Title

The FaCT System: A New Tool Linking Cognitive Science with Educators

Permalink

https://escholarship.org/uc/item/19t0h644

Journal

Proceedings of the Annual Meeting of the Cognitive Science Society, 29(29)

ISSN

1069-7977

Authors

Pavlik Jr., Philip I.
Presson, Nora
Dozzi, Giancarlo
et al.

Publication Date

2007

Peer reviewed

The FaCT (Fact and Concept Training) System: A New Tool Linking Cognitive Science with Educators

Philip I Pavlik Jr. (ppavlik@cs.cmu.edu), Human Computer Interaction Institute
Nora Presson (presson@cmu.edu), Psychology Department
Giancarlo Dozzi (gdozzi@gmail.com), Human Computer Interaction Institute
Sue-mei Wu (suemei@andrew.cmu.edu), Modern Languages Department
Brian MacWhinney (macw@cmu.edu), Psychology Department
Kenneth Koedinger (koedinger@cmu.edu), Human Computer Interaction Institute
Pittsburgh Science of Learning Center
Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213 USA

Abstract

The FaCT (Fact and Concept Training) System provides a general platform for delivering practice in the form of discrete flashcard-like drills. The system optimizes practice schedules according to model-based predictions and can be used to deliver various types of assessment. The system's features satisfy the real world goals of educators using a theory-driven approach that gives researchers control over the model of practice delivery. For educators it provides web deployment, automatic reporting of student practice and assessment, and the ability to tailor content for specific curricular needs. For researchers it provides data export to MySQL, pluggable model architecture, and generalized model fitting algorithms.

Keywords: Memory; Skill; Practice; Computer-Aided Instruction.

System Overview

The FaCT System is a general-purpose application to provide practice for learners in various domains. Practice in these domains takes the form of a sequence of discrete drill trials, each of which includes immediate corrective feedback for errors. This sequence of practice trials is selected with an algorithm that uses a cognitive model of skill learning and forgetting to predict the optimal item to practice for each trial. Although the system currently uses the ACT-R model for its declarative memory predictions and trial selections (Anderson & Schooler, 1991; Pavlik Jr. & Anderson, 2005), the FaCT architecture is designed to house any model that produces dependent measures that can be used to select practice (e.g., latency and probability correct). The FaCT System is written mainly as a Java applet and is delivered over the web to learners and experimental subjects when they navigate to a webpage where the Java applet is located.

This paper introduces the system capabilities and some preliminary data. The primary goal of this research has been to develop a flexible, configurable vehicle for testing cognitive theories of practice and applying these theories to advance concrete educational goals. The facts and concepts the system is designed to train are single-step problems rather than the multi-step ones trained by more complex cognitive tutors (Koedinger & Corbett, 2006). Despite their simplicity, these facts and concepts are important in many real world domains. Specifically, this paper summarizes FaCT System projects in foreign language vocabulary (the Chinese I and II work described below) and in French word gender identification (a newer project described in this paper). The system also has clear applications in domains such as chemistry nomenclature, basic facts in math (i.e. times tables), history facts, and geography facts.

Sample Student Run

The longest running installation of the FaCT system has been in introductory Chinese I and II at Carnegie Mellon University, in cooperation with the Pittsburgh Science of Learning Center. Students in these classes are assigned a portion of their grade for using the system for vocabulary practice for a minimum of 15 minutes per week.

For their weekly assignments students navigate to the URL for the system (demonstration versions are available at http://optimallearning.org/demos/). At this point the Java browser plug-in (freely available at the Sun Microsystems website) loads the system applet and a login window is displayed. Students complete the login according to the instructions. After these preliminaries, the instructions for practice are displayed, and practice begins after the user presses a button.

Chinese practice trials may occur in two canonical forms, passive presentation (a study trial) and drill presentation (a test trial that includes corrective review for failure). While there are options for fixed-schedule practice sessions that randomize items and conditions (called assessment sessions in FaCT), students in Chinese are administered practice according to efficiency predictions dynamically computed from an equation-based ACT-R model of declarative memory. The overall premise of this practice scheduling is that there is an ideal spacing between successive presentations of each fact item or skill exemplar. In the model, the best spacing is characterized by a tradeoff between the advantage of spaced practice (wider temporal spacing between repetitions of an item improves the long-term recall gains for each practice) and the disadvantage of wider spacing (wider spacing decreases recall during drill practice, causing slower recall and more failures, which require costly review feedback).

