eDrudgery

How legal tech left law(yers) behind

"There are official searchers, inquisitors. I have seen them in the performance of their function: they always arrive extremely tired from their journeys; they speak of a broken stairway which almost killed them; they talk with the librarian of galleries and stairs; sometimes they pick up the nearest volume and leaf through it, looking for infamous words. Obviously, no one expects to discover anything."
— Jorge Luis Borges, The Library of Babel

"Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. The human mind [...] operates by association."
— Vannevar Bush, As We May Think

The number of answers to the question "what is eDiscovery" is equal to the number of people working in eDiscovery to whom you pose the question. I've never been able to get a straight, or coherent explanation.[1] With that said, the underlying problems of document search, which launched the industry back in the 1990s, as well as contextualizing the results are very real.

Lawyers are continually dealing with increasing and varying volumes of information (documents, text messages, social media posts, etc). We are talking about millions, if not tens or hundreds of millions of items. A typical scenario can be described as follows. An imminent litigation triggers the exchange of data between opposing parties, and with a limited amount of time each respective team of lawyers is supposed to comb through it, extract the potentially relevant case-matter material, and hopefully find that "needle in the haystack" document key to winning the case.[2] Even imbued with the near parental love of billable hours, the attorneys know that looking through everything is impossible. They turn to technology, but as we'll see below the technology is really turned on them.

Hundreds of examples of eDiscovery software have hit the markets since the 1990s, with varying economic success, but universally the attorneys themselves hate using these systems.

---

[1] This I believe to be a key feature of industries that provide what are known as "professional" services but that in itself is another conversation.

[2] In reality both parties are doubly-tasked, since which data is exchanged, or "produced," is also a non-trivial matter. They are obligated to produce all case related materials, but make sure that nothing non-relevant, confidential or potentially damaging makes it to the other side. Even defining the scope of a legal matter, not to mention what data is relevant, can be somewhat complicated and as we'll see made even more so by the current software solutions.

The modus operandi is “avoid possible.” Even first year associates, those bottom of the barrel law firm slaves, have their exposure limited. Perhaps there is some sort of moral threshold to the sadism one can impose on a junior entering the legal field and eDiscovery software oversteps this boundary.

Legal tech rationalizes this revolusion by asserting that “lawyers are the world's worst adopters of technology,” a phrase that rings through every conference, gathering, or sales event in the industry. The fact that these same lawyers use all forms of computers (smartphones, laptops) and an endless myriad of software applications in their non-work life and do so with ease isn’t seen as any sort of contradiction.[3] The fact that when asked the attorneys describe the software as “crap,” or at best the vacuous “it’s better than the last version,” doesn’t set off any red lights that perhaps the entire approach might be faulty and that the entire industry is riding on some sort of bad decision.

To some this might look like a tragedy of the human spirit in the works, but to others an opportunity. In order to fill the gap between the lawyer and the software interface, eDiscovery has created an army of consultants, legal tech specialists, along with the usual gamut of training seminars, conferences and so on. The industry employs hundreds of thousands of people and the US market alone is expected to reach twenty five billion dollars in the next few years.[4] There are two types of people in the industry: the oblivious who drink the kool-aid and the others who know and say that all the software is crap, mostly utterly miserable to be part of the charade. A few drinks magically transforms 50% of the former category into the latter.

The truth is that legal tech is largely mired in solving the problems it itself has created. The attorneys are right to resent these solutions, and the people who provide them, because they simply have little to do with law, discovery, information, fact finding, case building, and more importantly thinking. Legal tech never considered the lawyer.

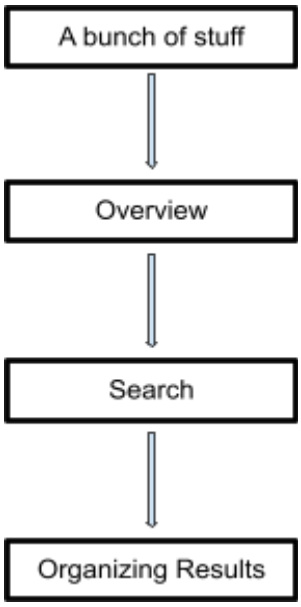
---

[3] ...neither is the inherent condescension. In my experience, with extremely few examples, the attorneys I have met at these firms were intelligent, technical, and highly competent learners. This was in general not the case for the legal tech “professionals.”

[4] Again the fact that you need an additional human interface layer between the human and the software interface should be a scathing review of the software, but here we are…

The process

Before we begin our story in earnest, let's take a look at how a typical eDiscovery process unfolds.



New litigation triggers a data exchange between the opposition parties. This data is loaded into an eDiscovery software.[5] Generally the software has collection overview features, consisting of various tally statistics (graphs and charts) related to the content, language and so on. Based on available information (case details, key people/custodians, overview etc) searches are run. Most of these searches are keyword and boolean filter based, although sometimes there are some slightly more sophisticated natural language features. Search results are reviewed and relevant examples are organized. Generally this is done by decorating the found documents with tags such as “relevant” or “hot,” moving them to specific folders on the system, and so on. When the search is exhausted, or simply nothing relevant has been found, the criteria is adjusted for an additional run and the search is restarted. Similarly when new information sheds light on the case, the search process is again adjusted and restarted.[6]

A bunch of stuff

How data is stored, what types of databases used, and how the records are indexed in eDiscovery is critical to any solution. However, these details are one step removed from the

[5] This can be a non-trivial process. There is no guarantee of compatibility between the two parties. Moving data securely from one store to another, processing it and loading into the software can be costly and complicated, especially if IT gets involved.

[6] I’d like to underline the linearity of the process here. Each search is new, starting from step one, even if conceptually it is a continuation of a previous set of explorations.

software interfaces themselves and at this point a little bit of a digression so I'll keep this section to the bare minimum with two observations.

Storage and indexing must reflect the desired interface and experience in discovering information, not the other way around. We must ask about the experience and process we desire, not about what is the optimal or natural way to extract examples from a given datastore. For example, running boolean queries in SQL is in one to one correspondence with how SQL indexing works. But as we'll see below, boolean constructs have little to do with exploration, discovery, or thinking. If we are going to make these systems more human we must forget about databases, indexing, scaling and related specificities and focus on the actual problem at hand. These technicalities will fall out of a proper framework, but a proper framework is never a derivative of technicalities.

