# The Mechanics of a Deep Net Metasearch Engine

Nigel Hamilton
Turbo10 Metasearch Engine
68 Manchester Road
London, E14 3BE, UK.
+44 (0) 20 7987 5460
nigel@turbo10.com

## ABSTRACT
The Deep Net refers to the thousands of topic-specific search engines on the Internet, including those that are inaccessible to traditional crawler-based search engines. Commercial metasearch engines have been slow to provide a simple, universal interface to these smaller topic-specific search engines. Turbo10 has developed a commercial metasearch engine that connects to these resources en masse (http://turbo10.com). Turbo10 automates the process of creating and maintaining software adapters that connect to, search, and extract results from a multitude of search engines. This poster outlines the functional mechanics of how Turbo10 searches the Deep Net.

## Keywords
Metasearch Engine, Deep Net, Information Retrieval

## 1. INTRODUCTION
Recent research has highlighted a large number of topic-specific search engines that are inaccessible to crawler-based search engines [1, 3, 4]. These engines have been variously grouped under the umbrella terms: invisible web [3], deep web [1], and hidden web [4]. The research has found that crawler-based engines cannot access the information stored in some of these engines, hence the monikers: invisible and hidden. Turbo10, however, prefers to use the term 'Deep Net' because some of these information sources are not web-based (e.g., peer to peer networks) and the contents of these databases are not hidden or invisible to metasearch engines. The challenges for a commercial metasearch engine are, first, to connect to these Deep Net sources, second, to select the most relevant, and third, to return relevant results as fast as possible.

## 2. ENGINE MECHANICS
To meet these challenges Turbo10's search engine is divided into three major subsystems: the Adapter Manager, Trawler Server and Browser (see Figure 1). The functional design of Turbo10 is different to other web-based search engines. In the interests of speed, the computational cost of information retrieval is mainly borne by the client web browser, not the server. Most metasearch engines do relevance ranking and results merging on the server-side. The problem is the server must wait for all the target engines to reply (or timeout) before sending the result to the browser. Waiting for the slowest target engine can hobble the response time of a metasearch engine.

Turbo10 performs relevance ranking, topic clustering and result merging in the client web browser, not the server. Rather than waiting for the slowest engine to respond, Turbo10 returns a result the moment the fastest engine responds. To achieve this, the server sends asynchronous messages to the browser and a client-side program caches all the results in memory. Because all the results are loaded at one time, displaying topic clusters and result pages does not require repeat trips to the server which makes browsing the results faster.

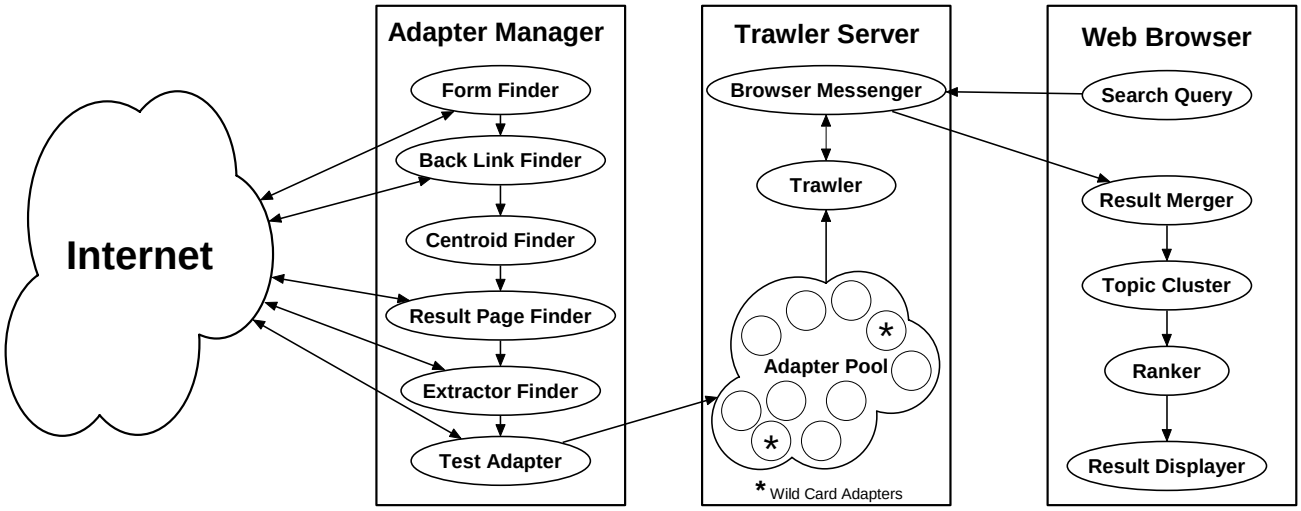


**Figure 1. Three major subsystems: Adapter Manager, Trawler Server and Browser**

On the server-side a bespoke web server, the Trawler Server, contacts target engines concurrently. As soon as the fastest engine responds the search results are sent to the browser. The results are also compressed to maximise the transmission speed. The Trawler Server holds a pool of adapters for connecting to search engines. Adapters encapsulate all the information required to connect to, query and parse the results of a target search engine. The adapters are automatically created and maintained by the Adapter Manager. Turbo10 initially created adapters using a manual process. A programmer would analyse the target search engine by inspecting the search form variables and craft a regular expression to extract details from the search results pages. However this process was found to be time consuming, error prone and difficult to maintain. To meet the challenge of creating and maintaining connections to thousands of engines in the Deep Net, Turbo10 required a fully automated process. The Adapter Manager fulfils this objective.

