

# 14

## CRUNCHING BIG DATA WITH FINGERTIPS

### How Typists Tune their Performance Toward the Statistics of Natural Language

*Lawrence P. Behmer Jr.*  
*Matthew J. C. Crump*

#### Abstract

People have the extraordinary ability to control the order of their actions. How people accomplish sequencing and become skilled at it with practice is a long-standing problem (Lashley, 1951). Big Data techniques can shed new light on these questions. We used the online crowd-sourcing service, Amazon Mechanical Turk, to measure typing performance from hundreds of typists who naturally varied in skill level. The large data set allowed us to test competing predictions about the acquisition of serial-ordering ability that we derived from computational models of learning and memory. These models suggest that the time to execute actions in sequences will correlate with the statistical structure of actions in the sequence, and that the pattern of correlation changes in particular ways with practice. We used a second Big Data technique,  $n$ -gram analysis of large corpuses of English text, to estimate the statistical structure of letter sequences that our typists performed. We show the timing of keystrokes correlates with sequential structure (letter, bigram, and trigram frequencies) in English texts, and examine how this sensitivity changes as a function of expertise. The findings hold new insights for theories of serial-ordering processes, and how serial-ordering abilities emerge with practice.

#### Introduction

The infinite monkey theorem says a room full of monkeys typing letters on a keyboard for infinity will eventually produce any text, like the works of Shakespeare or this chapter (Borel, 1913). This small space of natural texts has more predictable structure than the many other random texts produced by the typing monkeys. For example, letters and bigrams that occur in English appear with particular frequencies, some high and some low. The present work examines whether typists, who routinely produce letters by manipulating a keyboard,

**320** Behmer and Crump

become sensitive to these statistical aspects of the texts they type. Answering this question addresses debate about how people learn to produce serially ordered actions (Lashley, 1951).

***The Serial-Order Problem***

The serial-order problem cuts across human performance, including walking and talking, routine activities like tying a shoe, making coffee, and typing an email, to crowning achievements in the arts and sports where performers dazzle us with their musical, visual, and physical abilities. Performers exhibit extraordinary serial-ordering abilities. They can produce actions in sequences that accomplish task goals.

Three general features of serial-ordering ability requiring explanation are specificity, flexibility, and fluency. Performers can produce highly specific sequences, as in memorizing a piece of music. Performers can flexibly produce different sequences, as in language production. And, performers can produce sequences with great speed and accuracy. The serial-order problem is to articulate the processes enabling these abilities across domains of human performance, which is an overbroad task in itself, so it is not surprising that a complete account of the problem has remained elusive. However, theoretical landscapes for examining the problem have been put forward. We review them below.

***Associative Chain Theory***

Prior to the cognitive revolution (pre-1950s) serial-ordering ability was explained by associative chains whereby one action triggers the next by association (Watson, 1920). This domino-like process assumes that feedback from movement  $n$  triggers the next movement ( $n+1$ ), which triggers the next movement ( $n+2$ ), and so on.

Karl Lashley (1951; see Rosenbaum, Cohen, Jax, Weiss, & van der Wel, 2007, for a review) famously critiqued chaining theories on several fronts. The idea that feedback triggers upcoming actions does not explain action sequences that can be produced when feedback is eliminated, or rapid action sequences (like lightning fast musical arpeggios) where the time between actions is shorter than the time for feedback to return and trigger the next action.

The biggest nail in the coffin was that associative chains could not flexibly produce meaningful sequences satisfying grammatical rules for ordering—a requirement for language production. Consider an associative chain for ordering letters by associations with preceding letters. The associative strength between two letters could reflect the number of times or likelihood that they co-occur. Bigram co-occurrence can be estimated from large corpuses of natural text. For example, taking several thousand e-books from *Project Gutenberg* (see Methods) and counting the occurrence of all bigrams estimates the associations between

letter pairs in English. Generating sequences of letters using these bigram statistics will produce sequences with the same bigram frequency structure as the English language, but will rarely produce words, let alone meaningful sentences. So an associative chain theory of letter sequencing would take a slightly shorter amount of infinite time to produce the works of Shakespeare, compared to random monkeys.

Although associative chains fail to explain serial-ordering behavior in complex domains like language, the more general goal of explaining serial ordering in terms of basic learning and memory processes has not been abandoned. For example, Wickelgren (1969) offered that associative chains could produce sequences of greater complexity by allowing contextual information to conditionalize triggering of upcoming actions. And, as we will soon discuss, contemporary neural network approaches (Elman, 1990), which are associative in nature, have been successfully applied as accounts of serial-ordering behavior in a number of tasks.

Lashley's critique inspired further development of associative theories, and opened the door for new cognitive approaches to the serial-order problem, which we broadly refer to as hierarchical control theories.

### *Hierarchical Control Theory*

Hierarchical control theories of serial-ordering invoke higher-order planning and monitoring processes that communicate with lower-order motor processes controlling movements. Hierarchical control is aptly described by Miller, Galanter & Pribram's (1960) concept of TOTE (test, operate, test, exit) units. TOTEs function as iterative feedback loops during performance. For example, a TOTE for pressing the *space* key on a keyboard would test the current environment: has the *space* key been pressed? No; then, engage in the operation: press the *space* key; then, re-test the environment: has the *space* key been pressed? Yes; and, finally exit the loop. TOTEs can be nested within other TOTEs. For example, a TOTE to type a word would control sub-TOTEs for producing individual letters, and sub-sub-TOTEs, for controlling finger movements. Similarly, higher-level TOTEs would guide sentence creation, and be nested within even higher level TOTEs for producing whole narratives.

A critical difference between hierarchical control and associative theories is the acknowledgment of more complex cognitive functions like planning and monitoring. These assumptions enable hierarchical control frameworks to describe more complicated serial-ordering behaviors, but require further explanation of planning and monitoring processes. Hierarchical control theories are common in many areas including, language (Dell, Burger, & Svec, 1997), music (Palmer & Pfordresher, 2003), routine everyday actions (Cooper & Shallice, 2000), and skilled typing (Logan & Crump, 2011).

**322** Behmer and Crump

### *Emergence versus Modularism (A Short Coffee-Break)*

The history of theorizing about the serial-order problem is a dance between emergent and modularistic explanations. Emergent explanations seek parsimony by examining whether basic learning and memory processes produce serial-ordering abilities “for free”, without specialized systems for sequencing actions. Modules (Fodor, 1983) refer to specialized processes tailored for the demands of sequencing actions. Associative chain theory uses commonly accepted rules of association to argue for a parsimonious account of serial-ordering as an emergent by-product of a general associative process. Hierarchical control theories assume additional processes beyond simple associations, like those involved in plan construction, implementation, and monitoring.

The interplay between emergent and modularistic accounts was featured recently in a series of papers examining errors made by people brewing cups of coffee. Cooper & Shallice (2000, 2006) articulated computational steps taken by a planning process to activate higher-level goals (make coffee), and series of lower level sub-goals (grab pot, fill with water, etc.), and sub-sub-goals (reach and grasp for handle), and then described how their algorithms could brew coffee, and make human-like errors common to coffee-brewing. Then, Botvinick & Plaut (2004, 2006) showed how a general associative process, modeled using a serial recurrent neural network (Elman, 1990), could be trained to make coffee as accurately and error-prone as people. Their modeling efforts show that some complex routine actions can be explained in an emergent fashion; and that representational units for higher-order plans (i.e. TOTE units) are functionally equivalent to distributed collections of lower-level associative weights in a neural network. Thus, a non-hierarchical learning process can be substituted for a more complicated hierarchical process as an explanation of serial-ordering behavior in the coffee-making domain.

