Enterprise Search vs Google Search

- Google indexes **mostly homogeneous html pages** that contain **metadata and special tags**.

- Google's search **rankings** are based partly on the number of **incoming links** to a site.
  - Web search allows users to find public content on external websites.

- Google primarily focuses on **pages**, Enterprise search primarily focus on **documents**.
  - Indexing of repositories like **databases, content file management**, emails and so on, is different from web search indexing.
Enterprise Search vs Google Search

• Inbound links are rare in the enterprise

• Pagerank is not useful since there are very less number of hyperlink and they are quite discrete. This hyperlink architecture also effects crawling system.

• Google is designed around public use with anonymous access. It isn't designed to handle secure documents.
  – Security is top most requirement of enterprise search and so access control also varies with user roles and privileges.
Enterprise Search – A Few Examples

- **Customer problem solving** – in engineering, finding and linking technically related case studies inside large companies.

- **eDiscovery for litigation** – exhaustive search that include generating, selecting, and searching across variations of keyword terms leads to automated indexing of a broad variety of document types (files, email, attachments, etc) found in enterprises.

- **Intelligence gathering** on personal communications for law enforcement – focus on tracking activities and gathering evidence over email, instant messaging, voice transcripts, etc leads to automatic identification of people, concepts, actions, and relationships of any kind including social networks analysis.
What Enterprise Search Is
The Problem of Enterprise Search

Why can’t I find information within my company as easily as I find it on the internet?

All I want is...

- Immediate effective answers from my Intranet searches
- Covering all my sources
- Securely and cheaply
Enterprise search – Challenges

• Enterprise information comes from many different types of sources.

• Determining importance of enterprise documents requires different techniques than on the Internet.

• Effective answers must go beyond result hitlists.

• Searches need to access password protected sources: email, databases, document repositories and so on.
Enterprise search – Challenges

- Enterprise Search results rankings are based on a complex algorithm, which can be customised by many factors, including industry taxonomy, business rules, date, author, content type etc.

- Business rules can cover document retention requirements, how documents are imported, authenticated, accessed, and other concerns.
Why Enterprise Search is Important?

- Empower your employees to **find the information** they need to act **quickly** and **decisively**.
  - An organizations’ ability to share knowledge among employees is vital to its ability to compete.

- Delight your customers with an insightful search for product and service information on your website; increase your sale or reduce your call centre costs.

- Make search experience painless for all stakeholders: employees, customers, partners, management.
Users Are Lost in Information Overload

Intranet
Email
Collaboration System
File Shares
Records Management
Document Management
Content Management
Time Tracking
CRM

Connects people to the information they need to get their jobs done

COMP6205 - Enterprise Search
Why Enterprise Search is Important?

- Search helps when you have too much information to browse.
- Search helps to extract information from fragmented sites.
- Silos of content that make up so many intranets
  - Each business unit has gone ahead and done its own thing, developing content haphazardly with few (if any) standards, and probably no metadata to support any sort of reasonable browsing.
  - Search won’t solve all of your problems, but your priority should be to set up a search system to perform fulltext indexing of as much system content as possible.
Enterprise Search provides Insight

• Search is a learning tool
  – Through searchlog analysis you can gather useful data on what users actually want from your information environment,
  – and how they articulate their needs (in the form of search queries).

• Over time, you can analyse this valuable data to diagnose and **tune** your search system, and other aspects of **information architecture** to achieve better results.
ES System Anatomy – Basic View

- On its surface, search seems quite straightforward.
  - Look for the box with the search button, enter and submit your query, and mutter a little prayer while the results load.
ES System Anatomy – Full Stack

Result

User query

User Profile

Personalization

Enhanced user query

Indexing

Metadata SQL Index

Metadata Text Index

Full Text Index

Thesaurus

Information Processing

Categorization

Entity Extraction

Concept Extraction

Metadata Extraction

Feature Extraction

Language Recognition

Information Discovery

Web Crawler

Database Crawler

Personal Crawler

Custom Crawler

Information on the Intranet

Shared File System

Intranet

Document Management System

IM

Email

Content Management

Web sites

Message Boards

Blogs

Databases

News Feeds

Online Subscription Services

COMP6205 - Enterprise Search
Enterprise Search Value Propositions

• Simple: Bringing the Google Web experience to the enterprise end-user
  – Create virtuous usage cycles through successful Search experiences
  – Enable easy administration with an out-of-the-box solution

• Better Answers: Providing relevant, timely information
  – Meet highest standards for relevance across sources
  – Extract structure from unstructured content to connect the dots and provide answers

• Go Deep: Finding the right information -- across all your sources
  – Penetrate entire data source universe
  – Ensure Unbreakable, fully-globalized, all-platform scalable operation
Information Architecture

The art and science of **organising** and **labelling** the **CONTENT** (documents, web sites, blog posts, database entries, etc.) to support **findability** and **usability**.
Determining Search Zones

• Search zones are subsets of an information environment that have been indexed separately from the rest of the content.

