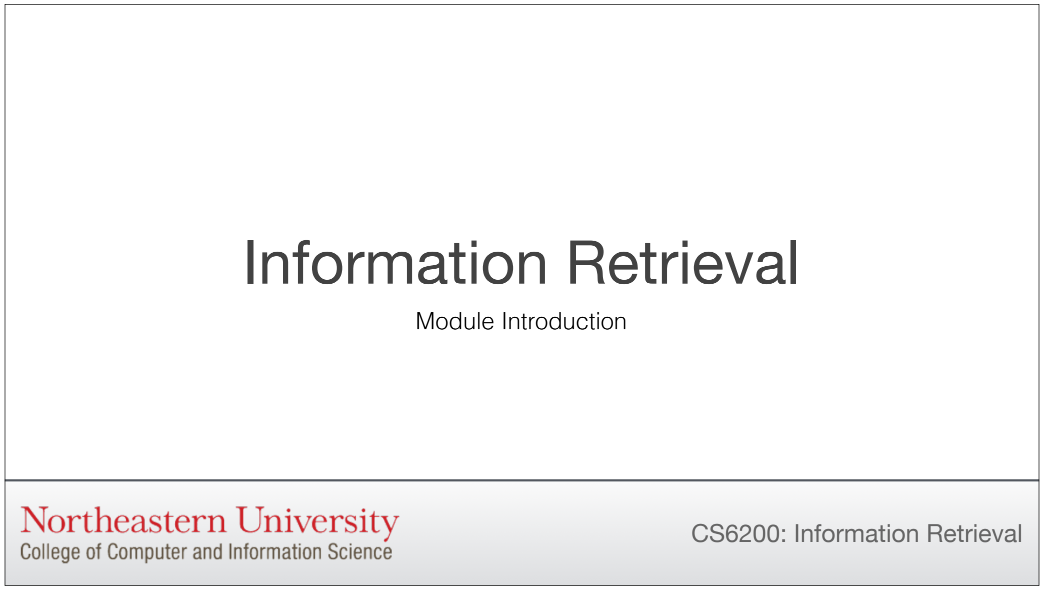


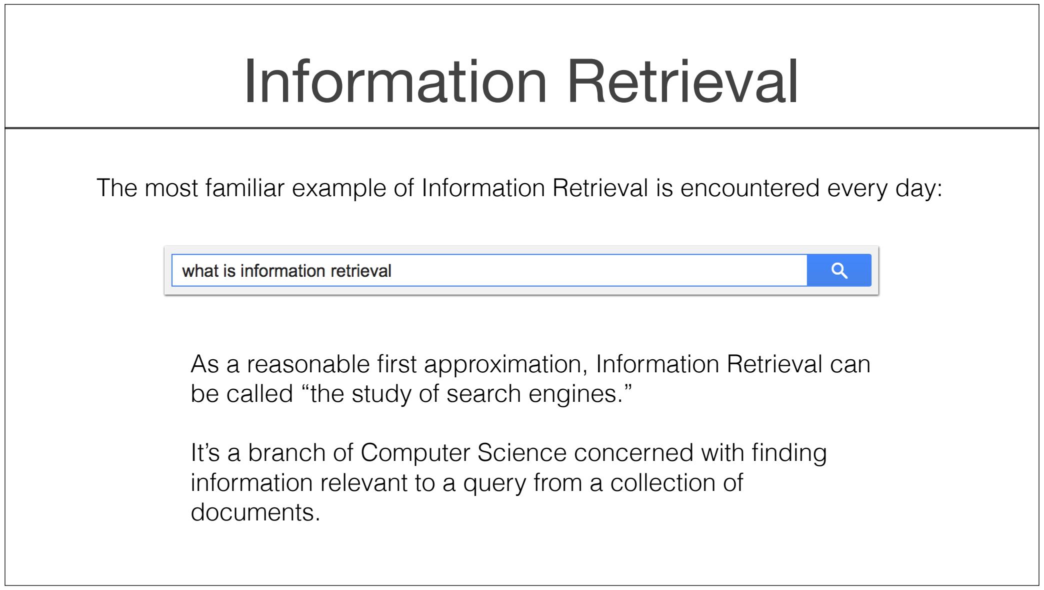
Information Retrieval

Module Introduction

Northeastern University
College of Computer and Information Science

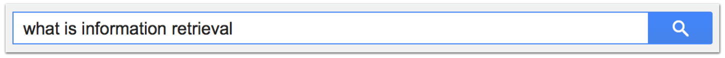
CS6200: Information Retrieval

Welcome to Information Retrieval. In this class, you'll learn about many of the exciting technologies that define life on the web as we know it. The Internet is changing the world by making vast amounts of information and new products available to everyone, and IR plays a key role in helping people navigate all that content. From Google search to Facebook feeds to Amazon product recommendations, IR helps make the changes brought by the Internet possible.



Information Retrieval

The most familiar example of Information Retrieval is encountered every day:



As a reasonable first approximation, Information Retrieval can be called "the study of search engines."

It's a branch of Computer Science concerned with finding information relevant to a query from a collection of documents.

But what is Information Retrieval, exactly?

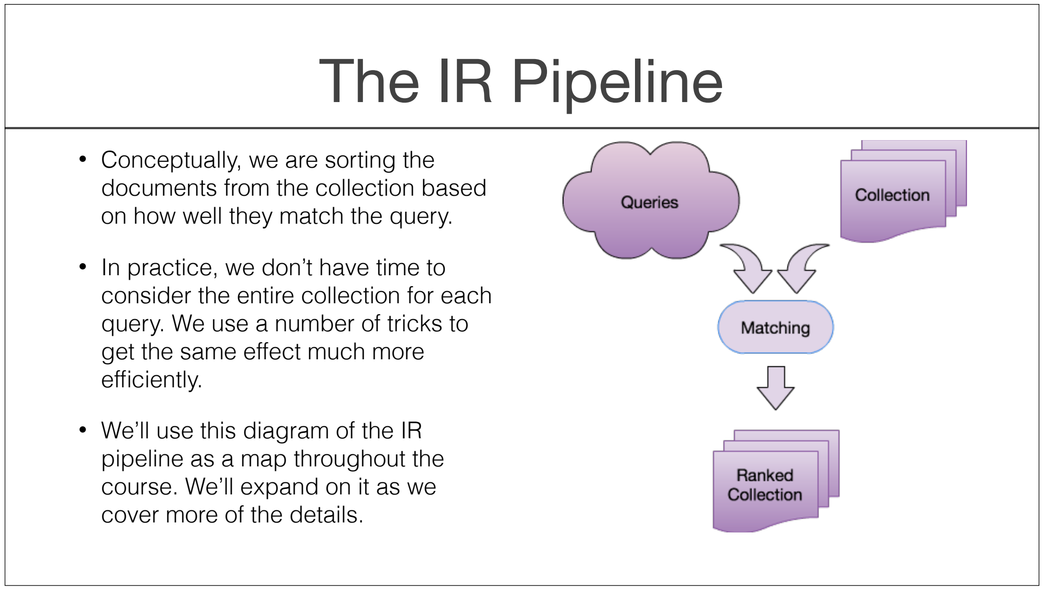
[show 1]

The most familiar and obvious example of it is web search. There are many other applications of IR, and we'll get to them in time, but let's start here.

[read 2]

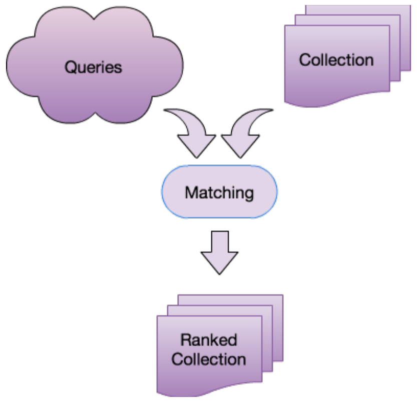
[read 3]

Let's set the stage to talk about how we do that.



The IR Pipeline

- Conceptually, we are sorting the documents from the collection based on how well they match the query.
- In practice, we don't have time to consider the entire collection for each query. We use a number of tricks to get the same effect much more efficiently.
- We'll use this diagram of the IR pipeline as a map throughout the course. We'll expand on it as we cover more of the details.



[read 1]

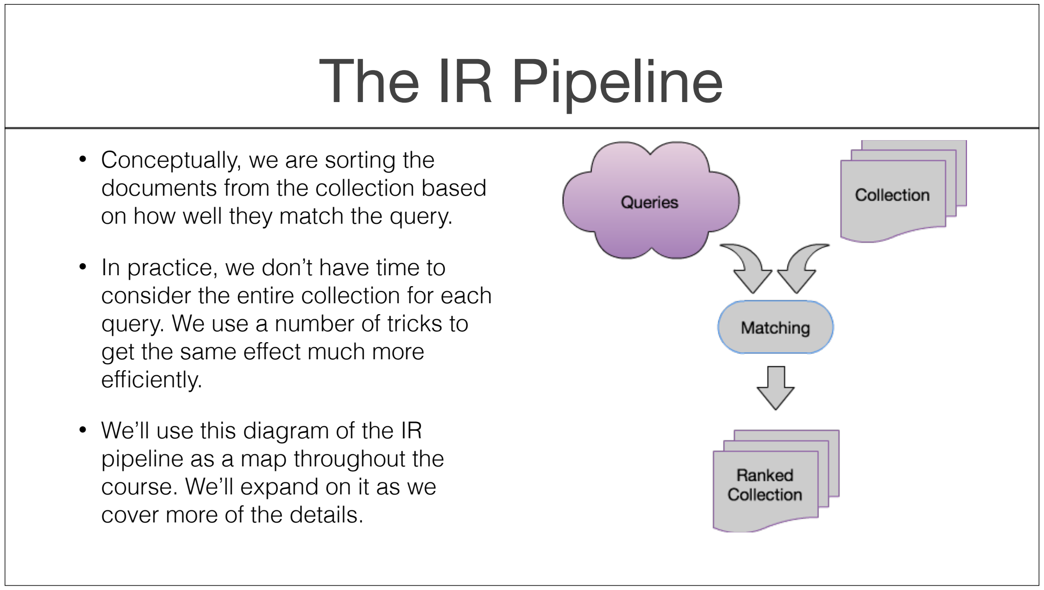
For web search the documents are web pages, and the collection is the entire Internet.

Obviously, [read 2]

We'll talk about many of those tricks throughout the course.

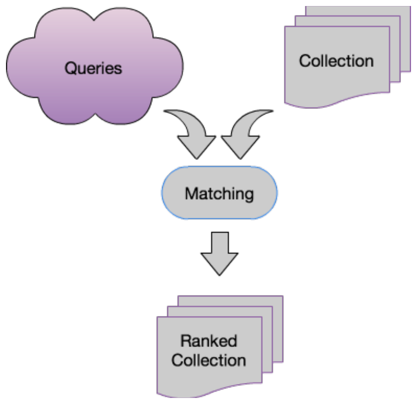
[read 3, show diagram]

Users give us queries they want us to answer. We have a collection of documents, and try to answer queries using the information in that collection. In order to find the answer, we perform some matching process which allows us to sort the collection from most relevant to the query to least relevant. Our final output is a ranking, or ordering, of the collection to present to the user. If matching went well, the user's answer will be at, or near, the top of the sorted list.



The IR Pipeline

- Conceptually, we are sorting the documents from the collection based on how well they match the query.
- In practice, we don't have time to consider the entire collection for each query. We use a number of tricks to get the same effect much more efficiently.
- We'll use this diagram of the IR pipeline as a map throughout the course. We'll expand on it as we cover more of the details.



[read 1]

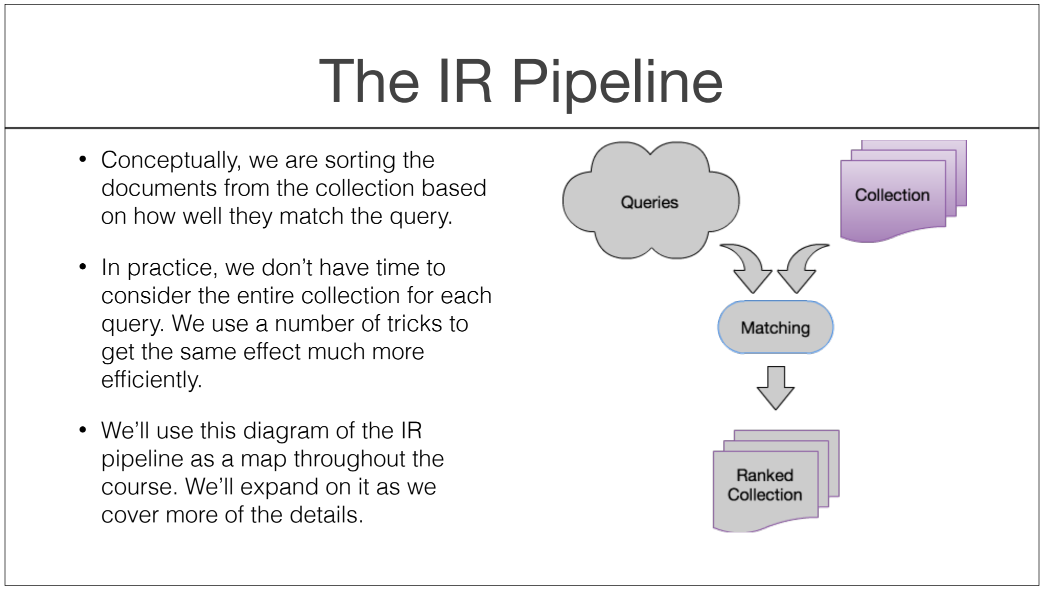
For web search the documents are web pages, and the collection is the entire Internet.

