“Beehive”

Or, “Business Intelligence and a Solution for Unstructured Data”

By Thomas Park

I was doing some reading on Business Intelligence, and was interested in what is seen as an unsolved problem, according to the wikipedia. That problem involves unstructured data, and what to do with it.

Much of this unstructured corporate data is e-mails. E-mails are traditionally stored on a company server, in “blobs” of data-- big chunks of text taken directly from transmissions.

It is estimated that workers spend 30-40% of their time trying to locate information found in one or several of these blobs.

I had an idea-- why not use the coding I learned at “Launchcode” to create a sort of “Hive Mind”, using data from text blobs? The code would break the blobs up into phrases, and store each one in a database. Each time the phrase was repeated in the database, it would get a larger numerical value.

Users could query the “Hive”, asking for key words. The code retrieves phrases in which key text appears and lists them in order of their number of appearances in the database. The code adds a numerical value for the number of times a phrase is queried.

This way, eventually, each company develops its own unique Hive Mind, its own set of phrases. The more often used, the easier they become to locate.

After a set amount of time, old e-mails are simply deleted, leaving the database itself, with its statistically-marked pieces of text.

To facilitate this switch, high-security or -importance e-mails could be flagged, and either not included in the Hive database, or included and saved for the long-term, as well.

Low- to mid- importance e-mails might stay for a year or two in the server, then remain as references in the Hive, which more and more serves as a resource for realizing what gets discussed in the organization and how often.

I have created a prototype for this kind of application, which I call “Beehive”. From its web page, you can add text (up to 10 pages at a time seems to work), and query the hive database, to see if particular words or pieces of text appear in stored phrases. The phrases will be listed for their frequency of appearance and of being queried.

Here is that prototype: https://www.thomasparksolutions7.com/