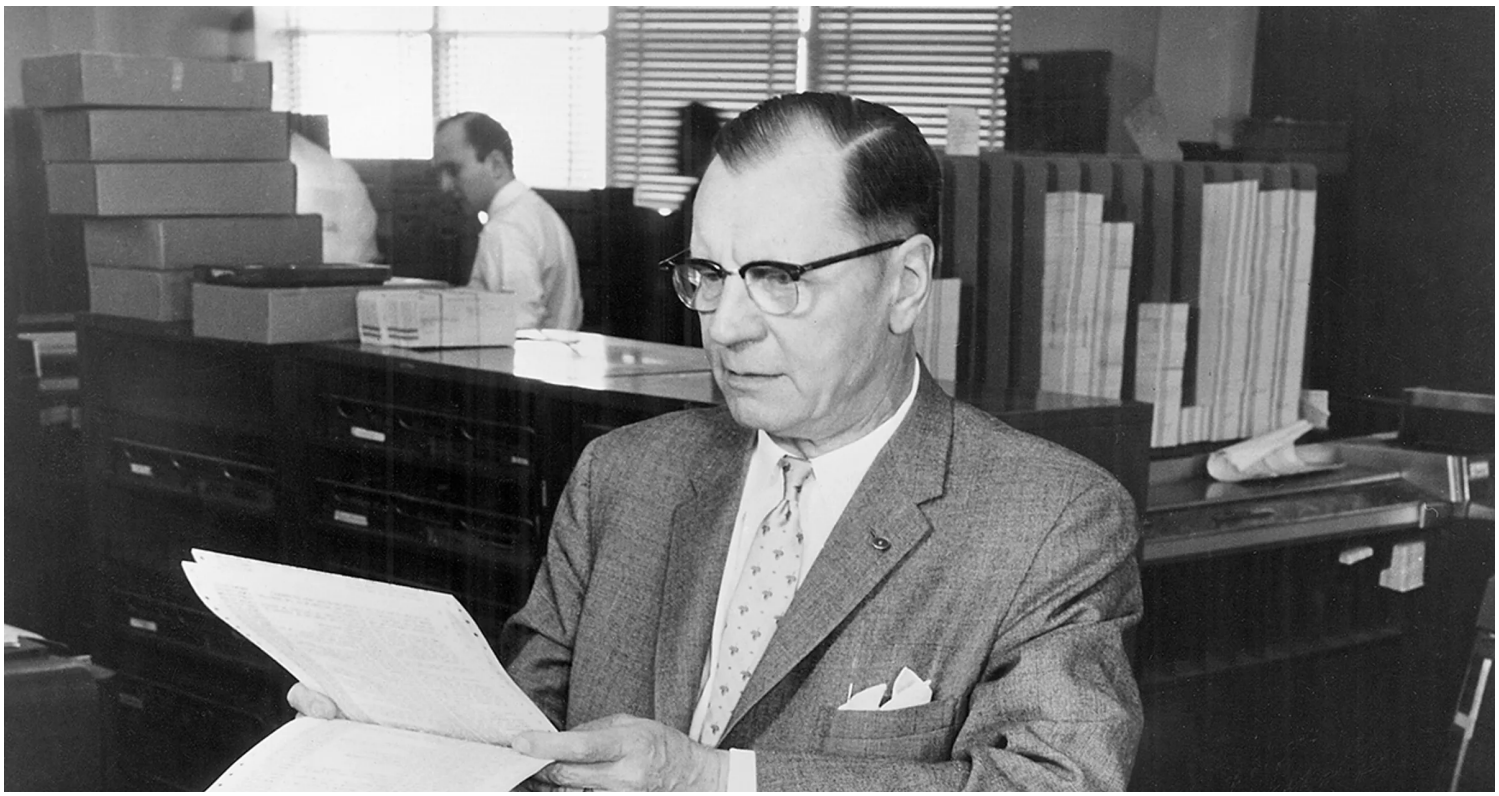
IEEE Spectrum

FEATURE HISTORY OF TECHNOLOGY

Hans Peter Luhn and the Birth of the Hashing Algorithm

The IBM engineer's hashing algorithm gave computers a way to quickly search documents, DNA, and databases

BY HALLAM STEVENS 30 JAN 2018



Information Scientist: Starting in the 1940s, Luhn devised machines and schemes for parsing information, most notably the now widely used hashing algorithm, which he suggested as a way to sort both numbers and text. PHOTO: IBM

IN NOVEMBER 1958, at a six-day international conference devoted to scientific information, the inventor Hans Peter Luhn demonstrated a series of his electromechanical machines. They looked rather ordinary. Much like other computing devices of the day, they were boxy and utilitarian, designed to scoop and sort tall stacks of punch cards into slots and bins.

