Spaced Repetition for Slow Learners

Devansh P. Shah, Nikhil M. Jagtap, Shloka S. Shah and Anant V. Nimkar
Department of Computer Engineering
Sardar Patel Institute of Technology
Mumbai, India
{devansh.shah, nikhil.jagtap, shloka.shah, anant_nimkar}@spit.ac.in

2020 IEEE Bombay Section Signature Conference (IBSSC) | 978-1-7281-8993-2/20/$31.00 ©2020 IEEE | DOI: 10.1109/IBSSC51096.2020.9332189

***Abstract*—Spaced Repetition has proven to be an effective way in learning and memorizing complex topics. An algorithm 'Spaced Repetition for Slow Learners' (SRSL) is described to schedule repetitions which eventually adapts to the capacity of the learner. SRSL computes the score of learners for a particular assessment based on factors such as response time, difficulty and dependency of questions. The exponential forgetting curve model is the memory model assumed by SRSL. Based on this algorithm, a model has been proposed with experimental analysis of the same. Further, comparison of SRSL and the Leitner System demonstrates the adaptability of the algorithm to the learning curve of the learner.**

***Index Terms*—Spaced Repetition, Slow learners, Forgetting curve, Exponential decay, Retention capacity**

I. Introduction

The task of learning anything often involves memorizing some or more parts of it. This task of memorizing is small in subjects which have more focus on practical applications like mathematics (e.g. remembering formulae) and is large in subjects like language (e.g. remembering genders of words in German or script in Japanese). There are two major methods which can be usually applied for any learning task namely, massed learning and spaced repetition. Massed Learning is when one tries to cram all the learning in a certain study period and then move on to the next topic. Spaced Repetition refers to creating time intervals between various learning sessions. But, it is found that the retention capacity is very low with massed learning compared to spaced repetition. Most of the things learnt via massed learning are forgotten quickly.

Although there is no strict rule for memorizing and learning, a generic approach is to repeat the learning procedure for specific times. Various approaches have been proposed to schedule these repetitions. The most notable of them being Spaced Repetition. Almost every algorithm in Spaced Repetition is a simple function or set of functions with trivial parameters like the correctness of the response. The first ever model proposed which explained the forgetting curve of the human mind was by Ebbinghaus [1] and since then spaced repetition has been applied to various fields like language learning, remembering strong passwords and so on. Various models like Multiscale Context Model [2], the Pimsleur Method [3], Half Life Regression by Duolingo [4] for language learning and the MEMORIZE algorithm [5] have been proposed which make use of spaced repetition.

The learning pattern among slow learners is a bit different from that of normal learners. A general trait found among them is that they cannot do multifaceted or complex problems and require timely revisions of the same. Also, they face difficulty in understanding the similarity between questions. Transfer learning is another problem faced by the slow learners. Most existing models do not adapt to the learning rate of the students. Considering some of these problems among the slow learners, there is a need of a Spaced Repetition algorithm optimized specifically for the slow learners.

The proposed algorithm 'Spaced Repetition For Slow Learners' (SRSL) consists of a number of parameters which can be easily fine tuned by the test setter. The number of times a topic should be repeated for effective memorization depends on various factors. It depends on the learning capacity of the learner, his/her interest in the topic, difficulty of the topic, dependency of the topic on another topic, and so on. There is a lack of an algorithm which adapts to the learning of students. The scheduling pattern generated by SRSL will be different for different type of learners and topics to learn. In SRSL, the score for the response by a student is calculated based on the correctness of the answer, the difficulty of the question (decided by the test setter) and the response time of the student. The equations for the same are discussed in detail in further sections.

The paper is organised as follows. In section II, there is a discussion about the current existing models and how spaced repetition has been used in various learning tasks. In section III, a top level view of the SRSL algorithm is given. Next there are subsections describing each function in detail like the score calculation, calculating the decay rate, updating the review queue, calculation of probability, discussing the dependency of questions, the criteria for the test to end and interpretation of parameters. In section IV, results and analysis of SRSL in the form of various graphs considering different test scenarios are shown. Section V provides with a conclusion to the paper.

II. Literature Review

The history of Spaced Repetition traces back to the nineteenth century, when Ebbinghaus [1] hypothesized that the rate at which humans forget information increases exponentially with time. However, if the information is revised repeatedly, it tends to be forgotten at lower rate, which also decreases with each revision. His experiments with himself as a subject (published as U¨ ber das Geda¨chtnis) are well known as the "forgetting curve".

