Yes You Can: Instantly Search Terabytes

Article contributed by dtSearch®

Yes you can: instantly search terabytes with a search engine. But first, let's take a look at the alternative, going through millions of files and emails individually scanning for specific keywords or phrases. First, you'd have to retrieve each file and email in its associated application. Then you'd have to read the full text of each from top to bottom so you wouldn't miss anything. Add to that the need to check all footnotes and click around for metadata. At some point, you might as well spend your time counting grains of sand on the beach.

Maybe there's a better way to spend one's time?

Installing a search engine like dtSearch® can transform the whole process. Instead of individually reviewing each item in its associated application, a search engine approaches everything in binary format. But the search engine doesn't just efficiently access each file; it also automatically indexes the contents. Indexing ingests the full text and metadata, recording every unique word and number and the location of each in the data. A single index can hold up to a terabyte of text and dtSearch can create and simultaneously search as many terabyte indexes as you need.

How do you get the software to build an index?

Just point to the folders and the like you want to index and the software does everything else. To start with, the search engine automatically figures out the relevant file type of each item so it can apply the correct parsing specification. The parsing specification is very different depending on whether something is a PDF, a Microsoft Word, Excel, Access, PowerPoint or OneNote file, a web-based text format, or an email format. After indexing, the search engine can search terabytes of full-text and metadata content instantaneously. And not only is indexed search immediate, but it is also vastly more thorough than any human file review process.

How so?

To start off with, hidden text like black on black or red on red text is just as apparent as any other text in binary format. Same with obscure metadata that may be easy to miss when viewing a file in its associated application. The search engine can also handle recursively embedded data, like an email with a ZIP or RAR attachment including a PDF as well as a Word document that fully embeds an Access database.

What if a standalone file or an embedded file has a mismatched file extension?

An individual going through files can be tripped up by having a file saved with a mismatched extension, like a PDF saved with a .DOCX extension. But for the search engine, the file extension is irrelevant to the determination of what type of file each item is. In fact, the search engine uses the binary format of the file to determine the correct file type, without reference to the file extension at all.

Can multiple individuals search at once?

Concurrent search can operate across a standard network or in an online environment. In a cloud or other online environment, search can proceed in a stateless manner. Each search thread processes independently without affecting other search threads, making it very easy to scale. End-users can see their own search results with highlighted hits. And updating an index to add new content does not block out individual or concurrent searching. That way, indexes can always be up-to-date.

What types of instant searching does indexed search enable?

While a human reviewing a large number of files can maybe scan each one for a few words or phrases, the search engine can perform elaborate Boolean and/or/not as well as proximity queries. Concept searching finds thesaurus or user-defined synonyms. For text that may have scanning or typographical errors, such as emails where mistyping is common, the search engine has fuzzy searching adjustable from 0 to 10. That way, if *butterfly* is mistyped *buttonfly* in an email, a search for *butterfly* with fuzzy search on can still find it. The search engine also offers numeric-oriented searches like looking for specific numbers or numeric ranges, as well as specific date or date ranges even across multiple different date formats like *May 12, 2022* and *5/12/22*.

What about other numeric sequences?

dtSearch can, for example, take any X digits that might represent a credit card number and run it through a credit card validation application, and then flag it if it is a valid credit card number. Or the search engine can generate hash values of all indexed files.

How does relevancy-ranking work?

Relevancy-ranking is key if a search gets a lot of hits. By default, dtSearch employs a vector-spaced relevancy-ranking model, sorting search results by search term density and rarity across the indexed data. Say your search request includes the words *secret*, *server* and *project*. If *server* and project are prevalent in the data, but *secret* appears just a handful of times, then *secret* mentions would rank more highly. And files or emails with the densest *secret* mentions along with *server* and *project* mentions would rank highest.

Can you customize relevancy-ranking?

You can override default relevancy ranking by, for example, giving *server* a positive weight of 9, and *secret* and *project* lower positive weights of 4. Then maybe add in some other elements, like adding *Chicago* with a positive weight of 6 and *Minneapolis* with a negative weight of 7. You can also adjust relevancy ranking so that terms that appear in certain metadata or at the top or bottom of a file have extra positive or negative weighting.

About dtSearch®. 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 different search features, 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®