In Defense of Hyperlinks: A Response to Dreyfus[1]

Ian Stoner
University of New Mexico

An international group of technophiles known as the Extropians promise us a future in which digitized humans live forever in cyberspace, all the world's accumulated knowledge just a thought away. Hubert L. Dreyfus, on the other hand, promises that if we were digitized, we would find ourselves in the position of having "to give up being able to retrieve most of the information we needed" (Dreyfus 2001, p. 94). He sees the roots of this disastrous state of affairs in current search engines. He suggests that, because the Internet is anarchically structured, entirely lacking in a central organizing authority, people must depend on search engines to locate information for them. Search engines, however, are computer programs, capable of manipulating purely syntactic symbols but unable to understand semantic content. They are then charged with the impossible task of locating relevant semantic content based on syntactic analysis. Thus, Dreyfus warns, to the extent that we make ourselves dependent on search engines, to that extent we cut ourselves off from the vast quantities of knowledge we have archived on the Internet.

As a cautionary fable for users of technology, Dreyfus' thoughts are invaluable. If, however, he is seriously suggesting that in using search engines we risk obscuring centuries of acquired human knowledge, it may be that he is guilty of the same sort of extremism as the Extropians whose cypertopia he delights in dissecting. In this essay, I suggest that the document organization present on the Internet—which is to say, no central or authoritative organization at all—has important strengths that Dreyfus ignores. I then describe the salient feature of a search engine that will be able to piggyback on human judgments of relevance, dramatically improving our ability to locate documents on the Internet. Contrary to Dreyfus' dim view, moving documents from libraries to the Internet might allow us to ask questions that have been difficult or impossible to ask in the past.

The strengths and weaknesses of hierarchies

Libraries are at the root of all Dreyfus' talk of hierarchies. It is not hierarchies in the abstract that are threatened by the advent of the Internet, but libraries in particular. Before undertaking an examination of the Internet, it is important to understand how, for Dreyfus, libraries, as archetypical hierarchies, draw their power from their very structure.

Three classes of people participate in the functioning of a library. The first is the expert who designs the hierarchy that the institution will use to organize its holdings. This expert is extremely well versed both in "the meanings of the terms involved, and the interests of the users" (p. 9). (Melvil Dewey is an example of such an expert.) The second class of people is the librarians that populate the hierarchy with books. They read enough of a given book to understand its contents, and then place it in the appropriate branch of the hierarchy. Finally, there are users who walk the hierarchy to find the subject in which they are interested. When they have found it, they will be presented with all the books that were placed there by those who populated the hierarchy.

The appeal of this method of storage is that information has been grouped together by a person who understands what information *ought* to be grouped together. We can imagine a hierarchical classification with "books" as the root. Each branch will be defined as "books about x," with 'x' growing increasingly specific with each branching iteration. The leaves of the hierarchy will be very specific branches, containing a few books. This way, if I seek a book about 'x,' I simply pluck that leaf (or branch, depending on how specific my interest is) from the tree, and I should have every single book in the hierarchy that is relevant to my query. For example, if I would like to learn about Sir Gawain, I need only descend from "Literature" to "Medieval Literature" to "Arthurian Romance" and so on, until I eventually come to the category labeled "Gawain." Here, the librarians will have placed every book they read and judged to be about that particular knight.

There are two significant problems with these sorts of hierarchies. The first is that certain documents resist being placed in them. But even an unattainably perfect hierarchy, in which this problem had been solved, would face a possibly more significant problem: the class of queries answerable by the hierarchy is very small, and probably doesn't capture the very queries that tend to interest us most as humans. I will discuss both of these problems in more detail, to make clear exactly what they are, and how it is that they both have as their source the hierarchical structure itself.

The first problem, that of documents resistant to placement, is a problem with the implementation of hierarchies. But it is a significant problem—in a strict hierarchical system, a book that will not "fit" into the hierarchy is either left out, or forced into a leaf that is not entirely appropriate. In either case, the thoughts in that book are as good as lost to browsers of the library.

