Personnel

Instructor

• Jiepu Jiang (jiepu@vt.edu), Assistant Professor
• Office hour: Wed 3-5pm, Torgersen 2160M
• Research interest: information retrieval; text & data mining

Graduate Teaching Assistant

• Siyu Mi (siyu6@vt.edu), Ph.D. student
• Office hour: Mon & Fri 9-10am, MCB 106
• Responsibilities
  • Help you with your homework
Enrollment and Course Websites

Force-add request
- Do it by the end of today; you can drop it later

Canvas
- https://canvas.vt.edu/ -- Just Google “vt canvas”
- Homework submission and feedback; discussion

Course Website
- Course schedule, readings, slides, etc.
Textbook & Readings

Textbook (free access online)


Readings

- From the textbooks (both are 10 years old); some research papers
- Readings for the next week will be posted by each Thursday
- Yes, you need to read the readings!
Grading

Homework (50%)
- 5 problem sets; each 10%
- The first 4 are programming-heavy; we use Java a lot

Midterm Exam (20%)
- In-class, 10/25 (Thursday)
- Open book (You probably can’t find answers in the book)

Final Project (30%)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90-100</td>
</tr>
<tr>
<td>A-</td>
<td>85-89</td>
</tr>
<tr>
<td>B</td>
<td>80-84</td>
</tr>
<tr>
<td>B-</td>
<td>70-79</td>
</tr>
</tbody>
</table>
Course Project (30%)

You can either work alone or in groups of two

Purpose: to demonstrate your ability to do IR research

• Work on some retrieval task, e.g., to implement a retrieval model
• Evaluate the approach through experimentation
• Analysis & discussion

Submissions

• You’ll need to submit a proposal (one-page) and a final report (about four pages by the ACM conference proceeding format)
• Presentation (the last week)
Course Policy

Don’t be late!

• Deadlines are deadlines!
• Start early!
• You may not be able to finish within the last day!
  • The datasets for each HWs are about several gigabytes …
• We may not be able to respond to your emails within the last few hours!

Well, you can ask for an extension …

• To allow for emergencies or unforeseen circumstances, once during the semester, you may use a 5-day extension for your homework.
• Only once; only for homework; does not apply to project report & exam
Academic Integrity

• Plagiarism and other forms of cheating will result in an Fail and/or other disciplinary action

• Always cite your sources
  • When in doubt, cite them (or at least ask us)
  • Includes information from people other than professor
  • Includes your fellow students
  • Includes anything someone else wrote, whether published or not, whether on the Web or not
  • Includes the textbooks
  • Exception is class lecture notes and presentations

• Encouraged to discuss homework issues with each other

• Handed in work must be your own
  • Do not copy, do not let someone read your work, do not leave it unprotected in your account
Course Schedule

http://people.cs.vt.edu/~jiepu/cs5604_fall2018/

You’ll be very busy before the midterm exam

• HW1-4 (programming-heavy; you’ll get a new one every two weeks)
• Midterm (open book)
• Form into groups and work out a project proposal

The rest are easy

• HW5
• A project presentation
• A project report
Getting help

You can meet us everyday
• TA’s office hour: Monday & Friday
• Instructor’s office hour: Wed
• Class meetings: Tue & Thur (after class)

Email is slow …
• We’ll try to respond within one or two days.

You are encouraged to discuss with your classmates
About the IR Community

Related Community

• ACM special interest group in information retrieval (ACM SIGIR)
• Mailing list: SIG-IRList (just Google it)

Related Conferences & Journals

• Annual SIGIR conference, and sub-conferences (ICTIR & CHIIR)
• Related: CIKM, WSDM, KDD, WWW, ACL, EMNLP, NAACL, JCDL …
• Regional: ECIR, AIRS, DIR, ADCS, …
• Evaluation workshops (like data challenges): TREC, CLEF, NTCIR, FIRE, …
• Journals: IR journal, ACM TOIS, JASIST, IP&M, J Doc, TKDE, …
What is Information Retrieval?