• Search zones allow a large body of content to be sliced and diced in useful new ways, providing users with multiple “views” of the environment and its content.

• For example in Windows 8.1 users can select search zones based on the type of content they are looking for, such as: Settings, Files, Web images, Web videos and so on.
Search Zones - Choosing What to Index

• You can create search zones in as many ways as you can physically segregate documents or logically tag them.
  – Content type
  – Audience
  – Role
  – Subject/topic
  – Geography
  – Chronology
  – Author
  – Department/business unit
Search Algorithms

• **Pattern-Matching Algorithms:** Most retrieval algorithms employ pattern matching; that is, they compare the user’s query with an index of, typically, the full texts of your system’s documents, looking for the same string of text.

• Some algorithms return numerous results of varying relevance, while some return just a few high-quality results. The terms for these opposite ends of the spectrum are *recall* and *precision*.

\[
\text{Precision Ratio} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of documents retrieved}}
\]

\[
\text{Recall Ratio} = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents in system}}
\]
Recall and precision – Examples

• If your system’s users doing legal research, learning about the current state of scientific research in a field, or performing due diligence about an acquisition.
  – In these cases, they’ll want high recall.

• On the other hand, a user who is looking for two or three really good articles on how to get stains out of a wool carpet will be hoping for high-precision results.
  – It doesn’t matter how many relevant articles there are if you get a good enough answer right away.
Search Algorithms – Detailed Approach

- Keyword Ranking
- Link Analysis
- Query Log Analysis
- Metadata Extraction
- Intelligent Matching
- Duplicate Elimination
Keyword Ranking

• Affect ranking by calculating relevance
  – Use term frequencies, position, and distribution information to determine the most relevant documents.

• Affect ranking by weighting terms
  – First retrieve documents containing both “Java” AND “XML”; if documents contain only one keyword, prioritise “Java”-only documents over “XML”-only.
Link Analysis

• The ‘Google Method’
  – The links pointing to a page determine its relevance

• Quality, not quantity
  – Authority and quality contribute more to ranking than sheer quantity of links
  – Links from PhD forums would contribute more to a technical page’s rank than links from general corporate directories
Query Log Analysis

• Leveraging search patterns
  – Which queries were made and how frequently
  – Which queries were successful/unsuccessful

• Refining response relevance
  – Spell checker can learn new terminology from certain query logs
  – What are popular documents – everyone searching for ‘Shuttle Schedule’ ended up on this page.

• Enabling advanced techniques
  – Personalization
Metadata Extraction

• Determining structure and significance
  – Extracting the semantic structure of a document from its physical representation enables heterogeneous documents to be collected in a common structure
  – Extracts title, author, table headings, subheadings, URL, domain, header, footer, self-described keywords, and more.
  – Keywords in the title and URL name suggest more relevance than in the body of a document

• Improving Relevance
  – If the search term ‘Package Shipping’ occurs in the HTML title of a page, or in the URL of a web site, that makes these sources more significant
  – If the search term ‘Error 3113 Causes’ occurs in a table heading in a piece of Email, that increases the likelihood of the mail being relevant
Intelligent Matching

• Exploit Linguistics
  – Stemming expands keyword search. By reducing a word to its stem, or root, form, variants with similar meaning are included in the search (Running = runs, run, ran)
  – Automatically removes stop words (the, and) from the query
  – Handles non-alphanumeric characters such as C++, 426-123-5611
  – Decompounding – search for bahn locates hauptbahnhof
  – Base letter conversion – crème = creme

• Application of keyword themes
  – Matches conceptually related terms
  – A search for ‘cars’ will return documents which do not contain the word ‘car’ but do include ‘coupes’, ‘convertibles’ or ‘sedans’

• Fuzzy and Soundex
  – Crispy Crème = Krispy Kreme
Duplicate Elimination

• Streamline results

  – Automatically remove URLs with duplicate or near duplicate content

  – A product development manager is looking for background information on a particular new product idea.