The problem of fit seems to appear in most, if not all, hierarchies of substantial complexity. It certainly appears in what is perhaps the most famous of all hierarchies: the taxonomy of living things. This hierarchy, the product of innumerable devoted scientists, all, presumably, experts, is far from perfect. Life is simply too complex and varied to be conveniently classified. Take, for example, the historical problem of the euglena, a single-celled organism that locomotes (an important feature of animals), but also photosynthesizes (an important feature of plants). For many years, the euglena was classified as a plant, despite its clearly animal characteristics. The result was that a strictly hierarchical search for "all creatures that self-propel" would miss the euglena, because it was filed in a branch of the tree that explicitly includes only those things that don't locomote.[2]

The source of the problem is that the structure of the taxonomic hierarchy insists that each organism be placed in one and only one category. (I will discuss, shortly, the most common method of addressing this problem in libraries.) It further insists that, while each category may have multiple children, it has one and only one parent. Under these conditions, the euglena is one of a number of examples of obvious problems of fit.[3]

Given the difficulty in classifying organisms, can we reasonably expect a rigorous taxonomy of ideas? Our ideas about animals are but a tiny subset of the entire corpus of human ideas. Classifying this subset has proved to be an intractable problem, and one may be confident that to classify the entire set will be much, much harder. This is challenge that libraries face.

Current libraries try to mitigate the problem of fit by permitting a single book to be filed in multiple categories. Dreyfus acknowledges this, but is adamant that the practice does not change the fact that, "there is an agreed-upon hierarchical taxonomy" (p. 110). His insistence that cross-referencing has not entirely undermined the hierarchical nature of the library makes him sound almost nostalgic for the Dewey Decimal System, which requires that one book have one and only one category. Such nostalgia would make sense, because the logical conclusion of the trend—the filing of a document in every category to which it is relevant—looks very little like hierarchy, and very much like the Internet. In other words, if the current techniques of cross-referencing were extended, the result would be the same leveling of content and undermining of meaning that Dreyfus sees on the Internet. If he is to argue for the power of hierarchies, he must resist their dilution through cross-referencing.

The second major shortcoming of a hierarchy (even a hierarchy that somehow managed to solve the problem of fit) is that it is not well suited to

many types of questions that interest us as human beings. There are at least two sorts of queries it cannot handle: queries that are interested in the relationships between objects, instead of the objects themselves, and queries that are interested in features of the objects that were considered incidental to the hierarchical branching principle.

Hierarchies are bad for relating information because there are no links other than the mother/daughter sort. If I am interested in the relationship between two leaves, I must trace up each of their branches until I find an ancestor node they share. While the common ancestor sort of relationship is important in some cases, it reveals nothing about the potentially numerous direct and meaningful connections between the leaves. For example, if I am interested in the relationships between the *Book of Job* and Chaucer's "Clerk's Tale" in *The Canterbury Tales,* the only sort of relationship the hierarchy can reveal is found in the common ancestral node. It is hard to imagine that such an ancestral node is any more specific than "Literature," which is not a very interesting relationship to note.

The history of databases reveals the practical shortcomings of hierarchical structures. Databases of one sort of another have been around almost as long as computers. First generation databases were "flat." That is, they were discrete tables of information, unable to relate the tables, or the data, to each other. This sort of data storage is very limited, both functionally and technically, and in the late 1950s, a great deal of research was done to develop a more powerful database. The result was a hierarchical database. Hierarchical databases dominated throughout the 1960s, but they were subject to shortcomings of their own. Although they did allow for some kind of relationship between data, they forced the user to represent all relationships as mother/daughter. This made it fantastically difficult to relate the stored bits of information in the innumerable ways that it was naturally related. During the 1960s, more research was done, and by the '70s, hierarchical databases had been entirely supplanted by relational databases. In relational databases, no relationships are dictated by the database's structure and the user of the database can define any relationship between any of the bits of data. Today, every widely-deployed database system (MySQL, Oracle, Access, etc.) is relational. The important thing to note from this brief history is that in relational databases, no piece of data is treated as more or less important by the structure of the database, and no relationships between the data are imposed or implied by the structure itself. Any piece of data can be linked to any other piece of data. In other words, these relational databases are, in spirit and in structure, much more closely related to the Internet than they are to hierarchical databases or libraries.

In addition to obscuring relevant relationships between objects, a hierarchy ignores the aspects of a work that are secondary to the hierarchy's branching principle. For instance, if I want to read up on breast cancer, a hierarchy might be very convenient. But if I am interested in works of fiction featuring characters with breast cancer, the hierarchy is useless. The same rigorous structure that makes the first query so convenient makes the second query impossible. Because most books will have many aspects, while each branch of the hierarchy can recognize only one, there are many more potential queries of the second sort than the first.

It is worth emphasizing that these "secondary" aspects are secondary in the hierarchy, and not necessarily in the book. Examples of books with multiple, important, levels of meaning abound, particularly in fiction. Is *The Crucible* about a puritanical town in 1692, or McCarthyism in the 1950s? In the attempt to decide on the book's "primary" subject, we are forced to favor either the intent of the author (in which case we would claim it is about American society in the 20th century) or the obvious subject matter of the book (in which case, it is about puritans in the 17th). Clearly, it is about both, and if a hierarchy forces us to choose a single subject under which to file it, it does us a disservice—whichever subject we choose, we will obscure an important aspect of the book.