Obviously, [read 2]

We'll talk about many of those tricks throughout the course.

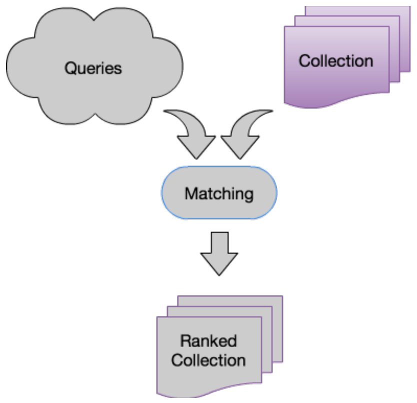
[read 3, show diagram]

Users give us queries they want us to answer. We have a collection of documents, and try to answer queries using the information in that collection. In order to find the answer, we perform some matching process which allows us to sort the collection from most relevant to the query to least relevant. Our final output is a ranking, or ordering, of the collection to present to the user. If matching went well, the user's answer will be at, or near, the top of the sorted list.



The IR Pipeline

- Conceptually, we are sorting the documents from the collection based on how well they match the query.
- In practice, we don't have time to consider the entire collection for each query. We use a number of tricks to get the same effect much more efficiently.
- We'll use this diagram of the IR pipeline as a map throughout the course. We'll expand on it as we cover more of the details.



[read 1]

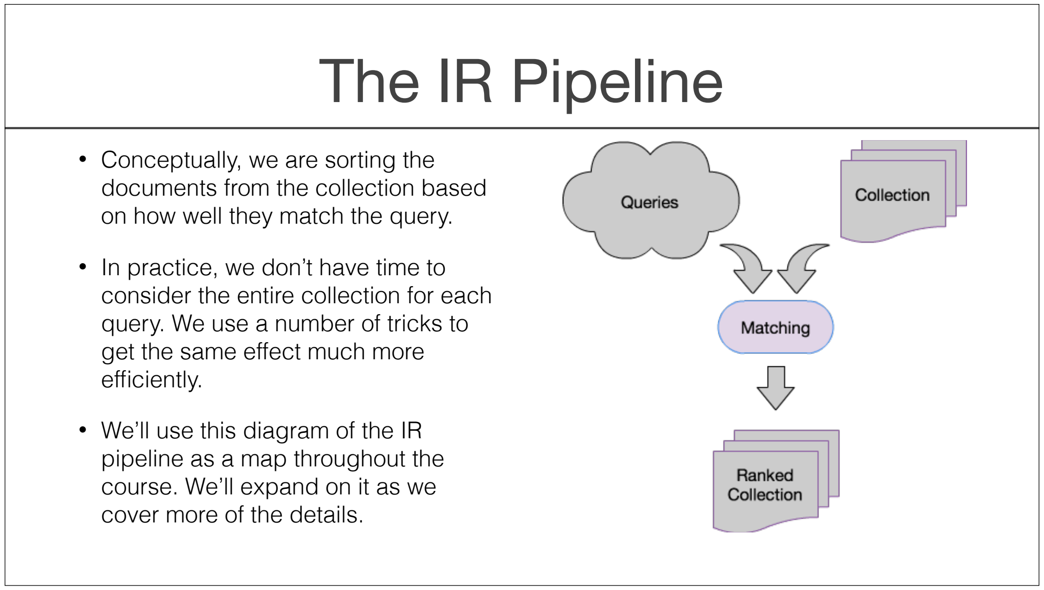
For web search the documents are web pages, and the collection is the entire Internet.

Obviously, [read 2]

We'll talk about many of those tricks throughout the course.

[read 3, show diagram]

Users give us queries they want us to answer. We have a collection of documents, and try to answer queries using the information in that collection. In order to find the answer, we perform some matching process which allows us to sort the collection from most relevant to the query to least relevant. Our final output is a ranking, or ordering, of the collection to present to the user. If matching went well, the user's answer will be at, or near, the top of the sorted list.



The IR Pipeline

- Conceptually, we are sorting the documents from the collection based on how well they match the query.
- In practice, we don't have time to consider the entire collection for each query. We use a number of tricks to get the same effect much more efficiently.
- We'll use this diagram of the IR pipeline as a map throughout the course. We'll expand on it as we cover more of the details.

[read 1]

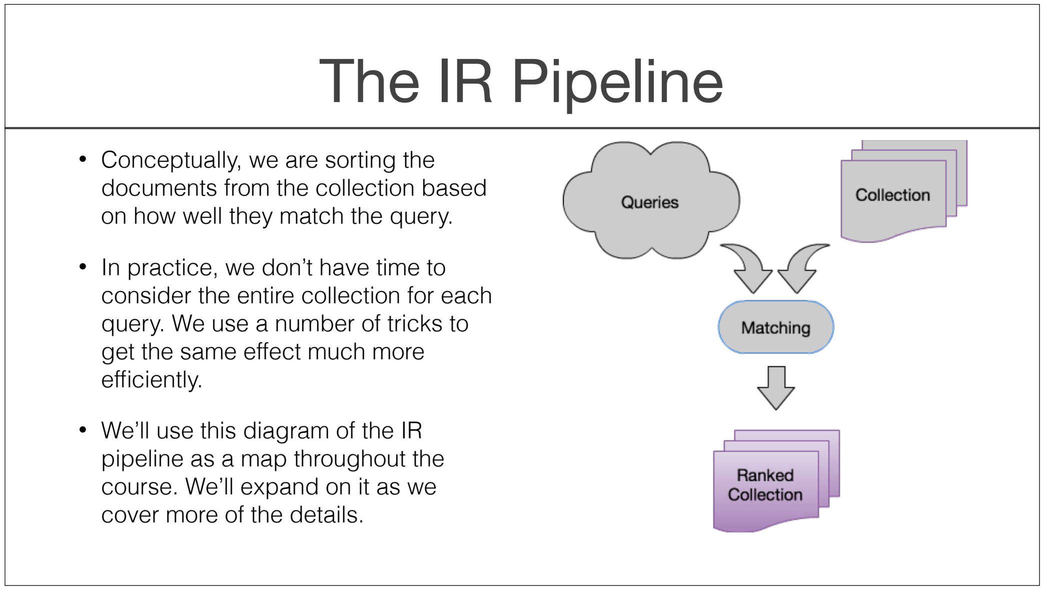
For web search the documents are web pages, and the collection is the entire Internet.

Obviously, [read 2]

We'll talk about many of those tricks throughout the course.

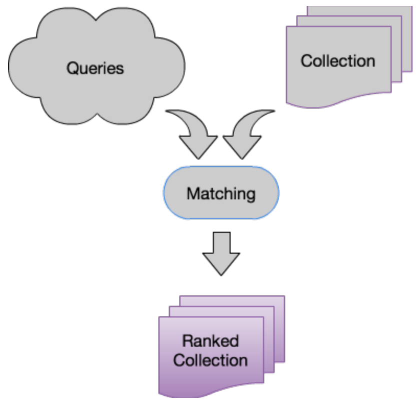
[read 3, show diagram]

Users give us queries they want us to answer. We have a collection of documents, and try to answer queries using the information in that collection. In order to find the answer, we perform some matching process which allows us to sort the collection from most relevant to the query to least relevant. Our final output is a ranking, or ordering, of the collection to present to the user. If matching went well, the user's answer will be at, or near, the top of the sorted list.



The IR Pipeline

- Conceptually, we are sorting the documents from the collection based on how well they match the query.
- In practice, we don't have time to consider the entire collection for each query. We use a number of tricks to get the same effect much more efficiently.
- We'll use this diagram of the IR pipeline as a map throughout the course. We'll expand on it as we cover more of the details.



[read 1]

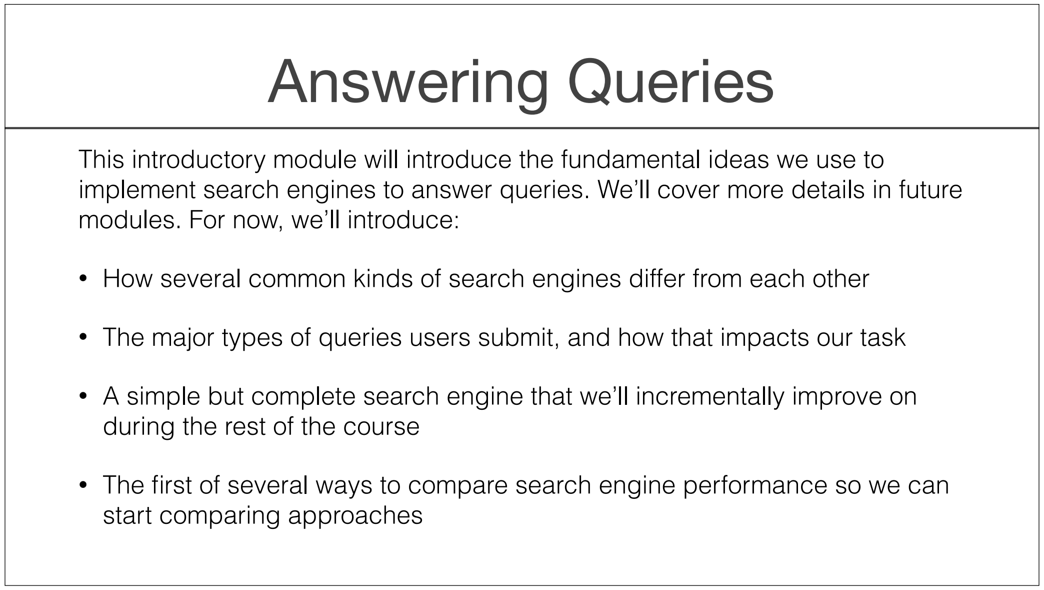
For web search the documents are web pages, and the collection is the entire Internet.

Obviously, [read 2]

We'll talk about many of those tricks throughout the course.

[read 3, show diagram]

Users give us queries they want us to answer. We have a collection of documents, and try to answer queries using the information in that collection. In order to find the answer, we perform some matching process which allows us to sort the collection from most relevant to the query to least relevant. Our final output is a ranking, or ordering, of the collection to present to the user. If matching went well, the user's answer will be at, or near, the top of the sorted list.



Answering Queries

This introductory module will introduce the fundamental ideas we use to implement search engines to answer queries. We'll cover more details in future modules. For now, we'll introduce:

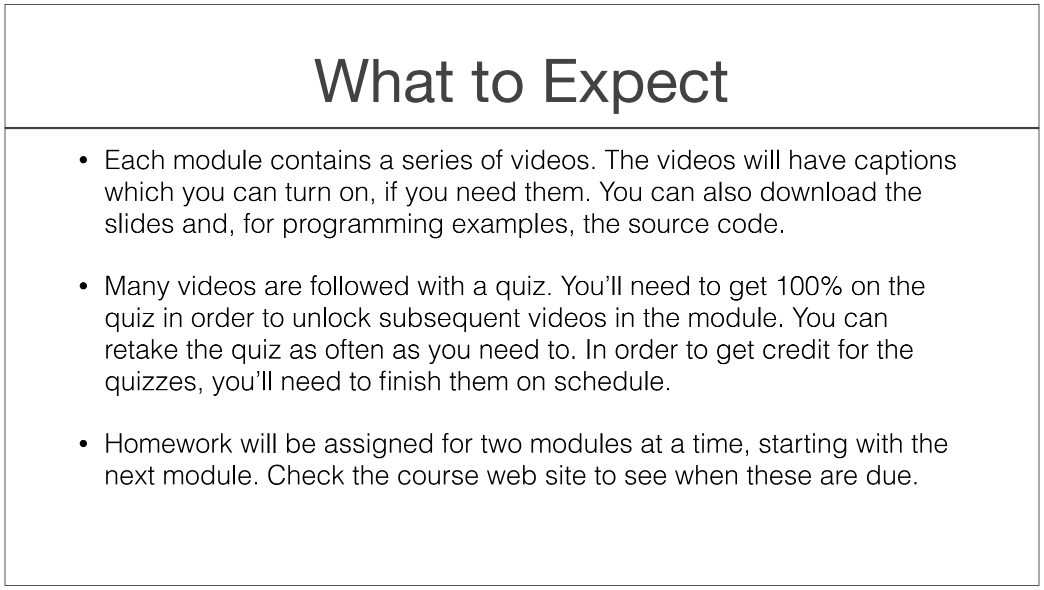
- How several common kinds of search engines differ from each other
- The major types of queries users submit, and how that impacts our task
- A simple but complete search engine that we'll incrementally improve on during the rest of the course
- The first of several ways to compare search engine performance so we can start comparing approaches

[read 1]

[read 2]

[read 3]

[read 4]



What to Expect

- Each module contains a series of videos. The videos will have captions which you can turn on, if you need them. You can also download the slides and, for programming examples, the source code.
- Many videos are followed with a quiz. You'll need to get 100% on the quiz in order to unlock subsequent videos in the module. You can retake the quiz as often as you need to. In order to get credit for the quizzes, you'll need to finish them on schedule.
- Homework will be assigned for two modules at a time, starting with the next module. Check the course web site to see when these are due.

