Error Identification and Correction in Human Computation: Lessons from the WPA

David Alan Grier
Center for International Science and Technology Policy
George Washington University
grier@gwu.edu

Abstract

Human computing promises new capabilities that cannot be easily provided by computing machinery. However, humans are less disciplined than their mechanical counterparts and hence are liable to produce accidental or deliberate mistakes. As we start to develop regimes for identifying and correcting errors in human computation, we find an important model in the computing groups that operated at the start of the 20th century.

Introduction

Far from being a new idea, Human Computation has a long history that predates Amazon Mechanical Turk, the Internet, software and even the development of the electronic digital computer. Organized computing groups appeared in the 18th century, flourished in the 19th, and reached their zenith in the first decades of the 20th. These groups were found in organizations such as *Royal Nautical Almanac*, the *Connaissance des Temps*, the British Association for the Advancement of Science, the U. S. Coast and Geodetic Services, and the Works Progress Administration. Working with tools no more sophisticated than pen and pencil, these offices were able to complete major computations, tasks that were far beyond the ability of a single individual working with the best the calculating machinery of the day.

All of these organizations were concerned with errors in their work, errors that could propagate from their results through the calculations of others. Rather than resort to the obvious techniques of multiple calculation or highly disciplined verification, these groups developed procedures for error identification and correction. These procedures involved careful numerical analysis and disciplined managerial instructions. Like so many operational procedures, these methods produced good results not by demanding the best qualities of the human computers but by providing symbolic checks against their worst lapses.

In reviewing the historical methods for error identification and correction, we see that they were designed for the kinds of labor markets that we see in modern human computation and that they provide a model for addressing new problems. This model includes a well-defined computing plan, tools for guiding the human computers, and a set of computational procedures that identify the kinds of errors that are most likely to be made. Most importantly, this model required that all that all human calculations be bounded.

Human Computing Organization

For this paper, a human computing organization will be defined to require three distinct elements: a labor pool or workers, tasks that are based on the fine division of labor, and a market mechanism that is used for assigning tasks to workers.

The labor pool consists of workers who have the minimal set of skills to do the work that is required of them. However, we make no assumptions about their ability to understand the goal or the computation or to identify results that are inconsistent with that goal.

The tasks are elementary units of work. By assuming that these tasks are based on a fine division of labor, we accept that they may they might convey no information about the overall goal of the calculation. For example, the computing rooms of large observatories divided the calculation of orbital ephemerides into a large number of simple additions. The computers in these rooms did the additions without knowing how they related to the final orbital positions.

The market mechanism is a place where the organization can place requests for calculation and human computers can accept those requests. In the historical computing offices, this mechanism was a set of shelves that held worksheets. The worksheets were pages that held the request for calculation. The human computers could select the sheets that they wanted to handle, take them to their desk or home, and return them when they were finished. Historically, these markets operated on piece –work rules. They paid the computers by the sheet. However, the offices that adopted factory methods usually paid workers by the day.

These basic elements of human computation were developed in the 18th century and were discussed by Charles Babbage (1791-1871) in his 1832 book, *On the Economy of Machines and Manufactures*. Building on the work of Adam Smith, Babbage argued that organized computation demonstrated that "the division of labour can be applied with equal success to mental as to mechanical operations, and that it ensures in both the same economy of time."(Babbage 1832)

In writing about human computation, Babbage recognized the problem of identifying and correcting errors and articulated a fundamental observations about human computation that has since been called "Babbage's Law of Errors." He noted that two individuals who used the same computing methods on a the problem were likely to make the same mistake.(Buxton 1988)

Babbage was a highly dedicated calculator. His 1827 table of logarithms is considered "one of the most accurate ever printed."(Fletcher 1946). In his frustration with computational errors, Babbage turned from human calculation to the design of computation machinery. This approach proved to be impracticable as almost a century would pass before his simplest machine, called the Difference Machine, would make a substantial impact on calculation.(Comrie, 1936) In the intervening years, scientists and engineers could only complete large calculations by organizing a computing and finding a way to control error.