Coffee-making is a difficult task for discriminating between theories. Although plan-based theories are not required, they may be necessary elsewhere, for example in tasks that require a larger, flexible repertoire of sequences as found in skilled typewriting (Logan & Crump, 2011). Perhaps the two approaches share a nexus, with general associative learning principles tuning aspects of the construction, activation, and implementation of plans for action. So the dance continues, and in this chapter, leads with the movements of fingers across computer keyboards.

### *Hierarchical Control and Skilled Typing*

Skilled typing is a convenient tool for examining hierarchical control processes and naturally suited to studying serial-ordering abilities. Prior work shows that skilled typing is controlled hierarchically (for a review see Logan & Crump, 2011). Hierarchically controlled processes involve at least two distinguishable levels in

which elements from higher levels contain a one-to-many mapping with elements in the lower levels. Levels are encapsulated. The labor of information processing is divided between levels, and one level may not know the details of how another level accomplishes its goals. Because of the division of labor, different levels should respond to different kinds of feedback. Finally, although levels are divided, they must be connected or coordinated to accomplish task goals.

The terms outer and inner loop are used to refer to the hierarchically nested processes controlling typing. The outer loop relies on language production and comprehension to turn ideas into sentences and words, passing the result one word at a time to the inner loop. The inner loop receives words as plans from the outer loop, and translates each word into constituent letters and keystrokes. The outer loop does not know how the inner loop produces keystrokes. For example, typists are poor at remembering where keys are located on the keyboard (Liu, Crump, & Logan, 2010; Snyder, Ashitaka, Shimada, Ulrich, & Logan, 2014), and their typing speed slows when attending to the details of their actions (Logan & Crump, 2009). The outer and inner loop rely on different sources of feedback, with the outer loop using visual feedback from the computer screen to detect errors, and the inner loop using tactile and kinesthetic feedback to guide normal typing, and to independently monitor and detect errors (Crump & Logan, 2010b; Logan & Crump, 2010). Finally, word-level representations connect the loops, with words causing parallel activation of constituent letters within the inner loops' response scheduling system (Crump & Logan, 2010a).

### ***Developing a Well-Formed Inner Loop***

We know that expert typists' inner loops establish a response scheduling process that is precise, flexible, and fluid: capable of accurately executing specific sequences, flexibly producing different sequences, and fluidly ordering keystrokes with speed. However, we do not know how these abilities vary with expertise and change with practice. The outer/inner loop framework and other computational theories of skilled typing have not addressed these issues.

Rumelhart & Norman's (1982) computational model of skilled typing provides a starting point for understanding inner loop development. Their model used word representations to activate constituent letter nodes, which then drove finger movements. Because video recordings of skilled typists showed fingers moving in parallel across the keyboard (Gentner, Grudin, & Conway, 1980), their word units caused parallel activation of letter units and finger movements. This created a problem for outputting the letters in the correct order. Their solution was a dynamic inhibition rule (Estes, 1972): Each letter inhibits every other letter in series, letter activation moves fingers to keys (with distance moved proportional to activation), a key is pressed when a finger reaches its target and its letter activation value is higher than the others, and letter units are de-activated when keys are

**324** Behmer and Crump

pressed. This clever rule is specialized for the task of response-scheduling, but has emergent qualities because the model types accurately without assuming a monitoring process.

The model explains expert typing skill, but says nothing about learning and skill-development. It remains unclear how associations between letters and specific motor movements develop with practice, or how typing speed and accuracy for individual letters changes with practice. Addressing these issues is the primary goal of the present work.

### ***Becoming a Skilled Typist***

The processes underlying the development of typing skill enable people to proceed from a novice stage where keystrokes are produced slowly, to an expert stage where they are produced quickly and accurately. Novices with no previous typing experience scan the keyboard to locate letters, and use visually targeted movements to press intended keys. Their outer loop goes through the motions of (1) intending to type a letter, (2) looking for the letter on the keyboard, (3) finding the letter, (4) programming a motor movement to the found key, and (5) repeating these steps in a loop until all planned letters are produced. Experts can type without looking at the keyboard, with fingers moving in parallel, and with impressive speed (100 ms per keystroke and faster). Experts use their outer loop for planning words and phrases, and their inner loop for producing individual keystrokes in series.

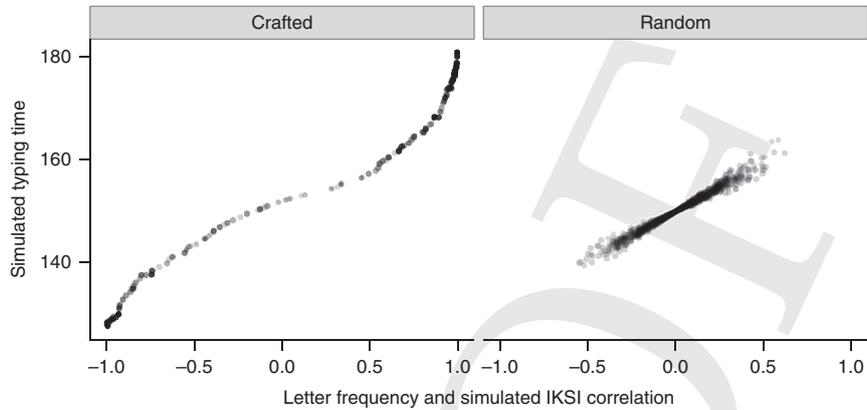
We divide the problem of how inner loops are acquired into questions about how response-scheduling ability changes with practice, and how the operations of response-scheduling processes work and change with practice. Abilities will be measured here in terms of changes in speed and accuracy for typing individual letters. Operations refer to processing assumptions about how one response is ordered after another. Here, we test theories about the development of response-scheduling abilities, and leave tests of the operations of the response-scheduling process to future work.

### ***More Than One Way to Speed Up a Typist***

We consider three processes that could enable a slow typist to improve their speed with practice. Each makes different predictions for how typing times for individual letters would change with practice.

### ***Adjust a Global Response Scheduling Timing Parameter***

If keystroke speed is controlled by adjustable timing parameters, then faster typing involves changing timing parameters to reduce movement and inter-movement times. Normatively speaking, if a typist could choose which letters to type more



**FIGURE 14.1** Simulations showing that mean letter typing time to type a text is shorter for simulated typists whose typing times for individual letters negatively correlate with letter frequency in the typed text.

quickly, they would benefit from knowing letter frequencies in texts they type. All things being equal, and assuming that typists are typing non-random texts, a typists whose letter typing times are negatively correlated with letter frequency norms for the text (i.e. faster times for more than less frequent letters) will finish typing the same text faster than a typist whose letter typing times do not negatively correlate with letter frequency.

