Web Search

Introduction

Chapter 1 of textbook plus more

Disclaimer: *These abbreviated notes DO NOT substitute the textbook for this class. They should be used IN CONJUNCTION with the textbook and the material presented in class. If there is a discrepancy between these notes and the textbook, ALWAYS consider the textbook to be correct. Report such a discrepancy to the instructor so that he resolves it. These notes are only distributed to the students taking this class with A. Gerbessiotis in Fall 2015 ; distribution outside this group of students is NOT allowed.*

1.1. Web Search: What is it? Web search is an instance of **information retrieval** (IR) and involves the search of web-based information. Information retrieval is a more generic term that concerns the retrieval of any piece of information web-based or otherwise, and it is different for example from the term **data retrieval** (which is more database-centric).

Since the mid-1990s, Web-based search engines have matched the explosive growth of the World-Wide Web to provide a tool for web search. However they have risen to prominence since the introduction of the Google search engine that facilitates web searching and makes web search capabilities commercially exploitable through advertising.

1.2. Information Retrieval (Definition). It is define (Salton, 1968) as "... the field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."

1.3. Data Retrieval vs Information Retrieval. The concept of **data retrieval** is database-related and we are not concerned with it. In data retrieval information is organized (eg. database tables), whereas in information retrieval, data (aka information) is less structured.

1.4. Online searching and Web search(ing). Online searching has been available for some time, since the 70s. Information retrieval systems such as LexisNexis, DIALOG/ProQuest, MEDLARS, MEDLINE, have provided full-text and bibliographic search capabilities for text and text-related documents. Several early web-based systems searched only lists of documents or their titles. With the development of systems such as WebCrawler and AltaVista or Excite and Infoseek web-users became capable of searching the full text of web pages.

1.5. Web Search can be realized in a variety of ways that includes the following.

- **A Search Engine**, that searches and indexes Web-based documents.
- **A List of Web Directories** that classify a list of Web documents by subject, and
- **Link Information** that can be used to exploit the hyperlink structure of the world-wide web, to facilitate the search (rather than rank the results of a search engine).

1.5.1. Web Search using a Search Engine is a Syntactic Search. Web-searching is mostly associated with its realization by a search engine which is an IR system and deals, among other things, with locating and then retrieving the relevant information, if it is not stored locally, organizing it and storing it in an efficient form, and then ranking the results of the retrieval process. It is noted that web-searching is syntactic search. We search user-specified words or patterns in the text of a web-based document. We do not (ordinarily) do a natural language analysis of the text to say extract the text semantics.

1.5.2. A Search Engine nowadays is more than Web Search realization or Syntactic Search. In several search engines nowadays one can do calculations (eg. writing a `10 + 20` in Google's text box won't JUST find for you the documents satisfying that query) or currency conversion(s) or even track postal packages!

1.6. Web search can be used to satisfy

- **ad-hoc** search requests based on user queries (in the search box of a search engine for example),
- **filtering** that involves the search of say news items that are of interest to a person,
- **classification** requests that help automate Web directories,
- **question answering** that answers specific user questions that utilized the same interface or mechanism that ad-hoc requests are using.

1.7. Specialized forms of (Web search or not) include the following.

- **vertical** search where the domain of search is a specific topic,
- **enterprise** search where the target is the intranet of an organizational entity,
- **desktop** search that involves the document/files of a particular user/computer,
- **peer-to-peer** search that involves finding information in nodes of interconnected computers without central control.

1.8. A search engine is the application of information retrieval techniques to large collections of texts/documents. Search engines come in a variety of configurations to satisfy a variety of user needs.

- **Web search engines** such as Bing, Google, Yahoo! should be able to regularly **crawl** the Web and capture terabytes of data and provide responses to millions of queries per day.
- **Enterprise search engines** might be accessing only a fraction of this information by filtering out non-relevant sites or documents but they also provide services beyond search such as **data mining** or **clustering** tasks that involve the automatic discovery of interesting structure in data.
- **Desktop search engines** have a smaller document base (a user's or computer's contents) yet they need to be more responsive to user needs (and data creation) and provide a more user-friendly interface.

1.9. A Web search engine is a search engine, where the **Corpus**, i.e. the body of text used, is the **Web** or the Corpus is **Web-based**. Therefore it is a search engine specialized for searching the Web.