The organization that probably did the best job of controlling errors was the Mathematical Tables Project, which operated in New York City from 1938 to 1948. As its name implies, the Mathematical Tables Project created large volumes of tables of higher mathematical functions and other complicated scientific calculations. As the group was a relief project of the Works Progress Administration, its leaders were very concerned about establishing a reputation or producing error-free calculation.(Grier 2005)

The leaders of the group regularly claimed that their work contained no mistakes. An independent assessment of the tables concluded that this claim was not true. However, they acknowledged the errors were both minor and rare. The 28 volumes of tables, most of which had more 500 pages of numbers, contained only a handful of errors.(Fletcher 1946)

Planning and Errors

The Mathematical Tables Project was managed by an operational group called the Planning Committee. This committee, lead by the mathematician Gertrude Blanch, analyzed proposed calculations and prepared plans for the calculation. Part of this work was similar to modern numerical analysis. They would look for an algebraic approximation for the function, study how quickly that approximation converged and determine the difference between any computed value and the underlying function. In developing these plans, Blanch and her lieutenants tended to utilized conventional approximations such as Taylor series, though Blanch had a fondness for continued faction expansions(Blanch 1945)

In addition to doing the standard numerical analysis, the Planning Committee had to divide the calculations into elementary tasks, prepare worksheets to guide the computers through the tasks, and create procedures for assembling the results of the computations into a complete table. For the most part, the Mathematical Tables Project utilized three elementary tasks: multi-digit addition, multi-digit subtraction, and the multiplication of a multi-digit number by a single digit. In a few cases, they identified long division as an elementary task but they had few human computers who were able to handle this task. Even subtraction proved difficult at times. To avoid confusion, the ultimately recorded all negative numbers with red pencils.(Grier 2005)

As they developed the plan, Blanch and her committee tried to identify errors by bounding key calculations. While they could not bound every single elementary calculation, they had identified bounded values for every step of the calculation including those steps that assembled the final values and transferred them to a typescript. For some of these values, they developed bounds using the underlying properties of the underlying calculation. For others, they utilized a general technique that was derived from the method of finite differences.

Bounding Calculations

Targeted Values

As the first step in its plan, Gertrude Blanch would divide the computation into a small set of open-ended calculations and a large group of interpolations. The open-ended calculations would be done by the members of the Planning Committee, all of whom held advanced degrees in mathematics, physics or astronomy. They generally computed these values with a high-order Taylor expansion. They would compare their results to known values and circulate the computations among themselves to check for errors. For their first large calculation, a book of the exponential function, they computed 28 such values.

The seed values created a box for the final tables. They included the first value of the table, the last value, and a regularly spaced set of points within those two values. Once they had produced these values, they would create a procedure for the bounded computations. These computations would start at one seed value and compute a series of function values at regular intervals. This series would end at the next seed value, giving the human computers a clear target. If the last computation did not equal the next seed value, the sheet contained an error.

Blanch required that all values be double computed. As she was knew about Babbage's Law of Errors she used a different equation for each computation. In general, she created one equation that would take the computations from the low seed value to a higher seed value and a second equation that would work backward from the higher seed value to the lower.

For the exponential function, she based her two equations on the observation that $e^{x+h}=e^x(O+E)$ and $e^{x-h}=e^x(O-E)$, where h, was the interval at which values of the function were to be computed. The values O and E were constants that were computed from the odd and even terms of the Taylor expansion, respectively. These values were computed and checked by the Planning Committee. Using this observation, the committee produced two sets of worksheets. One computed the values that began with the low seed S_1, and moved upwards through $S_1(O+E)$, $S_1(O+E)^1$, $S_1(O+E)^2$, and so on until they reached the second seed value S_2. The second set began at S_2 and moved downward through $S_2(O-E)$, $S_2(O-E)^1$, $S_1(O-E)^2$.