To illustrate, we conducted the simulations displayed in Figure 14.1. First, we created a vector of 26 units populated with the same number (e.g. 150 ms) representing mean keystroke times for letters in the alphabet. This scheme assumes that all letters are typed at the same speed. Clearly, overall speed will be increased by decreasing the value of any of those numbers. However, to show that sensitivity to letter frequency alone increases overall speed, we crafted new vectors that could be negatively or positively correlated with letter frequency counts consistent with English texts (taken from Jones & Mewhort, 2004). We created new vectors using the following rules: (1) randomly pick a unit and subtract  $X$ , then randomly pick a different unit, and add the same value of  $X$ ; (2) compute the correlation between the new vector and the vector of letter frequencies; (3) keep the change to the vector only if the correlation increases; (4) repeat 1–3. All of the crafted vectors summed to the same value, but were differentially correlated to letter frequencies through the range of positive and negative values. The random panel shows a second simulation where the values for simulated letter typing speed were simply randomly selected, with the constraint that they sum to the same value. Figure 14.1 shows that time to finish typing a text is faster for vectors that are more negatively correlated with letter frequencies in the text.

**326** Behmer and Crump

The simulation shows that typists have an opportunity to further optimize their typing speed by modifying individual letter typing speeds in keeping with the frequencies of individual letters in the text they are typing. Indeed, there is some existing evidence that, among skilled typists, letter typing times do negatively correlate with letter and bigram frequencies (Grudin & Larochelle, 1982). However, it is unclear how these micro-adjustments to the timing of individual keystrokes take place. If typists are simply changing the timing parameters for each keystroke whenever they can, without prioritizing the changes as a function of letter frequency, then we would not expect systematic correlations to exist between letter typing times and letter frequencies. The next two hypotheses assume that typists become sensitive to the statistics of their typing environment “for free”, simply by using the same general learning or memory processes they always use when learning a new skill.

*Use a General Learning or Memory Process*

The task of typing has processing demands equivalent to laboratory sequencing tasks, such as artificial grammar learning (Reber, 1969) and the serial-reaction time (SRT) task (Nissen & Bullemer, 1987). In those tasks, participants respond to unfamiliar patterned sequences of characters or shapes defined by a grammar. The grammar controls the probability that particular characters follow others. By analogy, the task of typing words is very similar. The letters in words in natural language occur in patterned fashion, with specific letters, bigrams, and trigrams (and higher-order  $n$ -grams) occurring with specific frequencies. In the artificial grammar task, as participants experience sequences they develop the ability to discriminate sequences that follow the grammar from those that do not. In the SRT task, as participants gain experience with sequences they become faster at responding to each item in the sequence. In the task of typing, the question we are interested in asking is how typists become sensitive to the frequencies of letters, bigrams, and trigrams, as they progress in acquiring skill in typing.

Computational models of learning and memory can account for performance in the artificial grammar and SRT task. For example, exemplar-based memory models (Jamieson & Mewhort, 2009a, 2009b), and serial recurrent neural network learning models (Cleeremans & McClelland, 1991; Cleeremans, 1993) explain sensitivity to sequential structure as the result of general learning and memory processes operating on experiences containing order information. The serial recurrent network (Elman, 1990) applied to the SRT task involves the same architecture used by Botvinick & Plaut (2006) to model action sequences for making coffee. These models make specific predictions about how people become sensitive to statistical structure in sequences (i.e.  $n$ -gram likelihoods) with practice.

*Serial Recurrent Network Predictions for Acquiring Sensitivity to Sequential Statistics:*

A serial recurrent neural network (see Elman, 1990) is a modified feed-forward neural network with input units, a hidden layer, and output units. Input patterns are unique numerical vectors describing elements in a pattern, like letters in a sequence. After learning, the distributed pattern of weights in the hidden layer can represent multiple input patterns, such that a particular input generates a learned output pattern. Learning occurs in the model by presenting input patterns, and making incremental adjustments to the weights that reduce errors between input and output patterns. The model learns sequential structure because the hidden layer is fed back into itself as part of the input units. This provides the model with the present and preceding input patterns at each time step. If the model is trained on patterned sequences of letters, then the model gradually learns to predict upcoming letters based on preceding letters. In other words, it becomes sensitive to the  $n$ -gram structure of the trained sequences.

In the context of learning sequences in the SRT task, Cleeremans (1993) showed that participants and network simulations gradually accrue sensitivity to increasingly higher orders of sequential statistics with training. Early in training, most of the variance in reaction times to identify a target is explained by target frequency, but with practice target frequency explains less variance, and higher-order frequencies (e.g. bigrams, trigrams, etc.) explain more variance. The model makes two important predictions. First, sensitivity to sequential structure is scaffolded: Novices become sensitive to lower-order statistics as a pre requisite for developing sensitivity to higher-order statistics. Second, as sensitivity to each higher-order statistic increases, sensitivity to the previous lower-order statistics decreases. The model makes the second prediction because the weights in the hidden layer overwrite themselves as they tune toward higher level sequential structure, resulting in a loss of sensitivity to previously represented lower level sequential structure. This is an example of catastrophic interference (McCloskey & Cohen, 1989) whereby newly acquired information disrupts older representations by changing the weights of those representations.

*Instance Theory Predictions for Acquiring Sensitivity to Sequential Statistics.*

Instance theories of memory make similar predictions for the development of sensitivity to sequential structure. Instance theories assume individual experiences are stored in memory and later retrieved in the service of performance. For example, the instance theory of automatization (Logan, 1988) models the acquisition of routine behavior and the power law of learning with distributional assumptions for sampling memory. Response time is a race between a process that remembers how to perform an action, and an algorithm that computes the correct

328 Behmer and Crump

action, with the faster process winning the race and controlling action. Instances in memory are not created equal, and some can be retrieved faster than others. As memory for a specific action is populated with more instances, that action is more likely to be produced by the memory process because one of the instances will tend to have a faster retrieval time than the algorithmic process. So, memory speeds responding as a function of the likelihood of sampling increasingly extreme values from increasingly large distributions. More simply, faster memory-based responses are more likely when the instance pool is larger than smaller.

We simulated predictions of instance theory for acquiring sensitivity to letter and bigram frequencies in English texts (Jones & Mewhort, 2004) with practice. Response times to a letter or a bigram were sampled from normal distributions, with the number of samples constrained by the number of instances in memory for that letter or bigram. The response time for each was the fastest time sampled from the distribution.

To simulate practice, we repeated this process between the ranges of 50 experiences with letters and bigrams, up to 1,000,000 experiences. To determine sensitivity to letter and bigram frequency, we correlated the vectors of retrieval times for presented letters and bigrams with the vectors of letter and bigram frequencies from the corpus counts.