The same method that was used to address the problem of fit, namely, cross-referencing, can be applied to mitigate the problem of secondary aspects. In the case of *The Crucible*, it looks like cross-referencing might be successful. File it under both relevant subjects, and the problem is solved. Such cross-referencing, however, can only be successful so long as we assume that all books have a small number of aspects. But this is not the case. The overwhelming majority of books will have many aspects, ranging from the centrally important to the tangential or trivial. The question we must struggle with, if we are to adopt the method of cross-referencing is: in how many categories is it reasonable to file the same object? Too many and the result is a hierarchy that has been undermined and leveled, too few and the result is the obscuration of relevant aspects of things.

What I mean by "leveled" here might need an illustration. A hierarchy such as a library is designed to have a large number of objects at the root, and very few in the leaves. For instance "literature" as a root node will have many books associated with it, but a terminal subject heading like "literature about regicide in 17th century Britain" will have relatively few. The problem with filing a book in multiple categories is that there are books like *The Brothers*

Karamazov, *The Iliad*, and *Moby Dick*, that are obviously relevant to a broad range of different subjects. If we suppose that there are many books of this sort, then we have many books each filed in many categories, and the terminal leaves of the hierarchy swell in size. A hierarchy in which the leaves have swollen to the size of the root is fully leveled and entirely useless. In general, a swelling of the leaves corresponds to a leveling of the hierarchy, and a reduction in its utility. Dreyfus sees the Internet much like a fully-leveled hierarchy. The structure of the Internet is such that the pool of documents that must be examined for any query is the entire body of documents on the Internet. This is no different than a "hierarchy" in which every document is filed in every terminal leaf.

The fact is that libraries *must* obscure relevant aspects of books in order to avoid this leveling and maintain their utility. Libraries are designed to answer the question "what is an example of a book about x?" and the only way this question can be answered is if a judgment about *the* topic of a book has been made. Cross-referencing is designed to avoid the problem of books that clearly have a small number of obvious, centrally important, subjects. The method of cross-referencing is simply not intended to allow a book to be filed in every category to which it is in some way relevant. Thus, even with cross-referencing, libraries are bound to obscure books that may be relevant to a user's query.

Dreyfus is right to claim that, in the evolution of the Internet, "no authority or agreed-upon catalogue system constrains the linker's associations" (p. 8). But we have seen the centrally organized, authoritative structure of even an ideal library can seal off important meaning by obscuring relationships between entities as well as by ignoring "incidental" or "secondary" aspects of objects. While the non-authoritative, non-agreed-upon, and seemingly chaotic structure of the Internet may make document retrieval more challenging, its structure does not, like that of hierarchies, seal off important areas of meaning.

The strengths and weakness of search engines

Search engines are subject to the same shortcomings that face every computer program. Thus, they succumb to Dreyfus' critique of strong artificial intelligence[4]: they will never (or at least not in the foreseeable future) be able to understand semantic content.[5] This is a major problem for search engines because, presumably, without being able to understand the documents they index, they will not be able to judge their relevance to any given query. This understanding of search engines is at the core of Dreyfus' grim presentiments

of the Internet's future. Dependent on search engines for all our document retrieval, we will never be able to know when better, more relevant documents were overlooked. Perhaps worse, we may be mollified by mediocre documents and never think to ask the question "how many better documents are out there?" As we move an increasing number of documents online, we risk losing them in a vast sea of other documents that neither we, nor the search engines, can navigate.

In developing his argument concerning the shortcomings of search engines, Dreyfus treats the matter of a computer's inability to judge matters of relevance in more detail. He gives the example of an allergy-prone jockey who finds himself on a racetrack covered in hay. The astute observer, noting the hay, will not bet on the allergic jockey. To get a computer to make the same move would be difficult, because the matter of hay on the track seems irrelevant to any discussion of shrewd betting strategies. The problem facing the computer is that "everything we know can be connected to everything else in a myriad meaningful ways" (p. 20). These myriad meaningful relationships cannot possibly be made explicit and programmed into a computer.

While Dreyfus is surely right that this is a problem for intelligent systems in general, and search engines in particular, this very claim is a hint that we might be on the right track with the radical interconnectedness of the Internet. It is certainly true that "everything we know can be connected to everything else in a myriad meaningful ways," and the Internet mirrors this with its anarchical structure. Hierarchies like the Dewey Decimal System, on the other hand, acknowledge exactly one meaningful connection: that between the mother and daughter node of the tree.