Let's talk about what to expect from this online course.

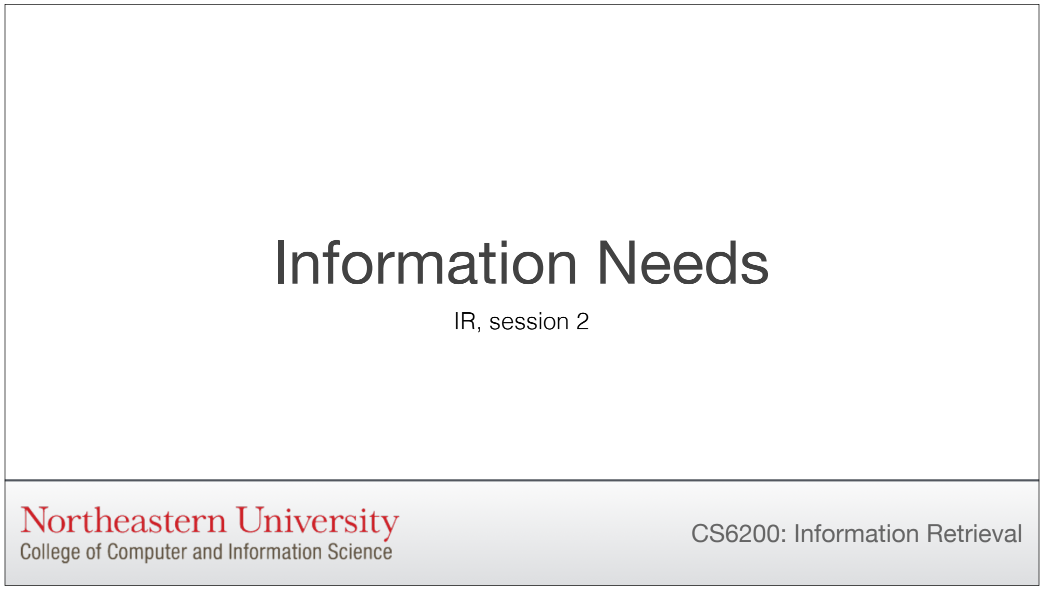
[read 1]

[read 2]

This is meant to encourage you to think about the material instead of passively watching a long lecture. It also helps give everyone a good reason to prepare for the week's classroom lecture.

[read 3]

As always, ask your professor if there's any confusion about the course policies.



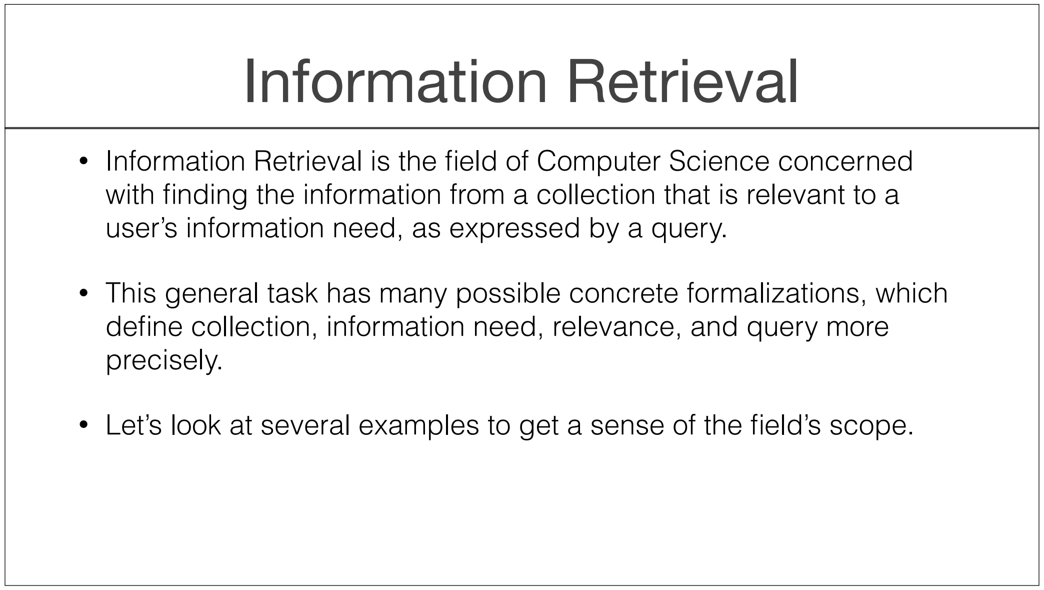
Information Needs

IR, session 2

Northeastern University
College of Computer and Information Science

CS6200: Information Retrieval

In this session, I’m going to define the term “Information Retrieval,” and then talk about several examples of IR systems in the wild. By the end of the video, you should have some idea of the scope of commercial IR systems.



Information Retrieval

- Information Retrieval is the field of Computer Science concerned with finding the information from a collection that is relevant to a user's information need, as expressed by a query.
- This general task has many possible concrete formalizations, which define collection, information need, relevance, and query more precisely.
- Let's look at several examples to get a sense of the field's scope.

[read 1]

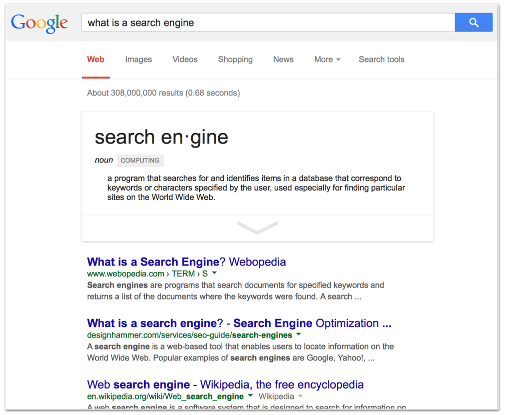
The critical flaw of this definition is that it's using several terms from the field to define the field. What is an information need, and how is that different from a query? What do we mean by "information," and what makes it "relevant?" It turns out that there are a lot of ways to define these terms.

[read 2]

[read 3]

Ad-Hoc Search

- Currently, the most important search task is ad hoc search on the Internet.
- The collection is the set of web pages indexed by the search engine.
- The information need is the web content the user is looking for.
- The query is an ordered list of keywords.
- A document is relevant if it contains text on the same topic as the query.



About 308,000,000 results (0.68 seconds)

search en·gine

noun COMPUTING

a program that searches for and identifies items in a database that correspond to keywords or characters specified by the user, used especially for finding particular sites on the World Wide Web.

What is a Search Engine? Webopedia
www.webopedia.com › TERM › S
Search engines are programs that search documents for specified keywords and returns a list of the documents where the keywords were found. A search ...

What is a search engine? - **Search Engine** Optimization ...
designhammer.com/services/seo-guide/**search-engines**
A **search engine** is a web-based tool that enables users to locate information on the World Wide Web. Popular examples of **search engines** are Google, Yahoo!, ...

Web **search engine** - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/Web_**search_engine** Wikipedia
A web **search engine** is a software system that is designed to search for information on

[read 1]

This is the main feature of web search engines, like Google or Baidu or Bing. These sites have evolved into very complex systems, but the core feature behind them all is ad-hoc search. In ad-hoc search, the user types in a bunch of keywords and gets back a bunch of documents. But let's be a little more precise.

[read 2]

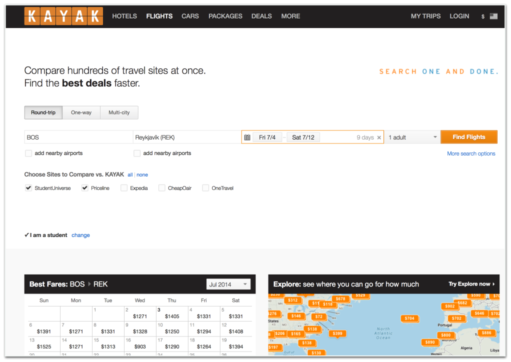
Conceptually, the collection for these systems is the entire Internet. In practice, though, the collection is that portion of the web which has been downloaded and processed by the search engine — what's called crawling and indexing. We'll talk about those steps later.

[read 3]

Each user comes to a search engine with an information need. In ad-hoc search, that information might be anything a person could need to know. That's the "ad-hoc" part — the system should be able to handle any information need.

Vertical Search

- Vertical Search focuses on information from a particular domain: flights, music, news, sports, etc.
- The collection might be the set of all airline fares, research papers, or blog posts.
- An information need can be very specific: “the cost of a flight to Iceland tomorrow”
- The query may be structured using a web form, providing specific property values to search for.
- Document relevance is sometimes less ambiguous: matching the search fields.



The next type of search engine is called Vertical Search.

[read 1]

Since they’re focused on one kind of information, they can generally present more query options and better-formatted results than an ad-hoc system could.

[read 2]

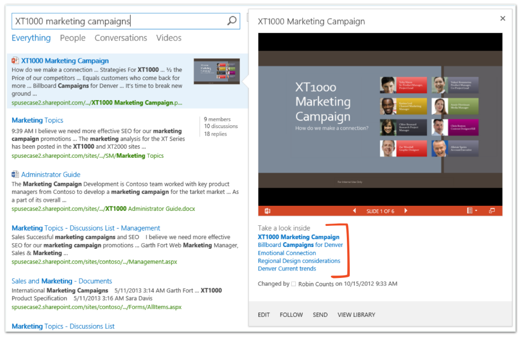
For these systems, the collection is not necessarily a set of web pages. It might be the set of airline tickets you could buy, or the set of news articles about sports teams, or the set of research papers published by the ACM. You use different methods to obtain this information: you might have a deal with airlines to tell you their ticket costs, or use a web crawler designed to focus on sports news and ignore all other web sites.

[read 3]

If you’re selling airline tickets, the user might need to know the cost of all possible flights to their destination within a certain time window. In order to handle these more complex information needs, a lot of vertical search sites present a

Enterprise Search

- Enterprise Search is vertical search run against a company's internal content.
- The collection is the set of documents, e-mails, forum threads, wiki pages, etc. in the company's internal network.
- Information needs, queries, and relevance are typically defined as for ad-hoc search.


XT1000 marketing campaigns

Everything People Conversations Videos

XT1000 Marketing Campaign
How do we make a connection ... Strategies For **XT1000** ... ½ the Price of our competitors ... Equals customers who come back for more ... Billboard **Campaigns** for Denver ... It's time to break new ground ...
spusecase2.sharepoint.com/.../**XT1000 Marketing Campaign**.p...

Marketing Topics
9:39 AM I believe we need more effective SEO for our **marketing campaign** promotions ... The **marketing** analysis for the XT Series has been posted in the **XT1000** and XT2000 sites ...
spusecase2.sharepoint.com/sites/.../SM/**Marketing** Topics
9 members
10 discussions
18 replies

Administrator Guide
The **Marketing Campaign** Development is Contoso team worked with key product managers from Contoso to develop a **marketing campaign** for the tarket market ... As a part of its overall ...
spusecase2.sharepoint.com/sites/.../**XT1000** Administrator Guide.docx

Marketing Topics - Discussions List - Management
Sales Successful **marketing campaigns** and SEO I believe we need more effective SEO for our **marketing campaign** promotions ... Garth Fort Web **Marketing** Manager, Sales & **Marketing** ...
spusecase2.sharepoint.com/sites/contoso/.../Management.aspx

Sales and **Marketing** - Documents
International **Marketing Campaigns** 5/11/2013 3:14 AM Garth Fort ... **XT1000** Product Specification 5/11/2013 3:16 AM Sara Davis
spusecase2.sharepoint.com/sites/contoso/.../Forms/AllItems.aspx

Marketing Topics - Discussions List

XT1000 Marketing Campaign

Take a look inside
XT1000 Marketing Campaign
Billboard **Campaigns** for Denver
Emotional Connection
Regional Design considerations
Denver Current trends

Changed by Robin Counts on 10/15/2012 9:33 AM

[read 1]

A large company will create a lot of material specific to their business. The HR department will create a bunch of documents addressing corporate policies and employees' needs, Sales will generate reports and marketing plans, and Engineering will create a lot of design and implementation information. All of these documents are valuable to various people in the company, and there's a thriving market for search engines for this content.

