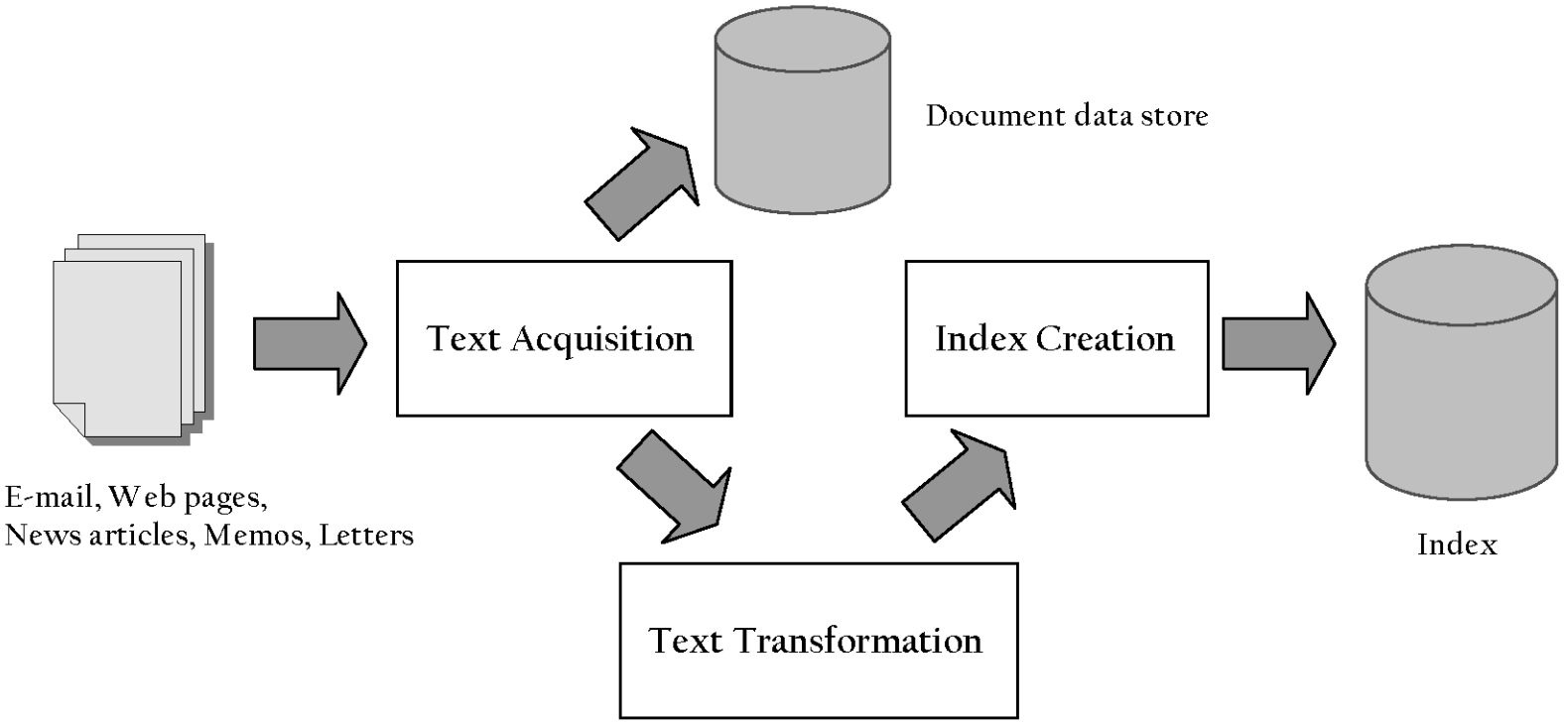
# Search Engines

## Information Retrieval in Practice

# Search Engine Architecture

- A software architecture consists of software components, the interfaces provided by those components, and the relationships between them
  - describes a system at a particular level of abstraction
- Architecture of a search engine determined by 2 requirements
  - effectiveness (quality of results) and efficiency (response time)