As $(O+E)$ and $(E-O)$ were different values, this procedure produced the same number in two different ways. Any disagreement between the two sets of numbers or any final calculation that did not match its final seed would send the worksheet back to be recalculated.

Checking Calculations

Once the worksheets were completed, the Planning Committee assembled the final values into the final tables. As they did, they subjected the numbers to two kinds of tests. The first was done before the transfer. The tests in this group relied on the mathematical properties of the underlying function. For the exponential function, they used two tests. The first computed weighted sums of 3 adjacent values. This value should small in absolute value, less than 10^{-25}. The second test used sums of 10 consecutive values. These sums should equal a known target.

These functional tests not only identified accidental errors, they quickly revealed deliberate errors. Such errors appeared from time to time when one of the computers would attempt to avoid work by inventing numbers to place in the worksheet but would attempt to protect themselves by entering a final number that equaled the target seed. In such cases, the functional tests would produce values well outside the expected range.

Once the raw calculations were tested they were typed on to a mimeographed stencil and tested once again. If the Planning Committee found an error, they would erase the bad number with hot wax and retype it. To check these values, they used a general-purpose test that would work for all functions and also identify errors of transcription. This method was based on the technique of forward differences.

These assistants would check the results by computing certain values from neighboring numbers. On the exponential table, for example, they employed two such tests. On the first, they computed a weighted sum of three adjacent values. On the second, they computed the sum of ten adjacent numbers. The first value would indicate an error if it was bigger than 10^{-6}. The second would indicate an error if it did not equal the original value times a known constant. By performing these calculations across different groups of adjacent numbers, they could isolate the problem calculations.

The Technique of Differencing

Though human computers regularly used the method of differencing to find errors, they left only a scant record of its theory and application. Blanch treated it slightly in her book about WPA computational methods. (Blanch 1945) In an article that was intended to be the first installment of multi-part series on the method, the mathematician J.C.P.

Miller noted that the "precise details of [differencing] and its pitfalls," were "never set out fully in print."(Miller, 1950) In spite of this observation, Miller never completed any article beyond the first of the series.

The technique of differencing to identify errors was replaced to the use of differencing to interpolate a function. To look for errors, a member of the Planning Committee would compute a series of first differences, which was done by computing the difference between any given number and the next number in the table.

After computing first differences, the Planning Committee would compute second differences by computing a similar set of differences from the first differences. Following this process recursively, they would compute a set of third, fourth and higher differences. In general, they computed at least fourth order differences. In many cases they computed differences up to the eighth or tenth order differences.

To use differences to search for errors, the Planning Committee made the assumption that the function was well approximated by a polynomial, at least within a small range. Just as the derivatives of a polynomial eventually become zero, the high order differences of a polynomial will also become zero. Therefore, the Planning Committee expected the high order differences to be within a certain narrow band that was centered on zero. (Blanch 1945).

Differencing worked fairly well with functions that were not close to polynomials, such as the exponential function, or for finding transcription errors. A difference of order k was actually a sum of k computed values that were multiplied by the binomial coefficients of order k. As binomial coefficients get large quickly as k increases, they would ultimately identify an erroneous value by inflating its role in some difference and producing large value that fell outside the target range.

Miller noted that differences should fall within fixed bounds but admitted that he had no fixed rule for calculating that range. "The determination of exact theoretical probabilities for differences of various sizes is a matter of some difficulty." He argued that human computers could learn how to use the technical of differencing "from experience and the examination of many tables."(Miller 1950) In her book, Blanch suggests that the Mathematical Tables Project not only used differences to identify errors but also to correct errors. She shows how to do this in a few simple settings but never develops a general theory. (Blanch 1945)