[read 2]

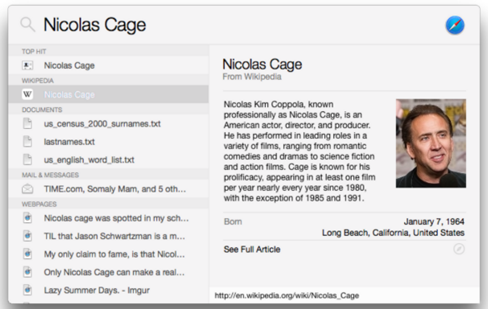
[read 3]

These search engines face a few big challenges. First, the total size of the collection is much smaller than in a typical web search engine, so accuracy can suffer from having less data to run your statistical algorithms on. In this case, having a bigger haystack can make it easier to find the needles.

Second, most search engines need experts to monitor them and tune performance over time, and many companies are really just looking to buy a product they can install once and not worry about.

Desktop Search

- Desktop Search focuses on searching the contents of your computer.
- The collection is the set of files (and contacts, messages, events, etc.) stored on your computer.
- An information need is generally either a file, or information stored in a file.
- A query can be a list of keywords as in ad-hoc search, or a list of property values in a custom query language.
- Relevance is defined as in ad-hoc search.


Nicolas Cage

TOP HIT
Nicolas Cage
WIKIPEDIA
Nicolas Cage
DOCUMENTS
us_census_2000_surnames.txt
lastnames.txt
us_english_word_list.txt
MAIL & MESSAGES
TIME.com, Somaly Mam, and 5 oth...
WEBPAGES
Nicolas cage was spotted in my sch...
TIL that Jason Schwartzman is a m...
My only claim to fame, is that Nicol...
Only Nicolas Cage can make a real...
Lazy Summer Days. - Imgur

Nicolas Cage
From Wikipedia

Nicolas Kim Coppola, known professionally as Nicolas Cage, is an American actor, director, and producer. He has performed in leading roles in a variety of films, ranging from romantic comedies and dramas to science fiction and action films. Cage is known for his prolificacy, appearing in at least one film per year nearly every year since 1980, with the exception of 1985 and 1991.

Born January 7, 1964
Long Beach, California, United States

See Full Article

http://en.wikipedia.org/wiki/Nicolas_Cage

[read 1]

Most of us are familiar with this class of search engines, but you might not have thought of it as an IR problem.

[read 2]

Most major operating systems have had various tools for performing full text search on the files in your computer for decades, but tools like Spotlight in Mac OS X have really pushed the technology forward. Modern implementations allow developers to write plugins to index and search their custom file types, and they can search more conceptual objects like contacts and calendar events that may not correspond to individual files. These tools are also being expanded to run web searches and mix web results with local results.

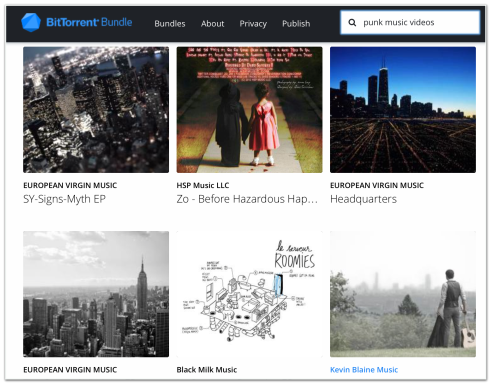
[read 3]

These information needs are often closer to the information needs from enterprise search than for ad hoc search. You're looking for a file, or a contact, or a program you want to run.

[read 4]

Peer to Peer Search

- Peer to Peer Search focuses on finding content shared on peer to peer networks.
- The collection is the set of all files currently shared by any peer on the network.
- An information need is a particular file, e.g. a music video.
- A query is often a keyword list, but may use an extended query language.
- A document is relevant only if it's the file the user wanted.



[read 1]

This is another interesting search task. These systems search for files hosted on peer to peer networks.

[read 2]

The files themselves are most commonly media files rather than text documents. A given file may be found on many different hosts, and the hosts themselves will connect and disconnect from the network unpredictably – especially back in the days of the dialup Internet. The particular challenge of this type of search is to index the content, identify which files are duplicated across which hosts, and allow the user to download the file efficiently as hosts appear and disappear on the network.

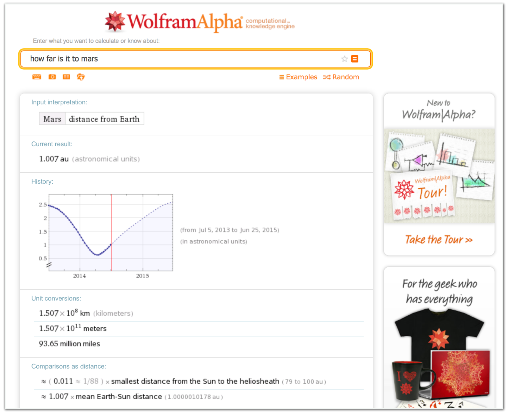
[read 3]

Searching for files is also seen in Enterprise or Desktop Search. However, Peer to Peer search engines may not support full-text searching of file contents, because the files are most often media files.

[show 4] Instead, you generally search against metadata, like the name of the file or its creator, and so a query is

Question Answering

- Question Answering tries to answer questions posed as normal dialog.
- Information needs are usually restricted to concisely-stated answers.
- Queries are posed as a single sentence of natural language text.
- A response is relevant if it answers the question correctly, and if it is expressed clearly (e.g. fluently).


WolframAlpha computational knowledge engine
Enter what you want to calculate or know about:
Examples
Random
New to Wolfram|Alpha?
Take the Tour »
For the geek who has everything

how far is it to mars

Input interpretation:

Mars | distance from Earth

Current result:

1.007 au (astronomical units)

History:

(from Jul 5, 2013 to Jun 25, 2015)
(in astronomical units)

Unit conversions:

1.507×10^8 km (kilometers)

1.507×10^{11} meters

93.65 million miles

Comparisons as distance:

≈ (0.011 ≈ 1/88) × smallest distance from the Sun to the heliosheath (79 to 100 au)

≈ 1.007 × mean Earth-Sun distance (1.0000010178 au)

Our last search engine example has strong ties to the Natural Language Processing research community.

[read 1]

It attempts to answer questions posed in natural language, or even math notation, and respond with facts rather than documents.

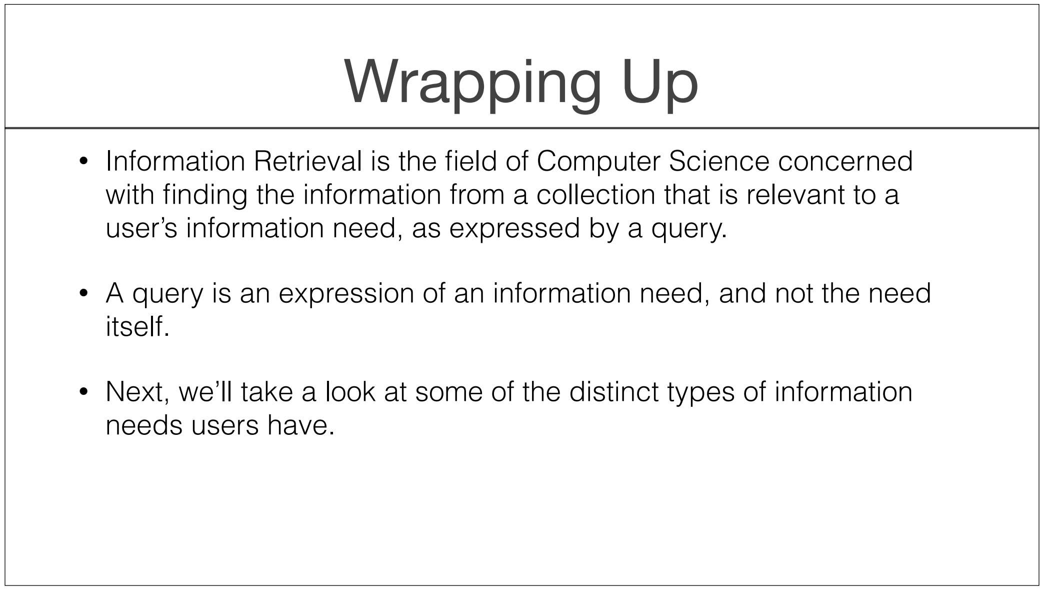
[read 2]

These answers are generally byte-sized facts which can be expressed with a single sentence, or maybe a graph.

[read 3]

[read 4]

Since the system generally composes the answer itself, the notion of relevance is somewhat conflated with the notion of fluency: an answer is more relevant if it is expressed more clearly.



Wrapping Up

- Information Retrieval is the field of Computer Science concerned with finding the information from a collection that is relevant to a user's information need, as expressed by a query.
- A query is an expression of an information need, and not the need itself.
- Next, we'll take a look at some of the distinct types of information needs users have.

Let's bring back the definition from the first slide.

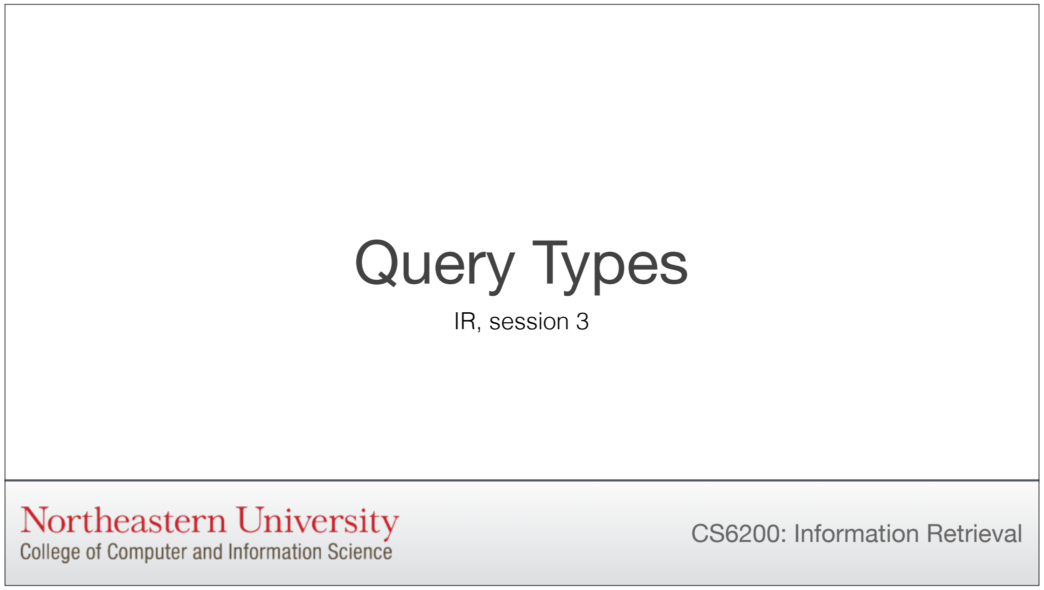
[read 1]

I hope you can see from these examples some of what IR systems have in common, and some of the differences. They all have some collection to search, but that collection might be web pages or music files or facts stored in a database. They all have different notions of information needs, and different ways users can express those needs as queries. That's worth emphasizing.

[read 2]

This distinction isn't just technical: it gets to the heart of a major challenge of IR. We want to find documents that satisfy the information need, and that need is often very poorly expressed by the query. Decoding the query is challenging and important.