The sequence of trials that students in Chinese I and II experience is determined by an algorithm derived from this

ACT-R model. Interestingly, practice for each item (or skill) tends to follow a schedule of expanding spacing (with each item being repeated again after an increasingly long interval). This happens because both of the effects being balanced (spacing effect and cost of recall) depend on memory strength, which tends to stabilize as practice accumulates. This increased stability after each repetition means more time can pass before the next repetition since both recall costs will grow more slowly (reducing the cost of wider spacing) and spacing benefits will grow more slowly (requiring wider spacing to show the same benefit). Further, because spacing for items increases with each successful repetition, new items can be continually introduced by interleaving them with repetitions of previously presented items. This algorithm is conceptually similar to work by Pimsleur (1967) in which he advocates a "graduated interval schedule" as being optimal for language learning.

Table 1 focuses on how the terms pair, cluster and drill relate in the two different tasks. The content for Chinese I and II vocabulary includes six possible drill trial types for each vocabulary "cluster," corresponding to the four cue types [English written word form, Chinese written pinyin form (using English orthography), Chinese written Hanzi form (using Chinese characters) the Chinese pronunciation (sound file)] crossed with two response types [English written and pinyin written]. Because the pinyin→English and English→pinyin drills address the same pairing, these 6 drill types correspond to the 5 target pairings, which together are referred to as a cluster. In French, the cluster structure is similar, but differs in the nature of the pairs: rather than two lexical items in different modes being paired together, the pairing in French consists of the target word (e.g., *maison*) and its gender response, according to some rule (e.g., -on words tend to be masculine).

Practice scheduling uses a model in which the knowledge components for each pair are mapped to ACT-R model memory strengths. These strengths are used to compute efficiency of all the pairs in the set before each trial. The program either selects a prior pair that is at its optimally efficient repetition interval or selects a new pair (or new cluster in French).

Practice continues in this fashion until a criterion is met or the student chooses to quit (the system can support time based, performance-based or model-based criteria). Finally, the student's data are saved. Students and teachers have access to online web reports that track the history of use and performance.

Presentation Structures

Practice in the system can be subdivided into trials and curriculum units. A trial is the smallest increment of practice, and curriculum units are composed of trials.

Trials

The system uses only two trial types (the study presentation and the drill presentation), although these trial types can take different forms depending on the domain. The current system distinguishes two types of responses: text and multiple-choice. These response modes currently correspond to the models that are used to select practice trials; specifically, we use text responses for the paired-associate model (in Chinese) and multiple-choice responses for the general-specific model of skills (in French). These models will be described in further on in the paper.

When responses are multiple choice, predictions do include the probability of guessing to account for the effect of the number of choices on probability of performance [i.e. p(success with guessing) = p(success w/o guessing) + p(fail) * p(guess)]. This is particularly important for the French gender identification model where guessing alone has a 50/50 chance of correctness.

Curriculum Units

Curriculum units are defined when the system configuration is specified. There are several varieties of curriculum units, the model driven learning sessions being the most interesting.

Learning Sessions Learning sessions begin with a short introduction screen that may contain curriculum unit-specific instructions. After clicking an "OK" button, practice trials are delivered in sequence according to the predictions of the model. Because the specific model (ACT-R) we used is not the subject of the paper and can vary across content domains, we will not describe the exact equations in this paper (Pavlik Jr., in press; Pavlik Jr. &

Table 1: Presentation structures in French gender and Chinese vocabulary.

Task	Pair	Cluster	Drill Trials
Chinese Vocabulary	Any combination of 2 stimuli that stand for the same semantic unit (e.g., tea-*cha*)	The set of all pairs that stand for the same semantic unit (5 pairs for each cluster: tea-*cha*, tea-茶, tea-"*cha*" sound, *cha*-茶, *cha*-"*cha*" sound)	All permutations of pairs with allowable responses (tea→*cha*, *cha*→tea, 茶→tea, "*cha*" sound→tea, 茶→*cha*, "*cha*" sound→cha)
French Gender	A target word and its grammatical gender response (linked by an inference rule) (e.g., *fromage*-M)	All exemplars of a particular rule (therefore of the same gender) (e.g., *fromage*-M, *ménage*-M, . . .)	A word stimulus and a gender response (e.g., *fromage*→M, *ménage*→M, . . .)