  – The first hitlist should contain highly relevant, distinct results – not ten versions of the same report
Presenting Results

• Which Content Components to Display

  – Display **less information** to users who **know** what they’re looking for, and more information to users who aren’t sure what they want.

  – show users who are clear on what they’re looking for only **representational** content components, such as a title or author, to help them quickly distinguish the result they’re seeking.

  – Users who aren’t as certain of what they’re looking for will benefit from **descriptive** content components such as a **summary**, part of an **abstract**, or **keywords**.
Different Ways of Viewing Search Results

- Yelp iPad app allows users to select three different ways of viewing search results: as listings, as locations on a map, or as images
Grouping Results - Clustering

- Clustering retrieved results by some common aspect.

- The obvious ways are, unfortunately, the least useful: we can use existing metadata, like document type (e.g., .doc, .pdf) and file creation/modification date, to allow us to divide search results into clusters.

- Much more useful are clusters derived from manually applied metadata, like topic, audience, language, and product family.

- Unfortunately, approaches based on manual effort can be prohibitively expensive.
Grouping Results - Clustering
Designing the Search Interface

• All the factors we’ve discussed so far—what to search, what to retrieve, and how to present the results—come together in the search interface.

• And with so much variation among users and search technology functions, there can be no single ideal search interface. Here are a few of the variables on the table:

• Level of searching expertise and motivation
  – Are users comfortable with specialised query languages (e.g., Boolean operators), or do they prefer natural language?
  – Do they need a simple or a high-powered interface?
  – Do they want to work hard to make their searches truly successful, or are they happy with “good enough” results?
  – How many iterations are they willing to try?
Designing the Search Interface – Cont.

• *Type of information need*

  – Do users want just a taste, or are they doing comprehensive research?

  – What content components can help them make good decisions about clicking through to a document?

  – Should the results be brief, or should they provide extensive detail for each document?

  – And how detailed a query are users willing to provide to express their needs?
Designing the Search Interface – Cont.

• **Type of information being searched**
  
  – Is the information made up of structured fields or full text? Is it navigation pages, destination pages, or both?
  
  – Is it written in HTML or other formats, including non-textual?
  
  – Is the content dynamic or more static? Does it come tagged with metadata, full of fields, or is it full text?
Designing the Search Interface – Cont.

• *Amount of information being searched*
  
  – Will users be overwhelmed by the number of documents retrieved?
  
  – How many results is the “right number”? 

Examples of ES - Microsoft SharePoint

• Microsoft SharePoint is probably the most widely installed of all search applications, with some organizations still using SharePoint 2010.

• The next versions, SharePoint 2013 and 2016 can be installed either on premise or in the cloud.

• There are two key points to take into consideration.
  – The first is that the full functionality of SharePoint 2013 search is only available against content that is stored in SharePoint.
  – The second is that specialised search expertise is needed to get the best out of SharePoint 2013.
Search architectures for SharePoint 2013
# Overview of Search Components

<table>
<thead>
<tr>
<th>Component name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawl component</td>
<td>Crawls content sources to collect crawled properties and metadata from crawled items and sends this information to the content processing component.</td>
</tr>
<tr>
<td>Content processing component</td>
<td>Transforms the crawled items and sends them to the index component. This component also maps crawled properties to managed properties.</td>
</tr>
<tr>
<td>Analytics processing component</td>
<td>Carries out search analytics and usage analytics.</td>
</tr>
<tr>
<td>Index component</td>
<td>Receives the processed items from the content processing component and writes them to the search index. This component also handles incoming queries, retrieves information from the search index and sends back the result set to the query processing component.</td>
</tr>
<tr>
<td>Query processing component</td>
<td>Analyzes incoming queries. This helps optimize precision, recall and relevance. The queries are sent to the index component, which returns a set of search results for the query.</td>
</tr>
<tr>
<td>Search administration component</td>
<td>Runs the system processes for search, and adds and initializes new instances of search components.</td>
</tr>
</tbody>
</table>
## Search Databases

<table>
<thead>
<tr>
<th>Search database name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crawl database</td>
<td>Stores tracking information and historical information about crawled items such as documents and URLs. It also stores information such as the last crawl time, the last crawl ID and the type of update (add, update, delete) during the last crawl.</td>
</tr>
<tr>
<td>Link database</td>
<td>Stores unprocessed information that is extracted by the content processing component and information about search clicks. The analytics processing component analyzes this information.</td>
</tr>
<tr>
<td>Analytics reporting database</td>
<td>Stores the results of usage analysis.</td>
</tr>
<tr>
<td>Search administration database</td>
<td>Stores search configuration data.</td>
</tr>
</tbody>
</table>
The Crawl Component

• The crawl component crawls the content sources.

