Natural Language Processing and Information Retrieval

Course Description

Alessandro Moschitti
Department of Computer Science and Information Engineering
University of Trento
Email: moschitti@disi.unitn.it

Course Schedule

- Lectures
  - Tuesday,   8:30 - 10:30
  - Thursday, 10:30 - 12:30
  - Room 107
  - Some lectures in lab
- Consulting hours:
  - My office at third floor
  - Monday since14:00 to 15:30
  - Sending email is recommended

Syllabus

- Introduction to Information Retrieval (IR)
  - Boolean retrieval, Vector Space Model, Feature Vectors, Document/Passage Retrieval, Search Engines, Relevance Feedback & Query Expansion, Document Filtering and Categorization, flat and hierarchical clustering, Latent Semantic Analysis, Web Crawling and the Google algorithm.
- Statistical Machine Learning:
  - Kernel Methods, Classification, Clustering, Ranking, Re-Ranking and Regression and hints to practical machine learning.

Syllabus

- Performance Evaluation:
  - Performance Measures, Performance Estimation, Cross validation, Held Out and n-Fold Cross validation
- Statistical Natural Language Processing:
  - Sequence Labeling: POS-tagging, Named Entity Recognition and Normalization.
  - Syntactic Parsing: shallow and deep Constituency Parsing, Dependency Syntactic Parsing.
  - Shallow Semantic Parsing: Predicate Argument Structures, SRL of FrameNet and ProbBank, Relation Extraction (supervised and semi-supervised).
  - Discourse Parsing: Coreference Resolution and discourse connective classification

Syllabus

- Joint NLP and IR applications:
  - Deep Linguistic Analysis for Question Answering: QA tasks (open, restricted, factoid, non-factoid), NLP Representation, Question Answering Workflow, QA Pipeline, Question Classification and QA reranking.
  - Fine-Grained Opinion Mining: automatic review classification, deep opinion analysis, automatic product extraction and review, reputation/social media analysis

Lab

- Search Engines
- Automated Text Categorization
- Syntactic Parsing and Named Entity Recognition
- Question Classification (Question Answering)

Where to study?

- Course Slides at http://disi.unitn.it/moschitti/teaching.html
  - NLP-IR section
- Book - IR:
  - Modern Information Retrieval Authors:Ricardo A. Baeza-Yates. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA ©1999 ISBN:020139829X
  - IIR: Introduction to Information Retrieval. Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. Cambridge University Press, 2008.

Where to study?

- Book – NLP:
  - Foundations of Statistical Natural Language Processing. Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999
  - SPEECH and LANGUAGE PROCESSING.An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition Second Edition by Daniel Jurafsky and James H. Martin

Where to study?

- Course Slides at http://disi.unitn.it/moschitti/teaching.html
- NLP-IR section:
  - Slides of IIR available at: http://informationretrieval.org

Teaching

Teaching by year

Year 2011-2012
Year 2010-2011
Year 2009-2010
Year 2008-2009
Year 2007-2008

Home
Department of Information and Communication Technology
iKernels

Accademic Year: 2011-2012

Informatica Generale

- Presentatione del corso
- Introduzione all'Informatica

...

