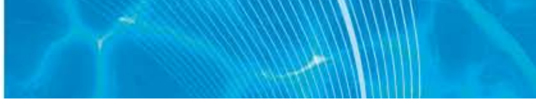
Drowning in Big Data? A Search Engine Can Throw You a Life Preserver



Article contributed by dtSearch®

Before I dive into this metaphor, I'd like to share a personal note. I was swimming with a group of people on a seemingly calm ocean day in Long Island when a sudden change in current pulled one swimmer into a large, jagged jetty, pulled another swimmer under the water, and pushed me out to sea. By some miracle, there happened to be three lifeguards on the shore, one to rescue each of us. But I learned that a relatively placid looking body of water does not mean that there is nothing going on underneath. Always swim—and have your kids swim—in a lifeguarded area. And hats off to lifeguards!

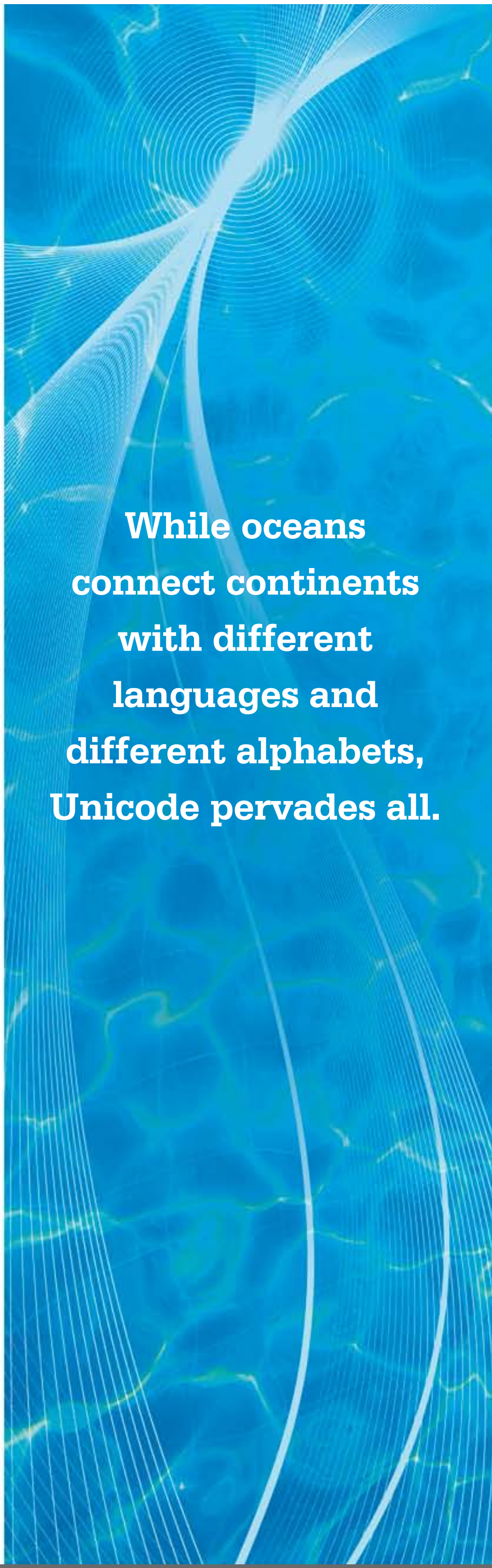
Back to the topic. If you are metaphorically drowning in Big Data, the type of search engine that can throw you a lifeline is not a scour-the-internet-for-the-best-beach-vacation-deal search engine. Rather, it is an enterprise search engine of which dtSearch® is an example. Enterprise search dives deep into terabytes of an organization's data, finding critical text even if it resides at the bottom of a long email exchange, in obscure metadata, or in a footnote on the 218th page of a file.

To instantly search terabytes, enterprise search first has to index the data. While resource intensive at the machine level, at a human level, indexing couldn't be easier. To kick off indexing, just point to the shared data folders, individual data folders, email archives, etc. to cover, and the search engine will do everything else. The data can be local or can reside in the cloud so long as the data appears as part of the Windows folder system. In all cases, the search engine will approach the files in their binary format, bypassing the need to retrieve each in its associated application, as that would be way too slow.

If you look at a binary format, you don't see easily readable text like you would in the same file in its associated application. Instead, you see a sea of binary codes. To identify all text and metadata, a search engine has to apply the correct parsing specification for that binary file. Parsing specifications can be hundreds of pages long and can update as a file format evolves, so matching the right one is essential. The search engine will look inside each binary format to make sure that it gets this determination right.

Applying the right parsing specification, a search engine can go much deeper in finding all text and metadata than a person ordinarily would:

- A misapplied file format extension like a OneNote file with a PDF extension that might confuse an end-user is no problem for the search engine as it can ignore the file extension in determining the applicable file type.
- Metadata that may take a huge amount of clicking around to uncover in a file's native application is readily available in binary format.



