

# Understanding Deep Web Technologies

A three part series of articles

Written by: Abe Lederman and Sol Lederman,  
Deep Web Technologies, LLC

## ***Mining the Deep Web***

*Issue 6 - January / February 2004*

## ***Challenges of the Deep Web Explorers***

*Issue 6 - March 2004*

## ***Beyond Information Clutter***

*Issue 9 - June 2004*

# Mining the Deep Web

In this first article in a series we introduce the deep web and tell you why, as a business or scientific professional you should care about mining its content. In later articles we will discuss in more depth some of the technical challenges to mining the deep web and how Deep Web Technologies and other companies are meeting those challenges.

The Internet is vast and growing - that's not news. Google does a great job of finding good information within it - that's not news either. What is news, and one of the dirty little secrets of Internet search engines, is that there's a huge collection of really useful content on the Internet that Google will never find - nor will any of its competitors, or any single search engine for that matter. We like to think that Google knows all, that if we click through enough of its search results we'll find whatever we need. This just isn't so. Beyond the 'surface web' of content that's continuously mined is the 'deep web'.

So, you're wondering, 'What is the deep web?' and 'Why haven't I ever heard of it?' In reality you've probably searched the deep web, maybe even surfed it, and never even realized it. The deep web is the collection of content that lives inside of databases and document repositories, not available to web crawlers, and typically accessed by filling out and submitting a search form. If you've even researched a medical condition at the National Library of Medicine's PubMed database [www.ncbi.nlm.nih.gov/PubMed/](http://www.ncbi.nlm.nih.gov/PubMed/) or checked the weather forecast at [www.weather.com](http://www.weather.com) then you've been to the deep web.

Three nice properties of deep web content are that it is usually of high quality, very specific in nature, and well managed. Consider the PubMed example. Documents cited in PubMed are authored by professional writers and published in professional journals. They focus on very specific medical conditions. The National Library of Medicine spends money to manage and make their content available. Weather.com provides timely and specific reports of weather conditions for all of the United States and much of the rest of the world as well. Both collections share the three properties.

The deep web is everywhere, and it has much more content than the surface web. Online TV guides, price comparison web-sites, services to find out of print books, those driving direction sites, services that track the value of your stocks and report news about companies within your holdings - these are just a few examples of valuable services built around searching deep web content.

So, why doesn't Google find me this stuff? The answer is that Google isn't programmed to fill out search forms and click on the submit button. The problem is that there are no standards to guide software like the smarts behind Google in how to fill out arbitrary forms. In fact, computers don't 'fill out' and submit forms, they instead interact with the web server that's presenting the form, and send it the information that specifies the query plus other data the web server needs. Each web form is different and there are too many of them so Google can't know how to search them all. Plus, it currently takes a human to 'reverse engineer' a web form to determine what information a particular web server wants. Standards are emerging to help with the content access problem

and software will certainly get better at filling out unfamiliar forms but we have a long way to go before most of the deep web is accessible to the next generation of web crawlers.

While filling out that web form is non-trivial it isn't the only barrier to accessing the deep web and it isn't even the hardest problem. Finding the best, or most relevant, content is harder. Within the deep web it means searching multiple sources, collating the results, removing duplicates and sorting the remaining results by some criteria that is meaningful to the person doing the searching. The problem of finding, aggregating, sorting and presenting relevant content is an involved one that we don't want to just gloss over so we will dedicate an entire article to discussing the issues.

As a professional you should care about What's in the deep web and about how to mine it effectively and efficiently. 'Why is that?' you ask. It's simple. In the worlds of business, science and other professional endeavors time is money. The slow and steady tortoise may win the race in fairy tales but it's going to get run over or left in the dust in today's competitive marketplace. The race to bring a new product to market, whether it be a new computer chip or a new drug, will be won by the company that can most quickly gather the most relevant information and intelligence and execute on it before its competitors do. A tool that can fill out forms on a number of web-sites with that high quality, specific and well managed content -- whether it be purchased, internal, or publicly available content -- then do the heavy duty processing to deliver the best of the best documents is worth its weight in gold. Such a tool will save you time and money and will make the best use of the content that you pay to acquire.

Imagine taking all of the intellectual property you possess or to which you have access and integrating its access into one simple to use form. Imagine further a system that knows what makes a certain document relevant to you as an individual. This system would be customized to scour your content plus all sorts of knowledge bases relevant to your needs and sift and sort information to present you with the very best of the deep web on demand. It would save you time. It would help you make money. This is the promise of deep web mining.