Unlike other computers, however, Luhn's devices were not designed to work with numbers and calculations but rather with words and sentences. One machine that drew particular attention implemented an algorithm that Luhn called KWIC, for Key Word in Context. Taking in a large number of texts—typically, articles from 500 to 5,000 words in length—the KWIC system could quickly and automatically construct a kind of index.

At the time, indexing, classifying, and organizing written information was a painstaking process, even for the most experienced specialists. And the volume of information in many fields was growing too rapidly for anyone to keep up. A better means for abstracting and summarizing was desperately needed.

For the otherwise staid gathering of librarians and information scientists in Washington, D.C., the demonstration of KWIC was nothing short of earthshaking, with newspapers across the United States reporting on Luhn's astounding invention.

By the early 1960s, KWIC had become central to the design of hundreds of computerized indexing systems, including those used by the Chemical Abstracts Service, Biological Abstracts, and the Institute for Scientific Information. One expert called KWIC "the greatest thing to happen in chemistry since the invention of the test tube." Luhn, a senior engineer at IBM, also built KWIC into an "intelligence system" for businesses, designed to identify and then deliver relevant information to specific individuals within a large organization. KWIC was basically the era's equivalent of a search engine: It allowed users to speedily locate the information they needed.

We now take for granted that computers can make sense of information, readily offering up restaurant reviews, sports scores, and stock prices on demand. In Luhn's time, though, computers were crude and simple. His attempts to manipulate text contributed to a more expansive way of thinking about computers and their capabilities, and his ideas still underpin modern-day algorithms that we use for online shopping, automatic translation, and genetic research. In the 1950s, of course, many of these applications weren't even conceivable. Here I'll explore what led Luhn to the solution of a problem that didn't yet exist—a solution called the hash function.

The years following World War II were formative ones for electronic computers. Various kinds of computers built during the war made vital calculations for ballistics, atomic weapons, and cryptography. Cold War tensions ensured continued funding for the development of computers, and as a result they grew faster, more accurate, and more powerful. But their main uses—crunching and storing numbers—changed little.

Within this nascent computer world, Luhn cut an unusual figure. An elegant dresser throughout his life, Luhn knew more about the textiles industry than computer science when he arrived at IBM in 1941. His many inventions seemed to belong to an earlier, predigital era of mechanical calculators and slide rules. Even in the 1950s, digital computers were supplanting his electromechanical devices. Nevertheless, his ideas, transformed and remixed for a variety of purposes, are now embedded in almost every kind of software we can think of.

Luhn was born in Barmen, Germany, in 1896. His father, Johann, was a master printer, prosperous and apparently very tolerant of his children's endeavors. At one point, Luhn and his younger siblings built a miniature railroad in the family garden, including 70 meters of track made by melting down printer's lead.

After high school, Luhn went to Switzerland to learn the family trade. But World War I and a stint in the German army interrupted his printing career, and after the war, he landed in the

textile trade. Luhn came to the United States in 1924 to scout for potential locations for textile mills. Even in textiles, Luhn's inventive bent was apparent. In 1927, he developed a rulerlike device that could be used to gauge the thread count of cloth. The Lunometer is still sold by H.P. Luhn & Associates, an engineering consulting company that Luhn founded.

Luhn was a quick study, absorbing information from a wide variety of fields and becoming in turn a proficient mountain climber, gourmet cook, and expert landscape painter. During the 1930s, his numerous patents included a foldable raincoat, a device for shaping women's stockings, a game table, and the "Cocktail Oracle"—a guide that told the user what drinks could be made from the ingredients on hand.

Drinks, Anyone? Luhn’s inventive mind roamed far and wide. In 1933, shortly before the end of Prohibition, he filed for a U.S. patent on a recipe guide that helped the user create cocktails from materials on hand. ILLUSTRATION: U.S. PATENT AND TRADEMARK OFFICE

But Luhn’s real interest was in the storage, communication, and retrieval of information, especially text, and it was largely to pursue those interests that he joined IBM. Given the title of “inventor,” Luhn was prolific—he ended up producing 70 patents

for IBM. Although he had the latitude to tackle whatever problems he liked, many of his inventions focused on using machines, including electronic computers, for manipulating information.

In 1946 and 1947, for example, Luhn worked on creating machine-readable typewritten documents. One device consisted of a metallic ribbon inserted into a typewriter, which punched magnetic patterns onto paper that could then be scanned by machine. Shortly afterward, he began to work with two MIT chemists, Malcolm Dyson and James Perry, on a machine that could automatically search through chemical compounds using punch cards. Each punch card was encoded with information about a particular compound. The user inserted a "question card" into the machine listing a set of criteria against which all the compound cards could be compared and sorted. Although Luhn's scanner was highly specialized, he continued to look for more general-purpose ways to automatically process information.