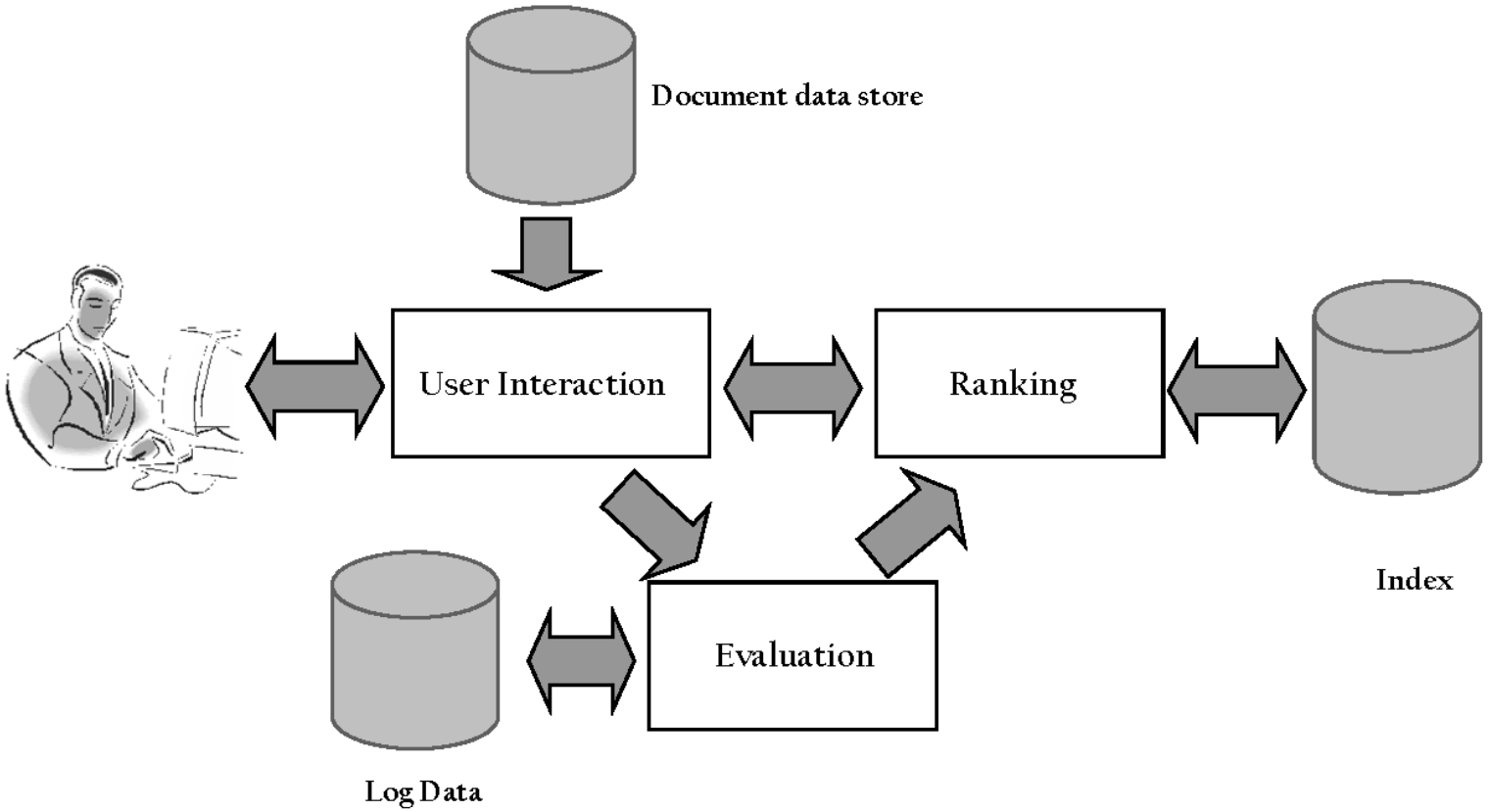
# Indexing Process



# Indexing Process

- Text acquisition
  - identifies and stores documents for indexing
- Text transformation
  - transforms documents into *index terms* or *features*
- Index creation
  - takes index terms and creates data structures (*indexes*) to support fast searching

# Query Process



# Query Process

- User interaction
  - supports creation and refinement of query, display of results
- Ranking
  - uses query and indexes to generate ranked list of documents
- Evaluation
  - monitors and measures effectiveness and efficiency (primarily offline)

# Details: Text Acquisition

- Crawler
  - Identifies and acquires documents for search engine
  - Many types – web, enterprise, desktop
  - Web crawlers follow *links* to find documents
    - Must efficiently find huge numbers of web pages (*coverage*) and keep them up-to-date (*freshness*)
    - Single site crawlers for *site search*
    - *Topical* or *focused* crawlers for vertical search
  - *Document* crawlers for enterprise and desktop search
    - Follow links and scan directories