From that point on the prefix **Web** *will be dropped and the term* **search engine** *would imply the instrument for performing Web search!*

1.10. Critical Featurs of (Web) Search Engines.

Several search engines can be scaled or adapted to be Web-oriented or desktop-oriented at the same time. Google is such an instance. Critical features of search engines include the following.

- **1.10.1 Efficiency or Performance** measured for example by response time, throughput time, and speed of indexing, a process that improves the speed of search.
- **1.10.2 Incorporation of new data** that is determined by the **coverage** and **freshness/recency** of the indexed information.
- **1.10.3 Scalability** that ensures that the search engine works without problems when data or users grow.
- **1.10.4 Adaptability** issues that ensure that when components or aspects change (e.g. ranking algorithm) such changes can be easily incorporated with out major restructuring. Such issues can also include the search engine's capability to adapt to threatening (or misleading) environments such as that forced to them by `spam indexing` attempts for example.
- **1.10.5 Effectiveness** measured in how good (aka relevant) the results are!

By the way what is . . . Web ?

1.11. What is "the Web"? The term "Web" seems to include all the information located on the World Wide Web.

How do we find it, locate it, retrieve its documents?

Most of this information might or not be available for a variety of reasons:

- Not accessible through a web-document (no link points to it).
- Directory restricted (aka password-protected) by a user.
- Almost the same as above but also behind a paywall.
- Filename-extension not recognized by Web Search Engine.
- File-content not recognized by any Web Search Engine (eg no filename extension).
- Hidden otherwise (a catchall phrase).

1.11.1. (Surface) Web In some sense we could say that Web search mostly refers to searching of the **Surface Web**, that is all the information of the World-Wide Web that is readily and easily accessible through hypertext links (and is part of neither the Deep Web nor the Dark Web).

1.11.2. Deep Web: Mostly unreachable Although Web-search capabilities have expanded in the past 10 years, there are several web-pages that are STILL unavailable to web searching. This is what is known as the **Deep Web**, information hidden behind commercial subscription-based products such as the information-retrieval systems mentioned earlier, or accessible through database-based accesses, or otherwise available selectively through login access or just not available through a link from the Surface Web. (The protected area of the course Web page falls into the Deep Web, and so do subscription pages, and also most of Facebook, for example.)

Needless to say that multimedia data are in their majority little searched as well. **So data files that Web Search Engines do now know how to search them fall under the umbrella of what is known as Deep Web.**

1.11.3. Dark Web It is Deeper than Deep, and Darker than Surface. The **Dark Web** is the portion of the Web that requires anonymous access through Tor (The Onion Router) or ther anonymized protocols.

1.12.1 Size of (What?) Web. It is estimated that
the Deep Web is approximately 500 times larger than the Surface Web.
Email alone and not counted as part of Deep Web is 2500 times larger than Surface Web and dwarfs both.
Instant messaging is approximately 3-4 times of Surface Web.

Note. Beware of published work on estimations of the size of Surface and Deep Web. Sometimes these estimates are using size as in GiB,TiB, PiB, sometimes size is in web-pages, or web-sites or web-servers.

Parenthesis: Bits and Bytes and Mega, Giga, Tera or Peta bytes. The minimal amount of information is a bit which as an acronym for binary digit. The minimal amount of information stored into the memory of a computer is a Byte (sometime referred to as an octet) which is 8bits. Informally a byte is the minimal amount of information that we can "bite" out of memory. A bytes can be represented with a capital-case B as in `10B` but a bit is always written in full as in `10bits`. A 10b is nonsense. For megabytes, kilobytes, etc, one is safe in determining size unequivocally by using the SI system in which a `1KiB`, `1MiB`, `1GiB`, `1TiB` and `1PiB` are respectivelly equal to 2^{10}, 2^{20}, 2^{30}, 2^{40}, 2^{50} and are read as kibi-bytes, mebi-bytes, gibi-bytes, tebi-bytes, and pebi-bytes. Sometimes they are all called kilobinary, megabinary, gigabinary, terabinary and petabinary. Note that `1KB` is not a kilo-byte as a kilo is always with a lower case `k` and a capital case `K` is degree Kelvin.

1.12.2 Host Name. A host name is a symbolic name to refer to a web-address/web-site (eg. `www.njit.edu`). (Note that a hostname can also be an ftp server, a mail server, or an Internet resource in general.) The portion `njit.edu` is known as the domain name; `njit` is the organization name, and `edu` is the organization type.