Anderson, 2005). However, because the system is flexible about the mathematical model it will accept, the next section (Models) describes how the model is integrated into the delivery system.

Assessment Sessions Assessment sessions can be mixed with learning sessions to monitor learning or provide periodic quizzes using an experimenter-controlled schedule. This capability allows the designer to create experiments which present specific pairs or randomized subsets of pairs at specific spacings, according to a particular research design. These experiments can be used to explore memory and skill learning theories at a micro level and are also useful to parameterize the model, because they can provide controlled parametric data that are more stable for model fitting than data from learning sessions. For instance, our lab is currently analyzing an experiment that compares 5 replications of a 21-condition paired-associate experiment designed to investigate transfer effects between possible pairings within clusters. (For example, one condition of this experiment tested whether practice on a Hanzi→Pinyin pair provides an advantage when learning the English→Pinyin pair for the same cluster.) The assessment session system allowed full randomization of pairs into conditions, counterbalancing, and provided systematic perturbation of schedules (to control sequence learning effects).

Survey Sessions, N-Back Sessions and Instruction Sessions Although these special unit types have been used infrequently in our lab, they show how easy it is to use the FaCT architecture to fit the needs of a curriculum or experiment. Survey sessions allow a researcher or teacher to deliver a sequence of Likert scale multiple-choice questions or text answer questions. N-back sessions deliver a version of the N-back task in which subjects must respond to a sequence of letters when the current item is repeated N-back in the sequence. Instruction sessions essentially allow the student or subject to page through a series of instructional slides created by the designer, which allows more detailed instructions than can be provided in the single instruction screen preceding learning or assessment sessions.

Models

Learning sessions in the system are controlled by a mathematical model of memory. This model can be conveniently subdivided into a structural model and a dynamic model. The structural model describes how the pairs for each trial are mapped to strengths (ACT-R based activations) in the dynamic model. The structural model allows the system to capture different relationships between pairs and within clusters. At present, we use two different structural models (described below). The dynamic model specifies how the components of the structural model are learned and forgotten with practice or the passage of time. Currently the system uses a version of ACT-R to determine these dynamic factors.

Structural Models

The basic assumption behind the structural level models is that very few domains contain collections of independent pairs (Asch & Ebenholtz, 1962). In fact, the probability of recall and latency of pairs in most sets are somewhat related. For example, in Chinese, the four stimulus modes the system handles result in 6 possible drill types (as discussed in the introduction), two of which are the English→pinyin and Hanzi→pinyin. In both cases, these drills depend not only on the specific linked pair, but also on the ability to produce the Pinyin. Because of this, the pairs are not independent.

Similarly, in the French gender identification work, there may be many words all of which are masculine because they share a common word ending (for instance, words that end in –age are most often masculine, as in *le fromage*). Each of these words constitutes a pair (when the response is also considered), but it is obvious that they share variance, because the same rule can be used to respond to any of pairs in a cluster (and in fact, it is this generalized responding, rather than exemplar-based recall, that we wish to teach).

To deal with this issue of dependence within clusters of related pairs, the system uses two structural models: the paired-associate structure and the general-specific structure.

The paired-associate model structure assigns three memory strengths to each pair: one for each stimulus and one for the link between them. Given this structural model, probability correct depends on the strength of the link and the strength of the response in a conjunctive function, p(link) * p(response). Latency is handled differently, as the sum of a fixed response time cost, variable word length time cost, link recall time cost and response recall time cost. Not only does this structural model handle the pinyin response example above, but it also handles the issue of drill directionality for reversible pairs. For example, consider an English-pinyin pair such as "sit—*zuo*" after it is given a single study presentation. It seems clear that recall probability for this single pair depends on how it is drilled. If one provides the Chinese, the student must recall the English word (which in the model has an assumed p=1.0) and the link, whereas if the English is presented, the pinyin response and the link must be recalled. This model captures research which suggests responding with English should be easier for an English speaker (e.g. Schneider, Healy, & Bourne, 2002).

The general-specific model, on the other hand, assigns two memory strengths to each pair, the specific component and the general component. General components are shared among all the pairs in a cluster, while each pair has a unique specific component. Given this structural model, probability of skill performance depends on the strength of both general and specific components in a disjunctive function, p(general) + p(specific) * (1-p(general)). Latency is currently calculated as a function of the general strength only, but this may change as model verification and testing of the system continues. As will be described, adjusting

these functions requires the designer to have only a small understanding of the Java computer language.