When we're lucky, relevance itself can be a clearly-defined concept – in file search, an audio file is relevant if it's the song you're looking for – but usually it's quite fuzzy. We'll be exploring these aspects of IR systems in more detail



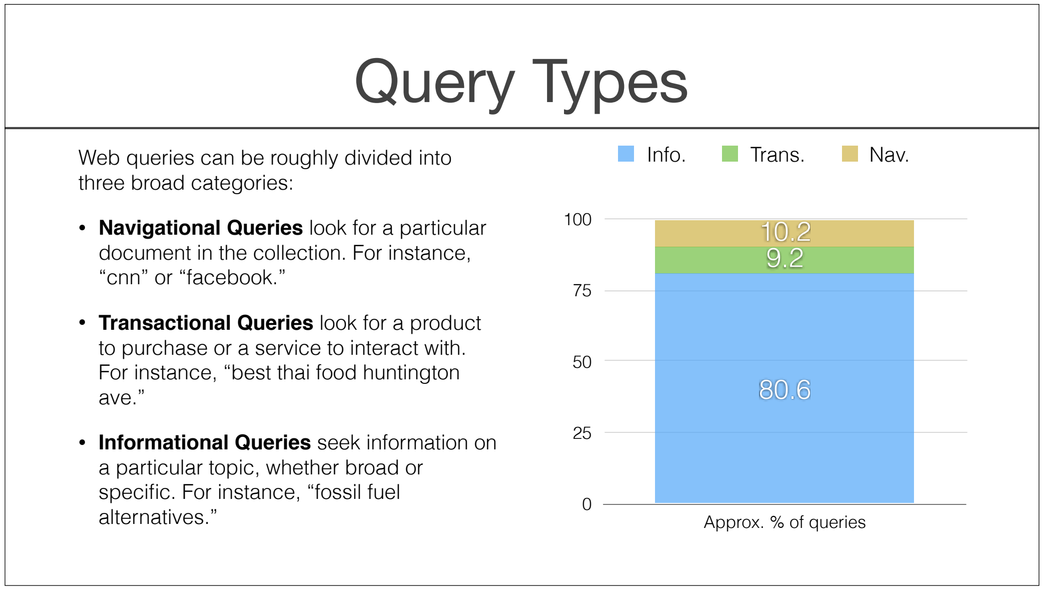
Query Types

IR, session 3

Northeastern University
College of Computer and Information Science

CS6200: Information Retrieval

This session introduces the most common types of queries people tend to run in ad-hoc search engines. It's important to explore the kinds of questions people ask because it can shed light on how to improve your IR system. There are quite a few ontologies of query types you can find if you go looking for them, but we'll cover one of the more commonly-used classification systems for IR.

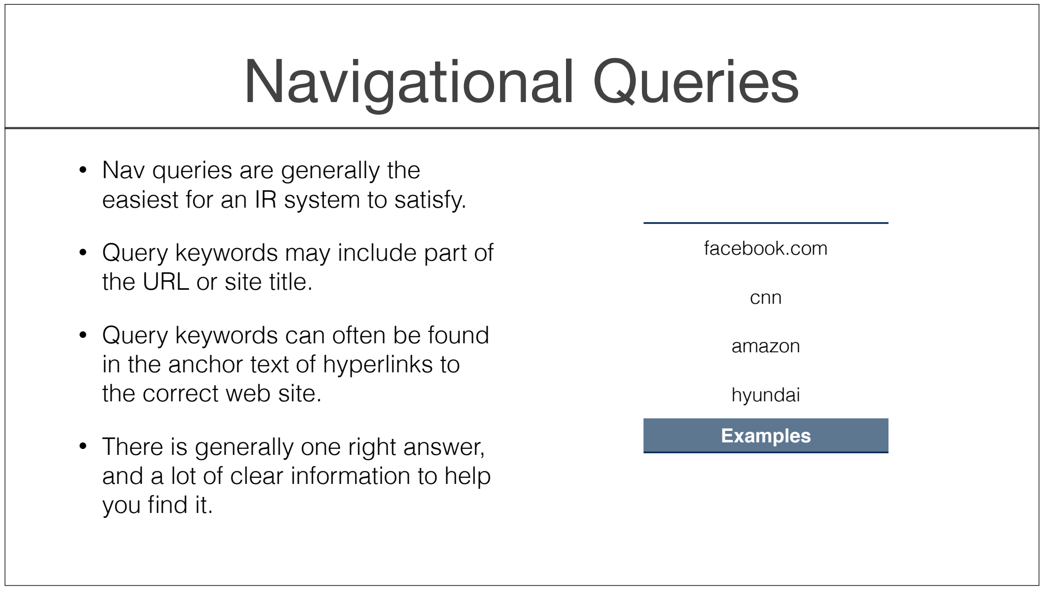


Query Types

Web queries can be roughly divided into three broad categories:

- **Navigational Queries** look for a particular document in the collection. For instance, “cnn” or “facebook.”
- **Transactional Queries** look for a product to purchase or a service to interact with. For instance, “best thai food huntington ave.”
- **Informational Queries** seek information on a particular topic, whether broad or specific. For instance, “fossil fuel alternatives.”

[1: don’t read] In general, queries can be grouped into three big classes. I’ll introduce them here, and then talk in a little more detail about each.
[2: don’t read] First are navigational queries. For nav queries, the user’s objective is to find a particular document from the collection. In web search, that means they’re looking for a particular web site. Quite a few people will type “facebook” into the search box of their web browser to get to facebook.com. That’s a nav query.
[3:don’t read] The second type is transactional queries. With these queries, the user is using the web as a tool to perform a particular action. They might want to purchase a product, get a map, or interact with a web service. Transactional queries are task-driven queries.
[4:don’t read] Finally, we have informational queries. These queries are all about trying to learn something. The query will express some topic, and the goal is to find documents that expound on that topic.
The graph here shows the ratio of queries of each type reported in a particular research paper. The exact proportions vary depending on when and how you measure, but the point is this. Most queries are informational queries, but



Navigational Queries

- Nav queries are generally the easiest for an IR system to satisfy.
- Query keywords may include part of the URL or site title.
- Query keywords can often be found in the anchor text of hyperlinks to the correct web site.
- There is generally one right answer, and a lot of clear information to help you find it.

Examples
facebook.com
cnn
amazon
hyundai

Let’s talk a little more about nav queries.

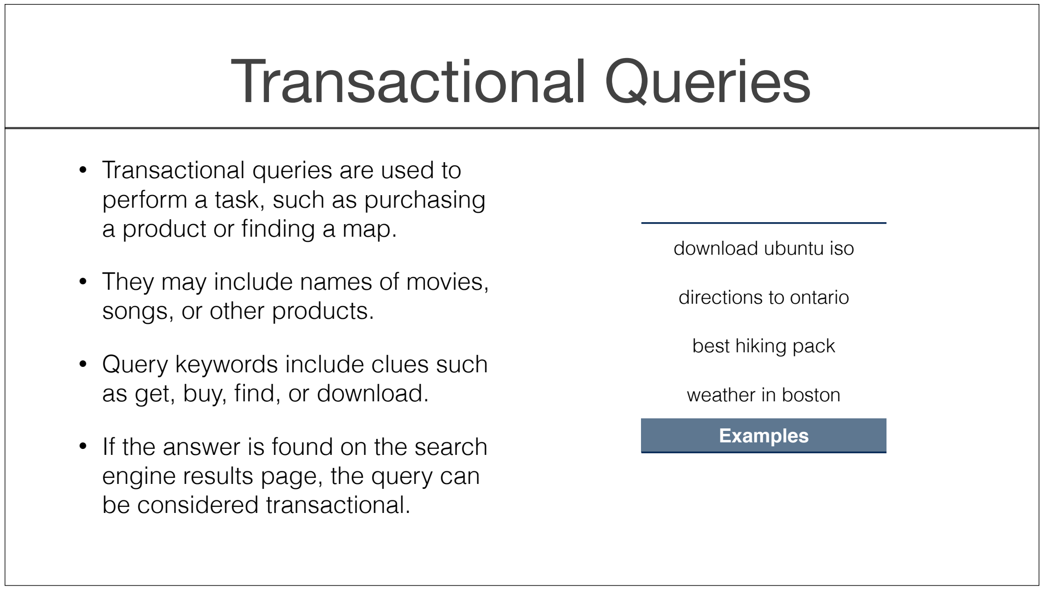
[1:don’t read] These are, by far, the easiest type of query. The user will often just tell you the name of the web site they’re looking for.

[2: read]

[3: read]

If you know you’ve got a nav query you can find the right answer easily, because the query text will be found on all the web site’s <title> tags, their URL, and in the anchor text of links pointing to the site. The Internet is full of big signs pointing to the right web site.

[4: don’t read] For a nav query, there is typically just one relevant document: the particular site the user wants to get to. This problem is easy enough that achieving nearly perfect performance on nav queries is more or less a solved problem.



Transactional Queries

- Transactional queries are used to perform a task, such as purchasing a product or finding a map.
- They may include names of movies, songs, or other products.
- Query keywords include clues such as get, buy, find, or download.
- If the answer is found on the search engine results page, the query can be considered transactional.

Examples
download ubuntu iso
directions to ontario
best hiking pack
weather in boston

Transactional queries are a little bit harder.

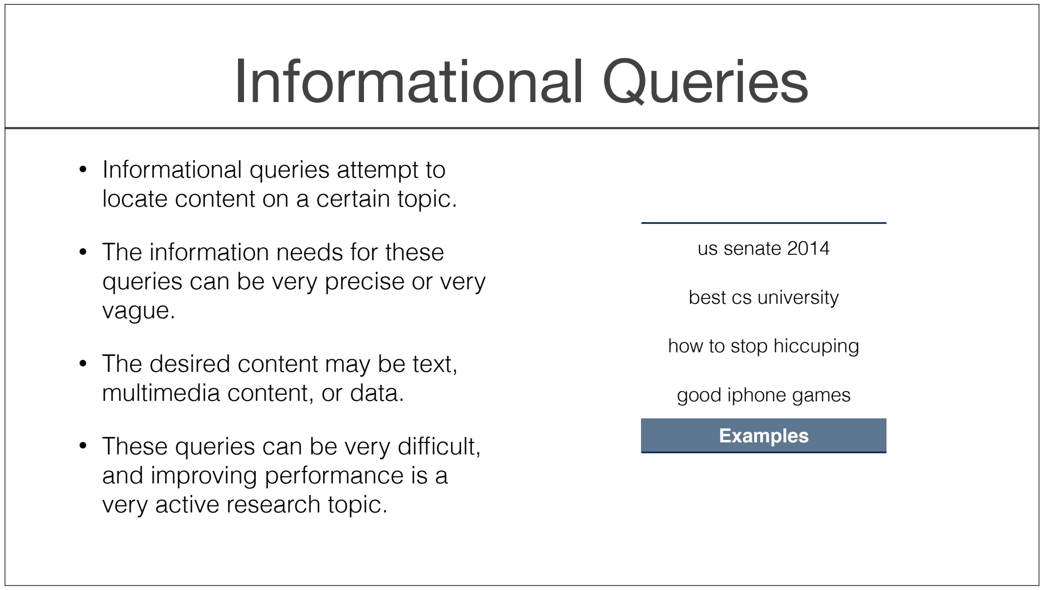
[1: read] These are distinguished from information queries because the user isn’t just trying to learn something; they’re trying to do something. As a rule of thumb, if the query is meant to lead to a sale, a download, or interaction with an online service of some kind, it’s probably a transaction query.

[2: read]

[3: read]

These terms can help filter out possible responses that are purely informational. They will often mention a product or entity from a known product class, such as “backpack” or “Chinese food.”

[4: read] In this case, the online service the user interacted with was the search engine itself.



Informational Queries

- Informational queries attempt to locate content on a certain topic.
- The information needs for these queries can be very precise or very vague.
- The desired content may be text, multimedia content, or data.
- These queries can be very difficult, and improving performance is a very active research topic.

Examples
us senate 2014
best cs university
how to stop hiccuping
good iphone games

Finally, we have Informational Queries.

[1: read]

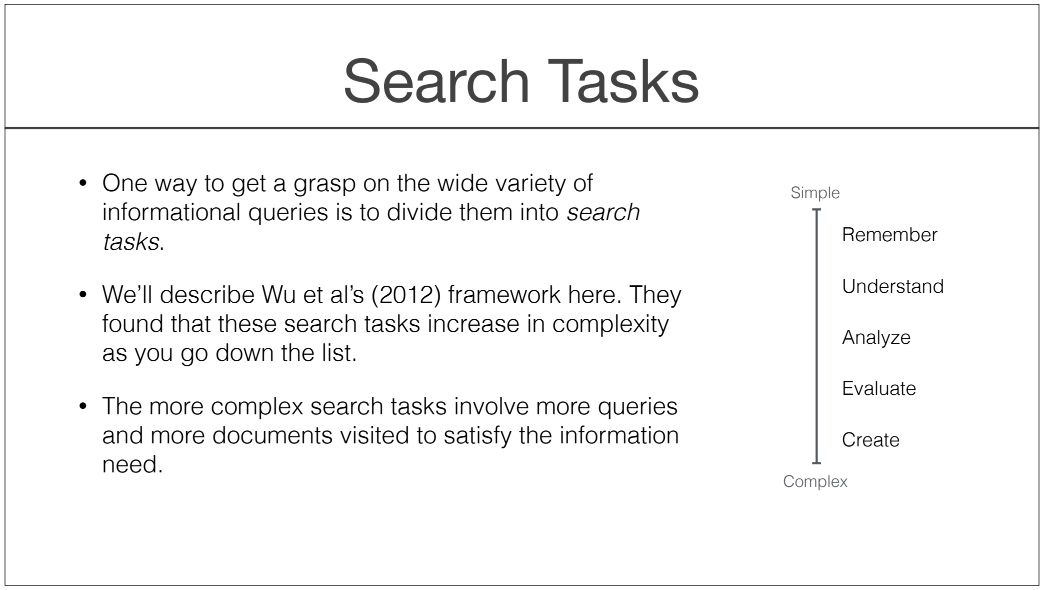
If you could get what you're looking for by talking to an expert, it's probably an informational query. These can be the hardest of all. Nav queries are limited to documents that exist in the collection, and Transactional queries are limited to tasks people have built tools to perform, but the set of possible informational queries is limitless.