One glaring weakness of the current approaches to storing information is adding new data, or new types of data. Items appended to a collection require new indexing, which often requires re-indexing of the entire store. New types of information can be completely incompatible with current storage and indexing setups, in turn forcing massive changes to the entire platform. Think of any text document (email, PDF, MS Word, etc) specific eDiscovery solution which now has to accommodate images, or audio, or worse yet something unknown. Fundamental changes are called for, but more often than not the chosen path is a patchwork integration of some other, usually, incompatible system. This is both costly from a resources standpoint, but more importantly undermines the potential for any sort of unified framework. If the solutions are to focus on people, then they must seamlessly handle not only new data but also currently unknown and unimaginable - yes unimaginable - types of information. IE what we need is an "any data datastore," which can handle any information of yesterday, today, and tomorrow. This might sound like science fiction but there are good frameworks in computing for such a thing. More critically there is a perfect example of such a flexible datastore which is you.[7]

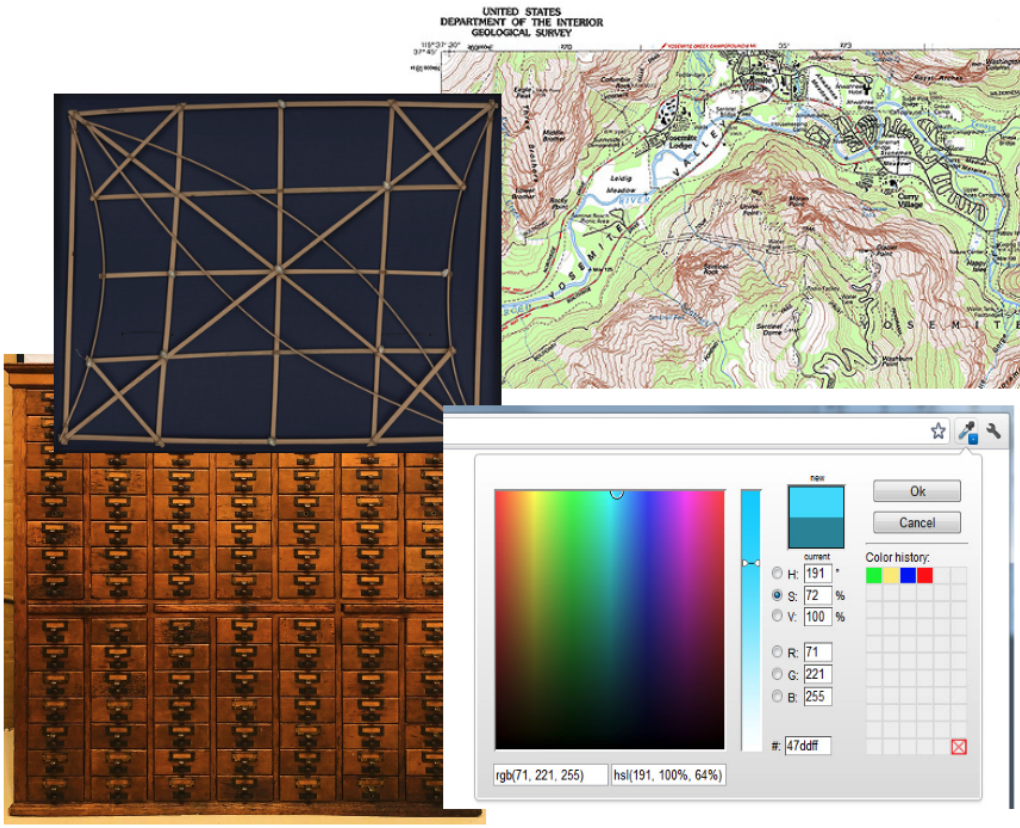
In this section I've been deliberately light on the details in order to get to the part of the process of eDiscovery which people are more familiar with: the experience of searching and organizing. The one additional observation I'll add is that there is nothing here which is really specific to eDiscovery, digital stores or computing at large. Storing, indexing, retrieving, adding, removing, and updating information has been at the core of archive technologies for millenia and there are deep lessons to learn from these historical examples.[8] And this is precisely our next task.

---

[7] For a more general discussion about the "any data datastore" you can look here.

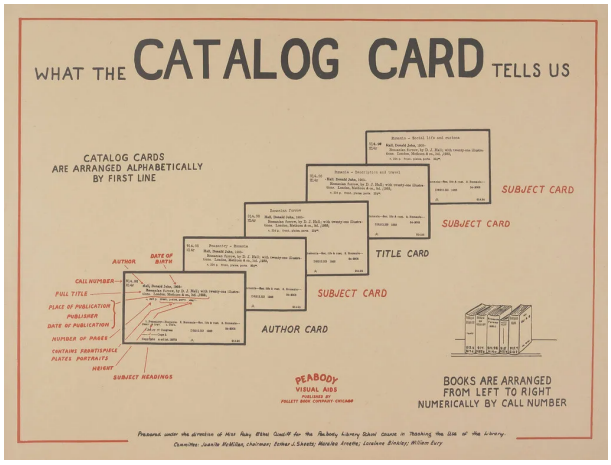
[8] For example, in the world of libraries and archives it took until the 19th century with the advent of ISBN combined with collection experts (librarians, archivists) to propose a workable solution. Unfortunately software developers in legal tech and elsewhere seem to be generally oblivious about anything that happened before they wrote their first computer program.

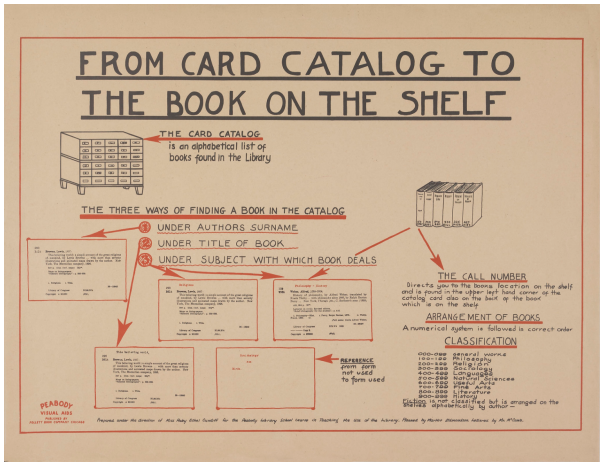
Finding aids



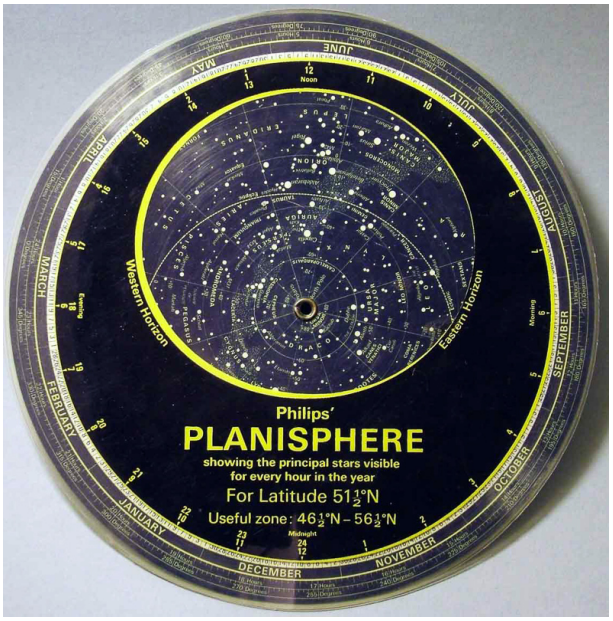
Clockwise from bottom left: card catalog cabinet, Marshall Islands navigation chart, topographic map, color picker

Getting around, whether a library, the oceans, or the stars has been traditionally done with finding aids. These have come in many flavors and have an interesting history of their own which I won't get into here. There are two elucidating examples I would like to look at in more detail: card catalogues and celestial maps.