In this general-specific model, presentations are different from the paired-associate structure. For study trials (and review after drill failures), it is necessary both to present the prompt item and the correct response as is done for paired-associates, but additionally the system presents the inference rule that explains why the answer is correct. This inference rule for a cluster of pairs is essentially a simple explanation of why the answer is correct. Drill trials do not refer to this inference rule except during the review feedback for errors. Therefore, although study trial information could conceivably be memorized (hence the specific component in the model), test trials of unseen exemplars in a cluster must use the general component for correct performance. General-specific implementation in the French gender identification experiment has used more than 10 exemplars for each gender rule, which allows the system to produce many unique drills of each inference rule. This prevents students from merely memorizing content by requiring that performances depend on general rule use.

Dynamic Model

The dynamic model provides the output to the structural model that is necessary to compute predicted latencies and probabilities necessary to make efficiency predictions about specific pairs. However, since this paper is not about the specifics of the ACT-R model, this section about models explains the model interface which specifies the functions that control practice item selection. The underlying mathematical model functions are not shown because they are considered to be theory specific (Pavlik Jr., in press; Pavlik Jr. & Anderson, 2005), and because other models could be used just as easily. In contrast, the functions below need to be defined regardless of the model (e.g. ACT-R, Markov, etc.). Table 2 shows the main functions that a model linked to the system needs to support.

The first two functions (updateStudyPair and updateDrillPair) are called after each practice to update the model based on the learning that occurred from that practice. Further, for a drill trial, the updateDrillPair function accounts for the success or failure of that practice. Success or failure may lead to differences in learning due to the difference between successful recall and passive study. Further, success means the pair was better learned *prior* to the drill and this can be used to adjust the model as described in the next section.

RecallLatency and probRecallPair are used for various purposes. Importantly, they are the values the model outputs that can be compared with learner performance data in the computeLLPair function to compute a loglikelihood statistic for fitting the model parameters or comparing different models. Also, both of these functions are used when computing the learning rate for drill trials.

The chooseTrial function is the function called by the learning session which provides the identity of the next pair and whether it should be studied or drilled. The choose trial

function depends on the learnRateDrill and learnRateStudy functions.

Finally, numMastered provides an output function the system uses to check if criterion performance has been met. This function's output is attached to a progress bar that is displayed during learning sessions.

Table 2: Model interface (supported functions).

Function Name	When Called (and Purpose)
updateStudyPair	After study presentation to increment learning
updateDrillPair	After drill presentation to increment learning and adjust attribution of proficiency
recallLatency	When computing learning rates and model likelihood
probRecallPair	When computing mastery, learning rates, and model likelihood
numMastered	When updating screen progress bars and checking unit progress criterion
learnRateStudy	During the process of choosing the next trial this is called for all pairs
learnRateDrill	During the process of choosing the next trial this is called for all pairs
computeLLPair	When fitting models (both during and after practice curriculum units) this function is called on all pairs
chooseTrial	Before each trial this function selects the optimal pair using the learning rates

Figure 1 shows how control flows during learning sessions. As the diagram shows, learning sessions largely involve the interaction of the chooseTrial function, which decides which pair of which cluster to practice, and the update functions (updateStudy and updateDrill), which adjust the model based on an attribution of the results of practice.
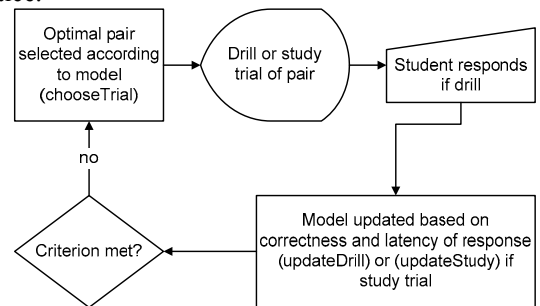

Figure 1: Learning session curriculum unit flowchart.

Knowledge and User Tracking

To make the system we have described behave robustly for different users in light of inherent human variability requires that the system track both the user's knowledge state for each pair and the overall performance characteristics of each user. This tracking allows real-time adjustments to system

performance so the system "learns" about the current user to tailor optimal practice to that user. Although specific pair knowledge estimates and overall model parameter estimation are both parameter fitting issues, they are handled separately to facilitate implementation.

