

Text Strings and What They Mean.

Dr. Tom Johnston
MindfulData.com

Ontologies and taxonomies are making their appearance in IT organizations primarily as means to organize and access unstructured data, data contained in such structures as emails, scanned documents, word processing documents, PDFs. They don't seem to have much to do with structured data, which for the most part means data contained in production databases.

This is a mistake, which I will be discussing frequently, as I consider the mistake a particularly debilitating one. But for now, let's briefly look at document management.

Document management has been around for awhile, albeit as a somewhat neglected stepchild. IT organizations have traditionally devoted nearly all of their time, attention and resources to the management of the production data contained in relational databases or, in some cases, in file-based legacy systems. When they did turn their attention to document management, they found that there wasn't much to do.

There were non-electronic documents to be scanned. If these documents contained text, they could be OCR'd. At that point, they could be accessed by word processing or other text management software. They could therefore be searched for text strings, e.g. for the string 'pre-adjudication' in a document open in MS Word.

Searches based on boolean combinations of text strings provided an incremental improvement in semantic power. One could now search a document for example, for:

'pre-adjudication' AND ('HMO' OR 'Health Maintenance Organization')

As an aside, notice that "HMO" and "Health Maintenance Organization" are almost certainly synonyms. The former is an acronym for the latter. This reflects the fact that the semantic burden of creating a search for documents which talk about both pre-adjudication and health maintenance organizations is squarely on the person doing the search. He has to think about synonyms. He has to be sure to include them in his search criteria.

Placing this semantic burden on the person writing the query makes different results for the "same query" almost inevitable. For example, we might consider another person searching for the same information, whose search criterion is:

'pre-adjudication' AND ('HMO' OR 'Health Maintenance Organization' OR 'health maintenance organization' OR 'Health maintenance organization')

It is very likely that both these persons have the same query in mind, i.e. that both are searching for the "same thing", that being documents which talk about both pre-adjudication and health maintenance organizations. Yet over enough documents, the second query is bound to find more hits than the first one.

Having conducted their searches, we may assume that each of these two persons then reads the documents which satisfied their search criteria, and writes up their findings. These findings, of course, may differ, as significantly as you like. Decisions made and actions taken on the basis of one write-up may be quite different from those made and taken on the basis of the other write-up. Yet both write-ups purport to give decision-makers the "same information".

So what happened? Here we perhaps deepen our understanding of the distinction between data and semantics, between text strings and what they mean. For in both cases, the software used simply searched for boolean combinations of text strings. That software had no "understanding" of what those strings meant. The persons writing the queries had to translate their search for information into a search for text strings. They had to translate semantics into data. And for the same semantics, they produced different text string searches.

Perhaps I belabor the point. But while many IT professionals can use the "semantics vs. data" distinction pretty accurately, most would be hard pressed to explain how it differs from the "information vs. data" distinction that has been in use for a lot longer. And so I think it is worth our time to take a very theoretical term like "semantics", and attempt to link it to very specific situations like searches for text strings.

{to be continued.}