1.12.3 IP Address. An IP adress, in particular an IPv4 adress, is a 32-bit unsigned integer expressed in the form of 4 numeric numbers separated by three periods which are in the range 0-255. Thus the IP address of the host name above is written as 128.235.251.25.

1.12.4 Host Name to IP address Mapping: One-to-Many. The same host name may have more than one IP addresses. As of this writing (August 2015), the host name www.google.com is mapped to several IP addresses including 74.125.141.103, 74.125.141.104, 74.125.141.105, 74.125.141.106, 74.125.141.107, 74.125.141.147, 74.125.141.99.

1.12.5 Host Name to IP address Mapping: Many-to-one Moreover `www.google.co.uk` , `www.google.fr` and say `www.google.gr` all share the same IP address 74.125.141.94.

1.13 Web-site, Web-address Web-page.

A Web-site or Web-address is usually recognized or identified by its host name or an address that uniquely identifies it (URL: Uniform Resource Locator) and also identifies the protocol supported (eg http vs https separate with colons from the rest). A web-site or web-address might contain one or more **Web-page**(s).

1.13.1 Web-server. A web-site or a web-address or a web-page is supported by a program that is known as a web-server that is running on the hostname and serves web requests. The same web-server may serve multiple requests for multiple **web-pages** on a **web-site**, and several web-sites or web-pages may be hosted on the same **web-server**.

```
        August 2014                                  August 2015
 Host   Name:  www.google.com     www.bing.com       www.google.com     www.bing.com
 IP address :  173.94.46.112      204.79.197.200     74.125.141.103     204.79.197.200
 IP address :  173.94.46.113                         74.125.141.104
 IP address :  173.94.46.114                         74.125.141.105
 IP address :  173.94.46.115                         74.125.141.106
 IP address :  173.94.46.116                         74.125.141.114
                                                     74.125.141.99
 Host   Name:  www.google.co.uk/.fr/.gr/.de
 IP address :  173.94.46.111                         74.125.141.94
 IP address :  173.94.46.119
 IP address :  173.94.46.120
 IP address :  173.94.46.127

Year     Size                            Web-pages    Web-servers, sites etc

1995      0.03TB
1996      0.20TB
1997      1.80TB
1998                                      150mil
2000      19TB(Surface), 7500TB (Deep)    1000mil
2003     167TB(Surface),90000TB (Deep)    2500mil         43,000,000 web-servers; 50,000,00 web-sites
2004     170TB
2005                                      550mil unique   100,000,000 web-sites
2013                                                      500,000,000 web-site; 300,000,000 Host names

         Deep Web is  500 x Surface Web
         Email    is 2500 x Surface Web
         IM       is 3-4  x Surface Web
```


14.1 Information Retrieval: Text-based of Texts, Web-pages, and Multimedia and email. The major emphasis of **information retrieval** up until 1995 has been text and text-oriented i.e. information that is relatively unstructured; however the term information is quite general and also includes, besides text, multimedia. Nowadays although we still deal mostly with text-rerieval, increasingly information retrieval is also involved with sources of information that are more than text documents such as images (photographic or scanned), video, audio in the form of speech or music. Searching such sources of information is possible with current technology if some text-based tag is attached to In addition a "text document" includes not just traditional texts, but also web pages, email messages, books, research scholarly/academic papers, news items etc. Several of these documents have structure: (a) a subject or title, (b) an author, (c) a date of first publication, (d) some abstract or summary if they are lengthy. These constitute the **attributes or fields** of these documents.

How does Web affect IR? Where is the info? How do we get it? How do we process it? Well we need to locate the useful information. But where do we search? How do we organize the search? Is the information linked? Since it is web-based it is very likely disorganized, hard to find, and obeys or follows no useful data model. What we will do here is follow not a human-centric (e.g. Information Systems approach) trying to understand how people use and interpret information, but a computation centric (e.g. Computer Science approach) on how to structure, store, retrieve and rank relevant information.

14.2. Query and Retrieve. Similarly to traditional information retrieval, web-search involves a query and retrieval round.

14.2.1 Query. A user of a retrieval system formulates an initially imprecise natural language specification into a more structured and less imprecise query. The query might or might not be equivalent to the originally defined space of relevant answers.

14.2.2 Retrieval. A retrieval task is then executed. Many times a browsing task is performed that is interchanged with the retrieval task e.g. a hit or an answer of the retrieval task, is further explored (e.g. browsing an html document).