The research associated with the cognitive psychology of memory and learning articulated that successfully recalling an item from memory after a delay is more effective than recalling it immediately. A successful recall from memory generates a better retention of the target item. These two findings are contradictory. The former one recommends longer intervals and later one recommends smaller intervals. Hence, it can be inferred that in optimal methodology, an item should be reviewed when they are just about to be forgotten. In other words, the revisions must be scheduled so that the maximum remembering has occurred. The question 'how far these repetition should be spaced' is addressed in multiple works since the last century. The Lag Effect in various studies [6], [7] showed that people learn and recall better if the spacing increases gradually. Through the course of time, various algorithms have been proposed to estimate the optimal revision schedule. Two prominent examples include the "Leitner's System" and the "Pimsleur Method".

The Pimsleur Method [3] uses a predefined and fixed schedule over which the information (in audio format) was repeated. The Leitner's System was more adaptive than Pimsleur's, since the spacing intervals in it can increase and decrease depending on the performance of the student. Reddy et al. in their work [8] formulated the mathematical model for Leitner's System, known as Leitner Queue Network. In the proposed methodology, the authors assumed the information instances to follow a Poisson process with some arrival rate. The problem of optimization was framed as having static, and not dynamic parameters, in their work. Sean H K Kang in his work [9] showed how the spacing of revisions depends on the learner's speed and abilities. The paper also highlighted how spaced repeated learning helps students to use the knowledge in similar related problems.

The major breakthrough in the spaced repetition in second language learning was marked by A Trainable Spaced Repetition Model for Language Learning [4]. In this work, the authors proposed the Half Life Regression (HLR) as a model for spaced repetition. Based on the feature vectors of students of Duolingo (online language learning platform) and lexeme tagger, HLR too changed exponentially like the Ebbinghaus model [1]. HLR is a generalisation of the Pimsleur Method [3] and the Leitner System. This was successfully used in the experiment on Duolingo where students using HLR Spaced Repetition recalled words from foreign languages than the students using massed repetition. In our opinion, this use of Spaced Repetition in Duolingo is the largest known application to us. The recent yet the most important work in Spaced Repetition is Enhancing Human Learning via Spaced Repetition Optimisation [5] where authors proposed MEMORIZE algorithm which changes review intensities dynamically.

Various experiments verified the practicality of the spaced repetition method in various learning objectives. A study [10] suggested that words remembered by students by the Spaced Repetition method is more efficient and long term than the Massed method. Other studies also support spaced repetition practice in mathematics problems (Rohrer Taylor, 2006) [11], ecology lessons (Gluckman et al., 2014) and meteorology (Kapler, et al., 2015) as well.

Another spaced repetition model titled Multiscale Context Model (MCM) [2] could predict how a study schedule affects the retention of specific learning material. Use of spaced repetition in complex tasks such as remembering strong passwords, is heavily discussed in [12] and [13]. Even in mobile games, the Spaced Repetition is used for memorization [14] [15]. The learning and remembering in these mobile phone based games is through simple tasks. These simple tasks generally decide the spacing of repetitions and try to optimise it. A recent paper studying the applications of spaced repetition by Yan et al. [16] showed the use of spaced repetition for memorising GRE words. The authors experimented on students with uniform and spaced repetition concluding spaced repetition providing better and more memorising and learning.

Thus, the use of spaced repetition is not only restricted to flashcards or foreign language, but has widespread applications in generic learning objectives. Though these studies clearly show that Spaced Repetition improves memorization, one thing is very common among them. The users or people remembering something using Spaced Repetition are normal learners. Their learning curve is very much similar. For slow learners, these ideas will not be as effective as they are for normal learners. Thus, there is a need of Spaced Repetition Model which is optimised for slow learners.

III. Spaced Repetition For Slow Learners

There is a lack of a spaced repetition system which schedules review events purely based on the student's response/activity while being on the lesson. We present a system where we assume the Ebbinghaus [1] memory model of the human brain, i.e. retention levels decay exponentially. Instead of giving a model for each subject for each student, we are focusing on the understanding of a particular topic by the student. SRSL can work on questions of individual topics, or combination of questions in form of topics, thus providing various levels of granularity. For the further explanation in this section, we are assuming that there are different questions of the same topic of a particular subject.