The No floor panel in Figure 14.2 shows increasingly negative correlations between retrieval time and corpus count frequency as a function of practice for letters and bigrams. The correlations plateau with practice, and letter and bigram sensitivity develop roughly in parallel. Bigram sensitivity is delayed because

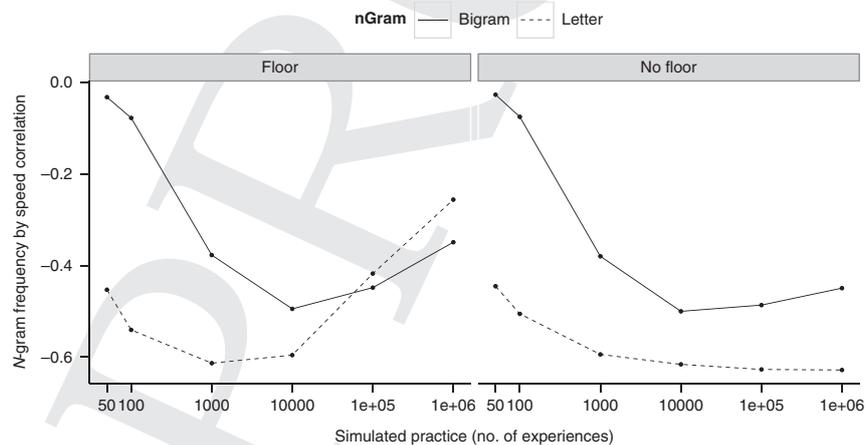


FIGURE 14.2 Simulated instance theory (Logan, 1988) predictions for how correlations between letter and bigram frequency, and their simulated typing times, would change as a function of practice. Floor versus No floor refers to whether simulated typing times were limited by some value reflecting physical limitations for movement time.

experience with specific bigrams occurs more slowly than letters (which repeat more often within bigrams). So, different from the SRN model, the instance model does not predict sensitivity to lower-order statistics decreases with increasing sensitivity to higher-order statistics. However, a modified version of this simulation that includes a floor on retrieval times to represent the fact that reaction times will eventually hit physical limitations does show waxing and waning of sensitivities, with the value of letter and bigram correlations increasing to a maximum, and then slowly decreasing toward zero as all retrieval times become more likely to sample the same floor value.

### ***Testing the Predictions in Skilled Typing***

The present work tests the above predictions in the real-world task of skilled typing. The predictions are tested by analyzing whether typists of different skill levels are differentially sensitive to letter, bigram, and trigram frequencies. This analysis is accomplished by (1) having typists of different skill levels type texts, (2) recording typing times for all keystrokes and for each typist computing mean keystroke time for each  $n$ -gram, (3) for each typist, sensitivity to sequential structure is measured by correlating mean typing times for each letter, bigram, and trigram with their respective frequencies in the natural language, and (4) by ordering typists in terms of their skill level, we can determine whether sensitivity to  $n$ -gram frequency changes as a function of expertise.

To foreshadow our analysis, steps three and four involve two different correlational measures. Step three computes several correlations for each individual typist. Each correlation relates  $n$ -gram frequency to mean keystroke times for each  $n$ -gram typed by each typist. This results in three correlations for each  $n$ -gram level (letter, bigram, and trigram) per typist. Because skill increases with practice, we expect faster keystrokes (decreasing value) for more frequent  $n$ -grams (increasing value). For example, typists should type high frequency letters faster than low frequency letters, and so on for bigrams and trigrams. So in general, all of these correlations should be negative. We take the size of the negative correlation as a measure of sensitivity to  $n$ -gram structure, with larger negative values showing increasing sensitivity.

Step four takes the correlations measuring sensitivity to  $n$ -gram frequency from each typist and examines them as a function of typing expertise. For example, one measure of typing expertise is overall typing speed, with faster typists showing more expertise than slower typists. According to the learning and memory models novices should show stronger sensitivity to letter frequency than experts. Similarly, experts should show stronger sensitivity to bigram and trigram frequencies than novices. These results would license consideration of how general learning and memory processes participate in hierarchically controlled skilled performance domains like typing.

**330** Behmer and Crump

### ***Using Big Data Tools to Answer the Question***

To evaluate the predictions we needed two kinds of Big Data. First, we needed access to large numbers of typists that varied in skill level. Second, we needed estimates of sequential structure in a natural language such as English.

We found our typists online using Amazon Mechanical Turk (mTurk), an Internet crowdsourcing tool that pays people small sums of money to complete HITS (human intelligence tasks) in their web browser. HITS are tasks generally easier for people than computers, like listing keywords for images, or rating websites. The service is also becoming increasingly popular as a method for conducting behavioral experiments because it provides fast and inexpensive access to a wide population of participants. More important, modern web-browser technology has reasonably fine-grained timing abilities, so it is possible to measure the timing of keypress responses at the level of milliseconds. For example, Crump, McDonnell & Gureckis (2013) showed that browser-based versions of several classic attention and performance procedures requiring millisecond control of display presentation and response time collection could easily be reproduced through mTurk. We followed the same approach and created a paragraph typing task using HTML and JavaScript, loaded the task onto mTurk, and over the course of a few days asked 400 people type our paragraphs.

To estimate sequential structure in natural language (in this case English) we turned to  $n$ -gram analysis techniques.  $N$ -grams are identifiable and unique units of sequences, such as letters (a, b, c), bigrams (ab, bc, cd), and trigrams (abc, bcd, cde). Letters, bigrams, and trigrams appear in English texts with consistent frequencies. These frequencies can be estimated by counting the occurrence of specific  $n$ -grams in large corpuses of text. For example, Jones & Mewhort (2004) reported letter and bigram frequency counts from several different corpuses, and the google  $n$ -gram project provides access to  $n$ -gram counts taken from their massive online digitized repository of library books. Generally speaking, larger corpuses yield more accurate  $n$ -gram counts (Kilgarriff & Grefenstette, 2011).

Because we were also interested in examining typists' sensitivity to trigram frequencies, we conducted our own  $n$ -gram analysis by randomly selecting approximately 3000 English language eBooks from *Project Gutenberg*, and counting the frequency of each lowercase letter (26 unique), bigram (676 unique), and trigram (17576 unique) from that corpus. We restricted our analysis to  $n$ -grams containing lowercase letters and omitted all other characters because uppercase characters and many special characters require the use of the shift key, which produces much slower typing times than keystrokes for lowercase letters. Both single letter ( $r = 0.992$ ) and bigram frequencies ( $r = 0.981$ ) between the *New York Times* (Jones & Mewhort, 2004) and *Gutenberg* corpuses were highly correlated with one another. Additionally, inter-text letter, bigram, and trigram counts from the *Gutenberg* corpus were highly correlated with one another, showing that

the sequential structure that typists may be learning is fairly stable across English texts.

### Methodological Details

All typists copy-typed five *normal* paragraphs from the Simple English Wiki, a version of the online encyclopedia Wikipedia written in basic English. Four of the paragraphs were from the entry about cats (<http://simple.wikipedia.org/wiki/Cat>), and one paragraph was from the entry for music (<http://simple.wikipedia.org/wiki/Music>). Each normal paragraph had an average of 131 words (range 124–137). The paragraphs were representative of English texts and highly correlated with *Gutenberg* letter (26 unique letters, 3051 total characters,  $r = 0.98$ ), bigram (267 unique bigrams, 2398 total bigrams,  $r = 0.91$ ), and trigram frequencies (784 unique trigrams, 1759 total trigrams,  $r = 0.75$ ).