Materiale aggiuntivo

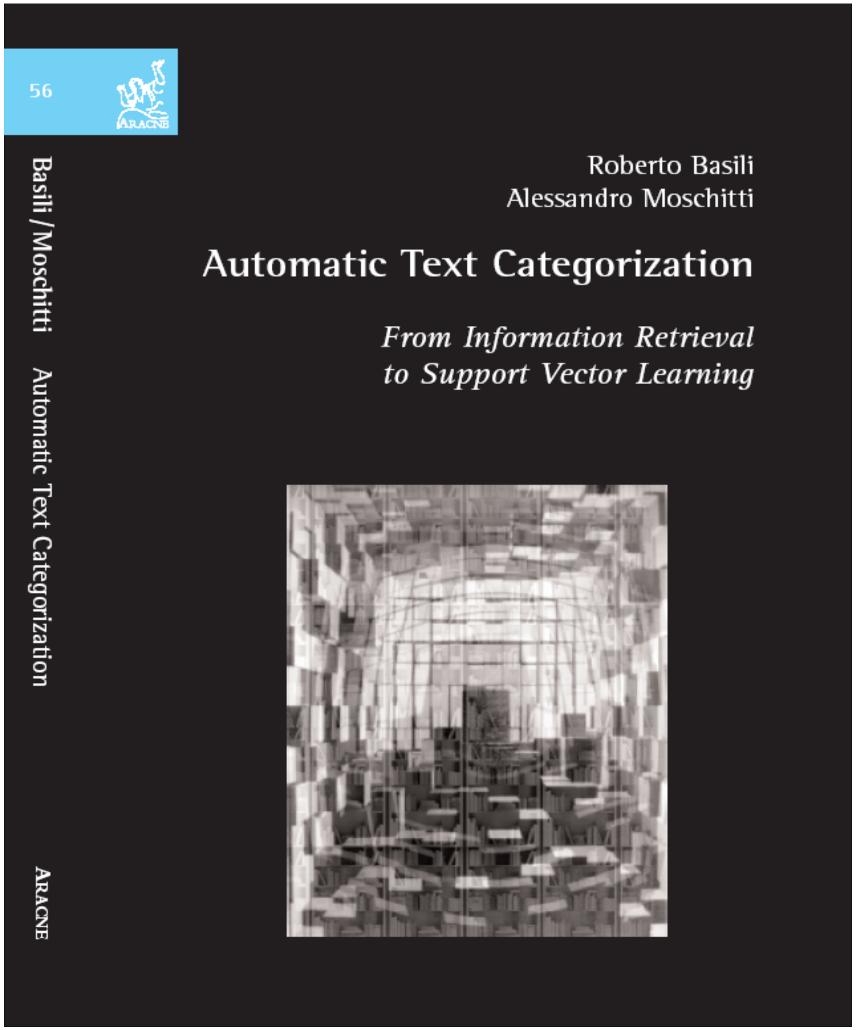
- Slides del corso (Prof. Bianchini)
- Altre slides recenti della Prof Bianchini

- Overflow
- Stack e Record di Attivazione
- Complessità Computazionale

Link alle lezioni di laboratorio

Natural Language Processing and Information Retrieval

Reference Book



Motivation

- Why NLP and IR?
- IR studies methods to search and retrieve information
  - Basic models based on words
  - Pretty much statistical-based
- NLP studies automatic approach to understand and language geneation
  - Use complex structures: syntax and semantics
  - Logic-based but nowadays pretty much statistical too

Motivation

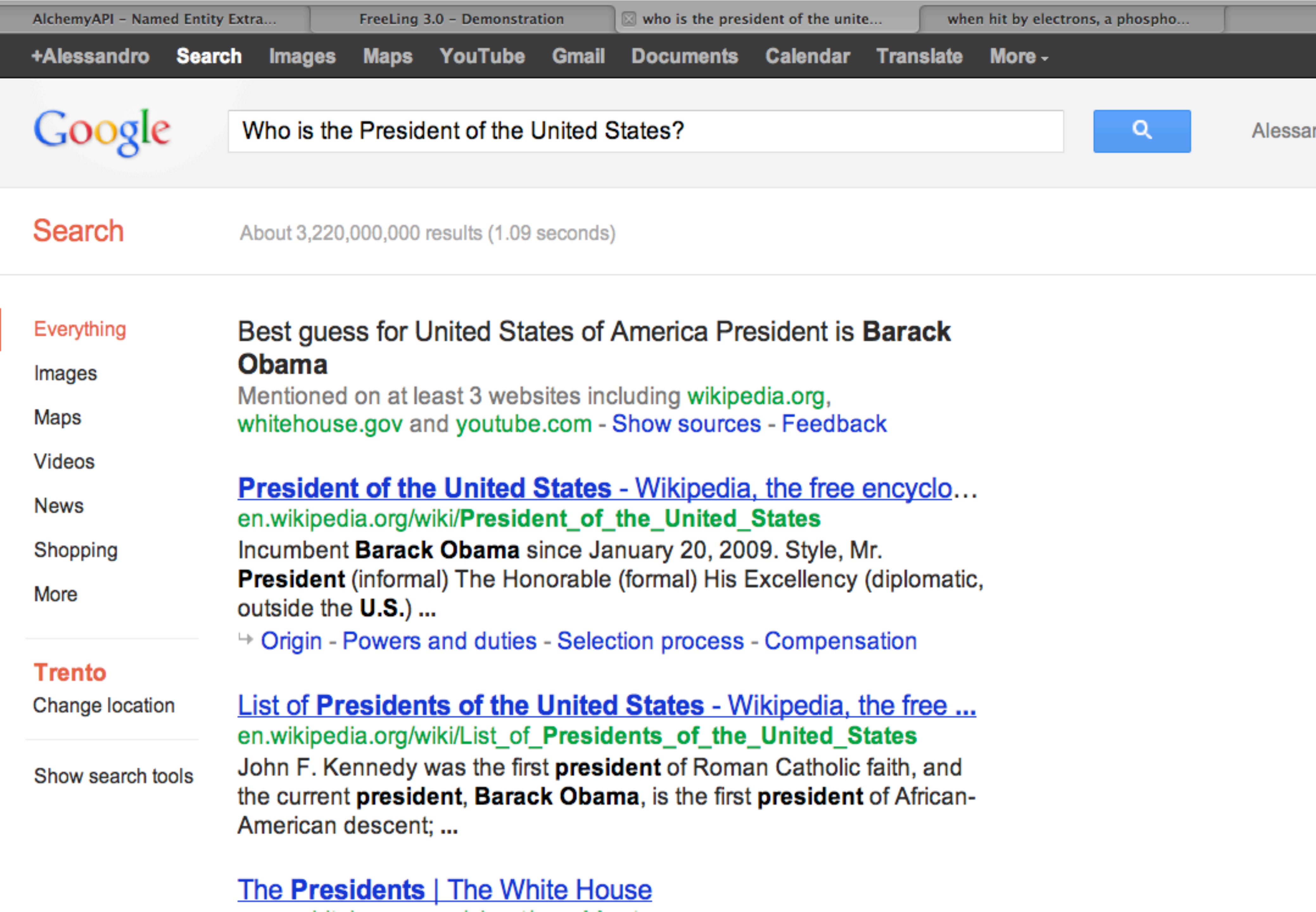
- IR very successful
  - Google Inc.
  - Altavista born in 1995
- NLP pretty much unsuccessful for company purposes
- Why using NLP?