# Text Acquisition

- Feeds
  - Real-time streams of documents
    - e.g., web feeds for news, blogs, video, radio, tv
  - RSS is common standard
    - RSS "reader" can provide new XML documents to search engine (See: https://edition.cnn.com/services/rss/)
- Conversion
  - Convert variety of documents into a consistent text plus metadata format
    - e.g. HTML, XML, Word, PDF, etc. → XML
  - Convert text encoding for different languages
    - Using a Unicode standard like UTF-8

# Text Acquisition

- Document data store
  - Stores text, metadata, and other related content for documents
    - Metadata is information about document such as type and creation date
    - Other content includes links, anchor text
  - Provides fast access to document contents for search engine components
    - e.g. result list generation
  - Could use relational database system
    - More typically, a simpler, more efficient storage system is used due to huge numbers of documents

# Text Transformation

- Parser
  - Processing the sequence of text *tokens* in the document to recognize structural elements
    - e.g., titles, links, headings, etc.
  - *Tokenizer* recognizes “words” in the text
    - must consider issues like capitalization, hyphens, apostrophes, non-alpha characters, separators
  - *Markup languages* such as HTML, XML often used to specify structure
    - *Tags* used to specify document *elements*
      - E.g., <h2> Overview </h2>
    - Document parser uses *syntax* of markup language (or other formatting) to identify structure

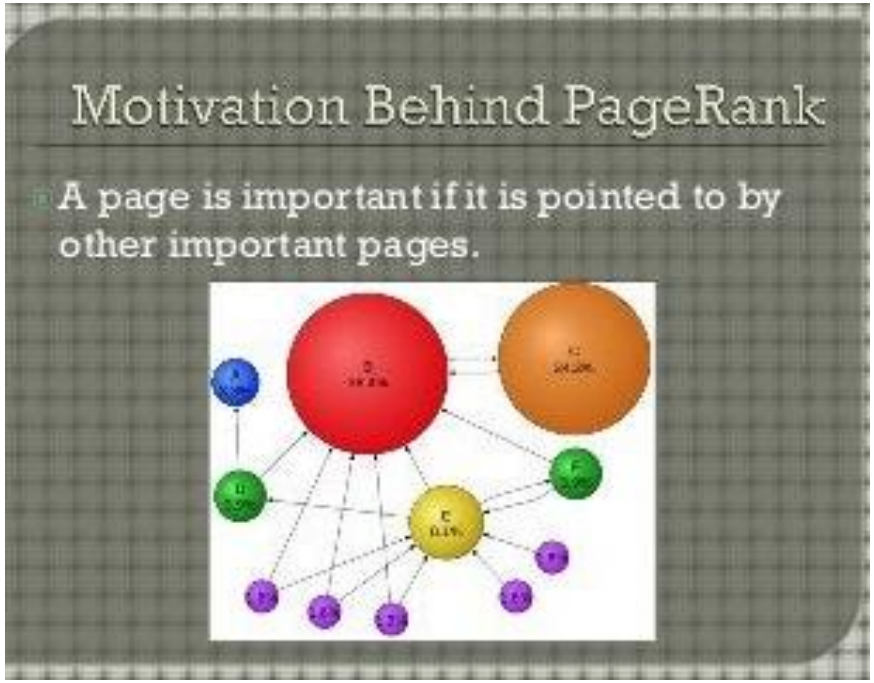
```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <companies>
  - <company>
      <companyname>Stanford and
        Son</companyname>
    - <employee>
        <code>1</code>
        <name>Joe Jackson</name>
        <street>14th street</street>
        <houseno>1</houseno>
        <areacode>1050 DD</areacode>
        <place>NoWhere</place>
        <phone>0100 987654</phone>
      </employee>
    - <employee>
        <code>2</code>
        <name>Peter de Wit</name>
        <street>ChurchLane</street>
        <houseno>4a</houseno>
        <areacode>9876 AB</areacode>
        <place>Whereever</place>
        <phone>0100 987654</phone>
      </employee>
    - <employee>
        <code>3</code>
        <name>John Brown</name>
        <street>1st street</street>
        <houseno>243</houseno>
        <areacode>5558 ZZ</areacode>
        <place>OutSide</place>
        <phone>0333 999888</phone>
      </employee>
  </company>
</companies>
```