We propose a general framework for spaced repetition which allows for flexibility in terms of scoring of questions, how often review events occur, etc. Then we discuss the specific functions and their parameters that we have chosen.

Suppose there are n different questions q_i, where $1 \leq i \leq n$. Each question has a some difficulty level d_i, where d_i can have values in some range, for example 1 to 10 (not necessarily discrete). As soon as the student answers a question, a score is calculated based on various parameters through a

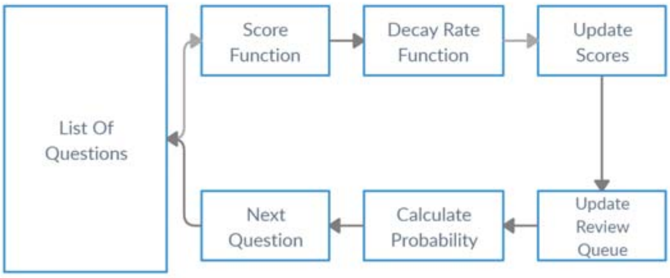

Fig. 1. Overview of Spaced Repetition For Slow Learners (SRSL)

'score' function. Based on the score obtained, a decay rate is calculated through a 'decay rate' function. The 'decay rate' function can also have additional parameters which can help to further accurately model the student. Once the scores and decay rates are calculated, we update the existing scores and decay rates. Once the scores are updated, they start to decay according to the decay rate calculated. We then insert all problems whose score is below a "threshold" score into the "review queue". The review queue contains all the problems who are to be considered for a review event. Next we calculate a probability that a review event occurs. This step determines the next question to be displayed to the student, and the whole process repeats. The process keeps on repeating until a criteria is met, which marks the end of the test. The criteria should include the basic condition of review queue being empty and all scores to be above threshold.

A. Score Calculation

In this section we address the calculation of the score for each question and topic. For every question a student answers, a score s_i is assigned to the question. This score is a measure of how well the student knows/understood/can calculate/can recall the answer to the question. This score s_i can be calculated for each question based on a number of parameters. We have considered time taken to answer the question (response time) t_i as a parameter to calculate the score. The very basic and rudiment parameter will be the correctness of the answer given by the student. The correct answer by the student clearly indicates that the student is aware of the concept tested in the question. In contrast, the wrong answer could be due to many possibilities. Loose understanding of the concept or poor recall being few of them. Thus the score will depend whether the answer was correct or not.

Now, our concern is how good or how bad the student is in that particular topic. That's why we have questions of different difficulties. Easy questions test primary knowledge of the student about that topic. More difficult and tricky questions will test how well the student has understood the concepts as well as how well he/she can recall and apply it when required. Thus, higher difficulty represents better understanding of the topic and less repetitions are needed. Hence, the score is directly proportional to the difficulty of the question.

$$s_{\mathrm{i}} \ \alpha \ d_{\mathrm{i}} \tag{1}$$

In conjunction with correctness, we also consider the timeliness of the answer, whether the answer is given quickly or after some time. If the student answers a question quickly and correctly, the corresponding score for that question should be high, indicative of good understanding and memory for that question and concept. Conversely, if the student answers a question quickly but the answer is incorrect, it indicates that the student was confident but wrong about the answer. Such cases should be penalized the maximum so that student will revisit the topic as soon as possible. If a student answers a question correctly but after a long time, it may be due to poor recollection or low confidence. Thus the score s_i of a question is inversely proportional to the response time.

$$s_{\mathrm{i}} \ \alpha \ \frac{1}{t_{\mathrm{i}}} \tag{2}$$

In cases where the students answers the questions wrongly, we want the student to revise that particular question again as soon as possible. Therefore in such cases, the score will be calculated such that it is directly proportional to the difficulty of question and the response time of the student.

$$s_{\mathrm{i}} \ \alpha \ d_{\mathrm{i}} * t_{\mathrm{i}} \tag{3}$$

B. Decay Rate

To estimate the decay factor λ_i , we have already calculated the score of the topic for the student. A high score indicates that the student is aware of the content of the topic and needs very less repetitions. Hence, the repetitions of such students can spread more i.e. the score should decay slowly and the decay rate should be low. Similarly, the decay rate should be high for low scores. Thus, decay rate is inversely proportional to the score.