# Challenges of the Deep Web Explorers

Web spiders these days, it seems, are a dime a dozen. Not to minimize the tremendous value that Google and other search engines provide, but the technology that gathers up or “spiders” web pages is pretty straightforward. Spidering the surface web, consisting mostly of static content that doesn’t change frequently, is mostly a matter of throwing lots of network bandwidth, compute power, storage and time at a huge number of web sites. Merely throwing lots of resources at the deep web, the vast set of content that lives inside of databases and is typically accessed by filling out and submitting search forms, doesn’t work well. Different strategies and a new kind of “deep web explorer” are needed to mine the deep web.

Surface web spiders work from a large list, or catalog, of known and discovered web sites. They load each web site’s home page and note its links to other web pages. They then follow these new links and all subsequent links recursively. Successful web crawling relies on the fact that site owners want their content to be found and that most of a site’s content can be accessed directly, or by following links from the home page. We can say that surface web content is organized by an association of links, or in HTML jargon, an association of <A HREF> tags. We should note that spidering is not without its hazards. Spiders have to be careful to not recrawl links that they’ve previously visited lest they get tangled up in their own webs!

If spidering the surface web is not an impressive achievement then what makes Google’s technology so highly touted? In the case of Google and of other good search engines what’s impressive is not the ability to harvest lots of web pages (although Google currently searches over four billion pages) but what the engine does with the content once it finds it and indexes it. Because the surface web has no structure to it good search technology has to make relevant content easy to find. In other words, a good search engine will create the illusion of structure, presenting related and hopefully relevant web pages to a user. Google’s claim to fame is its popularity-based ranking. It structures content by presenting first to the user web pages that are most referenced by other web pages. The deep web is a completely different beast. A web spider trying to harvest content from the deep web will quickly learn that there are none of those <A HREF> links to content and no association of links to follow. It will realize that most deep web collections don’t give away all of their content as readily as surface web collections do. It will quickly find itself faced with the need to speak a foreign language to extract documents from the collection. This need is definitely worth meeting since the quantity and quality of deep web content is so much greater than that of the surface web.

Deep web explorers approach content searching in one of two ways, they either harvest documents or they search collections on the fly. A deep web explorer may attempt to harvest content from a collection that doesn’t support harvesting but, for reasons cited below, the effort will likely not be very fruitful. Dipsie and BrightPlanet are harvesters. They build large local repositories of remote content. Deep Web Technologies and Intelliseek search remote collections in real time.

Harvesting and real time search approaches each have their pluses and minuses. Harvesting is great if you have adequate infrastructure to make the content you’ve collected available to your users and if you have a sufficiently fat network pipe plus enough processing and storage resources to get, index and save the content you’ve obtained. Harvesting isn’t practical if the search interface doesn’t make it easy to retrieve lots of documents or if it’s not easy to determine how to search a particular collection. If the collection doesn’t support a harvesting protocol then harvesting will not retrieve all documents. Additionally, not having the network bandwidth and other resources makes harvesting impractical. And, if a collection is constantly adding documents then either the collection is going to somehow identify new content or you’re going to waste lots of resource retrieving the documents already in your local repository just to get a few new documents.

OIA, the Open Archives Initiative, is an example of a harvesting protocol. OIA describes a client-server model useful for aggregating multiple collections into a single centralized collection. The server tells the client, among other things, what documents are new in its collection and the client updates its repository with them.

Deep Web Technologies’ (DWT) Distributed Explorit application implements the other approach, the real-time search approach, which also has its pluses and minuses. A tremendous plus is that most deep web collections lend themselves to real-time searching even if they don’t lend themselves to harvesting. This is because by not implementing a harvesting protocol the content owner doesn’t have to do anything to its documents to allow them to be searched; it doesn’t need to generate metadata or otherwise structure its content. An on-the-fly search client uses the simple HTTP protocol to fill out and submit a web-form that initiates a query against the content database. The client then processes (parses) the content returned and displays search results to the user. DWT’s Distributed Explorit does multiple simultaneous real-time searches against different collections then aggregates the results and displays them to the user. The minuses of the harvesting approach become pluses in real-time searching. That entire infrastructure you needed to retrieve, store, refresh and index remote content and to then provide access to it disappears.

Minuses of real-time searching are the ongoing demands placed on the remote collection, the reliance on the availability of the remote content, the vulnerability of depending on search forms that change or break, and the inability to rank documents in a homogenous or effective way. (Search engines are notorious for ranking poorly or not at all and even collections that do rank documents in a relevant way can’t deal with the fact that their well-ranked documents will likely be aggregated with documents from other poorly ranked documents.)

Now that we’ve tapped into the vast content of the deep web we quickly discover that we’re drowning in content and not all of it is so relevant. What’s a web explorer to do with so many documents? We’ll explore this question next time.