The library, or the archive, the “big data” store of the last two millenia became the powerful knowledge system that it is today with the arrival of the card catalogue. This technology enables exploration via self-referential views of the collection. Titles, authors, subjects (as many as needed) provide top-level entry points and each card contains full information for each corresponding item. For example, a title-based card will have the author as well as the subjects listed pointing to multiple immediate avenues of finding your next relevant item. (I can’t stress enough the importance of continuity here!) Moreover, the entire archive is organized, indexed, via call numbers which reflect the structure of the collection inline with the card catalogue (subject, author, title). When looking at a particular item on the shelf, browsing the records “around” is a very reasonable way to find other relevant information, as is browsing the card subject listings for a specific item.

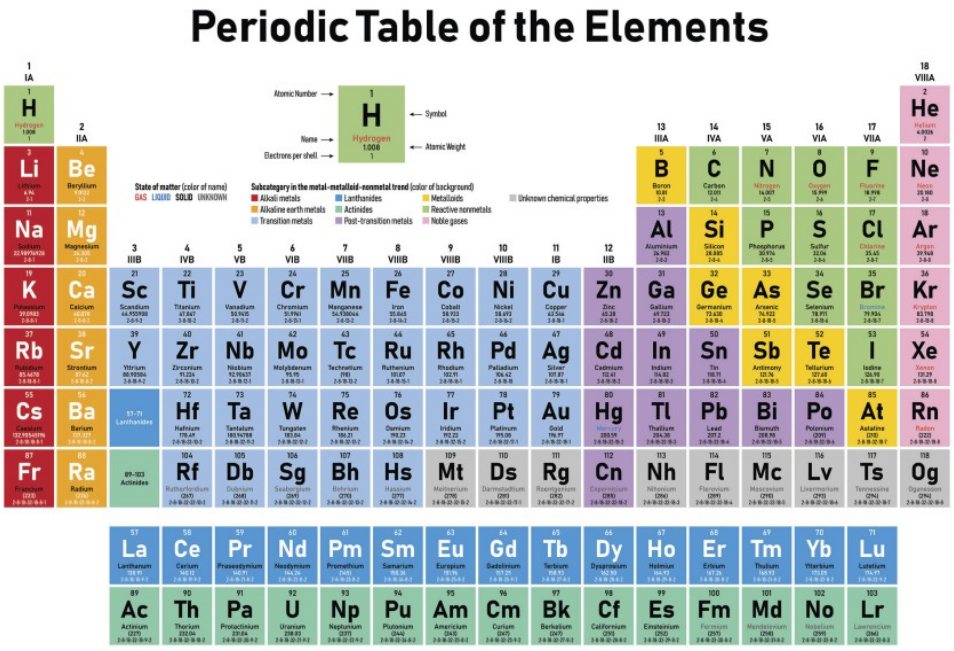


When we take a break from exploring knowledge spaces and look up to the skies, we inevitably turn to celestial maps. Finding your way around at night is aided by grouping the various stars

into recognizable figures. The location of one constellation is usually sufficient to start to trace out the others, provided you know where you are on the globe and what month of the year it is. Even missing data, such as obscured stars, can be filled in by a reasonably trained eye as you move across. As the seasons progress your finding aid can be adjusted to reflect the view.

Both the card catalogue and the celestial map require critical contextual additions. For the Planisphere you need two technologies already mentioned: an estimator for your coordinates and a calendar. For the card-catalogue it is the archivist or librarian. We'll spend a little more time on the librarian when we talk about search, but the main point here is that finding aids come along with a way to ground yourself, to find a clear starting point from which to guide you along. In addition, the "user interface" is optimized to get you to the next potentially relevant piece of information, never coming back to square one or restarting the search. This is precisely how you learn new constellations or discover a document you didn't even suspect existed.

Note that finding aids and overviews are very different things. The former is a guide, intended to allow you to traverse a collection and discover new information. It is inherently contextual and navigational, whether conceptually, temporally, physically etc. The latter is a complete description at the requisite level of knowledge.



A canonical example is the periodic table of elements that gives you all the taxonomic information necessary to understand the basic building blocks of our material world: name, type, atomic weight etc. It displays the set of data in an organized way. If there is a square missing you can imagine the existence of an element of specified atomic weight, which is for example how Americium was synthesized. But the table does not tell you what to do, where to go, or how to go about it.

Let's take a moment and look at a screenshot of the Relativity eDiscovery platform overview module.[9] What does it tell us about the collection? Is it a taxonomic breakdown of the document universe? If so, who chose the taxonomy and why is it relevant to the particular navigation. Even more broadly, why is a statistical breakdown useful here, in lieu of a guide. If the designers were imagining a “finding aid,” then how do you relate the pieces in this smorgasbord of visualizations? Despite the use of a uniform color palette there seems to be no immediate structure or connection between them. What’s worse the color scheme implies connection where there clearly isn’t one![10] It is not self-referential, there is no clear starting point, nor any sort of navigation trajectory. How do you find out where you are, and what is a reasonable definition of “where?”

The Relativity advertisement brochure, from which comes the screenshot, underlines the typical situation: a complete absence of a model, metaphor, or concept for arguably the most important component of the technology. Not only does eDiscovery software fail to address the basic questions above. It doesn’t even begin to ask them.

Building a card-catalogue inspired system for any digital archive, such as an eDiscovery one, is not particularly difficult. As a matter of fact something very similar, but more powerful, was

---

[9] Relativity is the most prevalent eDiscovery platform in the market, and as the “market leader” it serves well the general example (you can find the brochure here). To my knowledge there isn’t a single platform out there which embodies the concept of a finding aid.