15.1. Document. A **document** is a single unit of information in digital form. Documents are usually represented through a set of words. It includes text and metadata.

15.2. Low Level: Tokens and words. A document is a collection of tokens that include words, numbers, dates etc.

15.3. High Level: Keywords. The important words i.e. the key words of a document are known as **keywords**. Keywords can be automatically extracted from the document or be user-specified (e.g. an expert decides them).

15.4. Logical view of a document. It is the representation of a document through a collection of keywords rather than all of its tokens. The keywords then form the **logical view** of a given document.

15.5. Full-text logical view is when all words become keywords.

15.6. Higher Level: Index-term. All or a subset of the keywords become **index terms** and can be used for searching the document(s) using its logical view. Because of space problems, an **index-term logical view** is more preferable where few of the keywords become index-terms. This can be accomplished by the use of a variety of **text operations** that will transform the original (possibly unstructured) text into a collection of index-terms. The further processing of these index-terms is called **indexing**. The resulting text would be searchable (after being indexed) through those index-terms ONLY. Nowadays it is not just the words (or the keywords among them) that become index-terms: numbers, dates, and other tokens can become index-terms.

Index-terms: words, dates, numbers, acronyms, etc. What becomes an index-term is a complex decision. Do number become index-terms? For example, how do you treat 1776? Do dates become index terms? Are 7/4, 4 July , 4/7/1776 of some meaning? Is 9/11 a calculation or a date? Is capitalization important and if yes what is an `Apple`, a fruit or something else? How about `Windows`? Is it part of a house? How do you treat Web-search, WebSearch, Web-Searching? Is it online the same as on-line. Do you process `United States` as two separate words or ONE? Is a cat an animal? How about a CAT or C.A.T (medical device or a bulldozer)? How many millions in a billion in the UK or the US?

16.1 Relevance and Relevant documents. A major issue related to information retrieval and also to Web search is that of **relevance**. A document is relevant if it contains information of interest ("lookable") to the individual performing the query (aka the search) at the time the search was issued.

16.2 Topical and User relevance.

Relevance can be `topical relevance` or `user relevance`. For example a search for ``` ``extrem weather '' ``` might return a typhoon story in Asia, (and note the intended misspelling of `extreme`) which is `topically relevant`; in the first week of September 2011, a NJ-based resident would have been rather more interested in `hurricane Irene` (and in September 2010 of hurricane Earl and in November 2012 of hurricane Sandy) or other hurricanes developing in the eastern US that were more `user relevant` at the appropriate time frame.

17.1 Querying a search engine: Query and Query language. A user types a query in the text box of a search engine that is a Web search engine. Then the engine processes the query and returns the relevant documents. Depending on its capabilities, those relevant documents might be returned in an arbitrary order or they may get ranked according to a search engine-dependent ranking function. The user types a query that conforms to the syntax of the **query language** that is supported by the search engine. However most users are oblivious to it or its syntax.

17.2 Query language: Operators and Operands. The query language of popular Web search engines are mostly transparent to the end-user. They consist of a number of operands which are the index-terms and a number of operators; some of the operators are drawn from Boolean algebra. But before we get into more details an arithmetic expression such as $5 + 6$ contains one arithmetic operator (the plus sign) that indicates the arithmetic operation known as addition. It also contain two operands on the left and right side of the operator. The left operand is a number (5) and the right operand is a 6. For this reason the plus sign for addition is know as a binary operator. Note however that $+$ as well as $-$ can also behave as unary operators. Thus $+5$ or -6 assign a sign to 5 and 6 respectively. In Boolean algebra we talk about conjunction, disjunction and negation. The query language of google supports operators `AND` , `OR` and `-` for these three operations. So does Bing that also allows a `NOT` for negation. In both systems, a space between two words is an implicit `AND` as well. Note that operators are case sensitive (capital case); yet if they lack the appropriate operands, they might also be treated as index-terms. User tokens are not case sensitive; one cannot expect a user to be knowledgeable about the intrinsics of a query language. The systems that parses user informations converts those user-provided tokens into index-terms using case-folding, stemming, and other similar operations.

