Text Search and Relevancy

Article contributed by dtSearch®

Today's topic is search results and relevancy. Previously (please see, for example, Your Data: Finding the Forest through the Trees at USA Daily Times), I've talked about how a text search engine like dtSearch parses documents, emails and other data using its own document filters and then uses that information to automatically index any number of data repositories. After indexing, one user can instantly search—or multiple users can concurrently search—terabytes using over 25 different search features.

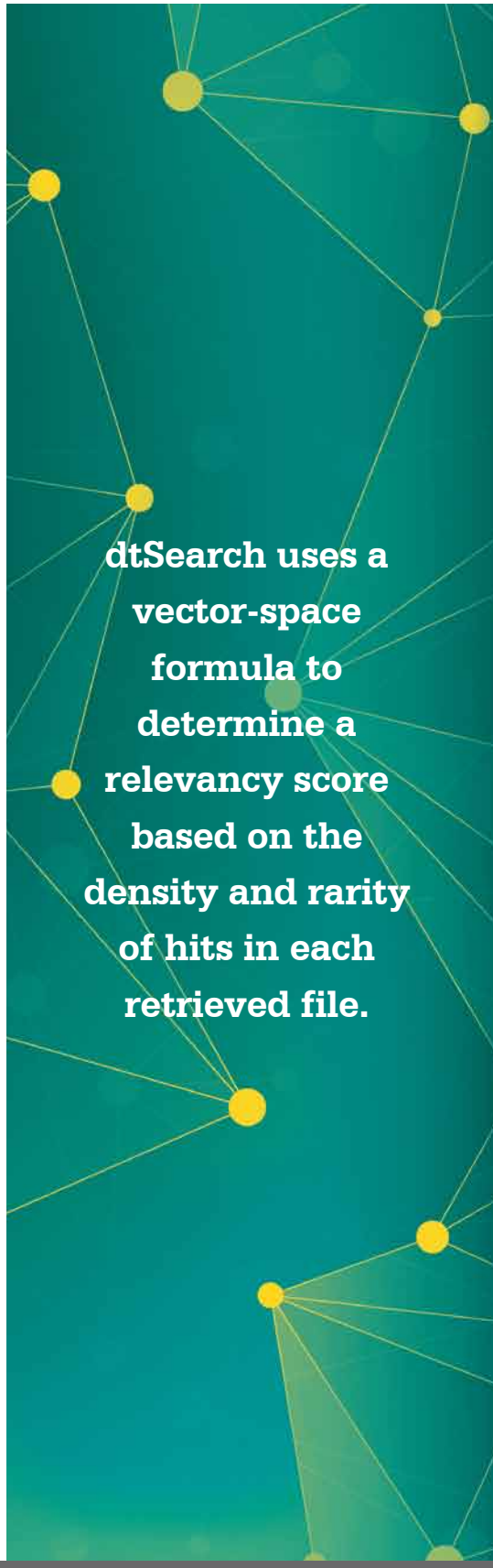
Users can browse the full text of retrieved files, jumping from one highlighted hit to the next. With only a few hits, reviewing all search results is easy. But when a search retrieves thousands or potentially millions of items, sorting and relevancy become important.

With dtSearch, you can sort—or instantly re-sort—search results across all indexed data by ascending or descending number of hits in each file, file date, file name, file location or file size. You can also sort or instantly re-sort based on metadata like subject or in the case of emails, sender and recipient metadata. Or you can comprehensively sort by relevancy score.

This type of relevancy scoring is different from simple sorting by number of hits. dtSearch uses a vector-space formula to determine a relevancy score based on the density and rarity of hits in each retrieved file. Let's start with the density part. Say you are searching for *pumpkins*. If there is a short file that contains a lot of references to *pumpkins*, that file would rank more highly than a longer file that only contains one reference to *pumpkins*.

The rarity part of the formula depends on the prevalence of the search terms across the indexed data. Suppose you are searching for *pumpkins* or *bananas*. If you have a huge number of references to *pumpkins*, but *bananas* only appears a couple of times, files with *bananas* would obtain a much higher relevancy ranking. As the rarity calculation is data-specific, this calculation will differ from one indexed data set to the next. For one data set, *bananas* may be the rarer term. For another, *pumpkins* may be the rarer term giving those references a higher relevancy weight.

dtSearch further lets you apply a variable term weight to override the vector-space default. If you are searching for *pumpkins, bananas* or *coffee*, you could give pumpkins a variable weight of 9, *bananas* a variable weight of 2 and *coffee* a negative weight of 7. That way, files with *coffee* would be less relevant in search results regardless of the prevalence of *coffee* across the indexed data.



You can also give a positive or negative weight to a search term only if it appears in certain specific metadata or at the beginning or end of a file. You could give *Smith* a high relevancy rank if *Smith* appears in the sender field of an email or near the top of an email, and a lower relevancy rank if *Smith* merely appears in the recipient field or elsewhere in the full text. Of course at the time of search, you could simply exclude everything that contains the word *Smith* in certain metadata or anywhere in the full text. But sometimes you don't want to exclude a search term entirely, just make it less relevant.

Also helpful for searches with a large number of search results, dtSearch has an option to generate a search report after a search. This search report collects all hits with as much context as you request. For example, you can use dtSearch to find any credit card numbers in a data set. Then you could use a search report to review each instance in context across all data.

Developer-implemented faceted search would also fall into the general relevancy navigation category. This lets a dtSearch Engine developer enable the end-user to filter search results instantly by clicking on specific metadata. On the dtSearch.com site, there is an SEC Filings web demo. After doing a search, you can filter search results using SEC metadata.

You could search for *aluminum*, for example, and see all filings mentioning *aluminum* jumping from one hit to the next within all retrieved filings. Or you could click to refine the search to show search results from just Form 10-Q and Form 10-K filings. You could further click to limit these to just filings originating from New York, expand to add filings from California – or go back to filings from any state.

Another common way to leverage metadata for developers is to use specific metadata residing in a separate database like SQL, NoSQL or SharePoint, or even full-text data in the files themselves and then use that to granularly filter search results based on the credentials of each end-user. For an end-user from HR, the developer could show search results that may be relevant to HR, while filtering out results with an e-discovery data tag. A developer can filter search results for end-users authorized for e-discovery work based on the specific cases the end-user is working on and the level of access that end-user has. For example, anything with *Secret* in specific metadata or even in the full text of retrieved files might be off limits to anyone without the highest data clearance.

dtSearch has enterprise and developer products that run "on premises" or on cloud platforms to instantly search terabytes of "Office" files, PDFs, emails along with nested attachments, databases and online data. Because dtSearch can instantly search terabytes with over 25 precision search options, many dtSearch customers are Fortune 100 companies and government agencies. But anyone with lots of data to search can download a fully-functional 30-day evaluation copy from dtSearch.com

Article contributed by dtSearch®