• It can crawl lots of content sources, for example file shares, SharePoint content, line of business applications and many more.

• To retrieve information, the crawl component connects to the content sources by invoking the appropriate indexing connector or protocol handler.

• After retrieving the content, the crawl component passes crawled items to the content processing component.

• For more information about crawling content sources, see Plan crawling and federation in SharePoint Server 2013.
The Content Processing Component

- The content processing component processes crawled items and sends these items to the index component.
- The content processing component performs operations such as document parsing and property mapping.
- It also performs linguistics processing such as language detection and entity extraction.
- The component transforms crawled items into artifacts that are included in the search index.
- The content processing component also writes information about links and URLs to the link database.
- For more information about content processing, see Plan crawling and federation in SharePoint Server 2013.
The Analytics Processing Component

• The analytics processing component performs two types of analyses: search analytics and usage analytics.

• This component uses information from these analyses to improve search relevance, create search reports, and generate recommendations and deep links.

• Search analytics is about extracting information, such as links, the number of times an item is clicked, anchor text, data related to people, and metadata, from the link database.
  – This information is important to relevance.
The Analytics Processing Component

- Usage analytics is about analyzing usage log information received from the front-end via the event store.
- Usage analytics generates usage and statistics reports.
- The results from the analyses are added to the items in the search index.
- In addition, results from usage analytics are stored in the analytics reporting database.
- For more information, see Overview of analytics processing in SharePoint Server 2013
The Index Component

• The search index can be divided into discrete portions, called index partitions.

• The search index is the aggregation of all index partitions.

• Each index partition holds one or more index replicas that contain the same information.

• To achieve fault tolerance and redundancy, create additional index replicas for each index partition and distribute the index replicas over multiple servers.
The Index Component

• The index component is the logical representation of an index replica. In the search topology, you have to provision one index component for each index replica.

• The index component:
  – Receives processed items from the content processing component and writes those items to an index file. Index files are stored on a disk in the server that hosts the index component.
  – Receives queries from the query processing component and returns result sets.

• For more information about the search schema and the search index, see Overview of the search schema in SharePoint Server 2013.
The Query Processing Component

• The query component analyzes and processes queries and results.

• It performs linguistics processing such as word breaking and stemming.

• When the query processing component receives a query from the search frontend, it analyzes and processes the query to optimize precision, recall and relevance.

• The processed query is submitted to the index component. It returns a result set based on the processed query to the query processing component, which in turn processes that result set, before returning it to the search frontend.

• For more information, see Plan to transform queries and order results in SharePoint 201
The Search Administration Component

- The search administration component runs the system processes for search.
- This component performs provisioning, which is to add and initialize instances of the other search components.
Other Elements

• **The crawl database**
  – The crawl database stores tracking information and historical information about crawled items. For example, it stores information about the last crawl time, the last crawl ID and the type of update during the last crawl.

• **The link database**
  – The link database stores information extracted by the content processing component.
  – In addition, it stores information about search clicks; the number of times people click on a search result from the search result page.
  – This information is stored unprocessed, to be analyzed by the analytics processing component.
Other Elements

• **The analytics reporting database**
  - The analytics reporting database stores the results of usage analytics.
  - In addition, it stores statistics information from the analyses.
  - SharePoint uses this information to create Excel reports that show different statistics.

• **The search administration database**
  - The search administration database stores search configuration data, such as the topology, crawl rules, query rules, and the mappings between crawled and managed properties.
  - It also stores the access control list (ACL) for the crawl component. There can be only one search administration database per search service application.
SahrePoint Small search farm: 0-10 million items
Medium search farm: 10-40 million items
Large search farm: 40-100 million items
Open Source Search System - Lucene & Solr

- **Lucene** is an Open Source Search Engine that was first written by Doug Cutting in 1999.

- He transferred the rights of Lucene and **Nutch** (a Web Crawler) to the Apache Foundation in 2001, and it became a top-level Apache project in 2005.