[2: read]

We'll explore some of the diversity of these queries in the next slides.

[3: don't read] Some of these queries can be satisfied by a single document, image, or graph. For others, the user will have to combine information from several pages. For instance, a user might want to know about all the possible treatments for some medical condition, and what people have to say about each. Since they're looking for different perspectives on their question, they probably won't find what they need in any single document.

[4: read]



Search Tasks

- One way to get a grasp on the wide variety of informational queries is to divide them into *search tasks*.
- We'll describe Wu et al's (2012) framework here. They found that these search tasks increase in complexity as you go down the list.
- The more complex search tasks involve more queries and more documents visited to satisfy the information need.

Simple

Remember

Understand

Analyze

Evaluate

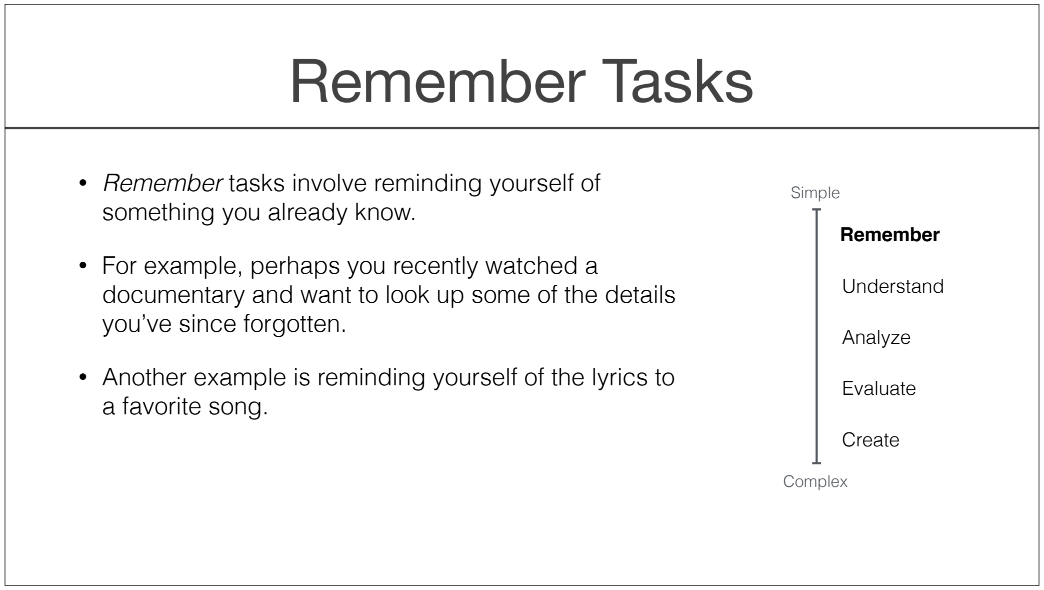
Create

Complex

[read 1]

[read 2]

[read 3]



Remember Tasks

- *Remember* tasks involve reminding yourself of something you already know.
- For example, perhaps you recently watched a documentary and want to look up some of the details you've since forgotten.
- Another example is reminding yourself of the lyrics to a favorite song.

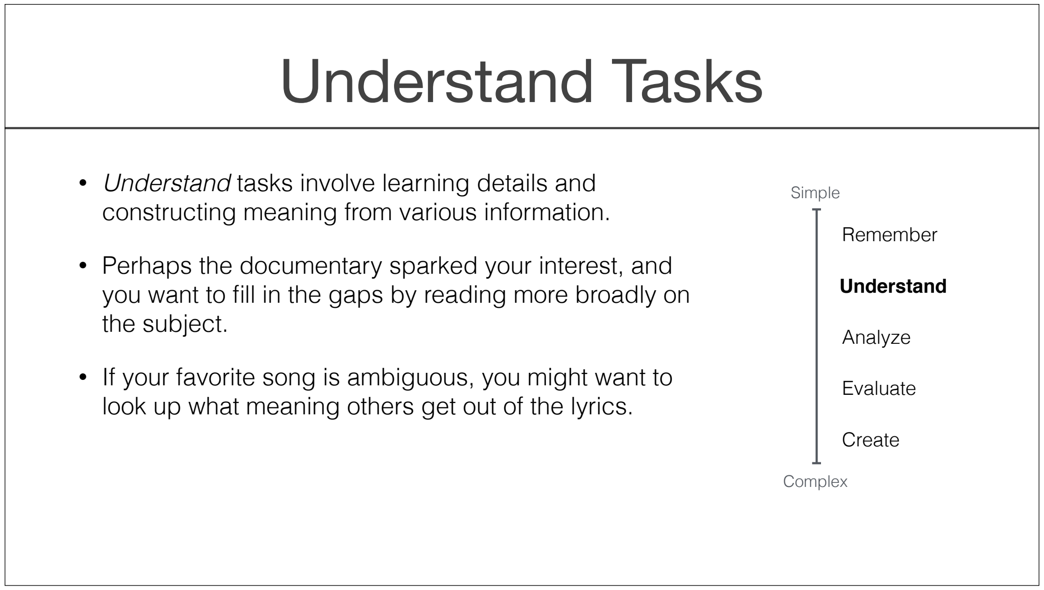
Simple
Remember
Understand
Analyze
Evaluate
Create
Complex

[read 1]

[read 2]

[read 3]

We'll continue with these two examples for the rest of the categories, so you can see how the search tasks increase in complexity.



Understand Tasks

- *Understand* tasks involve learning details and constructing meaning from various information.
- Perhaps the documentary sparked your interest, and you want to fill in the gaps by reading more broadly on the subject.
- If your favorite song is ambiguous, you might want to look up what meaning others get out of the lyrics.

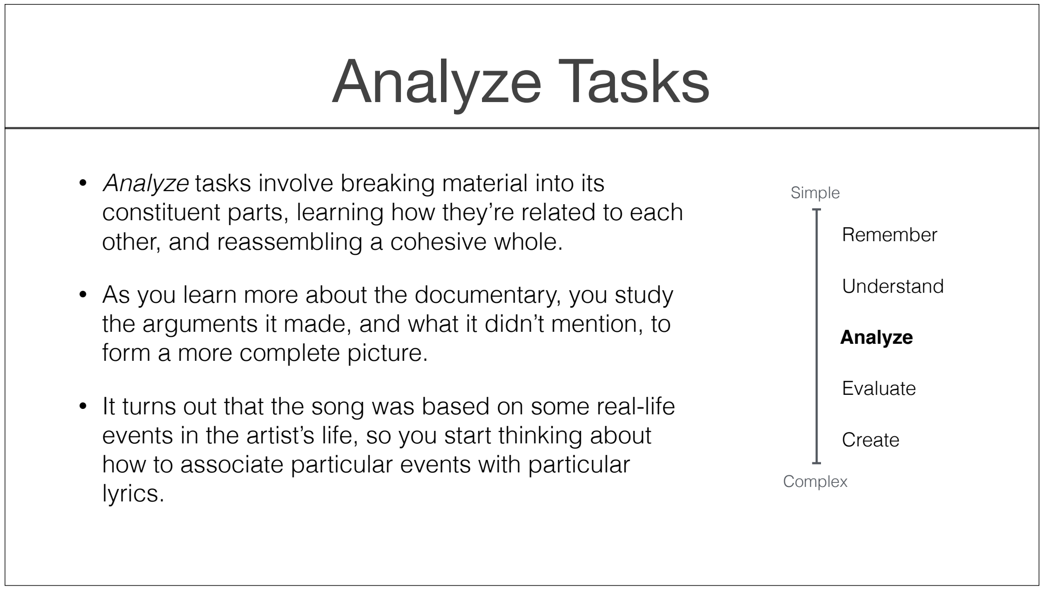
Simple
Remember
Understand
Analyze
Evaluate
Create
Complex

[read 1]

For instance, [read 2]

You're not just trying to remember what was in the documentary; you're trying to put it together with additional information.

For our song example, imagine that the lyrics to your favorite song are ambiguous. You might want to run a few queries to see what meaning other people get out of them.



Analyze Tasks

- *Analyze* tasks involve breaking material into its constituent parts, learning how they're related to each other, and reassembling a cohesive whole.
- As you learn more about the documentary, you study the arguments it made, and what it didn't mention, to form a more complete picture.
- It turns out that the song was based on some real-life events in the artist's life, so you start thinking about how to associate particular events with particular lyrics.

Analyze tasks are a little more complex.

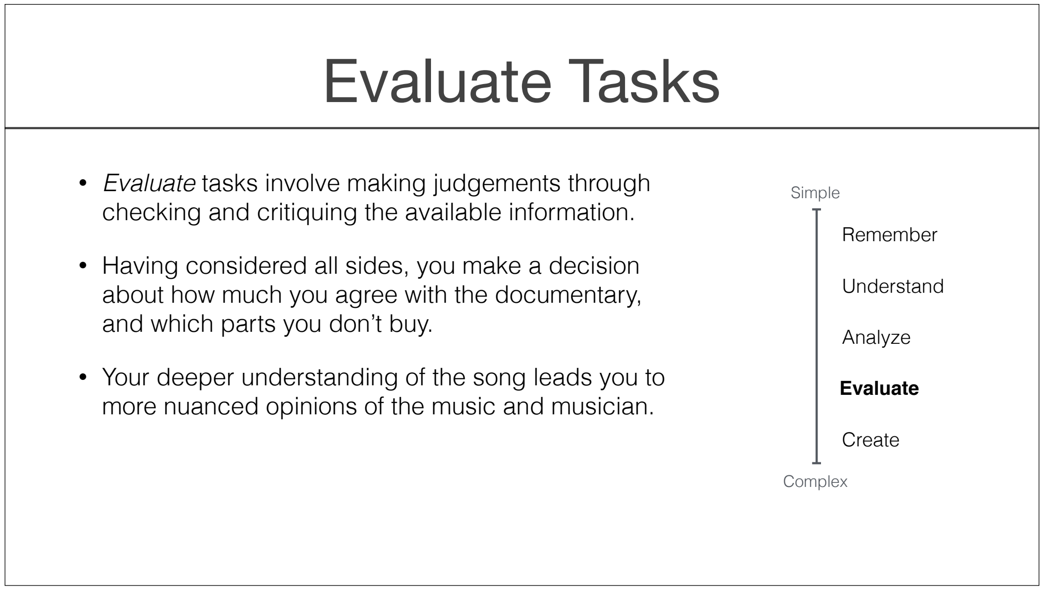
they involve… [read 1]

You're not just trying to understand what others have written. You're examining the individual parts and figuring out what's going on overall.

For a concrete example, [read 2]. Now you're thinking about the subject in more depth. You're not just reading for entertainment; you're putting the big picture together.

Or, for the song example, imagine that [read 3].

Now you're assembling your own meaning for the song, and relating that to the meaning you think the artist assigns to it. You're taking a deeper look and drawing new conclusions.



Evaluate Tasks

- *Evaluate* tasks involve making judgements through checking and critiquing the available information.
- Having considered all sides, you make a decision about how much you agree with the documentary, and which parts you don't buy.
- Your deeper understanding of the song leads you to more nuanced opinions of the music and musician.

Simple
Remember
Understand
Analyze
Evaluate
Create
Complex

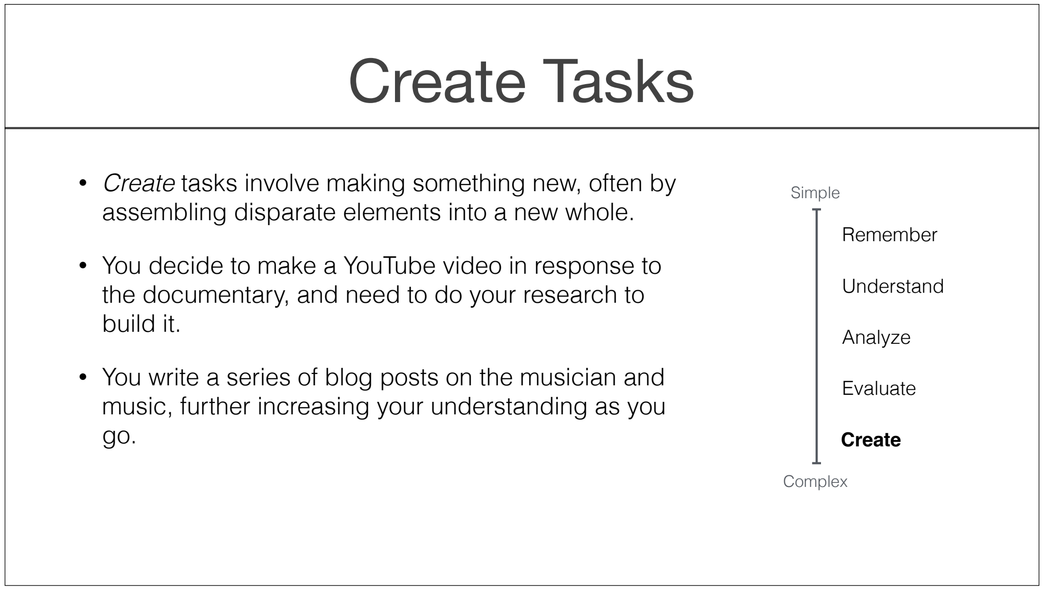
Next are evaluate tasks. [read 1]

These judgements are new opinions that take into account a lot of information, and a deeper understanding of the topic.