As part of an exploratory analysis we also had typists copy two paragraphs of non-English text, each composed of 120 five-letter strings. The strings in the *bigram* paragraph were generated according to bigram probabilities from our corpus counts, resulting in text that approximated the bigram structure of English text (i.e. a first-order approximation to English, (Mewhort, 1966, 1967). This paragraph was generally well correlated with the *Gutenberg* letter counts (24 unique, 600 total,  $r = 0.969$ ), bigram counts (160 unique, 480 total,  $r = 0.882$ ), and trigram counts (276 unique, 360 total,  $r = 0.442$ ).

The strings in the *random* letter paragraph were constructed by sampling each letter from the alphabet randomly with replacement. This paragraph was not well correlated with *Gutenberg* letter counts (26 unique, 600 total,  $r = 0.147$ ), bigram counts (351 unique, 479 total,  $r = -0.056$ ), or trigram counts (358 unique, 360 total,  $r = -0.001$ ).

Workers (restricted to people from the USA, with over 90% HIT completion rate) on mTurk found our task, consented, and then completed the task. The procedure was approved by the institutional review board at Brooklyn College of the City University of New York. Four hundred individuals started the task, however data were only analyzed for the 346 participants who successfully completed the task (98 men, 237 women, 11 no response). Participants reported their age within 5-year time bins, ranging from under 20 to over 66 years old (mean bin = 35 to 40 years old,  $\pm 2$  age bins). Two hundred and ninety-six participants were right-handed (33 left-handed, 11 ambidextrous, 6 no response). One hundred and thirty-five participants reported normal vision (202 corrected, 5 reported “vision problems”, 4 no response). Three hundred and twenty-nine participants reported that English was their first language (7 reported English being their second language, 10 no response). Participants reported that they had been typing between 1 and 60 years ( $M=20.2$  years,  $SE=9.3$ ), and had started typing at between 3 and 49 years old ( $M=13.3$  years old,  $SE = 5.5$ ). Two hundred and eighty

### 332 Behmer and Crump

participants reported being touch typists (63 not touch typists, 3 no response), and 187 reported having formal training (154 no formal training, 5 no response).

During the task, participants were shown each of the seven different paragraphs in a text box on their monitor (order randomized). Paragraph text was black, presented in 14 pt, Helvetica font. Participants were instructed to begin typing with the first letter in the paragraph. Correctly typed letters turned green, and typists could only proceed to the next by typing the current letter correctly. After completing the task, participants were presented with a debriefing, and a form to provide any feedback about the task. The task took around 30 to 45 minutes to complete. Participants who completed the task were paid \$1.

#### *The Data*

We collected inter-keystroke interval times (IKSIs; in milliseconds), for every correct and incorrect keystroke for each subject and each paragraph. Each IKSI is the difference between the timestamp for typing the current letter and the most recent letter. IKSIs for each letter were also coded in terms of their associated bigrams and trigrams. Consider typing the word *cat*. The IKSI for typing letter *t* (timestamp of *t* – timestamp of *a*) has the letter level *t*, the bigram level *at*, and the trigram level *cat*. In addition, each letter, bigram, and trigram has a frequency value from the corpus count.

In this way, for each typist we compiled three signatures of sensitivity to letter, bigram, and trigram frequency. For letters, we computed the vector of mean IKSIs for all unique correctly typed letters and correlated it with the vector letter frequencies. The same process was repeated for the vector of mean IKSIs for all unique correctly typed bigrams and trigrams. The resulting correlation values for each typist appear as individual dots in the figures that follow.

#### *Answering Questions with the Data*

##### *Are Typists Sensitive to Sequential Structure?*

Our first goal was to determine whether typing times for individual keystrokes are correlated with letter, bigram, and trigram frequencies. Looking at performance on the normal English paragraphs, for each subject we found the mean IKSIs for each unique correctly typed letter, bigram, and trigram, and then correlated (using Spearman’s rank correlation coefficient that tests for any monotonic relationship) these vectors with their respective frequency counts. A one-way analysis of variance performed on the correlations revealed a main effect for *n*-gram type [ $F(2, 1035) = 209, p = 0.001$ ]. Post-hoc *t*-tests ( $p = 0.016$ ) revealed that each mean was significantly different from one another, with mean correlations being greatest for letter ( $r = -0.410, SE=0.01$ ), bigram ( $r = -0.280, SE=0.004$ ), and then trigram

( $r = -0.220$ ,  $SE=0.003$ ). Additionally, one sample  $t$ -tests revealed that the mean correlation of each  $n$ -gram type was significantly greater than zero. All of the mean correlations were significant and negative, showing that in general, typing times are faster for higher than lower frequency  $n$ -grams. And, the size of the negative correlation decreases with increasing  $n$ -gram order, showing that there is more sensitivity to lower than higher-order structure. The major take home is that typists are indeed sensitive to sequential structure in the texts they type.

### ***Does Sensitivity to Sequential Structure Change with Expertise?***

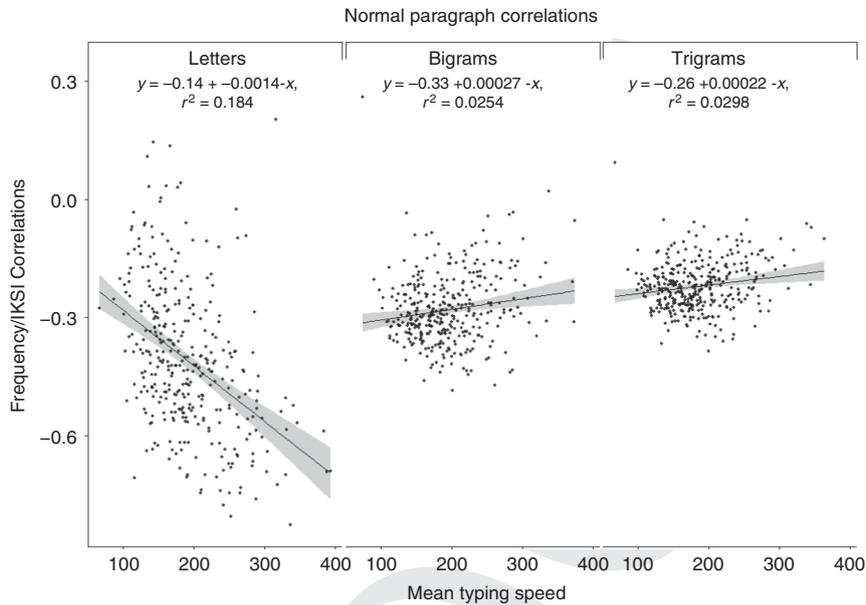
The more important question was whether or not sensitivity to  $n$ -gram frequency changes with expertise in the manner predicted by the learning and memory models. We addressed this question with a cross-sectional analysis. Our first step was to index the skill of our typists. As a simple proxy for skill we used the mean IKSI for each typist (overall typing speed), which assumes that faster typists are more skilled than slower typists. Overall typing speed is the x-axis in the following figures. The fastest typists are closer to the left because they have the smallest mean IKSI, and the slowest typists are closer to the right because they have the largest mean IKSI. Because we have many typists we expected to cover a wide range of skill, and indeed the figures show a nice spread across the x-axis.