The Adapter Manager runs once a day and is responsible for testing existing adapters and connecting new ones. If an adapter is found to be broken the Adapter Manager will automatically attempt to fix it. The only information required to adapt a search engine is the URL of the web page that contains the search box. All other information is gathered automatically. Turbo10 has been accepting submissions for inclusion in its database of Deep Net engines since late 2001 and now has a large list. The Adapter Manager passes URLs from this list to the Form Finder component. The Form Finder locates the search form and identifies the parameters required to drive the underlying search engine including the query parameters, form submission method, cookie settings and the search URL. Once a valid form is found, a test query term is required to retrieve results from the engine. The test query term is drawn from the terms used in the pages that 'link back' to the search form.

The Back Link Finder retrieves the top 50 web pages that point to a target engine's search form. The context of each back link is extracted using structural cues found in the text of each back link page. The contexts from all the back links are then combined and passed to the Centroid component. The centroid is the top 100 most distinctive terms that describe the engine used in back linked pages. Distinctiveness is measured by using the pre-computed Inverse Document Frequency (IDF) values from an inverted index of the Open Directory Project database (http://www.dmoz.org). This outer centroid is calculated by multiplying the term weight by the number of occurrences in back link pages, then sorting the terms in order of decreasing weight, and taking the top 100 terms. A test query term is taken from the outer centroid and used to search the target engine. If the test query succeeds, the Result Page Finder verifies that a valid result page is returned. The next step is to find an extractor definition to retrieve the search results from the page.

The Extractor Finder is the most complex component in the system. It locates semantic and structural information in the results page that match the search result list. Once the list has been identified, individual results can be selected as candidate links. From the candidate links a pattern emerges and it is this pattern that forms the basis of the extractor. These details are then encapsulated in an Extractor object. The extractor definition is accurate but at the same time flexible enough to match different result pages from the same engine. Many heuristics were tested during development until a balance was found. If the extractor test succeeds, the adapter is completed and moved into the pool of adapters used by the Trawler Server.

Turbo10 searchers can include a number of wild card adapters in the collection of engines they choose to search. The Wild Card Adapter attempts to match a search query to the best engine on which to conduct the search. Turbo10 initially tackled this problem by using a taxonomy-based approach but this required manually categorising engines. To search the Deep Net en masse Turbo10 needed a fully automated system. Turbo10's taxonomy-based source selection algorithm has been replaced by a system that maps search query terms to adapter centroids.

The Wild Card Adapter performs this function. It is spawned in parallel like the other adapters and searches a pre-defined traditional crawler-based engine (e.g., http://altavista.com). The results page is analysed for sub-topic clusters in a process similar to Lin *et al.* [3]. For example, searching on 'crime' may yield the sub-topics, 'government' and 'police'. These sub-topic clusters are combined with the search terms and matched against the adapter centroids. Whenever an adapter's centroid matches, the weight of the term (found in the centroid) is added to the score for that adapter. At the end of the process the adapters with the highest scores are selected as Wild Card Adapters. Turbo10 plans to improve the matching algorithm by probing target engines for an inner centroid.

## 3. CONCLUSION

A metasearch engine that connects to thousands of target source engines is an ambitious endeavour. Turbo10 found that it was not feasible to tackle the problem with manual processes. As a result Turbo10 automated the process of creating and testing adapters that connect to, search and extract results from the Deep Net. This fully automated system enables Turbo10 to connect to the Deep Net en masse.

## 4. REFERENCES

[1] BrightPlanet LLC. The Deep Web: Surfacing Hidden Value. http://www.completeplanet.com/Tutorials/DeepWeb/index.asp, 2000.

[2] Hirokawa, S., Watanabe, S., Koga, Y., and Taguchi, T. Automatic feature extraction of search sites. In Proceedings of the International Conference Advances in Infrastructure for Electronic Business, Science, and Education on the Internet. SSGRR 2001 L'Aquila, 2001.

[3] Lin, K. and Chen, H. Automatic Information Discovery from the "Invisible Web". In Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'02), 2002.

[4] Raghavan, S. and Garcia-Molina, H. Crawling the Hidden Web. In 27th International Conference on Very Large Data Bases. Rome, 2001.