17.3 Boolean Algebra. Be reminded that $A \vee B$ is the (inclusive) disjunction of A and B. It is true if A is true OR B is true. (This includes the case that both A is true and B is true.) The $A \oplus B$ denotes the exclusive disjunction (aka XOR). It is true if and only if exactly one of A and B is true (and the other false). (Thus if both A and B are true, it is false, and so will if both are false.) And $A \wedge B$ is the conjunction of A and B which is true if and only if A is true and B is also true (and otherwise, false). The negation of A is $\neg A$ or sometimes $\bar{A}$: If A is true, its negation if false, and if A is false, its negation is true! De Morgan's law establishes that $\neg(A \vee B)$ is the same as $(\neg A) \wedge (\neg B)$ i.e. $\neg(A \vee B) = \neg A) \wedge (\neg B)$
$\neg(A \wedge B)$ is the same as $(\neg A) \vee (\neg B)$ i.e. $\neg(A \wedge B) = (\neg A) \vee (\neg B)$. (The order of evaluation makes the parentheses redundant on the ride side of the equal sign.)

17.4. Query Language issues: Capitalization and Special Words. Every search engine has its limitations and idiosyncratic behavior.

```
data                                % Find all the documents relevant to data i.e. that
                                    % contain the word data
DATA                                % Same as above (DATA become data before query is processed)

                                    % A CONJUNCTION  of two (or more) words is satisfied if
                                    % all words appear in the document at least once. Whether
                                    % they appear exactly once, or more than once or next to
                                    % each other is not (very) relevant

data structure                      % Find all documents that contain the word data AND the word
                                    % structure (that's a Boolean conjunction  data AND structure)
                                    % Position is not relevant data might appear before or after
                                    % structure and many more times or fewer times than structure
data AND structure                  % Same as   data structure
data structures                     % Same as above: structures stemmed to structure before
                                    % evaluation
data         structures             % Same; spacing is irrelevant
structures data                     % Should return the same documents as the previous 4 queries
                                    % should it?
data and structures                 % and is not upper-case; it is treated as a word(index-term)
                                    % i.e. we have a three word query / conjunction equivalent to
                                    % data AND and AND structures
"data structures"                   % data must immediately precede structures; position is
                                    % important and in fact plural for structure might not get
                                    % stemmed (experiment with various engines
```

17.5. Query Language issues: Disjunctions and context.

```
                                    % Some query languages treat NOT as a negation, some not...
                                    % Several treat the dash - as a negation ; not many not! :-)

-data                               % Find all documents THAT DO NOT contain data
NOT data                            % for Bing same as -data
not data                            % Find all document thats contain the two words not and data
                                    % (order, occurrences, capitalization are irrelevant)
                                    % OR and not or become the disjunction operator if it can
                                    % be applied to two operands (left and right)
data OR structures                  % Return all documents that contain one or the other (or both)
structures OR data                  % Should be the same, should it?

OR                                  % OR is not treated as a binary operator
                                    % Is it Oregon or Operations Research?
or                                  % same as above
AND                                 % Return all docs containing AND (case independent)
and                                 % same as above

data or  structures                 % Return all docs containing all three terms
                                    % this is data AND or AND structures
                                    % lower-case   or  is not an operator!
                                    % space between words is equivalent to AND

data OR -data                       % Return all documents that contain
                                    % data OR do not contain data
                                    % (Google will not fall for it...
                                    %  Bing however is another story ...)

a b                                 % Observe and Understand ....
b a
a
a a
a a a
a a a a
a OR a
```

17.6 Pitfalls of a Query Language. For most web search engines an upper case `OR` is the logical operator dealing with disjunctions. Whether it is so or not might also depend on the context (existence of operands) and its position in the query. A lower case `or` most likely is not an operator! The same discussion applies to `AND` as well. For a `NOT` the case becomes a bit more complicated as some search engines does not recognize it as an operator.

Conclusion: Understand what query you are formulating or you will be surprised by the results!

17.7 To be or not be (capitalized)? The following example gives you an idea about the query language of Google and Bing around 2009. The situation nowadays is slightly (or not so) different.

Google @ July 2009		Bing @ July 2009	
Query Text	# hits reported	Query Text	# hits reported
TO BE OR NOT TO BE	7,200,000,000 hits	TO BE OR NOT TO BE	3,610,000,000 hits
to be or not to be	1,050,000,000 hits	to be or not to be	3,210,000,000 hits
to be OR not to be	11,400,000,000 hits	to be OR not to be	6,920,000,000 hits
"to be or not to be"	1,850,000 hits	"to be or not to be"	4,180,000,000 hits

17.8 Some more hacks. Even if Google and Bing allow the use of Boolean operators the query language supported but not be exactly one. Although conjunction and disjunction are commutative operators and thus $A \vee B = B \vee A$, this might not be the case for operators `AND, OR`. Repeating a term such as query `a` , or query `a a`, or `a a a` might give results not consistent with Boolean Algebra.