• An IR system helps people find the useful information.
• How it works: returns some units of information in response to a query
• Indexing: some effective data structure that makes it ultrafast
• Ranking: some formula to rank the most useful results on the top
• Evaluation: a set of techniques for distinguishing good and bad systems
• PageRank is NOT the core technique!
## IR ≠ DB

<table>
<thead>
<tr>
<th></th>
<th>Database</th>
<th>IR</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Types of data supported</strong></td>
<td>Mostly structured</td>
<td>Most anything (but mostly unstructured)</td>
</tr>
<tr>
<td><strong>System’s knowledge of data semantics</strong></td>
<td>Precise</td>
<td>Imprecise</td>
</tr>
<tr>
<td><strong>Nature of queries</strong></td>
<td>Specific</td>
<td>Vague</td>
</tr>
<tr>
<td><strong>Answer quality</strong></td>
<td>Exact</td>
<td>Likely</td>
</tr>
</tbody>
</table>
### Structured vs. Unstructured Data

- An example of structured data: a database table

<table>
<thead>
<tr>
<th>First Name</th>
<th>Last Name</th>
<th>Address</th>
<th>City</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mickey</td>
<td>Mouse</td>
<td>123 Fantasy Way</td>
<td>Anaheim</td>
<td>73</td>
</tr>
<tr>
<td>Bat</td>
<td>Man</td>
<td>321 Cavern Ave</td>
<td>Gotham</td>
<td>54</td>
</tr>
<tr>
<td>Wonder</td>
<td>Woman</td>
<td>987 Truth Way</td>
<td>Paradise</td>
<td>39</td>
</tr>
<tr>
<td>Donald</td>
<td>Duck</td>
<td>555 Quack Street</td>
<td>Mallard</td>
<td>65</td>
</tr>
<tr>
<td>Bugs</td>
<td>Bunny</td>
<td>567 Carrot Street</td>
<td>Rascal</td>
<td>58</td>
</tr>
<tr>
<td>Wiley</td>
<td>Coyote</td>
<td>999 Acme Way</td>
<td>Canyon</td>
<td>61</td>
</tr>
<tr>
<td>Cat</td>
<td>Woman</td>
<td>234 Purrfect Street</td>
<td>Hairball</td>
<td>32</td>
</tr>
<tr>
<td>Tweety</td>
<td>Bird</td>
<td>543</td>
<td>Itotltaw</td>
<td>28</td>
</tr>
</tbody>
</table>
Structured vs. Unstructured Data

- An example of structured data: JSON key-value pairs

```json
{
    "firstName": "John",
    "lastName": "Smith",
    "age": 25,
    "address": {
        "streetAddress": "21 2nd Street",
        "city": "New York",
        "state": "NY",
        "postalCode": "10021"
    },
    "phoneNumber": [
        { "type": "home", "number": "212 555-1234" },
        { "type": "fax", "number": "646 555-4567" }
    ]
}
```
Structured vs. Unstructured Data

- An example of unstructured data: a news webpage
Imprecise Knowledge of Data

Databases
- Fields with clear semantics
  - SSN is a social security number, BIRTHDAY is a Gregorian date

Information retrieval systems
- Allows for fields with semantics
- Mostly objects with incomplete or unspecified semantics
  - Free text fields, image content, ...
- Systems “knowledge” about objects is incomplete
DB vs. IR: Queries

Database

• Query is precise, unambiguous
• select model from phones
  where price < 300
  and camera-mp > 5
  and review-summary = “positive”

Information retrieval

• Find me a cheap mobile phone with a high MP camera
• Cell phone deal
DB vs. IR: Answers

Database
- Answers are tuples that match query
- What is returned is by definition “relevant”
- You are responsible to make your query correct
- System evaluation often focuses on efficiency
  - Including transactions, recoverability, etc.

Information retrieval
- Answers are estimates by system
- Typically ranked by likelihood of being “relevant”
- Need to measure system effectiveness
- Efficiency still an important issue at scale
  - Transaction processing rarely emphasized
Intents of IR Queries

Navigational search (homepage finding)
- Facebook
- wiki

Informational (know something)
- clinton trump polls

Transactional (get things done)
- Flight ROA to SFO

Many Other intents …
IR is not just for the web

**IR systems**
- FAST, Northern Light, Vivisimo, Funnelback, Endeca, …
- RecipeBridge, Monster, Healthline, PubGene, Realtor, …
- Lemur, Indri, Terrier, Zettair, Lucene, Galago, …

**Web search and In-house systems**
- West, LEXIS/NEXIS, Quicklaw
- Google, Bing, Baidu, Blekko, Yandex, Yahoo!…
- Ask.com (Ask Jeeves), Powerset
- eLibrary, GOV.Research_center, Inquira
IR is not just document retrieval

- Automatic organization (e.g., clustering)
- Question & answering
- Recommender systems
- Leveraging XML and other Metadata
- Text mining
- Novelty identification
- Personal information management
- Social media
- Product search
- Meta-search (multi-database searching)
- Summarization
- ...
A simple flow of the retrieval process

Information Need → Representation → Query

Text Objects → Representation → Indexed Objects

Comparison

Evaluation/Feedback

Retrieved Objects

Dashed line indicates feedback loop
User’s Mind
I plan to take CS 5604. What’s that?