Where does this leave us? It would be unfortunate if the anarchic structure of the Internet allowed for more meaningful connections than a hierarchy, but the poor quality of search engines left us bobbing in a sea of irrelevant documents nevertheless. Given that Dreyfus is right that a search engine will never be able to read and understand the contents of a document on the Internet, is there hope for web searches?

The hope lies in the very hyperlinks that Dreyfus declares overly hyped. Where he seems to go wrong is in claiming "everything *is* linked to everything else on a single level" (p. 10, emphasis mine). What he should claim is that everything *can* or *could* be linked to everything else. It is important that it is not. The crucial fact he misses is that the human beings who write and read the content of web pages create the links. They

understand the semantic content of the pages they work with, and they create links between them. There is meaning in these links. Every link is a marker of a relationship that was observed and hard-coded by a full-blown human being. Like the librarians who read books and place them in the appropriate branch of the tree, the authors of web pages read other web pages and place them in relationships with their own. Dreyfus is unfair to the Internet, then, when he asserts that everything *is* connected to everything else. This is plainly not the case. If everything *were* linked to everything else, meaning on the Internet would be undermined. Similarly, if the hierarchy in a library were randomly designed, meaning in the library would be undermined. Neither system functions this way in theory or in practice, and it is not revealing to accuse either system of potentially allowing such abuse.

If there is meaning in hyperlinks, then a search engine can use them to improve its results. One straightforward way that search engines can make use of links would be to observe them to identify groups of web pages. Presumably, such groups will naturally form around most any subject imaginable. For instance, a site devoted to Plato's *Republic* will have more links to sites about *The Republic* and closely related topics, and fewer links to sites about, say, evolutionary psychology or the latest Star Trek movie. Likewise, a Star Trek fansite will contain more links to Star Trek related sites than to cooking or Cubism sites. Generally, a page devoted to a given subject will contain more links to other pages on the same or related subjects than it will to unrelated pages.

By observing networks of links and looking for clusters, a search engine could identify groups of web pages. After a search engine has identified groups of related pages, it is in a much better position to guess at meanings and, piggybacking on human judgments of relevance, make relevance judgments of its own.

Search engines like Google[6] already make use of link tracing in their algorithms, although not in quite the way I am suggesting. Google, for instance, observes the links that point *to* each page it indexes. It treats each link as a "vote" for the indexed page. It then uses these votes, combined with the strength of the search-string match, to rank the pages that are returned to the user.[7] The idea behind Google's page-ranking algorithm is that the more links there are to a given page, the more people read that page, judged it to be of high quality, and linked their own pages to it. Thus, by observing the behavior of linking, Google can glean information about human judgments of quality—judgments it could not possible make for itself. The central

difference between Google's technique and the one I am proposing is that Google tries to piggyback on human judgments of *quality*, while I suggest piggybacking on human judgments of *relevance*.

An example: suppose I search for "Star Trek news information" on my proposed search engine. There are a number of Star Trek sites on the 'Net and, presumably, they tend to link to one another. In short, the search engine can identify a group of Star Trek pages. Now, suppose there is a web page that contains the text: "I hate Star Trek, because (among other reasons) the people who put up fansites rarely update their news, making it impossible to find reliable information." This page, written by someone who dislikes Star Trek fansites, will not link to any, and therefore will not be part of the group of Star Trek related sites. Therefore, despite the fact that the isolated page[8] contains all the words in my search string, the search engine can guess that it is likely irrelevant to my query. The search engine, then, has used the anti-Trek author's own judgment of relevance, made evident by his failure to link himself to a Star Trek group, to make a recommendation to the user: don't bother with this page.

Another example from the Star Trek milieu: suppose a dedicated fan of the Klingon language has put up a web page devoted to the syntax and vocabulary of this alien tongue.[9] Further suppose that this fan treats Klingon as if it were a real, autochthonous language and as such, he makes no reference to Star Trek anywhere on his web page. Despite the fact that Star Trek is not mentioned, the site will be of interest to fans of the show, and they will link to it. Thus, the Klingon page will be part of a Star Trek group. If I run a query like "Star Trek alien language resources," the string match with the Klingon page is fairly poor, as two of the significant words do not appear on it at all. However, the group-identifying search engine might still be able to return it as a relevant document, because it is part of a group of pages that feature the words "Star" and "Trek" with overwhelming frequency.