Next, we plotted the previous measures of sensitivity to  $n$ -gram frequency for each typist as a function of their overall typing speed. So, the y-axis in the following graphs are correlation values for individual typists' between the vector of mean IKSI for individual  $n$ -grams and their frequencies. A score of 0 on the y-axis shows that a given typist was not sensitive to  $n$ -gram frequency. A negative value shows that a given typist was sensitive to  $n$ -gram frequency, and that they typed high frequency  $n$ -grams faster than low frequency  $n$ -grams. Positive values indicate the reverse. In general, most of the typists show negative values and very few show positive values.

Figure 14.3 shows the data from the normal paragraph condition. The first panel shows individual typists sensitivity to letter frequency. We expected that typists should be sensitive to letter frequency, and we see that most of the typists show negative correlations. Most important, sensitivity to letter frequency changes with typing skill. Specifically, the fastest typists show the smallest correlations, and the slowest typists show the largest correlations. In other words, there was a significant negative correlation between sensitivity to letter frequency and skill measured by overall typing speed ( $r = -0.452$ ,  $p < 0.001$ ). This finding fits with the prediction that sensitivity to lower-order statistics decreases over the course of practice. Our novices showed larger negative correlations with letter frequency than our experts.

Turning to the second and third panels, showing individual typist sensitivities to bigram and trigram frequencies as a function of mean typing speed, we see a qualitatively different pattern. Here we see that the faster typists on the left show

**334** Behmer and Crump



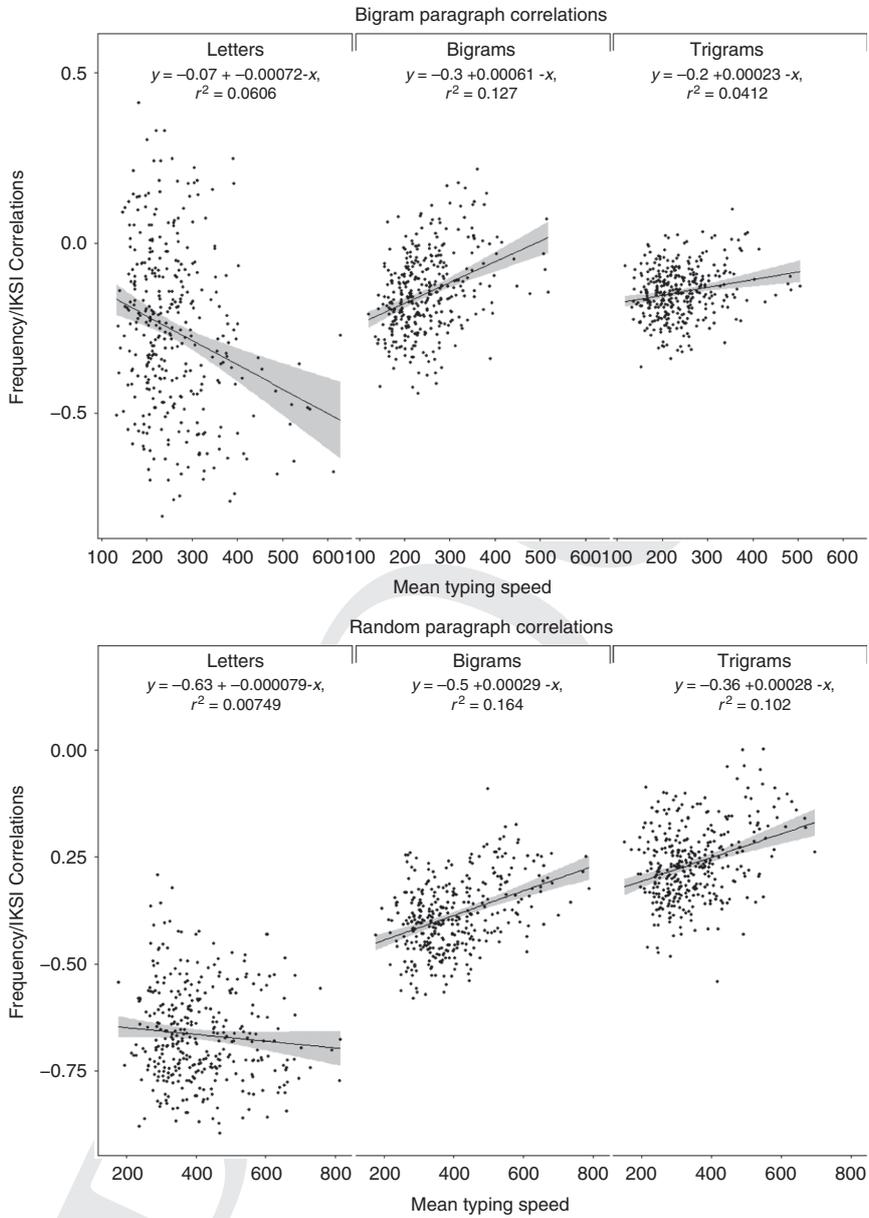
**FIGURE 14.3** Scatterplots of individual typist correlations between  $n$ -gram frequency and IKSI for each as a function of mean typing speed for normal paragraphs.

larger negative correlations than the slower typists on the right. In other words, there was a small positive correlation between sensitivity to bigram frequency and skill ( $r = 0.144$ ,  $p < 0.007$ ), and trigram frequency and skill ( $r = 0.146$ ,  $p < 0.006$ ). Again, predictions of the learning and memory models are generally consistent with the data, which show that highly skilled typists are more sensitive to higher-order sequential statistics than poor typists.

*Does Sensitivity to Sequential Structure Change when Typing Unfamiliar Letter Strings?*

Our typists also copy-typed two paragraphs of unfamiliar, non-word letter strings. The *bigram* paragraph was constructed so that letters appeared in accordance with their bigram-based likelihoods from the corpus counts, whereas the *random* paragraph was constructed by picking all letters randomly. A question of interest was whether our measures of typists’ sensitivity to  $n$ -gram structure in English would vary depending on the text that typists copied. If they do, then we can infer that utilization of knowledge about  $n$ -gram likelihoods can be controlled by typing context.

Figure 14.4 shows scatterplots of individual typists’ correlations between IKSI and letter ( $r = -0.214$ ,  $p < 0.001$ ;  $r = -0.080$ ,  $p = 0.13$ ), bigram ( $r = 0.345$ ,



**FIGURE 14.4** Scatterplots of individual typist correlations between  $n$ -gram frequency and IKSI for each as a function of mean typing speed for the bigram and random paragraphs.

### 336 Behmer and Crump

$p < 0.001$ ;  $r = 0.340$ ,  $p < 0.001$ ), and trigram ( $r = 0.171$ ,  $p < 0.001$ ;  $r = 0.244$ ,  $p < 0.001$ ) frequencies, respectively, for both the bigram and random paragraphs, and as a function of mean IKSI or overall typing speed. In general, we see the same qualitative patterns as before. For the bigram paragraph, the slower typists are more negatively correlated with letter frequency than the faster typists, and the faster typists are more negatively correlated with bigram and trigram frequency than the slower typists. For the *random* paragraph, the slope of the regression line relating mean typing speed to letter correlations was not significantly different from 0, showing no differences between faster and slower typists. However, the figures shows that all typists were negatively correlated with letter frequency. Typing random strings of letters disrupts normal typing (Shaffer & Hardwick, 1968), and appears to have turned our faster typists into novices, in that the faster typists' pattern of correlations looks like the novice signature pattern. It is noteworthy that even though typing random letter strings disrupted normal typing by slowing down mean typing speed, it did not cause a breakdown of sensitivity to sequential structure. We return to this finding in the general discussion.