Information was very much on people's minds. The postwar years saw an explosion in the number of published papers in science and engineering. Many experts worried that "information overload" threatened to overwhelm researchers and businessmen alike. Vannevar Bush, a leader of America's massive wartime scientific bureaucracy and one of the architects of the National Science Foundation, proposed a desk-size electromechanical device, the Memex, for storing and linking together information.

Bush's idea was never realized, but Luhn's ideas were. On 6 January 1954, for instance, he filed for a U.S. patent on a "Computer for Verifying Numbers" [PDF]. This handheld mechanical device aimed to solve a simple practical problem. At the time, various kinds of identification numbers, such as credit card numbers and Social Security numbers, were beginning to play an important role in public and private life. But the numbers were difficult to remember, and they could be transcribed incorrectly or deliberately falsified. What was needed was a means of quickly verifying whether an ID number was valid.

Luhn's handheld computer did that, using a checksum algorithm he developed. For a 10-digit number, the computer would perform the following steps:

- *Double every second digit*
- *If any result is 10 or greater, add up the digits of that result to get a single-digit number (for example, "16" would become 1 + 6 = 7)*
- *Add up all 10 digits of the new number*
- *Multiply by 9*
- *Take the last digit of that result*

This recipe produced a single-digit "check" number. In Luhn's original formulation, a 0 indicated the original number was valid. In later versions, the check was simply appended to the original number as a final digit, so that you could easily verify that the

final digit matched the check number produced by his machine. The underlying sequence of calculations, now known as the modulus 10 algorithm, is still widely used. The International Mobile Equipment Identity (IMEI) numbers assigned to cellular phones are verified in this way.

More significantly, the gears and wheels of Luhn's machine became the foundation for one of the most important algorithms of the digital age: the hash. This wide class of algorithms provides a powerful means of organizing information so that it's easy for a computer to find. Much like a culinary hash of corned beef and potatoes, a hash algorithm chops and mixes up data in various ways. Such mixing, when cleverly deployed, can speed up many types of computer operations.

In early 1953, Luhn had written an internal IBM memo in which he suggested putting information into "buckets" in order to speed up a search. Let's say you wanted to look up a telephone number in a database and find out whom it belonged to. Given the 10-digit number 314-159-2652, a computer could simply search through the list one number at a time until it found the relevant entry. In a database of millions of numbers, though, this could take a while.

Luhn's idea was to assign each entry to a numbered bucket, as follows: The phone number's digits were grouped into pairs (in this case, 31, 41, 59, 26, 52). The paired digits were then added together (4, 5, 14, 8, 7), from which a new number was generated, consisting of each single-digit result or, in the case of a double

consisting of each single-digit result or, in the case of a double-digit result, just the last digit (yielding 45487). The original phone number and the name or address corresponding to it would then be put into a bucket labeled 45487.

Looking up an entry from a phone number involved quickly calculating the bucket number using Luhn's method and then retrieving the information from that bucket. Even if each bucket contained multiple entries, sequentially searching through a single bucket was much faster than searching the entire list.

Over the decades, computer scientists and programmers have improved on Luhn's methods and pushed them to new uses. But the basic idea is still the same: Use a math problem to organize data into easily searchable buckets. Because organizing and searching for data are such widespread problems in computing, hashing algorithms have become crucial to cryptography, graphics, telecommunications, and biology. Every time you send a credit card number over the Web or use your word processor's dictionary, hash functions are at work.

Quick Indexing: At the 1958 International Conference for Scientific Information, Hans Peter Luhn (right) demonstrated an IBM system for automatically generating indexes of documents, based on an algorithm he'd developed called KWIC, for Key Word in Context. PHOTO: IBM

Luhn's ideas about computing went far beyond simple lookups. He saw that computers could be sophisticated text manipulators—for reading and understanding written language and then indexing and organizing that information so as to solve practical problems in science and business. By 1958, his chemical card sorter had evolved into the Universal Card Scanner and the 9900 Special Index Analyzer, which he demonstrated at the Washington, D.C., conference. These were electromechanical devices that could search and sort punched cards according to the user's criteria.

What really caused a stir, though, was KWIC, Luhn's computerized method of constructing concordances. A concordance is an alphabetical list of key words used in a book or a collection of writings. It's like an index, but it lists only actual

words that appear in the text, not the concepts (and it excludes trivial words, like *a* and *the*). Concordances have long been used in theology and philology. A concordance of the Bible, for instance, will show every instance of the word *love*, citing book, chapter, and verse. Before full-text computerized search came along, constructing a concordance was arduous and generally done only for major works like the Bible or the collected writings of Shakespeare.