Note that a user may issue many different queries for the same information need.
**System**

<table>
<thead>
<tr>
<th>Terms</th>
<th>Term Occurrences in Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 2)</td>
</tr>
<tr>
<td>information</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>query</td>
<td>(Doc#2, 2)</td>
</tr>
<tr>
<td>retrieval</td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
</tbody>
</table>
|            | .....

**“Bag-of-words” representation of documents**

Doc#1
- information x 1
- computer x 2
- data x 1
- science x 1

Doc#2
- retrieval x 1
- query x 2
- index x 1
- data x 1

Doc#3
- retrieval x 1
- information x 1
- data x 2
- search x 1
Matching query terms & indexed objects

Query: Information Retrieval

<table>
<thead>
<tr>
<th>Terms</th>
<th>Term Occurrences in Documents</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>information</td>
<td>(Doc#1, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>query</td>
<td>(Doc#2, 2)</td>
</tr>
<tr>
<td>retrieval</td>
<td>(Doc#2, 1)</td>
</tr>
<tr>
<td></td>
<td>(Doc#3, 1)</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
Boolean Search vs. Best Match Search

Boolean Search
• e.g., information AND retrieval, information OR retrieval
• Returns a set of results
• Results are similar to those by the following SQL:

```
SELECT * 
FROM articles
WHERE
  content LIKE "%information%"
OR
  content LIKE "%retrieval%"
```

Best Match Search (the most popular form of search)
• Returns a ranked list of search
• Sorted by some scores (indicating the inferred relevance of results)
Ranking Search Results

Query: Information Retrieval

Information Retrieval

information (Doc#1, 1) (Doc#3, 1)
retrieval (Doc#2, 1) (Doc#3, 1)

Merge (Boolean OR)

Doc#1 Doc#2 Doc#3

Scoring & Ranking (best match)

Doc#3
Score: 4.5

Doc#2
Score: 3

Doc#1
Score: 1.5
Although Google does not show any scores to you, it still computes certain scores for the search results and rank them by the scores.
it’s an old idea...
The old Tillamook County Library card catalog was recently put to new use as a holder of seed packets. (Photo by LeeAnn Neal)
**Compute Relevance score – An Example**

Many factors may contribute to score(q, d)
- Occurrence or non-occurrence
- Term frequency
- Document length
- Term importance
- Many others ....

Many models break down score(q, d) into a linear combination of score(w, d) – the score of each term w in the document d.

\[
\text{score}(q, d) = \sum_{w \in q} \text{score}(w, d)
\]
Occurrence or Non-occurrence

\[
score(w, d) = \begin{cases} 
1 & \text{if } w \text{ appears in } d \\
0 & \text{if } w \text{ does not appear}
\end{cases}
\]

Query: Information Retrieval

\[
score(q, d) = 1 \\
score("inform~", d) = 1 \\
score("retrieval", d) = 0
\]

\[
score(q, d) = 1 \\
score("inform~", d) = 0 \\
score("retrieval", d) = 1
\]

\[
score(q, d) = 2 \\
score("inform~", d) = 1 \\
score("retrieval", d) = 1
\]

Doc#1
- information x 1
- computer x 2
- data x 1
- science x 1

Doc#2
- retrieval x 1
- query x 2
- index x 1
- data x 1

Doc#3
- retrieval x 1
- information x 1
- data x 2
- search x 1
Matching a frequent term in a document is more important than matching a less frequent one.

“... the more frequently a notion and combination of notions occur, the more importance the author attaches to them as reflecting the essence of his overall idea.”

Hans Peter Luhn (1896-1964)

Term Frequency (TF)

\[ \text{score}(w, d) = \text{count}(w, d) \]

Query: Information Retrieval

- \( \text{score}(q, d) = 2 \)
- \( \text{score}(\text{"inform~"}, d) = 2 \)
- \( \text{score}(\text{"retrieval"}, d) = 0 \)

Doc#1

- information x 2
- computer x 2
- data x 1
- science x 1

Doc#2

- retrieval x 3
- query x 2
- index x 1
- data x 1

Doc#3

- retrieval x 1
- information x 2
- data x 2
- search x 1
Term Frequency (TF): Variants

\[
\text{score}(w, d) = \begin{cases} 
1 + \log \text{count}(w, d) & \text{count}(w, d) > 1 \\
0 & \text{count}(w, d) = 0 
\end{cases}
\]