# Text Transformation

- Stopping
  - Remove common words
    - e.g., “and”, “or”, “the”, “in”
  - Some impact on efficiency and effectiveness
  - Can be a problem for some queries
- Stemming
  - Group words derived from a common *stem*
    - e.g., “computer”, “computers”, “computing”, “compute”
  - Usually effective, but not for all queries
  - Benefits vary for different languages

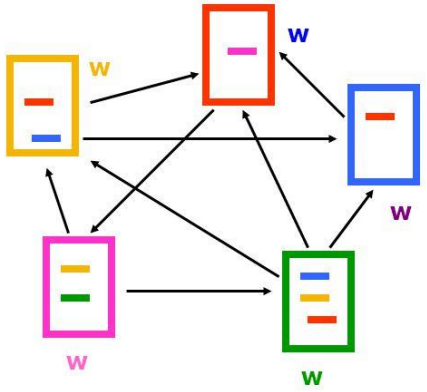
# Text Transformation

- Link Analysis
  - Makes use of *links* and *anchor text* in web pages
  - Link analysis identifies *popularity* and *community* information
    - e.g., PageRank
  - Anchor text can significantly enhance the representation of pages pointed to by links
  - Significant impact on web search
    - Less importance in other applications




## Link Analysis Ranking Algorithms

- Start with a collection of web pages
- Extract the underlying hyperlink graph
- Run the LAR algorithm on the graph
- Output: an authority weight for each node



# Text Transformation

- Information Extraction
  - Identify classes of index terms that are important
  - e.g., *named entity recognizers* identify classes such as *people, locations, companies, dates,* etc.
- Classifier
  - Identifies class-related metadata for documents
    - i.e., assigns labels to documents
    - e.g., topics, reading levels, sentiment, genre

# Index Creation

- Document Statistics
  - Gathers counts and positions of words and other features
  - Used in ranking algorithm
- Weighting
  - Computes weights for index terms
  - Used in ranking algorithm
  - e.g., *tf.idf* weight
    - Combination of *term frequency* in document and *inverse document frequency* in the collection
    - *to reflect how important a word is to a document in a collection or corpus*

# Index Creation

- Inversion
  - Core of indexing process
  - Converts document-term information to term-document for indexing
    - Difficult for very large numbers of documents
  - Format of inverted file is designed for fast query processing
    - Must also handle updates
    - Compression used for efficiency

# Index Creation

- Index Distribution
  - Distributes indexes across multiple computers and/or multiple sites
  - Essential for fast query processing with large numbers of documents
  - Many variations
    - Document distribution, term distribution, replication