### General Discussion

We examined whether measures of typing performance could test predictions about how learning and memory participate in the acquisition of skilled serial-ordering abilities. Models of learning and memory make straightforward predictions about how people become sensitive to sequential regularities in actions that they produce. Novices become tuned to lower-order statistics, like single letter frequencies, then with expertise develop sensitivity to higher-order statistics, like bigram and trigram frequencies, and in the process appear to lose sensitivity to lower-order statistics. We saw clear evidence of these general trends in our cross-sectional analysis of a large number of typists.

The faster typists showed stronger negative correlations with bigram and trigram frequencies than the slower typists. This is consistent with the prediction that sensitivity to higher-order sequential structure develops over practice. We also found that faster typists showed weaker negative correlations with letter frequency than the slower typists. This is consistent with the prediction that sensitivity to lower-order sequential structure decreases with practice.

### *Discriminating Between SRN and Instance Theory Models*

The instance theory and SRN model predictions that we were testing are globally similar. Instance theory predicts that sensitivity to higher-order sequential structure develops in parallel with sensitivity to lower-order sequential structure, albeit at a slower rate because smaller  $n$ -gram units are experienced more frequently than larger  $n$ -gram units. The SRN model assumes a scaffolding process, with sensitivity

to lower-order structure as a pre requisite for higher-order structure. So, both models assume that peaks in sensitivity to lower-order sequential structure develop before peaks in sensitivity to higher-order sequential structure. The data from our cross-sectional analyses are too coarse to evaluate fine differences in rates of acquisition of  $n$ -gram structure across expertise.

However, the models can be evaluated on the basis of another prediction. The SRN model assumes that experts who have become sensitive to higher-order sequential statistics have lost their sensitivity to lower-order statistics. Instance theory assumes sensitivity to lower-order statistics remains and grows stronger with practice, but does not influence performance at high levels of skill. Our data from the normal typing paragraphs show that faster typists had weaker sensitivity to letter frequency than slower typists. However, we also found that all typists showed strong negative correlations with letter frequency when typing the random paragraph. So, the fastest typists did not lose their sensitivity to lower-order structure. This finding is consistent with the predictions of instance theory. Experts show less sensitivity to letter frequency when typing normal words because their typing speed hits the floor and they are typing at maximum rates, however their sensitivity to letter frequency is revealed when a difficult typing task forces them to slow down.

### ***Relation to Response-Scheduling Operations***

We divided our questions about serial-ordering into issues of *abilities* and *operations*. Our data speak to the development of serial-ordering abilities in skilled typing. They are consistent with the hypothesis that general learning and memory processes do participate in hierarchically controlled skills, like typing. However, our data do not speak directly to the nature of operations carried out by a response-scheduling process controlling the serial-ordering of keystrokes. A central question here is to understand whether/how learning and memory processes, which clearly bias keystroke timing as a function of  $n$ -gram regularities, link in with theories of the response-scheduling process. We took one step toward addressing this question by considering the kinds of errors that our typists committed.

The operations of Rumelhart & Norman’s (1982) response-scheduling process involve a buffer containing activated letter-keystroke schemas for the current word that is being typed. The word activates all of the constituent letter schemas in parallel, and then a dynamic inhibition rule weights these activations enabling generally accurate serial output. We considered the possibility that learning and memory representations sensitive to  $n$ -gram statistics could bias the activation of action schemas in the buffer as a function of  $n$ -gram context, by increasing activation strength of letters that are expected versus unexpected according to sequential statistics. We assumed that such an influence would produce what we term “statistical” action

338 Behmer and Crump

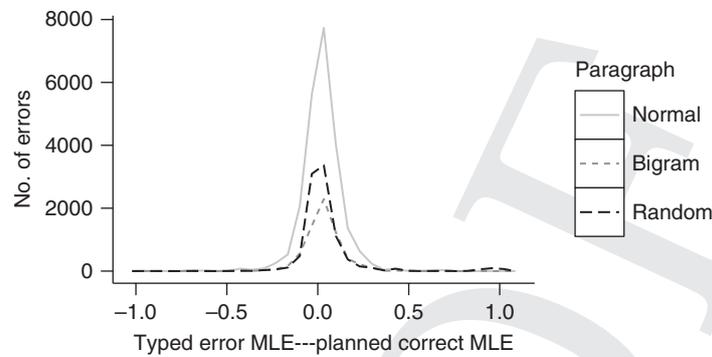


FIGURE 14.5 Histograms of the distribution of differences between the maximum likelihoods for the typed error letter and the planned correct letter based on the bigram context of the correct preceding letter.

slips (Norman, 1981), which might be detected in typists’ errors. For example, consider having to type the letters “qi”. Knowledge of sequential statistics should lead to some activation of the letter “u”, which is more likely to follow “q” than “i”. We examined all of our typists incorrect keystrokes following the intuition that when statistical slips occur, the letters typed in error should have a higher maximum likelihood expectation given the prior letter than the letter that was supposed to be typed according to the plan. We limited our analyses to erroneous keystrokes that were preceded by one correct keystroke. For each of the 38,739 errors, we subtracted the maximum likelihood expectation for the letter that was typed in error given the correct predecessor, from the maximum likelihood expectation for the letter that was supposed to be typed given the correct predecessor.

Figure 14.5 shows the distributions of difference scores for errors produced by all typists by paragraph typing conditions. If knowledge of sequential statistics biases errors, then we would expect statistical action slips to occur. Letters typed in error should have higher likelihoods than the planned letter, so we would expect the distributions of difference scores to be shifted away from 0 in a positive direction. None of the distributions for errors from typing any of the paragraphs are obviously shifted in a positive direction. So, it remains unclear how learning and memory processes contribute to the operations of response-scheduling. They do influence the keystroke speed as a function of  $n$ -gram frequency, but apparently do so without causing a patterned influence on typing errors that would be expected if  $n$ -gram knowledge biased activation weights for typing individual letters. Typists make many different kinds of errors for other reasons, and larger data sets could be required to tease out statistical slips from other more common errors like hitting a nearby key, transposing letters within a word, or missing letters entirely.

### ***Big Data at the Finger Tips***

We used Big Data tools to address theoretical issues about how people develop high-level skill in serial-ordering their actions. We asked how typists “analyze” Big Data that comes in the form of years of experience typing, and apply knowledge of sequential structure in that data to their actions when they are typing. Our typists’ learning and memory processes were crunching Big Data with their finger tips. More generally, Big Data tools relevant for experimental psychology are literally at the finger tips of researchers in an unprecedented fashion that is transforming the research process. We needed to estimate the statistical structure of trigrams in the English language, and accomplished this task in a couple of days by downloading and analyzing freely available massive corpuses of natural language, which were a click away. We needed hundreds of people to complete typing tasks to test our theories, which we accomplished in a couple of days using freely available programming languages and the remarkable mTurk service. We haven’t figured out how to use Big Data to save time thinking about our results and writing this paper. Nevertheless, the Big Data tools we used dramatically reduced the time needed to collect the data needed to test the theories, and they also enabled us to ask these questions in the first place. We are excited to see where they take the field in the future.