$$\lambda_{\mathrm{i}} \ \alpha \ \frac{1}{s_{\mathrm{i}}} \tag{4}$$

The decay rate should also change with each review event or repetition. The rate should decrease with each review event, as the student will have memorized it better. This also encourages eventual completion of the test, avoiding an infinite loop of questions. Let N_i represent the number of review events already occurred for the question.

$$\lambda_{\mathrm{i}} \ \alpha \ \frac{1}{N_{\mathrm{i}}} \tag{5}$$

With time, the score s_i obtained for the topic will decay with decay rate λ_i in an exponential fashion, mimicking the exponential forgetting curve. Here, s_i represents the score of the last occurence of the question.

$$s_{\mathrm{i}}(t) = s_{\mathrm{i}} e^{-\lambda_{\mathrm{i}} t} \tag{6}$$

With equation 6, we can easily calculate the time instance t_T when the student is just about to forget the contents of the topic. For threshold score T, we can calculate the value of t_T

and schedule the repetition.

While updating decay rates, we should be careful to avoid sudden changes which may abruptly change the repetition schedule of the student. To counter this, we update the decay rate through the following equation, where α is a constant.

$$\lambda_i(t+1) = \alpha\lambda_i(t) + (1-\alpha)\lambda_i(t+1) \quad (7)$$

C. Updating Review Queue

The review queue contains all the problems whose current score is less than its threshold score. We have assumed the review queue to be a priority queue, since there must exist an order to determine which question to review first. There can be many ways to determine the priority according to various strategies. We have considered the priority of a problem to be equal to the difference between its current score and its threshold value. At every time instant, the top element of the review queue is the problem to be considered for review, since it is the problem most likely to be forgotten till that time.

D. Calculation Of Probability

To prevent closed loops, which are cases where a student keeps answering a fixed number of questions and never really progresses in the test, we introduce a probability term. This represents the probability that a new question be introduced to the student instead of a review event. In some sense, this probability term can be used to control the speed of the test. In the initial stages of the test, the probability is kept high, so as to encourage introduction of new questions. The probability is slowly decreased to 0.5 at the half-way mark of the test. Thus in the second part of the test, there is an equal chance of a new question being introduced and a review event.

E. Dependency Of Questions

Another trait frequently found in slow learners is their inability to relate similar topics. In order to address this issue, we propose a directed acyclic graph of questions (not necessarily fully connected). Two problems q_1 and q_2 are connected by an edge (from q_1 to q_2) in the graph if q_1 is a prerequisite of q_2. The creator of the test also assigns weights to the edges (on a scale of 1 to 10) of the graph, which represent how closely the two questions are associated.

Whenever a student correctly answers a question, we reward the student by increasing the scores of all prerequisite questions of that particular question. The increase in the scores of prerequisite questions is proportional to the weight of the edge between the two questions. Thus this acts as a positive reinforcement and also ensures that the prerequisite question does not get reviewed if one keeps answering related questions correctly. Similarly, when a question is answered incorrectly by a student, we impose a penalty (proportional to the weight of the edge between two questions) to all the questions who have the current question as a prerequisite. This attempts to model the situation where a student answered a difficult question correctly just by chance and failed to answer its prerequisite questions. The whole prerequisite setup ensures that the test is connected and not just answering individual questions.

F. Criteria To End Test

As discussed earlier, the criteria to end the test is an important factor. We have considered the following criteria:
1) scores of all questions are above threshold.
2) no question is left in the review queue. We assign an overall score to the student based on his responses during the entire test. The overall score can be the average of the overall score at all time instants or it can be something which can better capture the overall performance of the student.

G. Interpretation Of Parameters

There are many independent parts of SRSL as described above, which leaves for enormous room for tweaking its behaviour. In this section we discuss some possible interpretations of parameters and functions.

The scoring function plays a very important part in determining the difficulty of the entire test. One can include more parameters which enables the system to better capture the student's response. One can also induce an inherent bias in the function by penalizing wrong answers more than awarding correct answers. This makes it difficult for the student to score in general. To overcome this, the student will have to answer the question in less time to get the same score, leading to more memorization. The reverse too might be applied in some cases. We can even make the proportionality constants change according to the responses of the student, making the above process dynamic.