[10] The same shade of blue appears in the bar chart where it is simply the color of the graph, the pie chart where it represents the percentage of records which are emails, and in the “donut” chart which we can assume to stand for a cluster (distribution) of words. The blue, yellow and green which appear in both the pie and “donut” charts don’t seem to mean anything. I would assume that the e-documents are a subset of the attachments which are subsets of the emails, but you can’t tell that from the pie chart. All in all this is horrendously bad design made by people who care more about marketing brochures than lawyers the software is intended for.

outlined in the 1945 *As We May Think* article by Vannevar Bush. Almost eighty years ago Bush imagined a machine of the future - the MEMEX - to be a dynamic self-referential storage system. Bush's proposal also adds what he calls "association trails" of (re)search. These trails capture not only the results but the search paths and process, which then become part of the knowledge base of the system. IE the "card catalogue" becomes a dynamic, growing, library of knowledge, which includes items, facts, categorization and the connections (paths) which link them.[11]

It's worth a moment to listen to Bush himself:

""

Selection by association, rather than by indexing, may yet be mechanized. One cannot hope thus to equal the speed and flexibility with which the mind follows an associative trail, but it should be possible to beat the mind decisively in regard to the permanence and clarity of the items resurrected from storage.

Thus he [the lawyer] goes, building a trail of many items. Occasionally he inserts a comment of his own, either linking it into the main trail or joining it by a side trail to a particular item. …

And his trails do not fade.

Wholly new forms of encyclopedias will appear, ready-made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified. The lawyer has at his touch the associated opinions and decisions of his whole experience, and of the experience of friends and authorities. The patent attorney has on call the millions of issued patents, with familiar trails to every point of his client's interest.
""

Take a moment to reflect on the above. You'll see an entire world of possibilities opening up: guided eDiscovery, prior art or case law dynamically inserted into the process, collaboration, defensibility, etc etc. All these become natural by simply taking a reasonable metaphor as opposed to a smorgasbord of "insights" and "features." But more importantly instead of the Library of Babel we get a library of (collaborative) knowledge.

By now I hope it is clear why we didn't see a lot of bar-charts in Bush's vision, just like there weren't any at your local library finding-aid station. Next we'll look at why the card-catalogues themselves were not keyword based, and why keyword search was not the framework adopted even when digital catalogues became commonplace in the 90's.

---

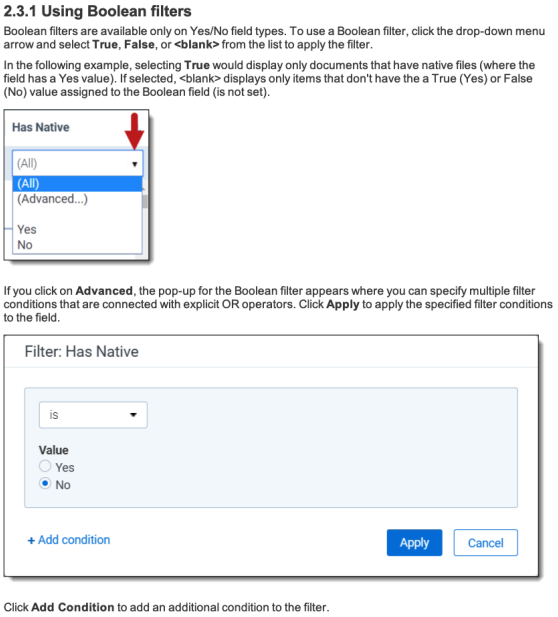
[11] In Marvin Minksky's *Society of Mind*, one of the best books on artificial intelligence by a pioneer in the field, you can find a similar description of knowledge and memory although in a much deeper developed theory. Minsky calls these "K-lines." The same idea is the theme of Marcel Proust's *In Search of Lost Time*, where memories are built and triggered via sequences of associative events (smells, tastes, sounds, names, places, people, etc).

The short answer comes again from Vannevar Bush:

```
*The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts…*
```

To search or not to search

eDiscovery's call to fame, the core component of any product, is search. Search here is based on boolean expressions, i.e. statements which evaluate to true or false, made up of keywords and various other filters. These come in flavors of pulldown menus and checkboxes, or boolean search strings such as the ones from the Relativity manual below. Sometimes they are peppered with pretty charts, statistics, natural language processing and so on, but the fundamental flow is largely deterministic.[12]



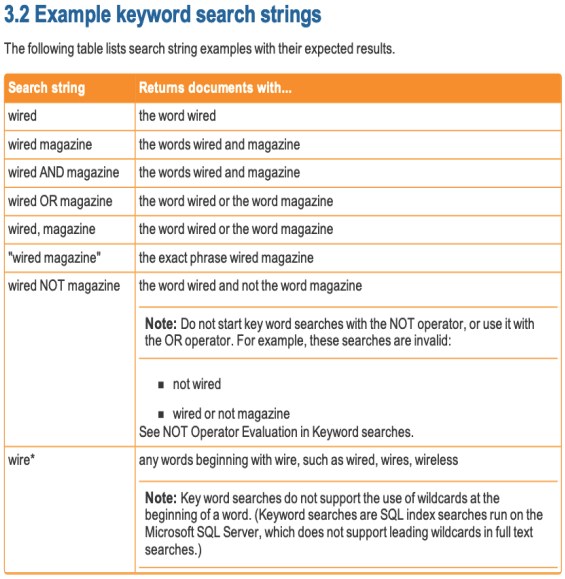
2.3.1 Using Boolean filters

Boolean filters are available only on Yes/No field types. To use a Boolean filter, click the drop-down menu arrow and select **True**, **False**, or **<blank>** from the list to apply the filter.

In the following example, selecting **True** would display only documents that have native files (where the field has a Yes value). If selected, <blank> displays only items that don't have the a True (Yes) or False (No) value assigned to the Boolean field (is not set).

If you click on **Advanced**, the pop-up for the Boolean filter appears where you can specify multiple filter conditions that are connected with explicit OR operators. Click **Apply** to apply the specified filter conditions to the field.

Click **Add Condition** to add an additional condition to the filter.



3.2 Example keyword search strings

The following table lists search string examples with their expected results.

Search string	Returns documents with...
wired	the word wired
wired magazine	the words wired and magazine
wired AND magazine	the words wired and magazine
wired OR magazine	the word wired or the word magazine
wired, magazine	the word wired or the word magazine
"wired magazine"	the exact phrase wired magazine
wired NOT magazine	the word wired and not the word magazine **Note:** Do not start key word searches with the NOT operator, or use it with the OR operator. For example, these searches are invalid: ▪ not wired ▪ wired or not magazine See NOT Operator Evaluation in Keyword searches.
wire*	any words beginning with wire, such as wired, wires, wireless **Note:** Key word searches do not support the use of wildcards at the beginning of a word. (Keyword searches are SQL index searches run on the Microsoft SQL Server, which does not support leading wildcards in full text searches.)