Lessons for Modern Human Computation

At this point in their development, human computers are not likely to repeat the same kinds of detailed numerical computations that were handled by the Mathematical Tables Project, or the Coast Survey, or the Connassiance des Temps. At the same time, if human computers represent "artificial artificial intelligence", as they are often characterized, then they will be employed in circumstances in which they will be making quantitative judgments that may be every bit as complicated as the traditional numerical computations. Furthermore, other tasks for human computation have useful parallels in these classic methods. (von Ahn, 2009)

The history of human computers offers a rich literature that could be used not only to recreate the computational methods of the 1930s but also provide a basis for error identification and correction techniques for modern computing groups. The techniques of the Mathematical Tables Project are documented in the prefaces of their 28 books of tables. Furthermore, the veterans of the group produced the well-circulated *Handbook of Mathematical Functions*, which presents many of the computation techniques used by the Mathematical Tables Project though only a limited discussion of error correction and detection.

A few articles that discuss error detection and correction can be found in the periodical *Mathematical Tables and Other Aids to Computation*, which is available on JSTOR. One of the leading computers of the age, L. J. Comrie, left an important discussion of errors and error detection in an index to mathematical tables that was complied by his colleagues. (Fletcher, 1946) Finally, Blanch recorded her thoughts on error identification in two manuscripts. (Blanch 1945, Blanch undated).

The accomplishments of the early computing groups provide four obvious lessons for the modern human computation of numerical values. The most important of these lessons is the one that provides a target for all calculations done by the human computers. We should never ask human computers to do a calculation unless we have prior estimate for that calculation. We can use that estimate to restrict the values that can be entered onto a computer screen or to check the final values.

The second lesson suggests that we need to have multiple ways of checking for errors, especially if we are looking to have highly reliable answers. We need to have tests that are based on specific properties of the underlying functions and as well as general methods that can capture many varieties of errors.

The third lesson addresses Babbage's Law of Errors and suggests that each value be computed at least twice by two different methods. This lesson suggests that human computation is going to be much less efficient than mechanical calculation. However, if we have no other technique, we will have to tolerate such inefficiencies.

Finally, the last lesson of the old human computing groups is the obvious lesson that human computation has to be well planned and that we need to identify all possible sources of errors and the means of identifying errors. When the Mathematical Tables Project prepared its computing plans, it generally circulated them to a few well-established mathematicians for their comments and criticism. Blanch and the other leaders of the group deeply desired to have the approval of the scientific community and knew that a poorly prepared table could permanently damage their reputation. From the start, the group was dismissed by leaders of the National Academy of Sciences, who reasoned that people on work relief could not be trusted to be accurate. Modern human computation could suffer the same fate if its leaders do not take the same care in preparing its computations.

References

Abramowitz, M., and Stegun, I. eds. 1964. *A Handbook of Mathematical Functions.* Washington, DC.: US Government Printing Office.

Babbage, C. 1832. *On the Economy of Machines and Manufactures.* London: Charles Knight.

Blanch, Gertrude. 1986. *Lectures on the Theory of Numerical Computation.* New York, NY.: WPA.

Blanch, Gertrude. Undated manuscript. *Numerical Analysis.* Gertrude Blanch Papers, Charles Babbage Institute.

Buxton, H. 1988(reprint). *Memoir of the Life and Labors of the Late Charles Babbage Esq.* Cambridge, Mass.: MIT Press.

Comrie, L. J. 1936. Inverse Interpolation and the National Accounting Machine. *Journal of the Royal Statistical Society.* 3:87+.

Fletcher, A., Miller, J. C., and Rosenhead, L. 1946. *An Index of Mathematical Tables.* London.: Scientific Computing Services.

Grier, David Alan, 2005, *When Computers Were Human.* Princeton, NJ.: Princeton University Press.

Miller, J. C. .. 1950. Checking by Differences. *Mathematical Tables and Other Aids to Computation.* 4:29 3-11.

Von Ahn, L. 2009. Human Computation. *Proceedings of the 46th Annual Design Automation Conference.*