What Luhn's bucket scheme did for numbers, his KWIC concordance system did for texts. Both made a large body of information easily searchable. To take a very simple example, let's say you wanted to generate a concordance of the words in the following four book titles: *Gone With the Wind*, *War and Peace*, *The Shadow of the Wind*, and *Shadows of War*.

A KWIC concordance of these titles would produce

```
                    Gone        With the Wind
War and              Peace
The                  Shadow     of the Wind
                     Shadows    of War
                     War        and Peace
Shadows of           War
Gone With the        Wind
The Shadow of the    Wind
```

The KWIC algorithm rearranged the words from the titles in all possible orders and then arranged each permutation

possible orders and then arranged each permutation alphabetically. The result was a complete list of keywords (meaning everything except prepositions, conjunctions, and articles) in the context they appeared.

Luhn's KWIC system was rapidly adopted throughout the scientific community. He knew it could be useful for business users, too. In 1958, he wrote an article for the *IBM Journal of Research and Development* entitled "A Business Intelligence System." In it, he proposed a system that could automatically generate article abstracts, extract "action points" from the abstracts, and then distribute the results to appropriate people within an organization. Luhn understood that solving the information overload problem meant devising a way to quickly sort through the crush of information without burdening people with irrelevant material.

The New York Times, in Luhn's 1964 obituary, described his auto-abstracting system this way:

> *"Mr. Luhn, in a demonstration, took a 2,326-word article on hormones of the nervous system from The Scientific American, inserted it in the form of magnetic tape into an I.B.M. computer, and pushed a button. Three minutes later, the machine's automatic typewriter typed four sentences giving the gist of the article, of which the machine had made an abstract."*

Luhn's abstracting program worked by first counting the frequency of all the words within an article. After discarding very

frequency of all the words within an article. After discarding very common words, the auto-abstracter located sentences in which several of the most frequent words occurred together. Such sentences were deemed to be representative of the overall content and so were placed into the abstract. This was a purely statistical method, making no attempt to "understand" the words in an article or the relationships between them. But like KWIC, it showed how computers could be fruitfully put to work organizing text into formats that humans could more easily understand.

Luhn retired from IBM in 1961 and died of leukemia three years later, and he didn't live to see the profound changes wrought by the Internet and the Web. Beyond a limited circle of information specialists, textile makers, and historians, his name is largely forgotten. But Luhn's ideas endure. Today, hashing plays a host of roles in managing and protecting our digital lives. When you enter your password on a website, the server is likely storing a hashed version of your password. When you interact with a website using a secure connection (where the URL begins with "https") or purchase something with Bitcoin, hashes are at work there, too. For cloud services like Dropbox and Google Drive, hashing makes storing and sharing files far more efficient. In genetics and other data-intensive research, hashing sharply reduces the time needed to computationally sift through vast quantities of data.

Hashes have turned computers into textual tools that can reason with letters and words. Google Translate, Google N-gram, Google

with letters and words. Google Translate, Google N-gram, Google AdWords, and Google Search are all devoted to determining, in one way or another, the meanings of texts. The explosion of information on the Web has made automated reading and understanding of central importance to business, to science, to everyone. The development of hashes was connected to texts, reflected in Luhn's thinking about words, sentences, concordances, abstracts, indexes, and digests.

This is Luhn's legacy: He helped show that computers and computation weren't just the province of mathematics, statistics, and logic but also of language, linguistics, and literature. In his day, this was a revolutionary way to think about machines.

Technology historian Michael Mahoney has called the computer "a protean machine": not just one thing but many things, a machine waiting to be shaped for different purposes. Even now, we tend to consider computers in a narrow way, as giant number crunchers, performing so many calculations and operations per second. Hans Peter Luhn's view of computers was more

This article is for

IEEE members only. Join IEEE to access our full archive.

Join the world's largest professional organization devoted to engineering and applied sciences and get access to all of Spectrum's articles, podcasts, and special reports. Learn more →

If you're already an IEEE member, please sign in to continue reading.

BECOME A MEMBER SIGN IN

MEMBERSHIP INCLUDES:

- Get unlimited access to IEEE Spectrum content
- Follow your favorite topics to create a personalized feed of IEEE Spectrum content
- Save Spectrum articles to read later
- Network with other technology professionals
- Establish a professional profile
- Create a group to share and collaborate on projects
- Discover IEEE events and activities
- Join and participate in discussions