Setting legal discovery aside, you might already be wondering what boolean queries, filters, check boxes and so on have to do with the way we think, organize, navigate etc. In reality, and as we'll see shortly, these have more to do with how programmers understand computers than how people interact with information.

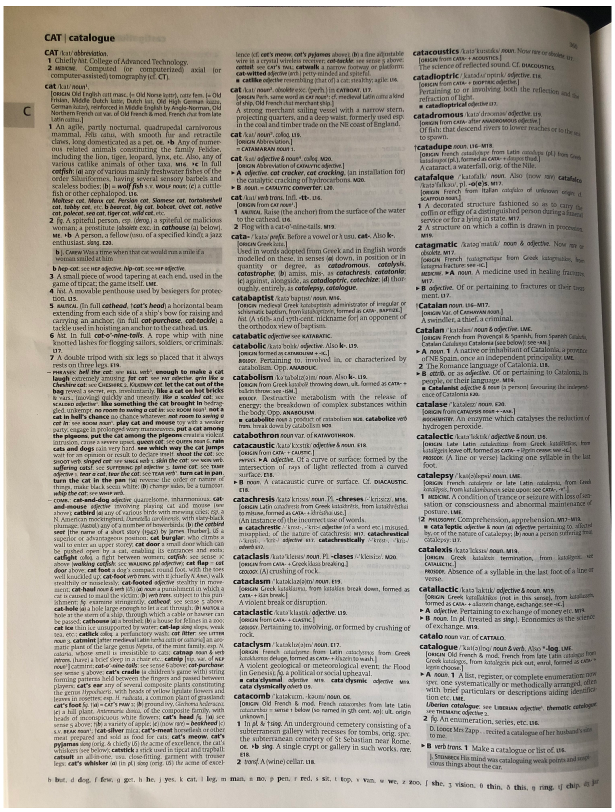
[12] Pulled from the RelativityOne eDiscovery search documentation (found here). The fact the document runs 150 pages should already be a tell sign of something wrong. Again Relativity is the top player in the legal tech space, but really any eDiscovery platform that I have seen follows the same approach.

But before all of that lets look at a common keyword search scenario to see how this “plays out” in eDiscovery. Faced with say ten million documents and some generally vague starting point description of the legal matter at hand, the lawyer’s goal is to come up with a set of words whose presence, or lack of presence, in a document determines its relevance to the case.

Stop and think about that for a second!

Two immediate issues jump out. First, the task of generating a precise deterministic description given limited information is a contradiction in terms. And the complex, nuanced, dynamics of legal disputes only exacerbate the situation. Second, words are not determinants of meaning, they are a way to meaning through context. Gluing them together with logical conjunctions and disjunctions is primarily helpful if you are cataloguing word appearances, which computer programmers just happen to be very good at, not knowledge or content.

Take a look at this page from the *Shorter Oxford English Dictionary*:


CAT | catalogue

There is almost a column and a half for “cat” - and this is the <<shorter>> dictionary - while single-entry words are relegated to the very technical or specific. IE the more likely you are to interact with a word, the more context and knowledge driven the meaning will be. A boolean string of words such as “cat OR (dog AND bird)” doesn’t increase meaning but dilutes it.

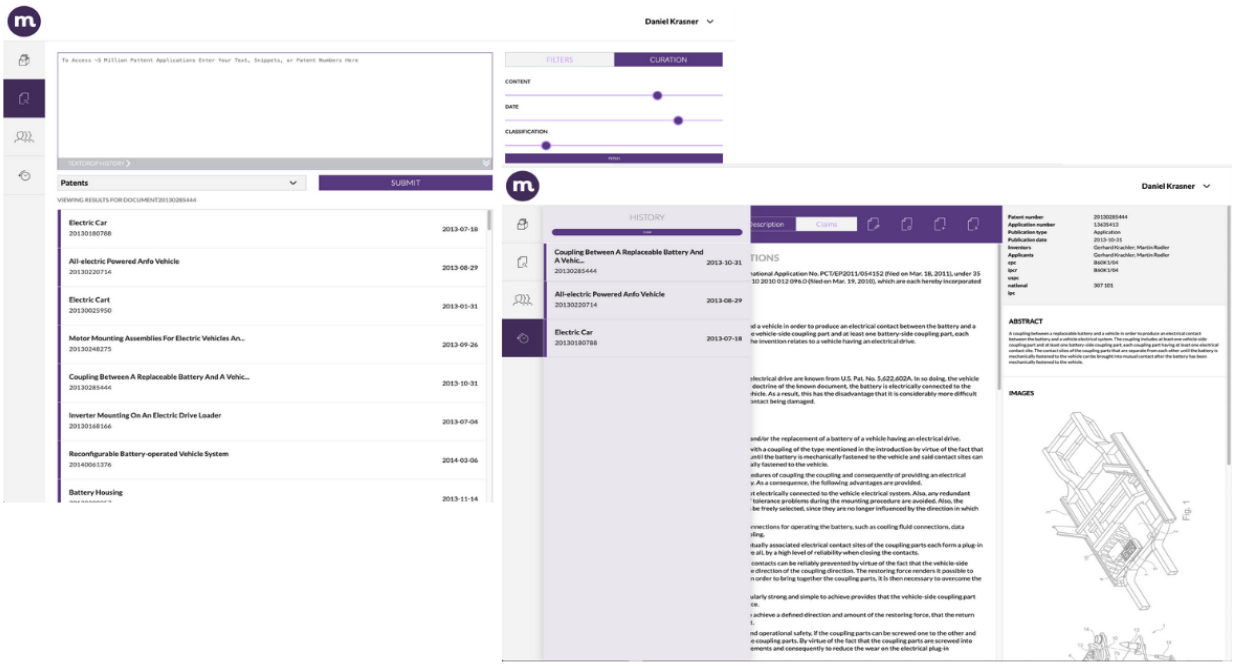
At no other time in history would such a thing be considered normal, workable, or reasonable. Just imagine coming up to your local librarian and saying “I have a research project, can you get me all the books, articles and audio tapes which contain words from list A but don’t contain words from list B.” If the librarian is having a good day he will gently explain that you are confused and ask you to describe the general theme of your research. After that you will be directed to a starting point and guided through the collection until all requisite materials have been found.

Let me be clear, there are good scenarios for deterministic, boolean, based search systems. A classic example is inventory. If you would like to know how many blue t-shirts of size six are currently in warehouse 23, then a SQL query is a reasonable thing to consider. But if you are looking to see if there was a discussion related to the purchase of clothing sometime before or

after some event, between person A, person B and maybe some others then it's not the way to go. This also doesn't mean that special situations where you might want to look up a record by an identifier do not come up. The call number of a book, or RGB color values, allows you to retrieve the corresponding item immediately. But the rest of the time the finding aids model the underlying associative fabric of the universe of information, and allow structured, guided exploration of the collections.