Motivations

- Let us ask
  - Who is the President of the United States?

(Yes) The president of the United States is Barack Obama

(no) Glenn F. Tilton is President of the United Airlines


AlchemyAPI – Named Entity Extra...
FreeLing 3.0 – Demonstration
who is the president of the unite...
when hit by electrons, a phospho...
+Alessandro Search Images Maps YouTube Gmail Documents Calendar Translate More
Google
Who is the President of the United States?
Alessar
Search
About 3,220,000,000 results (1.09 seconds)
Everything
Images
Maps
Videos
News
Shopping
More
Trento
Change location
Show search tools
Best guess for United States of America President is Barack Obama
Mentioned on at least 3 websites including wikipedia.org, whitehouse.gov and youtube.com - Show sources - Feedback
President of the United States - Wikipedia, the free encyclo…
en.wikipedia.org/wiki/President_of_the_United_States
Incumbent Barack Obama since January 20, 2009. Style, Mr. President (informal) The Honorable (formal) His Excellency (diplomatic, outside the U.S.) ...
Origin - Powers and duties - Selection process - Compensation
List of Presidents of the United States - Wikipedia, the free ...
en.wikipedia.org/wiki/List_of_Presidents_of_the_United_States
John F. Kennedy was the first president of Roman Catholic faith, and the current president, Barack Obama, is the first president of African-American descent; ...
The Presidents | The White House

Motivations

- TREC has taught that this model is too weak
- Consider a more complex task, i.e., a Jeopardy! Quiz show question
- *When hit by electrons, a phosphor gives off electromagnetic energy in this form*
  - Solutions: ***photons/light***
- What are the most similar fragments retrieved by a search engine?

le

When hit by electrons, a phosphor gives off electromagnetic energy ✕

About 194,000 results (0.22 seconds) Advan

[Cathode-Ray Tube - body, used, chemical, characteristics, **form** ...]

Sep 6, 2010 ... In order to **form** the **electron** beam into the correct shape, ... The actual conversion of electrical **energy** to light **energy** takes place on the ... For example, the **phosphor** known as yttrium oxide **gives off** a red glow ... complete explanation of electrostatic and **electromagnetic** focusing in the crt ...

www.scienceclarified.com › Ca-Ch - Cached - Similar

[Beta particle - Wikipedia, the free encyclopedia]

Beta particles are high-**energy**, high-speed **electrons** or positrons emitted by certain ... The beta particles emitted are a **form** of ionizing radiation also known as beta rays. ... by **electromagnetic** interactions and may **give off** bremsstrahlung x-rays. ... The well-known 'betalight' contains tritium and a **phosphor**. ...