# User Interaction

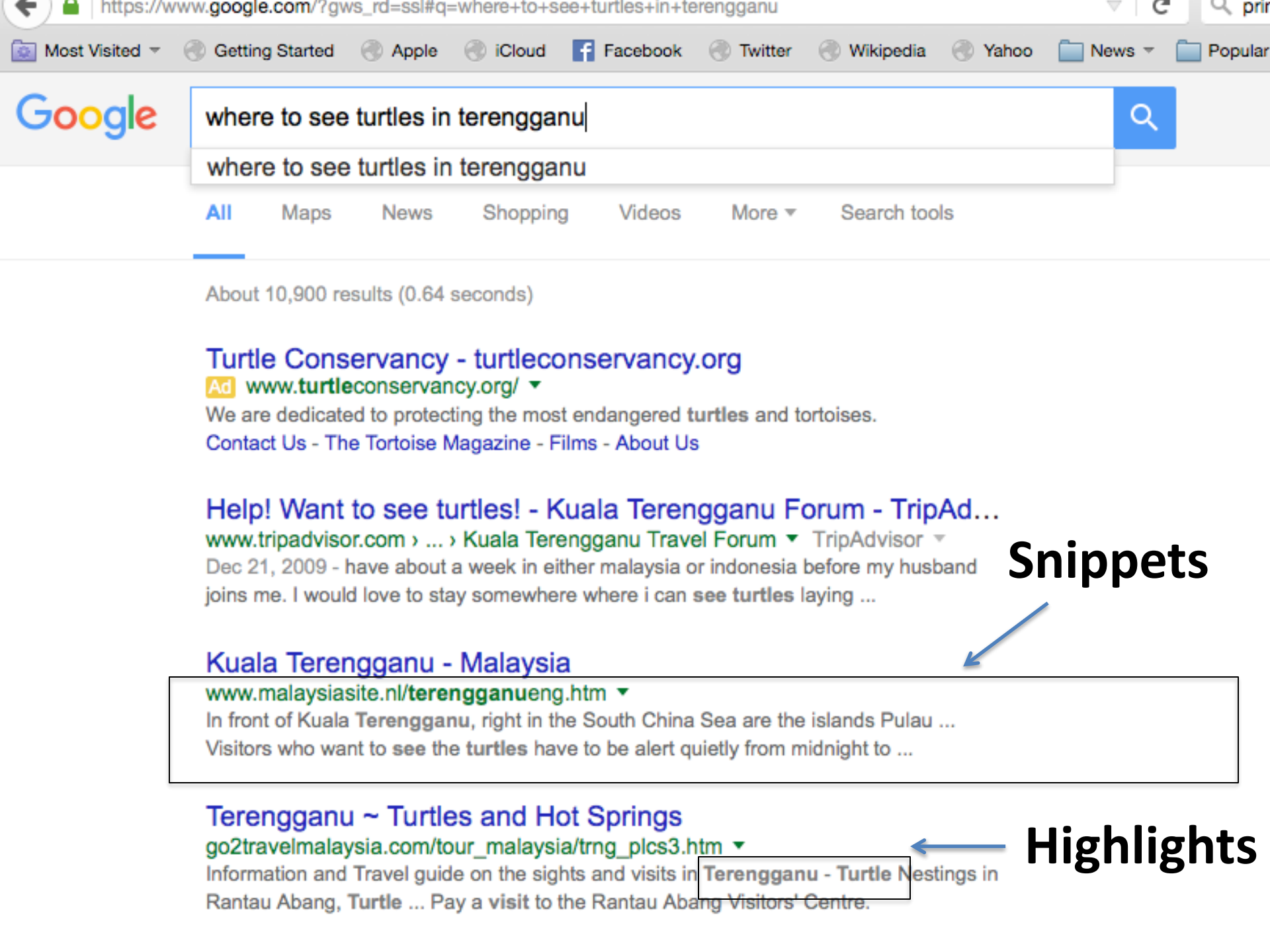
- Query input
  - Provides interface and parser for *query language*
  - Most web queries are very simple, other applications may use forms
  - Query language used to describe more complex queries and results of query transformation
    - e.g., Boolean queries, Indri and Galago query languages
    - similar to SQL language used in database applications
    - IR query languages also allow content and structure specifications, but focus on content

# User Interaction

- Query transformation
  - Improves initial query, both before and after initial search
  - Includes text transformation techniques used for documents
  - *Spell checking* and *query suggestion* provide alternatives to original query
  - *Query expansion* and *relevance feedback* modify the original query with additional terms