This all sounds nice, but the obvious question is whether an associative search system like the one Bush describes is possible. Perhaps the technology is not there yet, perhaps we are in some transitional moment and more research is needed. For starters, I along with a few colleagues built a fully working associative search engine. It took the librarian as the metaphor and allowed the complete spectrum of input, from the most sparse and vague snippets of text to the most precise identifiers. Much like talking to a real librarian, exploration was guided by indicating what you were looking for in terms of relative importance of context (text) and content (metadata such as time, people, subjects, topic etc etc). You could input a snippet of text or an entire stack of documents, and then tell the "librarian" why they were important. The system would guide you along, keeping your trail of exploration readily available and explorable. And of course you could use keywords and filters if you so desired. The search engine called *Merriam* was also magnitudes more efficient in terms of compute resource usage than any of the contemporary off the shelf systems. And it worked very well. It had no manual. It explained itself. It was built to production quality in six months by three people working part-time.[13]

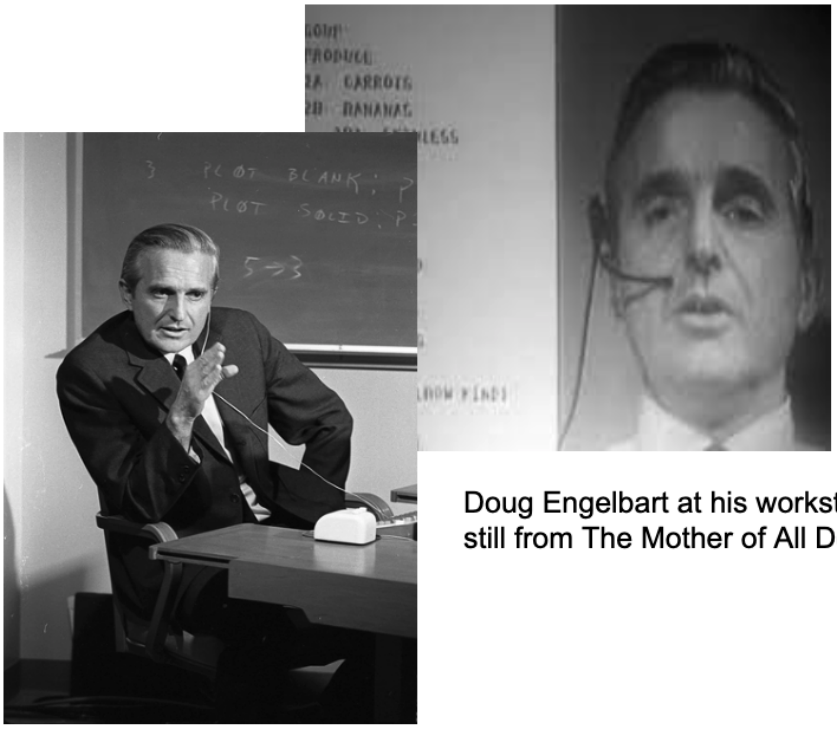
[13] Although the engine technically allowed for it, the ability for the search paths to become naturally explorable and part of the collection in the sense Bush describes was not built into Merriam. When we wrote the first version we hadn't read his article. It helps to know your field…



Associative search in the Merriam system: moving the sliders tells the system the relative importance of language and context. The text box accepts any text, from single words to full documents. Search paths are made available.

A more telling example is the fact that associative indexing and search was built into the *on-Line System* which debuted in the most important event in computing history: Doug Engelbart's 1968 *The Mother of All Demos*. Englebart was deeply inspired by Vannevar Bush, and knew that if computers were going to augment human thinking they would have to model how people think, discover, learn and collaborate. Associative search was **not** a feature of NLS. It was built in at the operating system level.[14]

[14] The entire demo can be found here. The ideas are laid out in Engelabart's 1962 report found here. Text search and associative search was a well studied topic in the following decade. See for example Gerald Salton's work on the subject.



Doug Engelbart at his workstation and still from The Mother of All Demos

At this point you might be asking the following: Why the status quo? Why when it is relatively easy to build associative search based systems, and moreover when such systems were part of core thinking about computing from the pioneers in the field, do we not see these today? The essence of most questions about technology is not technological and this one is no exception. The problem is largely cultural (more on this below).

In the meantime, let's ask the simpler question of where does this whole SQL framework come from?

Programmers tend to think more about how computers function than how people live.[15] Conveniently enough, computer *memory* (RAM/ROM) is completely deterministic; to retrieve what is stored in a given register you specify its address identifier. The framework which determines how a computer behaves at the lowest level, set in the arithmetic logic unit (ALU) found on the central processor, is boolean logic. Extrapolating computer memory and logic to a database system you get SQL.

[15] Even those in the industry who spend their time designing user experiences, really think more about how to make sure the computer influences the person into a specific action, or set of actions. This is very different from thinking about people or about how people think, explore, learn, discover and so on. Just the term “user” is indicative of the mindset, but that’s a story outside of this scope.

Now computer *memory* has nothing to do with human *memory* other than convenient word overloading. And logic plays a very small comparative role in human thinking and discovery; it is best seen as an after-thought, there to make sure you didn't make too many erroneous assumptions. To use a computing metaphor, logic is the interpreter for the implementation of our ideas, it's not the ideas themselves nor their genesis.

But if you are a programmer and you like how computers work then you'll make SQL.[16] Furthermore you'll declare that "this is how computers work and so the most natural solution for looking things up." The error lies in the understanding of what "looking up" means, and for whom this is a "natural" process.