Several search engines allow for `operator:term` arguments anywhere in a query. For that case `operator` is one of `filetype, site, link, cache, intitle, inurl, allinurl allintitle, related`, `info, blogurl, numrange`, `inanchor, allintext` etc. The term can be an index term or for the case of filetype one of `pdf, bak, bat, txt, bin, gz, tar ini, js, log, php, sql, tmp, ps, xls doc`. And these lists are not exclusive, nor all supported by all search engines. Some examples are listed below. Note that a period is a wild character to denote any character, and a star any word.

```
site:njit.edu CS435      intitle:Algorithms                      allintitle:Algorithms Data Structures
site:gov IRS             insite:gov Untangling_the_Web.pdf       intitle:index.of password site:njit.edu
inurl:server-info        inurl:backup                            intitle:"Test Page for the Apache"
intitle:"cv"             intitle:"Directory Listing" "tree view" intitle:"Error Occurred"
```

18.1 Ranking relevant information: retrieval-based modeling

Besides **relevance**, a **retrieval-based modeling** attempts to present a formal representation of the process of matching user queries and (relevant) documents. It consists of **ranking** algorithms that are being used to rank relevant documents. A satisfactory retrieval model, ranks and returns documents that are to be considered relevant by the user. It includes a language component (for dealing with phrases, synonyms, and typos), and thus deals more frequently with statistical issues (word frequency) rather than purely linguistic issues. It might also include a time model (the relevant document is a new one generated 30 minutes ago, or a 10 year-old document), and sometimes a personalization component.

18.2 Evaluation (quality assurance) thus becomes important to refine and further improve proposed models. Cleverdon's (1960s) measures of **precision** and **recall** become important.

18.3. Precision is the fraction of the retrieved documents that are relevant to the user (to the total number of retrieved documents). In general, one wants to maximize precision as much as possible.

$$\text{Precision} = P = \frac{\#\text{ of retrieved documents that are relevant}}{\#\text{ of retrieved documents}}$$

18.4. Recall is the fraction of the relevant documents that are retrieved. One wants to maximize recall.

$$\text{Recall} = R = \frac{\#\text{ of retrieved documents that are relevant}}{\#\text{ of relevant documents}}$$

18.5 Precision or Recall? If a search engine returns for a given query the whole corpus collection (e.g. the whole web if it is a web search engine), the recall factor is going to be perfect. However precision is going to be very very low.

Example. Say that 40 documents are relevant to a query q. The model tested retrieved for query q 50 documents; out of those 50 documents only 30 are relevant. The precision is 30 over 50 i.e. $P_{50} = 30/50 = 0.6$. The recall is 30 over 40 i.e. $R_{50} = 30/40 = 0.75$

Note. Precision usually decreases in a well-defined/structured system.

Precision and Recall: Example

Example 1 *Say we have the sequence of documents (in* **bold-face** *are the relevant documents) shown.*

$$\mathbf{d_1} d_2 d_3 d_4 \mathbf{d_5} d_6 d_7 d_8 \mathbf{d_9} \mathbf{d_{10}} d_{11} d_{12} d_{13} d_{14} d_{15} d_{16} d_{17} \mathbf{d_{18}} d_{19} d_{20}$$

Suppose we measure precision and recall every time a document is encountered left-to-right. (One can use percentages to express recall and precision as we do in Figure 1.)

The precision falls from 100 to $40, 33\%$, then goes up to 40% and eventually stabilizes at 25%.

The recall starts with 20% and eventually grows to 100% assuming that the relevant documents are only the ones listed i.e. $d_1, d_5, d_9, d_{10}, d_{18}$.

Thus one can draw a curve for precision and separately for recall during this experiment.

Sometimes we prefer to use Figure 1(a) were recall and precisions figures are computed after every document is considered in turn. We can also draw a smaller table such as Figure 1(b) in which recall and precision figures are derived from Figure 1(a) at incremental recall rates: in our example we used a 6-step 20% increment table with rows at `0% , 20% , 40% , 60 % , 80% , 100 %` recall rates. One could have used instead (if it was possible) an 11-step 10% increment table.

