장에 **to**를 썼음. 대신 **two**를 써야할 수도 있다.

쉼표 빠뜨림 이 단어 뒤에 쉼표를 써야 할 수도 있다.

빠뜨렸거나 불필요한 관사 이 단어 앞에 관사를 써야할 수도 있다. 관사 **the**를 쓰는 것을 고려하라.