- Avoid assigning a too strong impact on a very frequent term.
- Make sure the difference between 0 (no occurrence) and 1 (appearing once) is greater than that between 1 and 2 and so on

\[
\frac{dy_1}{dx} = 1 \quad \frac{dy_2}{dx} = \frac{1}{x} \leq 1 \quad \lim_{x \to \infty} \frac{dy_2}{dx} = 0
\]

\[
y_1 = x \quad y_2 = 1 + \log x
\]
Document length matters – It’s easy to match a query term in a long document

Many retrieval models normalize TF by document length.

\[ \text{score}(w, d) = \log \frac{\text{count}(w, d) + 1}{\text{length}(d) + 1} \]

Query: information retrieval

Which of the following seem more relevant?

Doc#1: a 5000-word paper
Information x 10
retrieval x 10
....

OR

Doc#2: a 300-page book
Information x 10
retrieval x 10
....

Information retrieval
Not All Terms are Equally Important

• A general belief is that more specific terms are more important, but we need a measure for specificity.

Query: information retrieval

Which of the following two docs seem more relevant?

Doc#1
- information x 1
- computer x 2
- data x 1
- science x 1

OR

Doc#2
- retrieval x 1
- query x 2
- index x 1
- search x 1
Inverse Document Frequency (IDF)

\[ \text{score}(w, d) = \text{IDF} = \log \frac{N}{n} \]

N: the total number of documents in the corpus
n: the number of documents containing the term
The log base is not important.

Rare terms are more discriminative and important.

Inverse Document Frequency (IDF)

\[
score(w, d) = IDF = \log \frac{N}{n}
\]

\(N\): the total number of documents in the corpus
\(n\): the number of documents containing the term

Rare terms are more discriminative and important.

<table>
<thead>
<tr>
<th>Word</th>
<th>N</th>
<th>n</th>
<th>IDF</th>
</tr>
</thead>
<tbody>
<tr>
<td>information</td>
<td>278123</td>
<td>60938</td>
<td>1.52</td>
</tr>
<tr>
<td>retrieval</td>
<td>278123</td>
<td>2034</td>
<td>4.92</td>
</tr>
</tbody>
</table>
Why IDF works?

\[ \hat{P}(w) = \frac{n}{N} \quad \text{IDF} = \log \frac{N}{n} \]

• The ratio \( n/N \) is an estimate on how likely the term is to appear in a document by chance.

• A term with a low IDF is one that is likely to appear in documents just by chance.

• So the occurrence of a term with a low IDF is less important/informative compared with one with a high IDF.


Stephen Robertson (Gerard Salton Awards, 2000)
TF-IDF Weighting

\[ \text{score}(q, d) = \sum_{w \in q} \text{tf}(w, d) \cdot \text{idf}(w) \]

• Since both TF and IDF are important factors, we just combine them.


Gerard Salton (1927-1995)
(Gerard Salton Award, 1983)
TF-IDF Weighting: Variants of TF

\[
\text{score}(q, d) = \sum_{w \in q} \text{tf}(w, d) \cdot \text{idf}(w)
\]

<table>
<thead>
<tr>
<th>Variant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binary</strong></td>
<td>0 (not occur) or 1 (any occurrence).</td>
</tr>
<tr>
<td><strong>Raw TF</strong></td>
<td>Just the raw frequency.</td>
</tr>
<tr>
<td><strong>Log TF</strong></td>
<td>(1 + \log \text{freq})</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
TF-IDF Weighting: Variants of IDF

\[
\text{score}(q, d) = \sum_{w \in q} \text{tf}(w, d) \cdot \text{idf}(w)
\]

<table>
<thead>
<tr>
<th>Variant</th>
<th>(\text{idf}(w))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Do not weight</td>
<td>1</td>
</tr>
<tr>
<td>KSJ</td>
<td>(\log \frac{N}{n})</td>
</tr>
<tr>
<td>BM25</td>
<td>(\log \frac{N - n + 0.5}{n + 0.5})</td>
</tr>
<tr>
<td>....</td>
<td></td>
</tr>
</tbody>
</table>
Similarity implies relevance

“The more two representations agreed in given elements and their distribution, the higher would be the probability of their representing similar information.”

Ranking results: query-document similarity

Vector Space Model (SMART system)

- Represent both query and document as vectors of terms (words).
- Measure similarity using cosine.

Vector Space Model

- A document similar to the query has a small angle in the N-dimensional space, and thus a greater cosine(q, d).