For each such line the corresponding precision rate is also copied from Figure 1(a). An interpolated precision can then be derived, which is the maximal precision value encountered in that line or in lines below the given recall level. For example at 20% recall level the interpolated precision is the maximum of all precisions. At 60% it is the maximum of $33\%, 40\%, 27\%$. We can then compute a full-point average of interpolated precision rates. Thus a 6-point average gives us $(100 + 100 + 40 + 40 + 40 + 27)/6 = 57\%$. A 3-point average can usually be drawn at recall rates of $20\%, 50\%, 80\%$. In our case, the 50% recall rate uses a 40% interpolated preceision figure which was taken either from the 40% or 60% recall rate to determine a 3-point effectiveness.

The `3-point, or 11-point, or full-point` average of interpolated precisions are known as `3-point` **effectiveness**, `11-point` **effectiveness**, etc.

Precision, Recall and Effectiveness: Example

r	dr	Relevant	Rr (recall)	Pr (precision)
1.	d1	yes	20%	100%
2.	d2		20%	50%
3.	d3		20%	33%
4.	d4		20%	25%
5.	d5	yes	40%	40%
6.	d6		40%	33%
7.	d7		40%	28%
8.	d8		40%	25%
9.	d9	yes	60%	33%
10.	d10	yes	80%	40%
11.	d11		80%	36%
12.	d12		80%	33%
13.	d13		80%	30%
14.	d14		80%	28%
15.	d15		80%	26%
16.	d16		80%	25%
17.	d17		80%	23%
18.	d18	yes	100%	27%
19.	d19		100%	26%
20.	d20		100%	25%

(a)

Recall	Precision	Interpolated precision
0%	-	100%
20%	100%	100%
40%	40%	40%
60%	33%	40%
80%	40%	40%
100%	27%	27%
3-point average		100+40+40/3 =60%
6-point average		57%

(b)

Figure 1: Example: Precision and Recall and 3-point and 6-point effectiveness

High performance computing

Web search capabilities and performance is strongly related or affected by **high-performance computing**. Web search can not easily exist absent of a strong high-performance parallel computing environment; in most cases the latter is materialized in the form of PC clusters with hundreds or even thousands of PCs working cooperatively to complete a computational task (such as web search). Several such clusters might be used and spread in a variety of location all over a country (USA) and the world.

The location of these clusters can be dictated by a variety of reasons:

1 corporate criteria such as location of the mother company,

2 commercial such as closeness to the end-user, and

3 financial or hardware reasons, such as closeness to areas that offer high-speed networking and also cheap and reliable electricity that will fully support the data centers that are or will be built to house these clusters of computers that support web search.

In the context of Web searching and hardware, one needs to be reminded by several simple facts.

- A typical computer used for Web search is a dual-processor or dual dual-core or quad-core system with 4-8Gb of main memory and a pair of hard disks,
- It has approximately 400W of power consumption,
- On an annual basis this results to energy consumption of 3500KWh,
- With 10cents/Kwh this is approximately $350, or about one third to one quarter of the purchasing cost of such a configuration (air-conditioning, maintenance, networking and bandwidth are extra).

There are several issues related to web-searching in general:

- the end user has varying needs expressed in structured or unstructured queries and interested in structured or unstructured results,
- the data themselves may be structured or not.

We shall discuss several topics related to web-searching and also how they relate to high performance computing. Such topics include

- **Web search** How can we effectively search the web? What are its components?
- **Web crawlers** How can one build an efficient web-crawler that facilitates web-searching, and what issues does this involve? Why do we need such a component for web-searching?
- **Search Engines** How do they work ? How do they interact with other components of web-searching?
- **High Performance Search Engines?** Why do we need to optimize search engines? What are the issues involved?

The web-searching problem as it relates to search engines can be summarized into the following.
Say one is interested in searching for

to be or not to be

One can just type in the interface of a search engine or within the search toolbar of a web browser this sentence or text. One's expectations might be very different from the observed outcome of this search for a variety of reasons (an assignment will help you identify some of these reasons).

Who needs it?

Do we need Web search? Who needs it?

- Students who need to register for a course on Monday mornings!
- Instructors who need to offer a course that same day!

