# SEARCH ENGINE SOCIETY

DIGITAL MEDIA AND SOCIETY SERIES

EXHIBIT 2087
*Facebook, Inc. et al.*
*v.*
*Software Rights Archive, LLC*
CASE IPR2013-00480

ALEXANDER HALAVAIS

First published in 2009 by Polity Press

Polity Press
65 Bridge Street
Cambridge CB2 1UR, UK.

Polity Press
350 Main Street
Malden, MA 02148, USA

ISBN-13: 978-0-7456-4214-7
ISBN-13: 978-0-7456-4215-4 (paperback)

A catalogue record for this book is available from the British Library.

Typeset in 10.25 on 13 pt FF Scala
by Servis Filmsetting Ltd, Stockport, Cheshire
Printed and bound in Great Britain by MPG Books Ltd, Bodmin, Cornwall.

The publisher has used its best endeavours to ensure that the URLs for external websites referred to in this book are correct and active at the time of going to press. However, the publisher has no responsibility for the websites and can make no guarantee that a site will remain live or that the content is or will remain appropriate.

Every effort has been made to trace all copyright holders, but if any have been inadvertently overlooked the publishers will be pleased to include any necessary credits in any subsequent reprint or edition.

For further information on Polity, visit our website: www.polity.co.uk.

providers: finding the least expensive airfares for a given route, for example.

Most crawlers make an archival copy of some or all of a webpage, and extract the links immediately to find more pages to crawl. Some crawlers, like the Heritrix spider employed by the Internet Archive, the "wget" program often distributed with Linux, and web robots built into browsers and other web clients, are pretty much done at this stage. However, most crawlers create an archive that is designed to be parsed and organized in some way. Some of this processing (like "scraping" out links, or storing metadata) can occur within the crawler itself, but there is usually some form of processing of the text and code of a webpage afterward to try to obtain structural information about it.

The most basic form of processing, common to almost every modern search engine, is extraction of key terms to create a keyword index for the web by an "indexer." We are all familiar with how the index of a book works: it takes information about which words appear on any given page and reverses it so that you may learn which pages contain any given word. In retrospect, a full-text index of the web is one of the obvious choices for finding material online, but particularly in the early development of search engines it was not clear what parts should be indexed: the page titles, metadata, hyperlink text, or full text (Yuwono et al. 1995). If indexing the full text of a page, is it possible to determine which words are most important?

In practice, even deciding what constitutes a "word" (or a "term") can be difficult. For most western languages, it is possible to look for words by finding letters between the spaces and punctuation, though this becomes more difficult in languages like Chinese and Japanese, which have no clear markings between terms. In English, contractions and abbreviations cause problems. Some spaces mean more than others; someone looking for information about "York" probably has little use for pages that mention "New York," for instance. A "stop words" and not included in the index because they are so common. Further application of natural language processing (NLP) is capable of determining the parts of speech of terms, and synonyms can be identified to provide further clues for searching. At the most extreme end of indexing are efforts to allow a computer to in some way understand the genre or topic of a given page by "reading" the text to determine its meaning.[1]

An index works well for a book. Even in a fairly lengthy work, it is not difficult to check each occurrence of a keyword, but the same is not true of the web. Generally, an exhaustive examination of each of the pages containing a particular keyword is impossible, particularly when much of the material is not just unhelpful, but – as in the case of spam – intentionally misleading. This is why results must be ranked according to perceived relevance, and the process by which a particular search engine indexes its content and ranks the results is really a large part of what makes it unique. One of the ways Google leapt ahead of its competitors early on is that it developed an algorithm called PageRank that relied on hyperlinks to infer the authority of various pages containing a given keyword. Some of the problems of PageRank will be examined in chapter 4. Here, it is enough to note that the process by which an index is established, and the attributes that are tracked, make up a large part of the "secret recipes" of the various search engines.

The crawling of the web and processing of that content happens behind the scenes, and results in a database of indexed material that may then be queried by an individual. The final piece of a search engine is its most visible part: the interface, or "front end," that accepts a query, processes it, and presents the results. The presentation of an initial request can be, and often is, very simple: the search box found in the corner of a webpage, for example. The sparse home page for the Google search engine epitomizes this simplicity. However, providing people with an extensive set of tools to tailor their search, and to refine their search, can lead to interesting challenges, particularly for large search engines with an extremely diverse set of potential users.

In some ways, the ideal interface anticipates people's behaviors, understanding what they expect and helping to reveal possibilities without overwhelming them. This can be done in a number of ways. Clearly the static design of the user interface is important, as is the process, or flow, of a search request. Westlaw, among other search engines, provides a thesaurus function to help users build more comprehensive searches. Search engines like Yahoo have experimented with autocompleting searches, anticipating what the person might be trying to type in the search box, and providing suggestions in real time (Calore 2007). It is not clear how effective these particular elements are, but they exemplify the aims of a good interface: a design that meets the user half-way.

Once a set of results are created, they are usually ranked in some way to provide a list of topics that present the most significant "hits" first. The most common way of displaying results is as a simple list, with some form of summary of each page. Often the keywords are presented in the context of the surrounding text. In some cases, there are options to limit or expand the search, to change the search terms, or to alter the search in some other way. More recently, some search engines provide results in categories, or mapped graphically.

All of these elements work together to keep a search engine continuously updated. The largest search engines are constantly under development to better analyze and present searchable databases of the public web. Some of this work is aimed at making search more efficient and useful, but some is required just to keep pace. The technologies used on the web change frequently, and, when they do, search engines have to change with them. As people employ Adobe Acrobat or Flash, search engines need to create tools to make sense of these formats. The sheer amount of material that must be indexed increases exponentially each year, requiring substantial

# Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

## Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

## Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

## Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

## API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

### LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

### FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

### E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

**WHAT WILL YOU BUILD?** | **sales@docketalarm.com** | **1-866-77-FASTCASE**