The issue of tracking learning for individual pairs is handled by the update functions. Of course, because study trials do not produce feedback to the system, they do not result in anything more than an increment in memory strength. Drills, on the other hand, are handled in one of two ways. The original method (and current method in the French gender identification work) for drill trials uses a Bayesian procedure that assumes that the distribution of variability for each pair's learning for each user is a normally distributed random variable with a fixed mean and variance. Given this information (the prior), and the model's assumptions about distribution of activation in the case of success or failure (the data), it is possible to use numerical integration to track the expected posterior distribution of activation as a function of performance.

More recent versions of the program (Chinese vocabulary), however, have shown that simply weighting learning from successful drill trials approximately 3 times more strongly than failed drill trials closely approximates this Bayesian procedure and is much easier to implement. This is the current method of accounting for performance in the model for individual pairs for individual users.

The issue of tracking user learning more globally is handled by a simple gradient descent algorithm that runs during the time that the system is given review for failed drill trials. For any model parameter, this algorithm computes the loglikelihood for the overall model at points one step above, equal to, and one step below the parameter's current value (step size is configurable for each parameter). Then, if the model finds that the above or below LL scores indicate the model is improved at the new parameter value, the algorithm adjust the parameter value a half step toward the optimal value (moving ½ step means that algorithm does not tend to oscillate between two parameters values with similar loglikelihoods when it has reached a local minimum). Although this simple optimizer is quite limited, it tends to be powerful enough to provide fast, course-grained improvements in parameters during learning. For students, this means that the system will become easier (if they are performing below model expectations) or more difficult (if they are performing better than model expectations) in response to their individual performance.

System Architecture

The FaCT System is primarily a client-server architecture as can be seen in Figure 2. The client side consists of a jar file that is automatically loaded and run when a user navigates to an HTML page found on a remote server. After the jar file is detected and run by the user's Java plug-in (version 1.4.2.10 or later), the user sees the FaCT GUI which instructs the user to login. Multiple types of logins are available, ranging from unrestricted login (that accepts any

text string as a user name) to passworded login that requires users to create both a login and password (which are both encrypted on the server).

After successful login, the applet allows the user to choose the training definition file (TDF) that specifies the stimuli set. Each HTML installation can point to multiple TDFs (or a single TDF). Each TDF specifies a sequence of curriculum units for the stimulus set. Each TDF corresponds to only one stimulus set, but the ability to address multiple TDFs at a single HTML installation location means that each installation is unlimited in the number of stimulus sets that it can train.

After selecting a TDF that the user has practiced previously, the system will automatically load the data files or create them if the user is new. After the instructions, practice commences until criterion is met or until the user initiates a save by pausing the system (Esc key). Saving involves writing four files (stimuli.xml, progress.xml, pairs.xml, and acts.xml) to the users/ directory in the main server filespec path. A CGI script accepts the files from the client and writes them to the server.
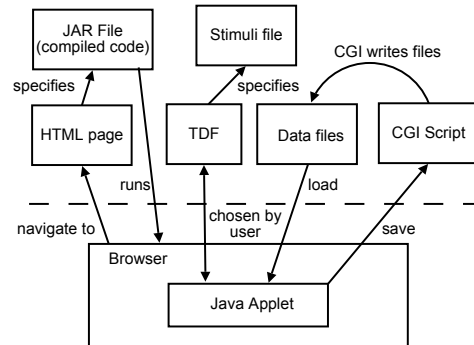

Figure 2: System architecture with server side above dashed line and client side below.

It is also important to understand the contents of the four data files that are saved for each user of a specific TDF. The progress.xml file is a short file of summary information It includes running totals of correct responses, times of starts and stops of the system, a record of which curriculum unit the user was on, and the experimental condition (if any). The pairs.xml file contains the pair level record of performance for the set of stimuli. This record includes latency of initiation of response, latency of submission of response, correctness, times of performances and study trials, and a number of reference values for lookup of stimuli and memory strengths. The acts.xml file is essentially ACT-R specific and contains the individual memory strengths (activations for link, single stimuli, general and specific components). Despite the current ACT-R specificity, the acts.xml file could be adapted for use with other models simply by storing different values in it. The stimuli.xml file is a catalog and ordering of the stimulus set. There are actually two stimulus files, the original, which the TDF points to, and the processed version, which is saved with the user files.