The probability factor described in section D was introduced to have some randomness in the system, which avoids infinite loops. However the probability can also be interpreted as a strategy for the test. Low probability of occurrence of a new question leads to more review events happening. This can be a valid strategy if the test setter wants the student to fully memorize all attempted questions before attempting new questions. The converse is also true.

The proportionality constants in all the equations can help to control the scale of the review time. By changing those constants we can change the scale of the algorithm from seconds/minutes to days/months.

IV. Results and Discussions

Scheduling and performance are two major aspects of SRSL. The schedules generated by SRSL and the schedule generated by Leitner system are compared with each other. A few model cases are considered and the review times generated by both systems are compared. Next a simulation is done while modelling an average student by some assumptions discussed below. We plot the length of review queue with time while varying the number of problems in the test and the response time of the student. This gives an insight into the behaviour of the algorithm. Finally, we look at the result of one simulation, where we plot the score and decay rate of a single question with time.

A. *Comparison of SRSL with Leitner System*

When trying to compare the scheduling of two systems, it is helpful to compare the review times generated for a single question. Here, the Leitner system is assumed to have review times of 2^n minutes, where 'n' represents the number of review events. For sake of simplicity, all answers are answered correctly. Three cases are considered for comparision between two systems. Case 1 models a student who answers the question in the same amount of time (10 seconds) in every review event. Case 2 models a student whose response time decreases with each review event by 1 second, indicating that the student recollects the answer better. Case 3 models a student whose response time increases with each review event by 1 second, indicating that the student is unable to recollect the answer as quickly as he/she could do before. Case 3 represents a student who is relatively slower than the students modeled by Case 1 and Case 2. The question has a difficulty of 10 with a threshold score of 50. The proportionality constant of eq (1) and eq (2) combined is set to 100. Similarly, the proportionality constant of eq (4) and eq (5) combined is set to 1.15. Using these values of constants, we plot review time generated by both systems.

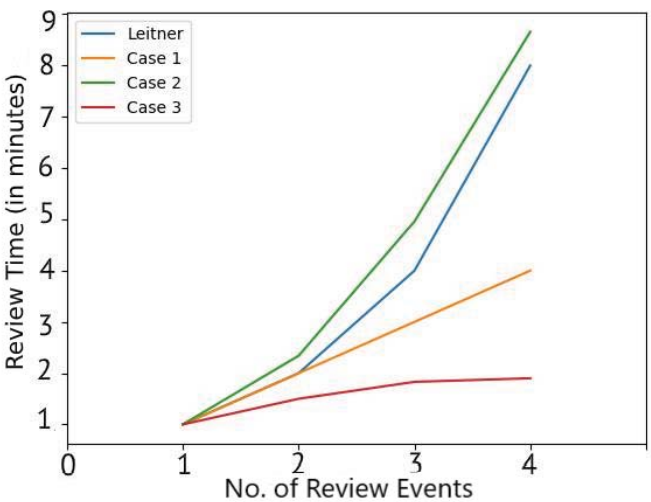

Fig. 2. Review times generated by Leitner system and SRSL

From the Fig 2, it is clear that Case 2 closely resembles the Leitner curve. Thus if a student memorizes the answer to the question in a better way than before, the review time generated by both the systems are very close to each other. However since the Leitner system does not consider response time of a question, the results of Case 1 and Case 3 vary. In Case 1, the review time generated by our system is linear in shape. Whereas in Case 3, we can see the review time is very less than the review times proposed by Leitner system. Due to the fact that the student is taking more time to answer, the review times were generated to be lower than normal, so that the question appears for review in lesser time than normal. We calculated the review times for different set of constants and found the general pattern to be the same, the only exception being the case where the score never decays below the threshold score. The difference between the two systems can be clearly seen as Leitner system provides fixed schedule while SRSL adapts to the responses of the student.

In the next sub section we demonstrate the behaviour of the algorithm in different cases through simulations. For the purpose of experimenting, we have set the difficulty of each question to be 5. The threshold score for each question is set to 20.