For instance, [read 2]

Or [read 3]

You're running a series of queries to help you form new opinions.



Create Tasks

- *Create* tasks involve making something new, often by assembling disparate elements into a new whole.
- You decide to make a YouTube video in response to the documentary, and need to do your research to build it.
- You write a series of blog posts on the musician and music, further increasing your understanding as you go.

Simple
Remember
Understand
Analyze
Evaluate
Create
Complex

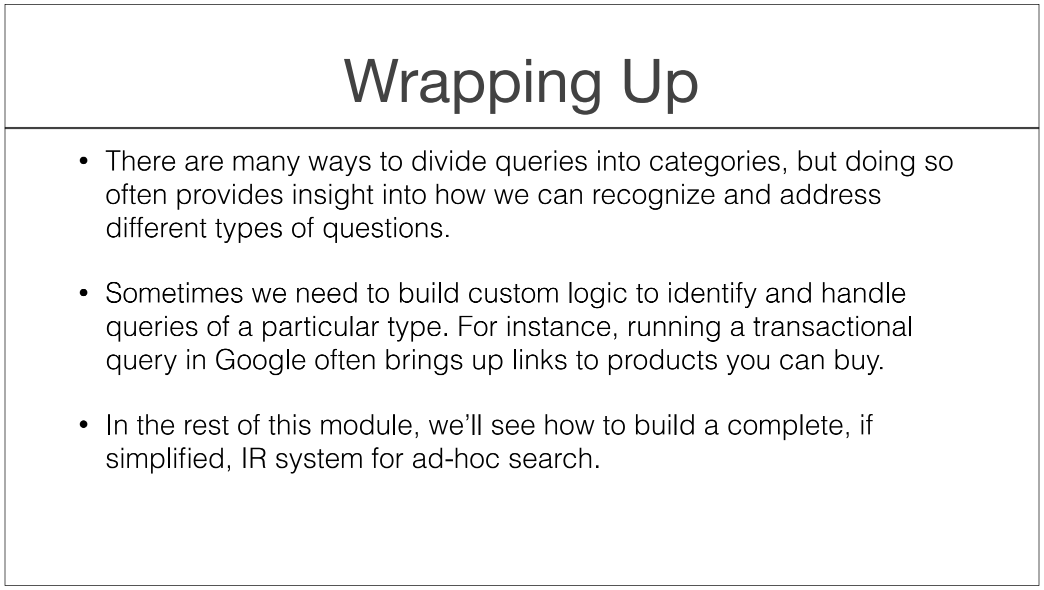
Finally, we have create tasks. [read 1]

These are the most complicated of these information gathering tasks. These information needs are quite detailed, and can change as you learn about the material and create whatever you're building.

For our two examples, maybe [read 2].

Or maybe [read 3].

You are making something new, and you'll typically figure out what you need as you go along.



Wrapping Up

- There are many ways to divide queries into categories, but doing so often provides insight into how we can recognize and address different types of questions.
- Sometimes we need to build custom logic to identify and handle queries of a particular type. For instance, running a transactional query in Google often brings up links to products you can buy.
- In the rest of this module, we'll see how to build a complete, if simplified, IR system for ad-hoc search.

Let's wrap up.

Information Retrieval is concerned with satisfying a user's information need, as expressed by a query. The types of queries users run vary, and many different classification schemes have been suggested. Examining these schemes is useful, because it often provides insight into how we can recognize and address the different types of questions.

[read 2]

In the rest of this module, we'll see how to build a complete, if simplified, IR system for ad-hoc search. That will give us a starting place to explore each of its components in more detail in future modules.

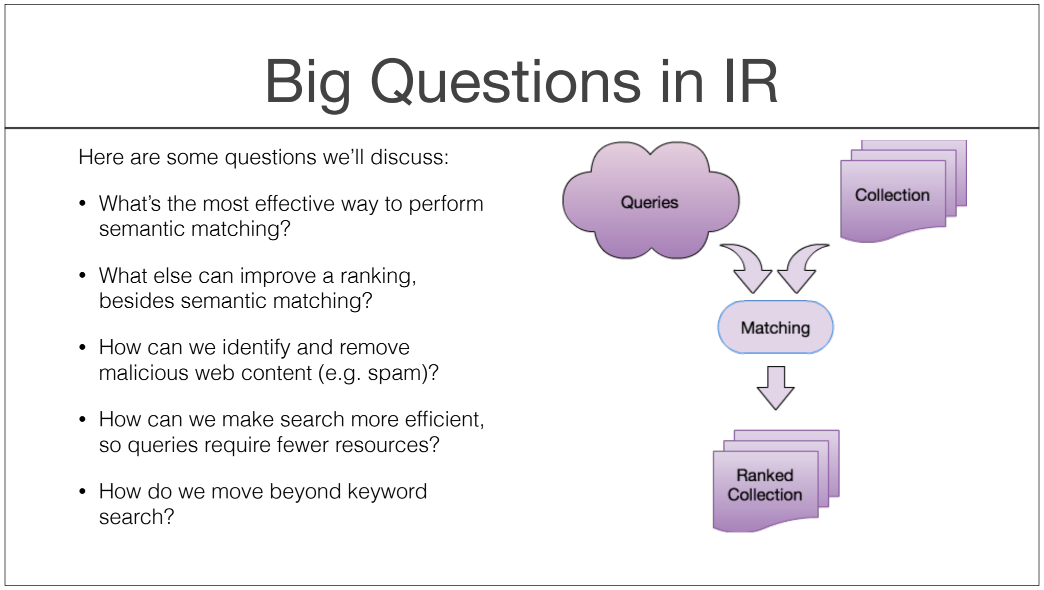
Course Content

IR, session 8

Northeastern University
College of Computer and Information Science

CS6200: Information Retrieval

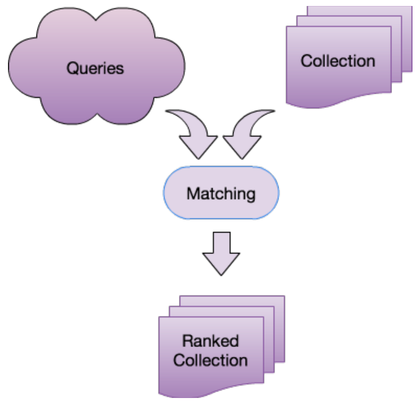
This session is a quick preview of the rest of the course. There’s a lot to cover, so let’s get started.



Big Questions in IR

Here are some questions we'll discuss:

- What's the most effective way to perform semantic matching?
- What else can improve a ranking, besides semantic matching?
- How can we identify and remove malicious web content (e.g. spam)?
- How can we make search more efficient, so queries require fewer resources?
- How do we move beyond keyword search?



Here are some of the questions we'll cover throughout the course.

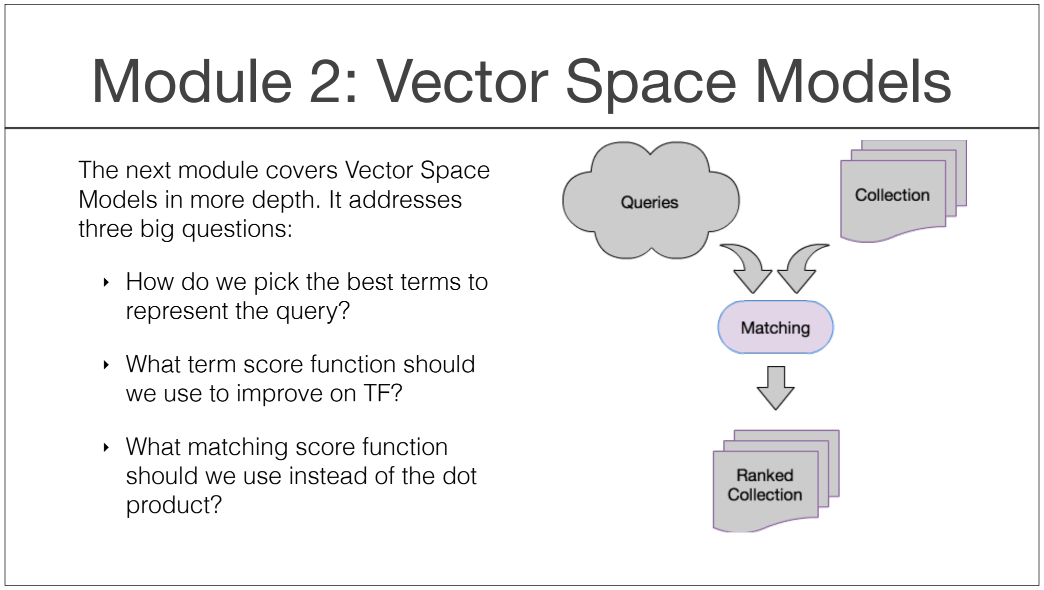
First, what's the most effective way to perform semantic matching? That is, if I give you a query and a document, what are some mathematically principled ways for you to tell me how similar their meanings are?

What else can improve a ranking, besides semantic matching? How do we include information about a document's quality, or authority, or genre, or writing style, to select better search results?

What can we do to protect our users from malicious web content, such as spam, that's designed to take advantage of the weaknesses of IR systems?

How can we minimize the resources it takes to run a query? It's estimated that Google search averages more than 40,000 queries per second. When you're dealing with that kind of volume, shaving off a fraction of a second from each query adds up to a lot of extra processors you don't need to buy. Optimization makes a big difference in production systems.

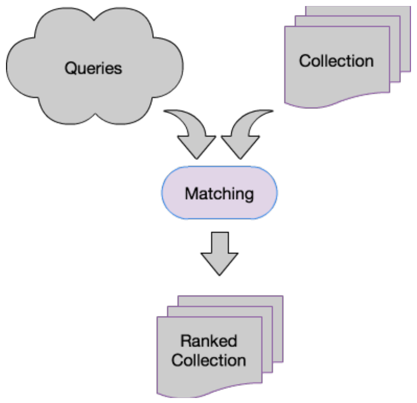
Finally, how do we move beyond keyword search and text documents? What's next for IR, after the search engine?



Module 2: Vector Space Models

The next module covers Vector Space Models in more depth. It addresses three big questions:

- How do we pick the best terms to represent the query?
- What term score function should we use to improve on TF?
- What matching score function should we use instead of the dot product?