- **Solr** was created in 2004 by Yonik Seeley at CNET Networks as an in-house project to add search capability for the company website.

- **Lucene** and **Solr** projects were merged in 2010 and are now generally referred to as Apache Lucene/Solr.
What Is Solr

• A full text search server based on Lucene
• XML/HTTP Interfaces
• Loose Schema to define types and fields
• Web Administration Interface
• Extensive Caching
• Index Replication
• Extensible Open Architecture
• Written in Java, deployable as a WAR
Concepts & Terminology

- **Apache Lucene** – is a full text search engine library written entirely in Java. Lucene is embedded with Solr.

- **Apache Solr** – is an enterprise search platform written in Java. It exposes web services that can manage the lifecycle of documents in the index.

- **Document** – is Lucene/Solr’s primary unit of storage – representing a flat collection of fields (no nesting).

- **Field** – definition consists of a name and configurable type (text, integer, double, date).

- **Core** – separate index and configuration. A single server can support multiple cores and it is used for data partitioning. Supports multitenant applications.
Concepts & Terminology

- **Shard** – Is a chunk of a larger index. They are created to scale an index horizontally across machines.

- **SolrCloud** – refers to a set of features that enable your search index to be scaled across a cluster of nodes.

- **Synonyms** – is a query expansion feature where (e.g. MB => megabyte)

Stop Words – are words that should be filtered from index storage and queries

- **Structured Content** – refers to content that has been richly tagged with metadata.

- **Unstructured Content** – MS Office, PDF documents, emails, instant messages, etc.

- **ACL** – access control list used to capture document permissions
Concepts & Terminology

- **Early Binding** – an authorization enforcement model where the document ACLs are stored in the index.

- **Late Binding** – an authorization enforcement model where document authorization is not determined until query time.

- **ETL** – extract (content source), transform (normalize the data), load (into index)

- **Search Based Application** – built on top of search platforms and they are designed to deliver unified information access.
Simple Search Architecture

[Diagram showing components of a simple search architecture, including:
- Solr Web Services
- FS Feed Utility
- File Share
- Index]
Enterprise Search Architecture
Extract-Transform-load (ETL) Process

- **Extract**
  - Content Source

- **Transform**
  - Centralize
    - Field Filtering
    - Field Mapping
    - ACL Mapping
    - Consider Groovy and Drools

- **Load / Publish**
  - Extensibility
    - Handle one or more search platforms

---

COMP6205 - Enterprise Search
Solr Architecture

Source: *Solr In Action*
Solr Architecture
Solr Features

- **Keyword Searching** – queries of terms and Boolean operators Ranked
- **Retrieval** – sorted by relevancy score (descending order) Snippet
- **Highlighting** – matching terms emphasized in results
- **Faceting** – ability to apply filter queries based on matching fields Paging
- **Navigation** – limits fetch sizes to improve performance
- **Result Sorting** – sort the documents based on field values
Solr Features

• **Spelling Correction** – suggest corrected spelling of query terms

• **Synonyms** – expand queries based on configurable definition list

• **Auto-Suggestions** – present list of possible query terms

• **More Like This** – identifies other documents that are similar to one in a result set

• **Geo-Spatial Search** – locate and sort documents by distance

• **Scalability** – ability to break a large index into multiple shards and distribute indexing and query operations across a cluster of nodes
Solr Feature Example
Solr Installation

- Tutorial
  

- Installing Solr

- Solr is available from the Solr website at: http://lucene.apache.org/solr/

- Running Solr

- Start the Server

- If you didn't start Solr after installing it, you can start it by running from the Solr directory bin/solr.

  $ bin/solr start

- If you are running Windows, you can start Solr by running bin\solr.cmd.

  bin\solr.cmd start

- This will start Solr in the background, listening on port 8983.
References

Information Architecture *For the Web and Beyond*
4th Edition – Chapter 9 *Search Systems*

Enterprise Search – Enhancing Business Performance, 2nd Edition
Solr Resources

- [https://cwiki.apache.org/confluence/display/solr/](https://cwiki.apache.org/confluence/display/solr/)
- *Apache Solr 4 Enterprise Search Server, 3rd Edition*
  by [David Smiley](http://example.com), [Eric Pugh](http://example.com), [Kranti Parisa](http://example.com), [Matt Mitchel](http://example.com)

- **Solr In Action**
  [Trey Grainger](http://example.com) and [Timothy Potter](http://example.com)
  Manning Publications