B. *Result of simulation*

Fig 3 shows the plot of the length of the review queue vs time for various number of problems. The problems were assumed to have no dependency between them. For this plot, we simulated the activity of a student while taking the test. The student answers each question in 10 seconds with 70% chance of it being correct. In the plot, it can be seen that the number of questions in the review queue increases in the beginning of the test and gradually decreases till the review queue is empty. One more thing to be observed is the time taken for each simulation test to end. Even between individual runs of the simulation there was some variation in the time taken to complete the test due to the inherent randomness in the algorithm. The graph shown here is smoothened using the lfilter function in scipy package.

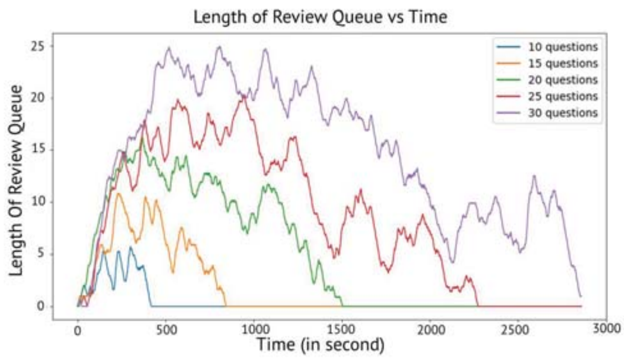

Fig. 3. Length Of Review Queue Vs Time (in seconds)

Fig 4 shows the length of the review queue vs time for various response times of the student. The setup of the test was similar to the one described above. Through literature available, we hope that the student memorizes an item with more number of review events. This assumption is modelled here. We increase the probability of answering correctly from 0.7 to 1 as the number of review events increases. We can see in Fig 4 that the distribution of length of review queue get wider as the response time increase.

In Fig 5, we can see the result of another model simulation. Here, the student is modelled to answer correctly every time within a random amount of time. The random time was

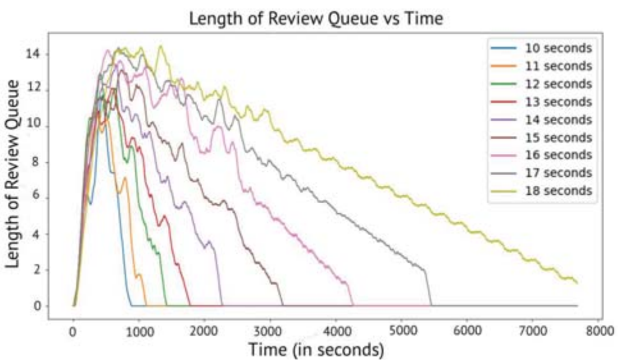

Fig. 4. Length Of Review Queue Vs Time (in seconds)

sampled from a pseudo-random number generator and was in between the range 10 seconds and 15 seconds. There were 10 total questions with no dependencies between them, out of which results for one of them are shown in Fig 4. We can clearly see the scores decaying below the threshold score, depicted in red. We can also see the decay rate for that question at the exact instant. Notice that after the first review event, the obtained score is less than the previous score, which leads to an increase of the decay rate. The score obtained during the next review event is similar to the 1st review event. However the decay rate decreases, which is worth noting. This is in accordance with equation 5.

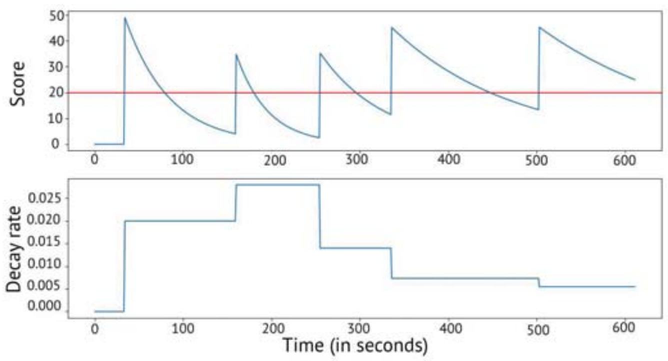

Fig. 5. Score and Decay Rate Vs Time (in seconds)

V. Conclusion

The proposed algorithm 'Spaced Repetition For Slow Learners' (SRSL) was experimented on different scenarios. The algorithm is shown to better adapt than the Leitner System to the responses of the student. It is verified that adaptive spaced repetition is more effective than fixed rule based spaced repetition. SRSL can adapt itself with the pace and memorizing capacity of the learner. On changing the various constants and functions in the algorithm, the system can be tailored to suit specific as well as generic use cases. Education is going more towards application of knowledge. The proposed system can play a huge role where memory dependency is still present, like learning a new language. SRSL can be effectively implemented in multiple choice type quizzes of online learning platforms further enhancing the learning experience of the student.