- We’ll talk more details in Lecture 6.
These ideas are still being used today, but are implemented in a more complex (and probably more effective) way.
Remember they are just ideas/heuristics. They may not be correct and they may not work well for all occasions.
In this course, we will introduce…

• More on Boolean search & Vector Space Model (Lecture 5)
  • VSM is still quite popular today!
  • The default model of Apache Lucene is a variant of VSM.

• Probabilistic retrieval models and BM25 (Lecture 6)
  • BM25 is a variant of TF-IDF
  • Becomes popular since 1995
  • Still one of the best-performing ad hoc search model.

• Language modeling approach for IR (Lecture 7 and Lecture 8)
  • Comes from the statistical language models in NLP
  • Becomes popular since 1998
  • Representation + retrieval model
  • Still one of the best-performing ad hoc search model.
Other ideas: query term proximity

- A document where the query terms appeared adjacent to each other is more likely relevant than another where the terms are far away from each other.
- Example: phrase match
- Sequential dependence model (Lecture 11)

**Query:** information retrieval

Which of the following two docs seem more relevant?

Doc#1

... information retrieval ...

OR

Doc#2

... information ... (1000 words)... retrieval ...
Vocabulary Mismatch

- Searchers and authors of documents use different words to refer to the same notion, causing difficulties in matching.

Score(q, d) = 0
Query representation

Improving query representation

• (Automatic) query and document expansion (Lecture 10)

---

<table>
<thead>
<tr>
<th>“Monica Lewinsky Case”</th>
<th></th>
<th>“Rats in Space”</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(w</td>
<td>Q) )</td>
<td>( w )</td>
</tr>
<tr>
<td>0.041</td>
<td>lewinsky</td>
<td>0.062</td>
</tr>
<tr>
<td>0.038</td>
<td>monica</td>
<td>0.030</td>
</tr>
<tr>
<td>0.027</td>
<td>jury</td>
<td>0.020</td>
</tr>
<tr>
<td>0.026</td>
<td>grand</td>
<td>0.018</td>
</tr>
<tr>
<td>0.019</td>
<td>confidant</td>
<td>0.014</td>
</tr>
<tr>
<td>0.016</td>
<td>talk</td>
<td>0.012</td>
</tr>
<tr>
<td>0.015</td>
<td>case</td>
<td>0.012</td>
</tr>
<tr>
<td>0.014</td>
<td>president</td>
<td>0.011</td>
</tr>
<tr>
<td>0.013</td>
<td>clinton</td>
<td>0.010</td>
</tr>
<tr>
<td>0.010</td>
<td>starr</td>
<td>0.010</td>
</tr>
</tbody>
</table>

Improving document representation (Lecture 9)

- Latent semantic indexing (LSI) and Probabilistic LSI (PLSI)
- Cluster-based retrieval and topic model (e.g., LDA)

A topic model for “information retrieval”

| Word     | P(w|D) |
|----------|--------|
| Information | 0.087  |
| Retrieval   | 0.068  |
| Search      | 0.043  |
| Engine      | 0.032  |
| index       | 0.028  |
| ....        |        |
Distributed representations of words

Word embeddings (Lecture 14)

• Representing words as vectors (of a low-dimensional space comparing to the vocabulary size).
• A guest lecture on deep learning for IR (from CIIR grad students)
Learning-to-rank

Since 2005, a popular method is to combine many different factors into search result ranking and learn the weights of factors automatically by machine learning techniques.

An example:

\[
\text{Score}(q, d) = 3.2 \times \text{score(}\text{information, d)} + 2.8 \times \text{score(}\text{retrieval, d)} + 5.9 \times ("\text{information retrieval}", d) + 0.8 \times \text{PageRank}(d) + 1.2 \times \text{HITS}(d) - 0.8 \times \text{Spam}(d) + 3.5 \times \text{similarity(}\text{previous queries, d)} + 6.5 \times \text{similarity(}\text{previous clicks, d)}
\]
Deep Neural Nets for Ranking
Contextual and Personalized Search

Taking into account context

• Who you are
• Where you are
• Short-term interest
• Long-term interest
• Your previous search activities

Example: Google always ranks “VT canvas” at the top when you search the query “canvas” in Blacksburg.
Not all documents are equally important

Link analysis

- PageRank and variants
- HITS and variants

Spam detection

- Determine whether a web page is a spam web page or not
Query suggestion

Searches related to information retrieval

information retrieval course  information retrieval algorithms
information retrieval definition  information retrieval journal
information retrieval system  information retrieval ppt
information retrieval techniques  information retrieval pdf
Q&A & Conversational Systems
Next Lecture

• Some basic ideas of IR evaluation
• Getting started – programming-related tutorials