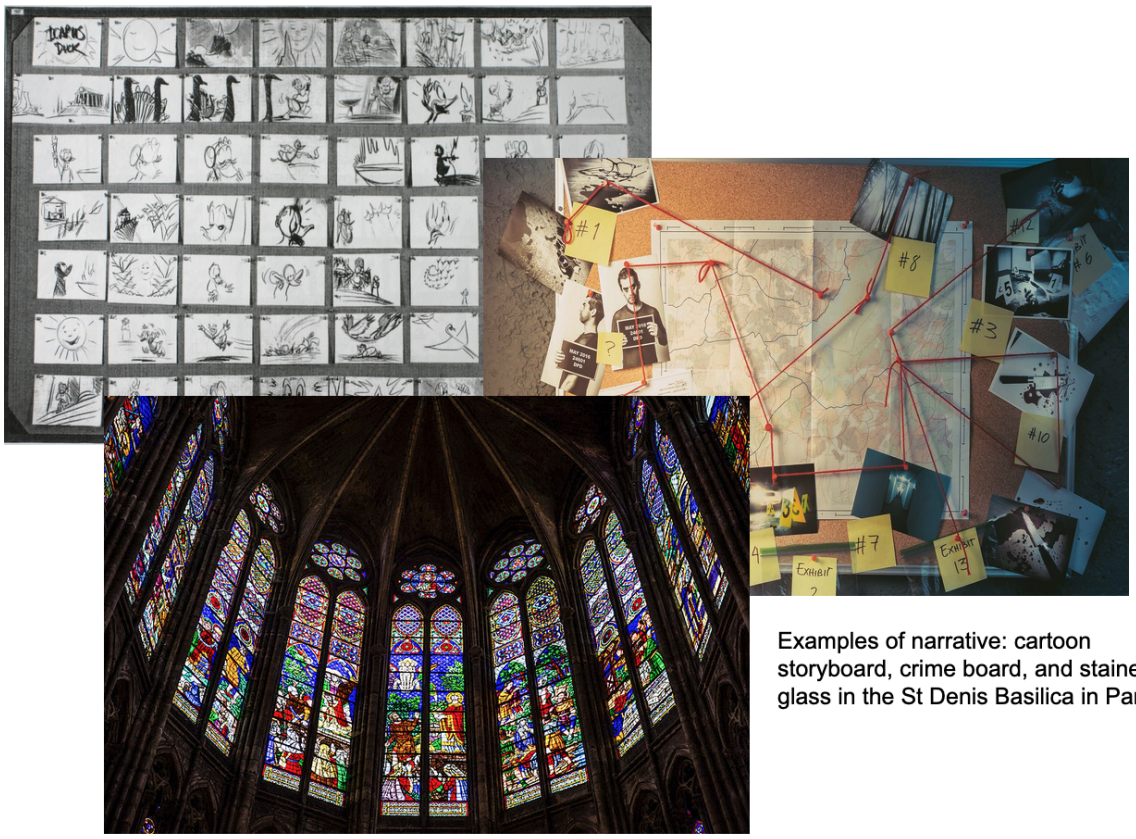
Ironically lawyers, besides being human, are some of the best stewarts of associative and referential based deduction. They are great investigators, a skill requiring the association of disparate pieces of information to build a clear narrative. And the good ones are all experts in case law, i.e. legal precedent is a critical component of any matter. So in essence, when presented with a gauntlet of keyword queries, filters and bar charts, they are being told that everything they know about building a case is largely irrelevant and moreover *this* is the only way discovery can work.

Telling a story - narrative

The very act of objectifying associated events, individuals, or the legal code is fundamentally an exercise in conventional literacy. The standalone "fact" devoid of anything else which we can then represent, manipulate, argue about and relate to other such instances of codified law or legal precedent is made possible by the flexibility and power of our reading and writing media.[17] As Goethe put it, "the highest wisdom is to understand that every fact is already a theory." Association, nomenclature, semiotics are critical components of how we reason about the world. Marrying up (informal) logic with objectification is in large part what legal argument is. With that said, whatever the organizational structure of fact and findings might be, the case itself is a type of story, a narrative, that the lawyer is to piece together and which is to be supported with evidence and legal precedent. To this brings us to arguably the most critical component of the process.

[16] I didn't do a deep historical study of this, but if I am wrong on some of the details the mentality described is prevalent.

[17] For more on the connection between conventional literacy and law see for example Eric Havelock's *The Muse Learns to Write*.



Examples of narrative: cartoon storyboard, crime board, and stained glass in the St Denis Basilica in Paris.

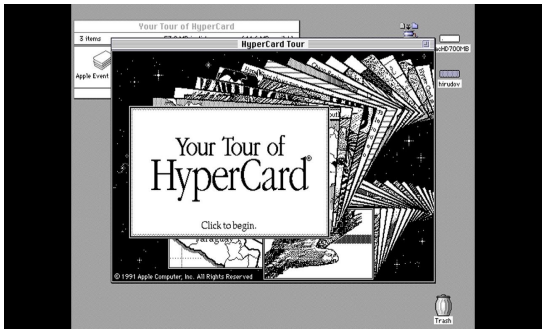
Narrative comes in many flavors. Some is linear, such as the cartoon storyboard; some is associative, such as the detective layout; and some is all engrossing, such as the gesamtkunstwerk of the cathedral stained glass. How you choose to lay out the narrative largely depends on what effect you are looking for and what message you are trying to send or receive.

The wonderful thing about computers is that they are exceptionally well suited not only for objectification and logical argument, but also for narrative. Moreover, a well designed environment can allow any imaginable narrative structure, and let you choose the more appropriate one for the reasoning and case at hand. Specific parts of the narrative can be directly associated, or linked, to case-law, legal precedent or anything else deemed relevant. But even more importantly, you can “run the narrative,” which in computer or scientific circles is referred to as “simulation.”

Imagine being able to label, organize and associate all your findings into a timeline which you can then run (backwards and forwards). You can see your entire case play out. You can argue about the validity of some association, or causal inference, by removing the underlying

components. For example, to judge whether a particular communication (call, email etc) was the watershed moment you can remove it, or remove some of the custodians appearing in the exchange, and add them back one at a time. To see whether a specific legal statute applies you can “toggle” some of the events and see if the critical ones are still linked to it. You can switch the components and glide over time, geography, economic values or whatever other “dimensions” make up the information landscape.

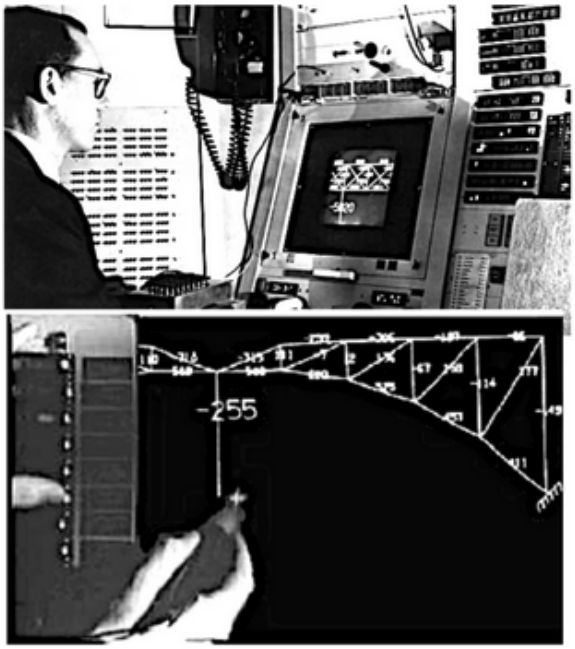
There are a number of examples of computer environments in which the above hyper-storytelling would be quite natural, but my favorite is *Hypercard*. This was a system developed by Bill Atkinson at Apple and extremely popular in the 1980s and 1990s. *Hypercard* had millions of authors who built and catered anything from business accounting, documents databases, games (*Myst* being the best known), presentation to multimedia systems.[18] It was incredibly flexible, expressive and perfectly suited for the sort of dynamic, hyperlinked, document/database supported, narrative system described above.