References

[1] H. Ebbinghaus, "Memory: A contribution to experimental psychology," *Annals of Neurosciences*, vol. 20, no. 4, Oct. 2013.

[2] H. Pashler, N. Cepeda, R. V. Lindsey, E. Vul, and M. C. Mozer, "Predicting the optimal spacing of study: A multiscale context model of memory," in *Advances in Neural Information Processing Systems 22*. Curran Associates, Inc., 2009, pp. 1321–1329.

[3] P. Pimsleur, "A memory schedule," *The Modern Language Journal*, vol. 51, no. 2, pp. 73–75, 1967.

[4] B. Settles and B. Meeder, "A trainable spaced repetition model for language learning," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Aug. 2016, pp. 1848–1858.

[5] B. Tabibian, U. Upadhyay, A. De, A. Zarezade, B. Schölkopf, and M. Gomez-Rodriguez, "Enhancing human learning via spaced repetition optimization," *Proceedings of the National Academy of Sciences*, vol. 116, no. 10, pp. 3988–3993, 2019.

[6] A. W. Melton, "The situation with respect to the spacing of repetitions and memory," *Journal of Verbal Learning and Verbal Behavior*, vol. 9, no. 5, pp. 596–606, Oct. 1970.

[7] S. A. Madigan, "Intraserial repetition and coding processes in free recall," *Journal of Verbal Learning and Verbal Behavior*, vol. 8, no. 6, pp. 828–835, Dec. 1969.

[8] S. Reddy, I. Labutov, S. Banerjee, and T. Joachims, "Unbounded human learning: Optimal scheduling for spaced repetition," in *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, 2016, pp. 1815–1824.

[9] S. H. K. Kang, "Spaced repetition promotes efficient and effective learning: Policy implications for instruction," *Policy Insights from the Behavioral and Brain Sciences*, vol. 3, no. 1, pp. 12–19, 2016.

[10] A. R. Lotfolahi and H. Salehi, "Learners' perceptions of the effectiveness of spaced learning schedule in l2 vocabulary learning," *SAGE Open*, vol. 6, no. 2, p. 215824401664614, Apr. 2016.

[11] D. Rohrer and K. Taylor, "The effects of overlearning and distributed practise on the retention of mathematics knowledge," *Applied Cognitive Psychology*, vol. 20, no. 9, pp. 1209–1224, 2006.

[12] J. H. Reynolds and R. Glaser, "Effects of repetition and spaced review upon retention of a complex learning task." *Journal of Educational Psychology*, vol. 55, no. 5, pp. 297–308, 1964.

[13] J. Blocki, S. Komanduri, L. Cranor, and A. Datta, "Spaced repetition and mnemonics enable recall of multiple strong passwords," in *Proceedings 2015 Network and Distributed System Security Symposium*. Internet Society, 2015.

[14] F. Schimanke, R. Mertens, O. Vornberger, and S. Vollmer, "Multi category content selection in spaced repetition based mobile learning games," in *2013 IEEE International Symposium on Multimedia*. IEEE, Dec. 2013.

[15] F. Schimanke, S. Ribbers, R. Mertens, and O. Vornberger, "Implications of short term memory research for the design of spaced repetition based mobile learning games," in *2015 IEEE International Symposium on Multimedia (ISM)*. IEEE, Dec. 2015.

[16] V. X. Yan, L. G. Eglington, and M. A. Garcia, "Learning better, learning more: The benefits of expanded retrieval practice," *Journal of Applied Research in Memory and Cognition*, vol. 9, no. 2, pp. 204 – 214, 2020.

[17] M. Kumar, S. Shambhu, and P. Aggarwal, "Recognition of slow learners using classification data mining techniques," vol. 2, Jan. 2016.

[18] T. Z. and A. M.Mahmoud, "Clustering of slow learners behavior for discovery of optimal patterns of learning," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 11, 2014.

[19] L. Averell and A. Heathcote, "The form of the forgetting curve and the fate of memories," *Journal of Mathematical Psychology*, vol. 55, no. 1, pp. 25–35, Feb. 2011.