### **Acknowledgment**

This work was supported by a grant from NSF (#1353360) to Matthew Crump. The authors would like to thank Randall Jamieson, Gordon Logan, and two anonymous reviewers for their thoughtful comments and discussion in the preparation of this manuscript.

### **References**

- Borel, É. (1913). La mécanique statique et l’irréversibilité. *Journal de Physique Theorique et Appliquee*, 3, 189–196.
- Botvinick, M. M., & Plaut, D. C. (2004). Doing without schema hierarchies: A recurrent connectionist approach to normal and impaired routine sequential action. *Psychological Review*, 111, 395–429.
- Botvinick, M. M., & Plaut, D. C. (2006). Such stuff as habits are made on: A reply to Cooper and Shallice (2006). *Psychological Review*, 113, 917–927.
- Cleeremans, A. (1993). *Mechanisms of implicit learning connectionist models of sequence processing*. Cambridge, MA: MIT Press.
- Cleeremans, A., & McClelland, J. L. (1991). Learning the structure of event sequences. *Journal of Experimental Psychology: General*, 120, 235–253.
- Cooper, R. P., & Shallice, T. (2000). Contention scheduling and the control of routine activities. *Cognitive Neuropsychology*, 17, 297–338.
- Cooper, R. P., & Shallice, T. (2006). Hierarchical schemas and goals in the control of sequential behavior. *Psychological Review*, 113, 887–916.

**340** Behmer and Crump

- Crump, M. J. C., & Logan, G. D. (2010a). Hierarchical control and skilled typing: Evidence for word-level control over the execution of individual keystrokes. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*, 1369–1380.
- Crump, M. J. C., & Logan, G. D. (2010b). Warning: This keyboard will deconstruct—The role of the keyboard in skilled typewriting. *Psychonomic Bulletin & Review*, *17*, 394–399.
- Crump, M. J. C., McDonnell, J. V., & Gureckis, T. M. (2013). Evaluating Amazon's Mechanical Turk as a tool for experimental behavioral research. *PLoS ONE*, *8*, e57410.
- Dell, G. S., Burger, L. K., & Svec, W. R. (1997). Language production and serial order: A functional analysis and a model. *Psychological Review*, *104*, 123–147.
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.
- Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). Washington, DC: V. H. Winston & Sons.
- Fodor, J. A. (1983). *The modularity of mind: An essay on faculty psychology*. Cambridge, MA: MIT Press.
- Gentner, D. R., Grudin, J., & Conway, E. (1980). *Finger movements in transcription typing*. DTIC Document. San Diego, CA: University of California, San Diego, La Jolla Center for Human Information Processing.
- Grudin, J. T., & Larochelle, S. (1982). *Digraph Frequency Effects in Skilled Typing*. DTIC Document. San Diego, CA: University of California, San Diego, La Jolla Center for Human Information Processing.
- Jamieson, R. K., & Mewhort, D. J. K. (2009a). Applying an exemplar model to the artificial-grammar task: Inferring grammaticality from similarity. *The Quarterly Journal of Experimental Psychology*, *62*, 550–575.
- Jamieson, R. K., & Mewhort, D. J. K. (2009b). Applying an exemplar model to the serial reaction-time task: Anticipating from experience. *The Quarterly Journal of Experimental Psychology*, *62*, 1757–1783.
- Jones, M. N., & Mewhort, D. J. (2004). Case-sensitive letter and bigram frequency counts from large-scale English corpora. *Behavior Research Methods, Instruments, & Computers*, *36*, 388–396.
- Kilgariff, A., & Grefenstette, G. (2003). Introduction to the special issue on the web as corpus. *Computational Linguistics*, *29*, 333–347.
- Lashley, K. S. (1951). The problem of serial order in behavior. In L. A. Jeffress (Ed.), *Cerebral mechanisms in behavior*. New York: Wiley.
- Liu, X., Crump, M. J. C., & Logan, G. D. (2010). Do you know where your fingers have been? Explicit knowledge of the spatial layout of the keyboard in skilled typists. *Memory & Cognition*, *38*, 474–484.
- Logan, G. D. (1988). Toward an instance theory of automatization. *Psychological Review*, *95*, 492–527.
- Logan, G. D., & Crump, M. J. C. (2009). The left hand doesn't know what the right hand is doing: The disruptive effects of attention to the hands in skilled typewriting. *Psychological Science*, *20*, 1296–1300.
- Logan, G. D., & Crump, M. J. C. (2010). Cognitive illusions of authorship reveal hierarchical error detection in skilled typists. *Science*, *330*, 683–686.
- Logan, G. D., & Crump, M. J. C. (2011). Hierarchical control of cognitive processes: The case for skilled typewriting. In B. H. Ross (Ed.), *The Psychology of Learning and Motivation* (Vol. 54, pp. 1–27). Burlington: Academic Press.

- McCloskey, M., & Cohen, N. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24, 109–165.
- Mewhort, D. J. (1966). Sequential redundancy and letter spacing as determinants of tachistoscopic recognition. *Canadian Journal of Psychology*, 20, 435.
- Mewhort, D. J. (1967). Familiarity of letter sequences, response uncertainty, and the tachistoscopic recognition experiment. *Canadian Journal of Psychology*, 21, 309.
- Miller, G. A., Galanter, E., & Pribram, K. H. (1960). *Plans and the structure of behavior*. New York, NY: Adams-Bannister-Cox.
- Nissen, M. J., & Bullemer, P. (1987). Attentional requirements of learning: Evidence from performance measures. *Cognitive Psychology*, 19, 1–32.
- Norman, D. A. (1981). Categorization of action slips. *Psychological Review*, 88, 1–15.
- Palmer, C., & Pfordresher, P. Q. (2003). Incremental planning in sequence production. *Psychological Review*, 110, 683–712.
- Reber, A. S. (1969). Transfer of syntactic structure in synthetic languages. *Journal of Experimental Psychology*, 81, 115–119.
- Rosenbaum, D. A., Cohen, R. G., Jax, S. A., Weiss, D. J., & van der Wel, R. (2007). The problem of serial order in behavior: Lashley's legacy. *Human Movement Science*, 26, 525–554.
- Rumelhart, D. E., & Norman, D. A. (1982). Simulating a skilled typist: A study of skilled cognitive-motor performance. *Cognitive Science*, 6, 1–36.
- Shaffer, L. H., & Hardwick, J. (1968). Typing performance as a function of text. *The Quarterly Journal of Experimental Psychology*, 20, 360–369.
- Snyder, K. M., Ashitaka, Y., Shimada, H., Ulrich, J. E., & Logan, G. D. (2014). What skilled typists don't know about the QWERTY keyboard. *Attention, Perception, & Psychophysics*, 76, 162–171.
- Watson, J. B. (1920). Is thinking merely action of language mechanism? *British Journal of Psychology, General Section*, 11, 87–104.
- Wickelgren, W. A. (1969). Context-sensitive coding, associative memory, and serial order in (speech) behavior. *Psychological Review*, 76, 1–15.