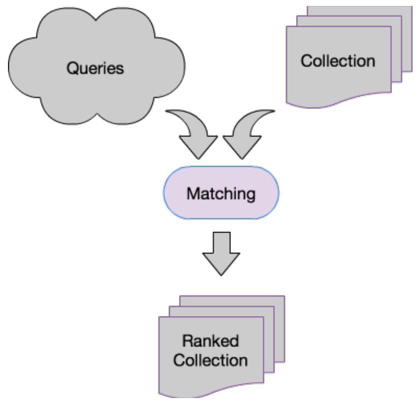


[read all]

Module 3: Language Models

This module does probabilistic semantic matching using NLP-style language models. It addresses:

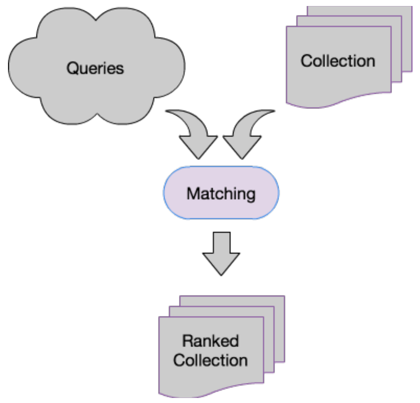
- How to build a probabilistic model of word usage
- How to use these models to estimate the likelihood that the query and document are on the same subject
- How to “fix” your model when you don’t have enough data to train it (e.g. for short documents, or queries)



Module 4: Combining Evidence

Here we discuss improving a ranking by adding extra information to the semantic matching scores:

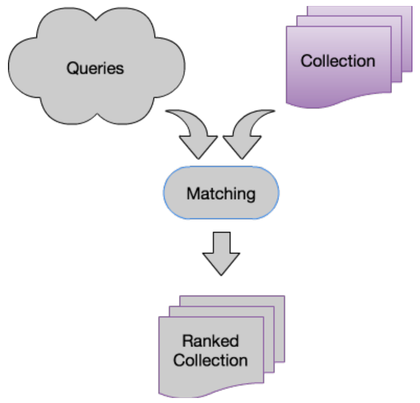
- Estimating the overall quality of a document
- Identifying document types using Machine Learning
- Mixing together many sources of relevance information to produce a final ranking



Module 5: Document Understanding

This module discusses ways to get a stronger signal of the document's topic:

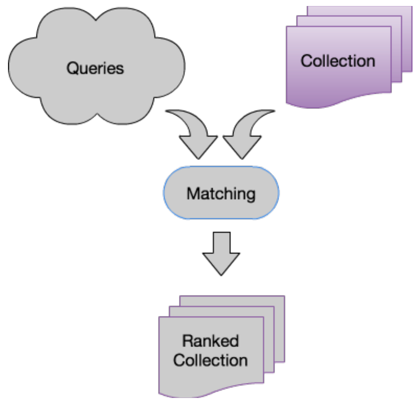
- Finding text emphasized by the document's structure
- Finding named entities (proper nouns) mentioned in the document
- Mathematical models of document topics
- Clustering similar documents together



Module 6: Crawling

Here we move to the mechanics of search, and discuss how to find documents on the Internet:

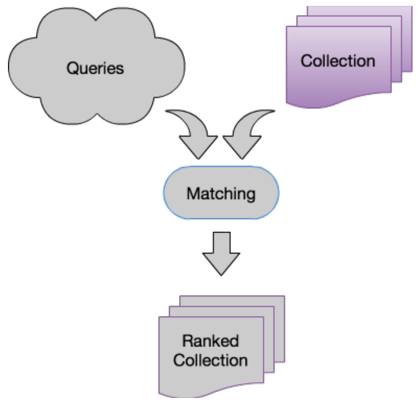
- Selecting the right documents to crawl (because you can't crawl everything)
- Deciding when to re-crawl documents you've already crawled
- Avoiding some of the common pitfalls of crawling the web



Module 7: Indexing

This module discusses the inverted index in depth:

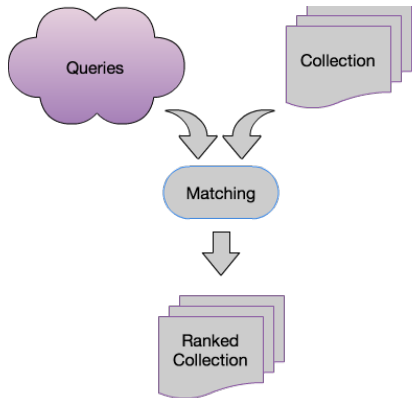
- Creating an inverted index from raw documents
- Storing term, document, and corpus level content in your index
- Efficiently reading the index at search time



Module 8: Interfaces and Logs

Here, we cover ways to improve the user interface and use recorded user interaction to improve search quality:

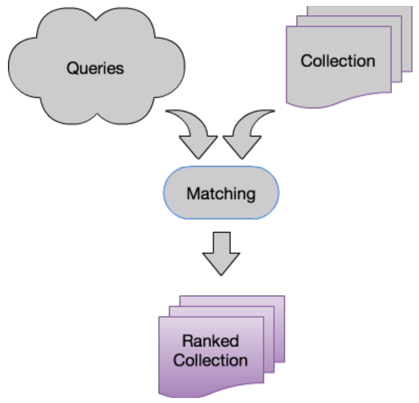
- Giving users more information about documents, so they can decide what to click on
- Using click data to decide whether documents are relevant
- Generating user profiles, and using them to customize search
- Performing location-specific queries



Module 9: Evaluation

How can you tell whether your search engine is good, whether it's improving, and whether it can get better?

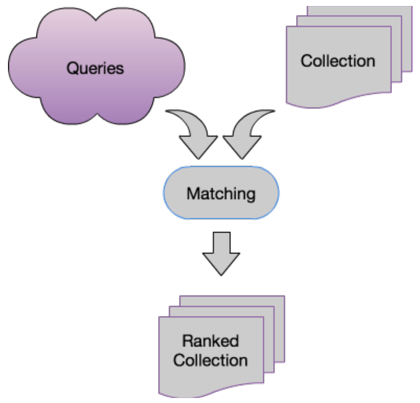
- Mathematical models of user interaction to compare rankings
- Measuring actual user interaction to compare rankings
- Choosing the best evaluation approach for your specific task



Module 10: Beyond Keywords

We explore interesting query types that move beyond keyword search:

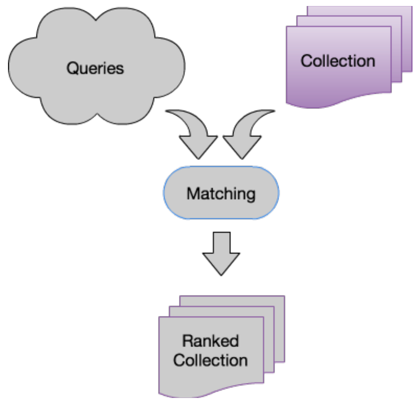
- Answering questions posed in natural language
- Generating summaries of the available information in the collection
- Building a knowledge graph from information on the Internet, and performing logical inference on its contents



Module 11: Beyond Text

This module discusses searching for non-textual content:

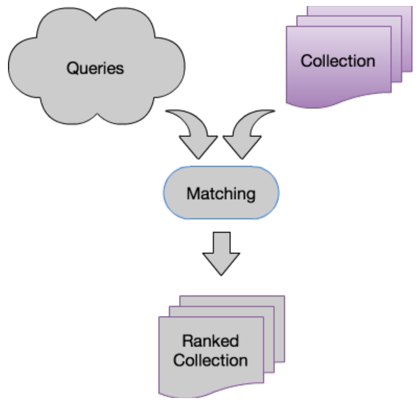
- Searching for images, video, and music
- Finding other objects “like this one”
- Product recommendation based on user ratings

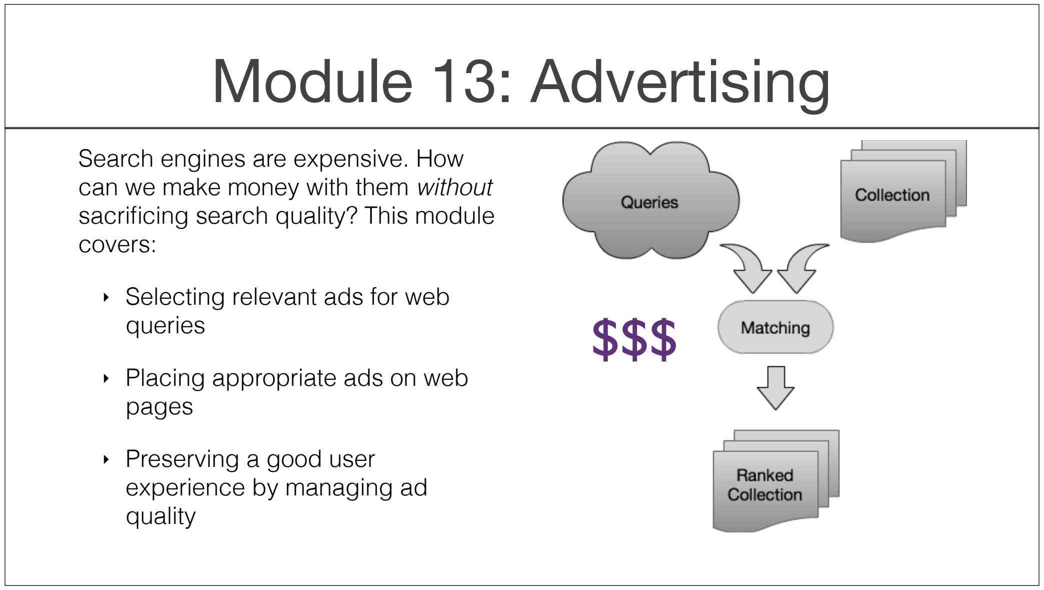


Module 12: Adversarial IR

Many users on the web seek to exploit IR systems to make money at the expense of search quality. This module covers:

- The tricks of the trade for malicious web users
- Various ways to identify spam on the web
- Detecting and responding to link farms

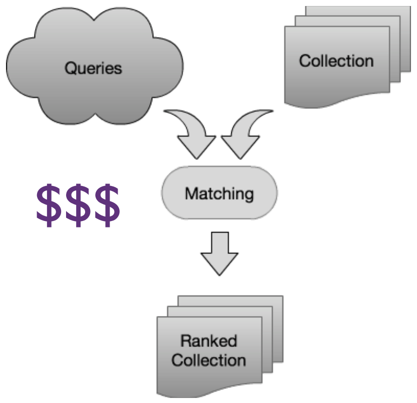




Module 13: Advertising

Search engines are expensive. How can we make money with them *without* sacrificing search quality? This module covers:

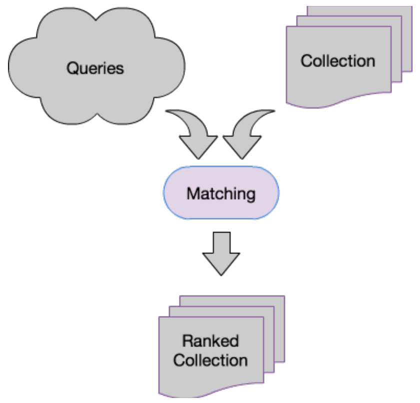
- Selecting relevant ads for web queries
- Placing appropriate ads on web pages
- Preserving a good user experience by managing ad quality



Module 14: Learning to Rank

This module discusses modern approaches of Machine Learning to IR ranking:

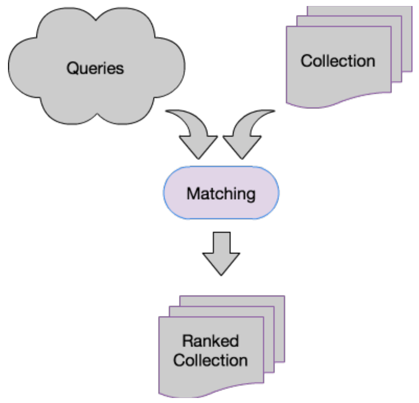
- How to cast ranking as a Machine Learning problem
- Various major approaches taken by Learning to Rank algorithms
- Features used by LtR



Module 15: Semantic Matching

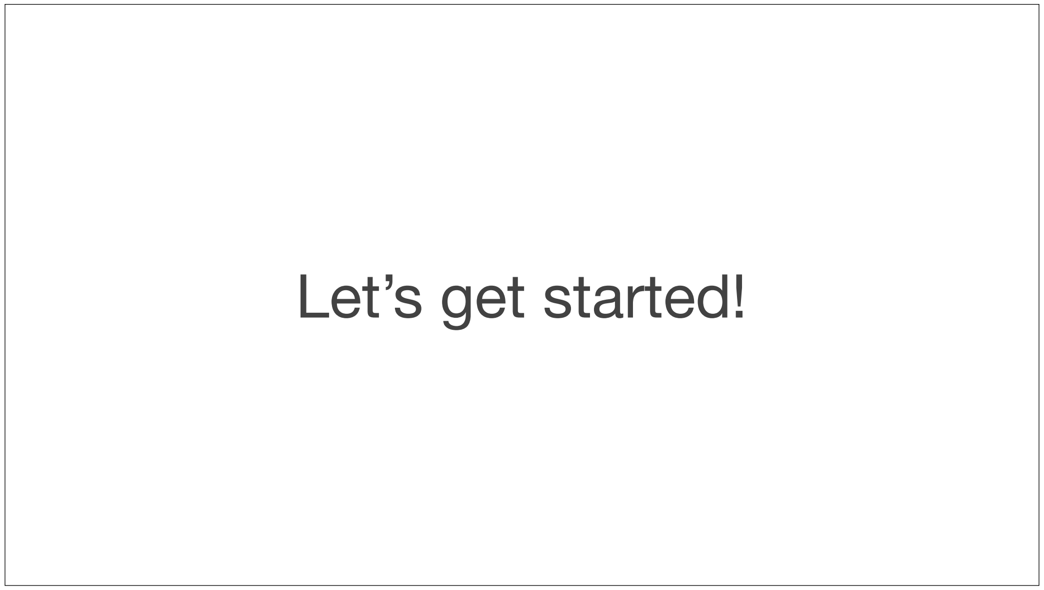
Our final module covers advanced and experimental approaches to semantic matching:

- A deeper discussion of the semantic matching problem
- Projecting documents and queries into a latent space
- Casting semantic matching as a Machine Learning problem (with applications far beyond ranking!)



Wrapping Up

- That’s it!



Let’s get started!

That’s it for the introductory video! Let’s get started.