Distributed Functionality

Although the main system is client-server, there are also functions that are distributed to other computers. Specifically, the following subsystems access the user files (and TDFs for model fitting) from a remote location, typically a desktop machine used for data analysis.

Model Fitting

A model fitting module integrates a GPL (general public license) function optimizer that uses a BFGS optimization algorithm to maximize the loglikelihood of the model given a list model parameters to optimize. The optimizer can be run for a list of subjects and a list of TDFs for those subjects. Using initial seed parameters, the algorithm estimates the optimal parameters for each subject for each TDF. Output is in the form of a comma separated text file with each row corresponding to a subject and each column a final parameter estimate for that subject. Each row also has columns that keep a record of final loglikelihood scores which simplifies calculations of likelihood ratio tests used to evaluate fits.

While the system operates reasonably well with a wide range of parameter settings, it is typically necessary to reestimate a subset of the model parameters for each new domain. The model fitting function provides a principled means to estimate the new values for the new domain.

MySQL Export

A MySQL (http://www.mysql.com/) export module transfers specific components of the data files into a structured MySQL database making these record quickly available for transfer into statistical analysis software. MySQL queries can easily be used to compute by-subject or by-trial-type averages for particular curriculum units or particular subsets of pairs (conditions in experiments).

Preliminary Results

Classroom testing of the system is currently in progress. Fall 2006 course results show that the system provides significant advantages in both Chinese and French.

In Chinese, 7 sections of Chinese I class participated in an experiment in which students were randomized to either a) have unit 3 voluntary and unit 4 required or b) have unit 3 required and unit 4 voluntary. This crossover within-subjects experiment tested whether there was an advantage for requiring students to use the system 15 minutes compared to not requiring usage. For each student we computed the score advantage for the required unit vs. voluntary unit on a paper and pencil test of both units (10 items for each unit given approximately one month later). Errors were less ($M = 0.90$, $SD = 1.4$) for required compared to voluntary usage ($M = 1.5$, $SD = 1.7$) $t(53) = 3.0$, $p < .005$, with a Cohen's d effect size $= 0.41$.

In French, two participating French I classes were assigned to use the system for one 15-25 minute session each week, for five weeks. Both these classes and two comparison classes were given a 25-item pre-test and post-test on pencil and paper evaluating their ability to assign grammatical gender to unfamiliar words, showing that they had extracted the gender/spelling rule. Independent of class, there was a larger increase in performance from pre- to post-test in the classes that used the system (M(improvement) $= 2.63$, $SD = .53$), compared to those that did not ($M = .25$, $SD = .44$), and this effect was significant ($F (1, 59) = 3.851$, $p < .05$). A more detailed report is in preparation.

Conclusion

The FaCT System blends goals and principles from psychology, artificial intelligence, education and linguistics in a system that addresses real world needs for efficient learning systems. Even as the system is designed to address the needs of specific domains and curriculums, it is also designed to become increasingly faithful to cognitive science by providing a platform that allows different theories and models of practice to compete and be refined.

Acknowledgments

References

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. Psychological Science, 2(6), 396-408.

Asch, S. E., & Ebenholtz, S. M. (1962). The principle of associative symmetry. Proceedings of the American Philosophical Society, 106, 135-163.

Koedinger, K. R., & Corbett, A. (2006). Cognitive tutors: Technology bringing learning sciences to the classroom. In R. K. Sawyer (Ed.), The Cambridge handbook of the learning sciences. New York, NY, US: Cambridge University Press.

Pavlik Jr., P. I. (in press). Timing is an order: Modeling order effects in the learning of information. In F. E., Ritter, J. Nerb, E. Lehtinen & T. O'Shea (Eds.), In order to learn: How order effects in machine learning illuminate human learning. New York: Oxford University Press.

Pavlik Jr., P. I., & Anderson, J. R. (2005). Practice and forgetting effects on vocabulary memory: An activation-based model of the spacing effect. Cognitive Science, 29(4), 559-586.

Pimsleur, P. (1967). A memory schedule. The Modern Language Journal, 51(2), 73-75.

Schneider, V. I., Healy, A. F., & Bourne, L. E., Jr. (2002). What is learned under difficult conditions is hard to forget: Contextual interference effects in foreign vocabulary acquisition, retention, and transfer. Journal of Memory and Language, 46(2), 419-440.