With that said, declarative style narratives, i.e. where the case is constructed out of the discovered components, put on a dimension like time, or otherwise organized by essential attributes, is not the only option. A constraint based storyline is potentially more powerful and natural in many situations. For example, you create the narrative by first making the relevant known events (email, call, transaction, etc), then input the possible custodians or people involved, add a general timeline, and finally insist that some of the events are coupled in the sense that if one comes first then the other has to follow or precede. Then you ask the computer to determine whether such a configuration is possible, and if so how potential scenarios exist. Then you can play each one out, toggling and sliding as above.

Another seminal computer system *Sketchpad* was designed on a similar constraints based principle. In the screenshots below you see its creator Ivan Sutherland in 1963 telling *Sketchpad* that he wants all these straight-line pieces to fit together, their lengths to be fixed, and their weights and weight bearings to be related in a trestle bridge layout. Then the system figures out how all these relationships fit together, if the bridge will hold, collapse and so on. Don’t be distracted by the architectural examples on the screen. A constraint based approach to building bridges can apply just as well to our legal narratives.

[18] I’ve heard numbers ranging from 4-6.5 million authors at the peak early 1990s days of Hypercard. The choice to call people *authors* instead of *users* and *programmers* was a deliberate one, since the creators purposefully obliterated any distinction between the two. You can see Bill Atkinson himself talking a bit about the system here: https://youtu.be/v9o5Ld8hpug?t=1345

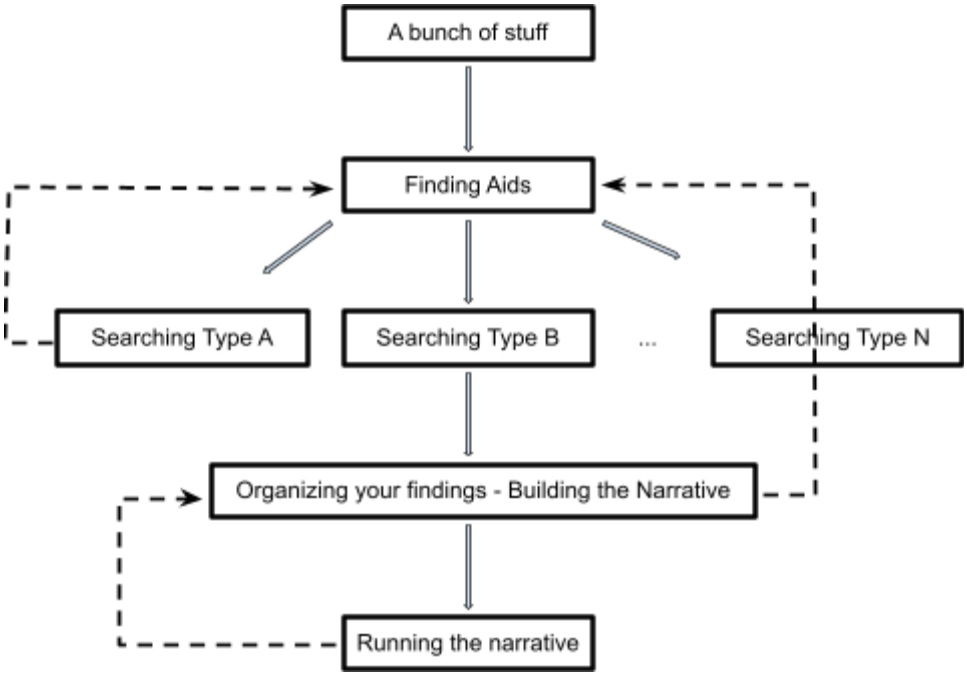


Finally, there is no reason that an eDiscovery system would not support multiple narrative styles. You can imagine either toggling or sliding between declarative and constraint based narratives depending on the situation.[19]

Adding an associative search system, such as the ones described above, would get us very close to what a lawyer might have imagined before the programmers came along and told everyone how it should be. It might also be close to what Vannevar Bush imagined almost 80 years ago.

Putting all of this together our eDiscovery flow might now look something like this:

[19] This is in many ways how many video games are designed.



Why or why not?

You might be wondering why, given all of the above, the ideas don't seem to permeate into the legal tech space.[20] One reason already mentioned is the fact that programmers tend to build tools for themselves, not for people. In essence, tech people are not interested in technology, they are interested in using technology to cause effect. Another, is the fact that computing, computer science, Silicon Valley - however you want to refer to computing related tech today - is largely unaware of its past, of its pioneers, of the brilliant ideas that gave birth and impetus to the systems we use today. We have a field where you can graduate from a top school without any knowledge of how, why and by whom the field in question was created. It is as if we have PhD's in physics who have never heard of Newton, or musicians in the NY philharmonic who have never heard of Mozart. If you try to imagine law without case law, without legal precedent, without a past, with only a trailblazing future, then you get something akin to what has been happening in tech for the last 40 years. In short, computing is a "pop" culture.[21]

But that does not have to be the case. Furthermore, besides torturing the law community there are greater implications to imposing terrible technology.

---

[20] Legal Tech is not special here. It is simply one example of a general trend of ignorance you find in computing.

[21] This is borrowed directly from Alan Kay, one of the most important of the computing pioneers. Besides being the leader of the group that invented much of everything we use in today, he has written and spoken extensively on the subject of technology, computing, culture, education and related topics. There are many lectures online, but this is a reasonable place to start.

The status quo of our current legal system underpinned by a (statically) written code, facts and causal inferences of events will likely not be sufficient to properly argue about the ever complex world of the future. A law founded not solely on the codified, but also on our ability to reason and argue the dynamic, the simulated, is something that we are in desperate need of. In the most present example, our failure to properly grasp and argue about climate change largely rests in the fact that we are dealing with a dynamic, continually evolving system. It is not describable by a fixed collection of causal fact-events. The set is a moving target, and categorically dissecting the dynamics only dilutes the entire picture, undermining any argument as opposed to putting it in focus. But a dynamical law “code” is something we can only begin to imagine, and certainly won’t be able to fully define until we acquire a sufficient level of computing literacy with which to supersede the present written framework.

Whether or not computing technology will allow for a new foundation on which to build the code of conduct of our civilization is to be determined. However, we can take the much smaller steps in creating more humane environments for those of us tasked with implementing and keeping the current societal fabric in check. There are no technical limitations to bringing law and lawyers back into the technology itself. The coders, programmers, designers just have to be more honest, more humble, look back to the many good ideas left by the wayside, and focus on the people.