# User Interaction

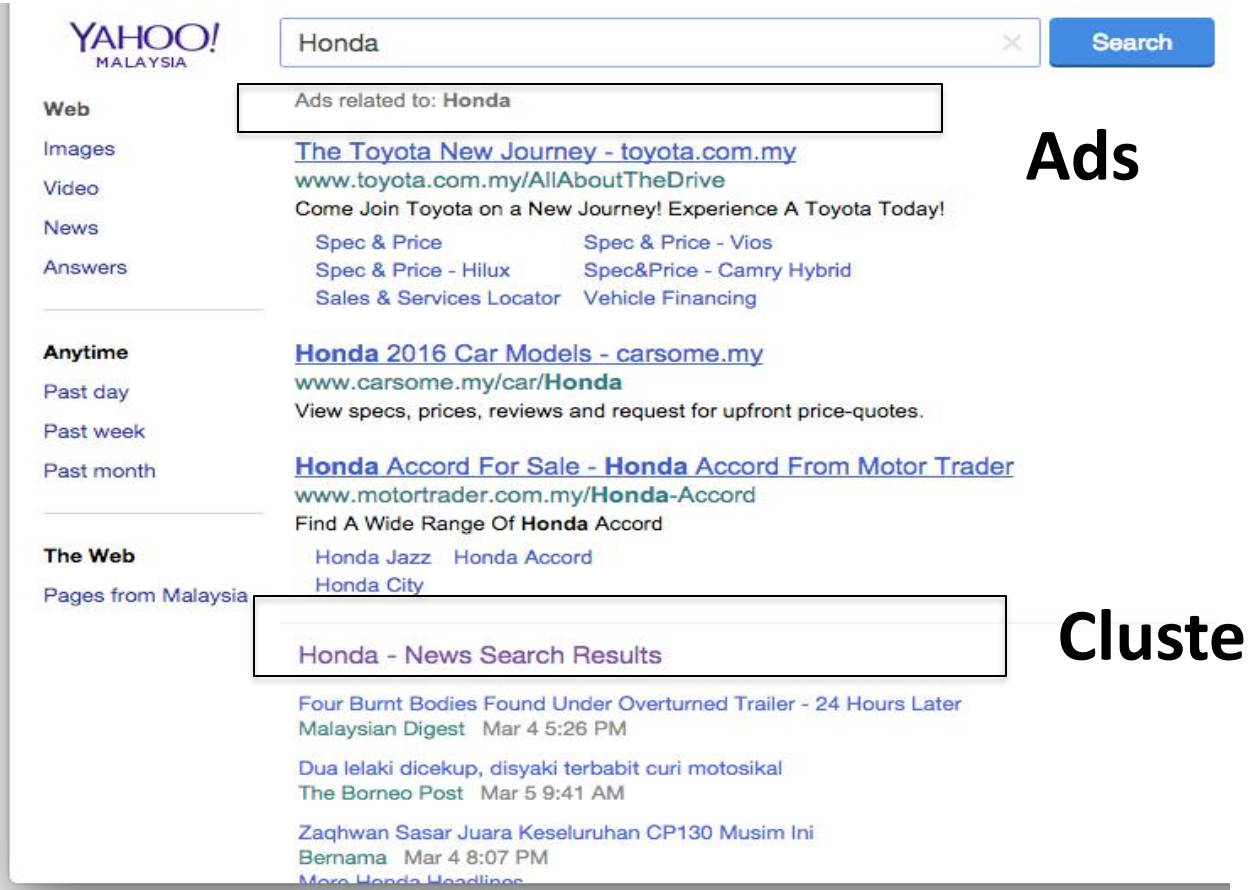
- Results output
  - Constructs the display of ranked documents for a query
  - Generates *snippets* to show how queries match documents


https://www.google.com/?gws_rd=ssl#q=where+to+see+turtles+in+terengganu
Most Visited
Getting Started
Apple
iCloud
Facebook
Twitter
Wikipedia
Yahoo
News
Popular
Google
where to see turtles in terengganu
where to see turtles in terengganu
All
Maps
News
Shopping
Videos
More
Search tools
About 10,900 results (0.64 seconds)
Turtle Conservancy - turtleconservancy.org
Ad www.turtleconservancy.org/
We are dedicated to protecting the most endangered turtles and tortoises.
Contact Us - The Tortoise Magazine - Films - About Us
Help! Want to see turtles! - Kuala Terengganu Forum - TripAd…
www.tripadvisor.com › ... › Kuala Terengganu Travel Forum TripAdvisor
Dec 21, 2009 - have about a week in either malaysia or indonesia before my husband joins me. I would love to stay somewhere where i can see turtles laying ...
Snippets
Kuala Terengganu - Malaysia
www.malaysiasite.nl/terengganueng.htm
In front of Kuala Terengganu, right in the South China Sea are the islands Pulau ...
Visitors who want to see the turtles have to be alert quietly from midnight to ...
Terengganu ~ Turtles and Hot Springs
go2travelmalaysia.com/tour_malaysia/trng_plcs3.htm
Information and Travel guide on the sights and visits in Terengganu - Turtle Nestings in Rantau Abang, Turtle ... Pay a visit to the Rantau Abang Visitors' Centre.
Highlights

– *Highlights* important words and passages

– Retrieves appropriate *advertising* in many applications

– May provide *clustering* and other visualization tools



# Ranking

- Scoring
  - Calculates scores for documents using a ranking algorithm
  - Core component of search engine
  - Basic form of score is $\sum q_i d_i$
    - $q_i$ and $d_i$ are query and document term weights for term i
  - Many variations of ranking algorithms and retrieval models

# Ranking

- Performance optimization
  - Designing ranking algorithms for efficient processing
- Distribution
  - Processing queries in a distributed environment
  - *Query broker* distributes queries and assembles results

# Evaluation

- Logging
  - Logging user queries and interaction is crucial for improving search effectiveness and efficiency
  - *Query logs* and *clickthrough data* used for query suggestion, spell checking, query caching, ranking, advertising search, and other components
- Ranking analysis
  - Measuring and tuning ranking effectiveness
- Performance analysis
  - Measuring and tuning system efficiency

- END