In both of these examples, the search engine has used links to infer human judgments of relevance. In the first case, the program could guess that a page that was not part of a Star Trek group was not relevant to a query seeking Star Trek information. In the second case, the program could guess that a page included in the Star Trek group was relevant to Star Trek queries, despite the fact that the name of the show did not appear anywhere on the site.

Perhaps the most dramatic way to see the power of such a search engine is to imagine how it would stand up to Spammers. A Spammer is an author of irrelevant web pages—usually advertising—who would like his or her

irrelevant documents returned at the top of search engine rankings. The authors of search engines, meanwhile, want to do everything they can to keep Spam out of their results; if they return too much Spam, people will stop using their service. The basic tension, then, is that the engine authors want to return the most relevant documents possible, while the Spammers wish to have their own, useless documents returned. According to Dreyfus, "the ongoing war between the search-engine designers and the Spammers" illuminates the hopelessness of the searching situation (p. 94). He suggests that the search engines are at a disadvantage because they cannot understand the contents of the documents they index and thus, they will always be vulnerable to the cleverly innovative tricks of human Spammers.

Clever tricks or not, a search engine that could identify groups of web pages would have an insurmountable upper hand in the Spam war. Because no legitimate web site would intentionally link to a Spam page, Spam pages could only have outgoing links, or incoming links from other Spam pages. In this situation, a clear differential between internal and external links could be identified—Spam pages could be well-linked to one another, but not to authentic documents with meaningful content. Thus, a node of Spam would be recognized. The whole node would then be considered irrelevant to any query. This search engine would be extremely difficult for the Spammers to defeat. No matter how inviting the content of the page might appear after a simple syntactic analysis (for instance, some Spam pages accomplish this by including hundreds of bogus keywords), if it is recognized as Spam by the human beings who write web pages, it will not be linked; it will be relegated to a node of Spam, and ignored by the search engine.

The conclusion of all of this is that hyperlinks are not only markers of human judgements of relevance, but they are also readily recognized and processed by search engines. They are machine-readable relevance judgements. Thus, hyperlinks will allow search engines to piggyback on human judgements of relevance, greatly improving the quality of search results. Once search engines successfully exploit the meaning in links, the archiving of documents on the Internet might usher in a future considerably rosier than the one Dreyfus fears. It would be a future in which we had access to the undeniably relevant aspects of things, and the undeniably relevant relationships between things, that have been systematically obscured by hierarchies.

References

Dreyfus, Hubert. *On the Internet.* London: Routledge, 2001.

---

[1] Thanks to Hubert Dreyfus, for discussing some of these issues with me, and to Iain Thomson, Mark Stoner, and Gabriel Gryffyn for their comments on an earlier version of this paper.

[2] The only search method that could undercover the euglena in this hierarchy is one that goes to every node in turn and asks the question "do any of the things in this node self-propel?" But such a method of searching renders the hierarchy useless.

[3] Taxonomists eventually addressed the problem of the euglena by creating a new kingdom, Protista, which is essentially a dumping ground for small things that aren't quite animals, and aren't quite plants. The creation of this new kingdom has resulted in a host of new difficulties for the taxonomist. This illustrates an important point about classifications that are this complex: changing the hierarchy to address one set of problems tends to create a new set of problems.

[4] "Big" or "strong" AI, as opposed to "weak" AI, is artificial intelligence that attempts to fully replicate the mental abilities of human beings. Weak AI, a more recent endeavor, has largely sprung up from the ashes of strong AI's spectacular failure. Weak AI attempts to create a machine that is very good at a single, clearly defined task, such as playing chess. In order for search engines to be able to read and understand the documents they index, they would have to be dramatic examples of strong AI.

[5] According to Dreyfus, the central problem facing computers is that they don't have bodies. Without bodies, they cannot acquire the vast amounts of common sense knowledge that humans naturally accumulate through the trial and error experiences that necessarily follow upon being a living being. This sort of knowledge is as basic as: "when [George Washington] was in the Capitol, so was his left foot, and that, when he died, he stayed dead" (p. 16). Without this body of knowledge, a computer cannot hope to understand anything that was written which assumes it, i.e., anything ever written by a person. But the size of this body of common sense knowledge is inconceivably large, so to make it explicit and program it into a computer seems impossible.

[6] http://www.google.com

[7] See http://www.google.com/technology/index.html for a very brief overview of Google's page ranking method.

[8] "Isolated" only in the sense that it is isolated from the Star Trek group. It may well be linked in to any number of other groups, with different areas of focus.

[9] Klingon is a fictional language, spoken by the alien race of the same name in some Star Trek television shows and movies. The linguist Marc Okrand developed its syntax and vocabulary in 1984.