- Ok, let's get serious,

- Someone (or everybody) who looks up for something really important (e.g. Britney Spears, hm is this keyword so yesterday?, or Lady Gaga perhaps).
- Students looking for a quick homework answer!
- Researchers or journalists looking for something but lazy enough to go to the library (what is a library? when was the last time that you visited one?)
- Computer Scientists who want to apply their parallel computing or high performance computing skills to an area more interesting (and lucrative) than say parallel linear algebra.
- Computer Science students looking for an edge in the job market.
- Computer Science problem solvers who want to solve some interesting real-life problems related to web-searching.
- Everyday people addicted to getting answers really quickly.

What's in this course?

- Web-searching
  1. Web-crawling (collect web-pages from web for indexing in a higher performance environment)
  2. Indexing (parsing, word identification, stemming, data structures etc.) i.e. conversion of raw data (web-pages) into a searchable form.
  3. Query and evaluation of query results.
     (a) Link Analysis (sometimes part of query result evaluation). Ranking (e.g. Google's PageRank). If a query has 20 hits which one comes first? How does this affect the user experience and his/her evaluation of the search experience?
     (b) **To be OR not To be?** What happens to a search engine if you type this? Explain.
  4. Scalability issues.
- Special Topics
  - Google: How do they do it?,
  - Mapping the Web : Web structure,
  - Google's MAPREDUCE model.
  - etc.

What might not be in this course? Objectives.

What we will not be teaching you in these notes are the following.

- How to use Google, Bing, or other search engines! (You will have to figure it out soon!)
- How to format or write or design a web-page!
- How to use a markup language such as html!
- Programming!
- Technologies (SOAP/AJAX/XML)!

What you might learn instead includes the following.

1. Comprehend the science of the topics already mentioned related to web-searching.
2. Motivate yourself to explore and learn more about web-searching as an area that is still not fully defined.
3. Debug this course (first time offered in this format).
4. Understand the complexity and the science of the topics involved through experimentation (assignments).
5. Get acquainted with some recent or not so recent technologies/developments (papers).

Google and Bing and a, b

Google and Bing and a and b results (in millions). (As of Sep 1, 2015, 13:10.)

	Google@2015	Bing@2015
a	25270	14100
b	18120	3140
a b	3750	3170
a AND b	13870	3150
b a	3200	171
b AND a	13860	3150
a OR b	25270	3160
b OR a	25270	15000
-a	-	242
-b	-	289
a -b	25270	523
a AND -b	25270	315
b -a	5820	1310
b AND -a	615	1310
-a -b	-	205
-a AND -b	9110	179
-a OR -b	-	43000

Google and Bing and `tom`, `jerry`

Google and Bing and tom and jerry results (in millions).

	Google@2015	Bing@2015
tom	1300	95
jerry	321	18
tom jerry	126	14
jerry tom	127	11
tom AND jerry	52	17
jerry AND tom	127	17
tom AND -jerry	213	83
-tom AND jerry	60	12
-tom AND -jerry	25270	351
tom OR jerry	1600	123
-tom OR -jerry	-	43000

Google and Bing and alpha, beta

Google and Bing and alpha and beta results (in millions).

	Google@2015	Bing@2015
alpha	496	33
beta	682	33
alpha beta	20	12
beta alpha	17	10
-alpha	0	317
-beta	0	264
alpha OR beta	1160	127
alpha AND beta	212	6500
beta AND alpha	212	32600
alpha OR -beta	0	43100
alpha AND -beta	89	33
-alpha OR -beta	0	43100
-alpha AND -beta	25270	322

The most popular state .... after New Jersey of course. (September 2014.)

	Google	Bing
FL	167	354
CA	587	770
OR	1730	3930
FL OR	938	353
OR FL	1240	569
FL CA	764	316
CA FL	765	169
CA OR	2670	826
OR CA	2670	7
FL CA OR	1590	316
OR FL CA	1590	316
OR CA FL	2670	138
CA OR FL	1700	835
FL OR CA	1700	806

Exercise 1 *Identify the syntax of the query language for popular search engines such as Bing, Google, Yahoo!. How are* `OR`, `NOT`, `AND`, `-` *treated by them? Explain.*

Exercise 2 *Estimate the number of web-pages indexed by Bing and Google. Justify your answer. (Hint: Form a query whose list of relevant documents might be very close to the list of all documents indexed by these two search engines.)*

Exercise 3 *Formulate or try to formulate a "return all documents" query in Google and Bing, something close to* `a OR -a`.