# Beyond Information Clutter

Let's face it, there's way too much content in the Internet. While finding enough of it was once a big deal that's not the case anymore. Huge armies of web crawlers now make their home on the Internet, continuously examining and cataloging documents from all corners of the Web, making them all accessible through their search engines. Today's problem is sorting through all that content, finding what you want, and keeping the clutter down.

Google is my favorite search engine but it has some serious limitations. In 0.23 seconds Google can identify more documents than I can read in a thousand lifetimes! Fine, you say, Google ranks them for you so what you're likely to be most interested in will show up in the first 10 hits, right? Not always. Google mostly searches the surface web. That's the collection of documents that web crawlers can easily catalog. It doesn't include the deep web, which is the collection of content (much, much larger than the surface web collection) that typically lives inside of databases and has the higher quality scientific and technical information. Web crawlers don't know how to search deep web collections so they miss much of the content that serious researchers are seeking. So, the millions of documents that Google finds might not include the ones you're looking for, either in its first page of hits or in the first thousand pages of hits.

A more serious problem that Google faces, and I'm not picking on Google – all web crawlers have limitations – is that it doesn't necessarily rank documents the way you would rank documents. Google ranks a web page highly if it's popular, i.e. if lots of web pages reference it. Popular documents are not necessarily the ones most relevant to you. Google should be praised, however, in that it does rank in a way that is very useful to many people in many situations. Other search engines rank very poorly and some don't rank in any useful way at all. This is especially true for deep web search engines.

Let's consider one more problem that keeps you from finding what you really want, especially if you're looking for technical or scientific content. For this type of content you'll likely use a deep web search engine, like the one that Deep Web Technologies developed for [www.science.gov](http://www.science.gov) which searches a number of technical and scientific databases and aggregates results into a single results page. If we consider that each of the document sources that returns documents ranks them in its own way then we have the problem of how to rank the documents within the aggregate result set. In other words, if one source ranks documents more highly if they've been published recently and a second source ranks documents in alphabetical order by title and a third source ranks documents by frequency of search terms within them how does one rank the aggregate set of documents returned from the three sources?

This relevance ranking problem as it's called is a messy one but a very important one to solve because without good relevance ranking the result set becomes filled with clutter, i.e. with documents you didn't want. The solution has two parts to it. First, identify what makes a document relevant to a researcher and,

second, analyze all documents against those criteria. The first part is fairly straightforward. Good current and envisioned search utilities can allow the user to select from a set of criteria. Examples are publication date of article, length of article, frequency of search terms within it, proximity of search terms to one another, and presence of search terms early in the document.

Solving the second part of the problem, the rank and aggregation part, turns out to be difficult to do and requires lots of CPU, storage and network resources and it places a burden on the computer hosting the documents. It essentially requires retrieving the full text of the documents being compared in the ranking process. Without retrieving and analyzing entire documents we have no way of measuring the worth of the document against the user-specified criteria since the collection itself may rank documents poorly and most likely not according to our standards. Additionally, because different document sources rank differently the only way to rank all documents against one set of criteria, regardless of which collections they came from, is to ignore the order in which the source returns (ranks) the documents and apply our own ranking approach.

This approach which we call "deep ranking" is not for the impatient but for he who values a thorough search. Retrieving and analyzing the full text of thousands, perhaps tens of thousands of documents in response to a single query is best done in a batch-oriented environment, as it doesn't lend itself to real-time processing. The model is one of submitting a search and receiving an email when the processing is completed with links to the most relevant documents. There is some instant gratification here, however, as our approach will retrieve and rank a number of documents quickly using our "QuickRank" technology, currently in production at [www.science.gov](http://www.science.gov). The second benefit is the knowing that a large number of potentially relevant documents have been scoured and that only the best ones have been retained. Additionally, that small set has been optimally ranked and tailored to the needs of the researcher. It's worth the wait.

Deep Web Technologies has researched different ways to perform relevance ranking and has created a novel approach that can effectively mine large numbers of text document from heterogeneous sources and document type and produces a single set of well-ranked documents. The approach utilizes different algorithms for processing different types of documents at varying degrees of thoroughness. This approach can yield great benefits to pharmaceuticals, legal firms, biotechnology companies and other enterprises needing to effectively separate the clutter from the content.

[ Abe Lederman is founder and president of Deep Web Technologies, LLC, (DWT), a Los Alamos, New Mexico company that develops custom deep web mining solutions. His brother Sol supports Abe in a variety of ways. Visit [www.deepwebtech.com](http://www.deepwebtech.com) for more information including links to sites successfully deploying DWT's sophisticated search applications. ]