While oceans connect continents with different languages and different alphabets, Unicode pervades all.

Article contributed by dtSearch®

- Blue text against a blue background or white text against a white background that may be invisible in the file's native application is just text to the search engine.
- Recursively embedded attachments are fully accessible to the search engine, like an email with a ZIP or RAR attachment containing a Word document and a PowerPoint with an Excel spreadsheet inside—even if the spreadsheet would not be fully visible by default from PowerPoint.
- A search engine can even flag "image only" PDFs that may look like regular PDFs inside a PDF viewer like Adobe Acrobat Reader but that don't have any text accessible for functions like copy, paste and text search. Run these "image only" PDFs through an OCR engine like Adobe Acrobat and then return them to the search engine.

Once a search engine finishes indexing the Big Data, search threads can operate statelessly and independently. This allows them to proceed concurrently without impacting each other's instant operation. And with over 25 different search features, each search can be as simple or intricate as the searcher wants.

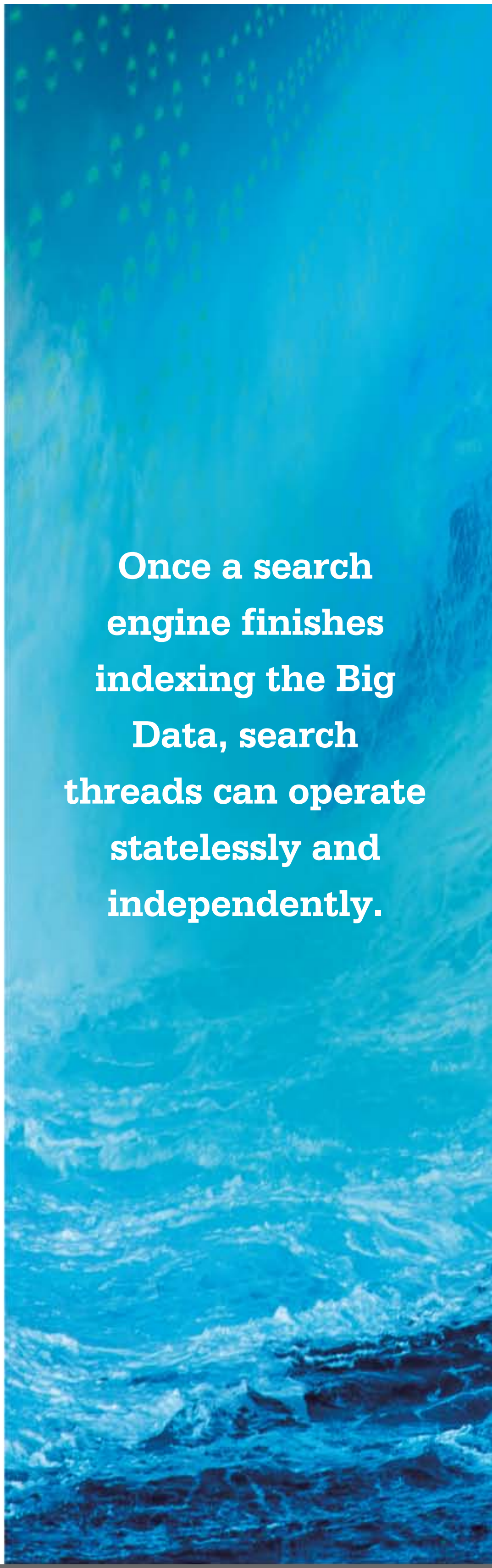
While oceans connect continents with different languages and different alphabets, Unicode pervades all. A search engine will auto-detect Unicode, including multiple languages in the same file. Unicode search covers not only European languages but also right-to-left languages like Hebrew and Arabic and even double-byte character languages like Chinese, Japanese and Korean.

Beyond searching for different combinations of words and phrases through natural language, Boolean (and/or/not), and proximity formulations, a search engine can also search for numbers and numeric ranges. A search engine can locate date or date ranges automatically in both full-text and metadata across popular date formats, like *July 2, 2022* to *10/25/23*. And a search engine can identify some number patterns, like finding any credit card numbers that may be lurking in data.

After a search, the search engine can display the full text of retrieved items with highlighted hits for convenient browsing. By default, the search engine will sort search results by vector-space relevancy ranking. What that means is if *minnows* are common in the ocean of indexed data but *angelfish* are rare, then angelfish will get a higher weight. And files with the densest mentions of angelfish will get the highest ranking.

Or a searcher can go beyond the default relevancy ranking with custom positive or negative terms-based weightings. The weighting can extend across any mention of a search term, or apply more heavily to specific metadata or positionally in files. Or a searcher can decide to instantly re-sort by some other unrelated metric like file date or file location for a new window on search results.

With a search engine, you are drowning in Big Data no more. And if you are out in the actual water this summer, use caution!



Once a search engine finishes indexing the Big Data, search threads can operate statelessly and independently.