en.wikipedia.org/wiki/Beta_particle - Cached - Similar

[luminescence: Definition from Answers.com]

Included on the **electromagnetic** spectrum are radio waves and microwaves; ... Though the Sun sends its **energy** to Earth in the **form** of light and heat from the .... Thanks to the **phosphor**, a fluorescent lamp **gives off** much more light than an ... The tube itself is coated

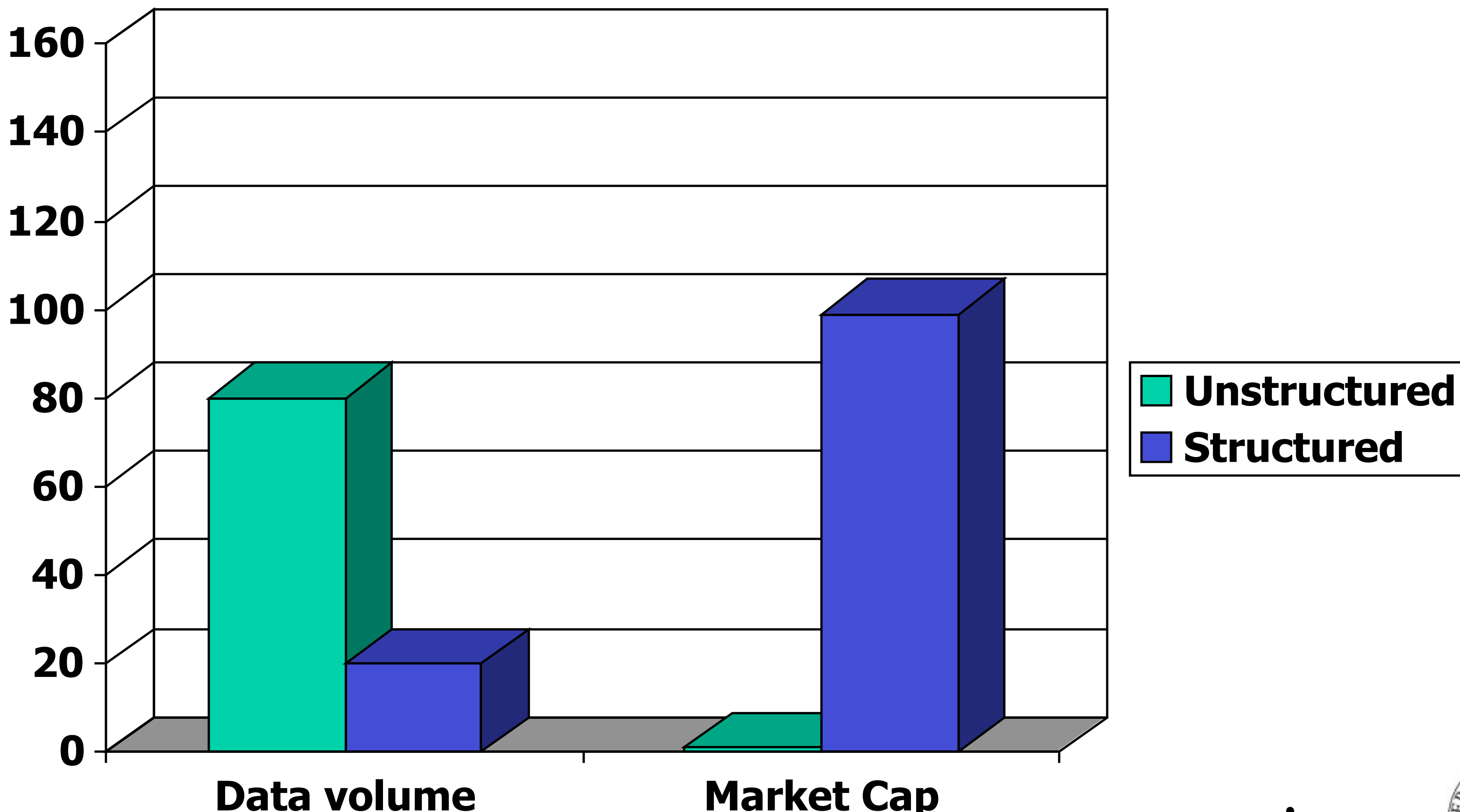
Motivations

- This shows that:
  - Word matching is not enough
  - Structure is required
- What kind of structures do we need?
- How to carry out structural similarity?
  - Still not complete solved problem but…

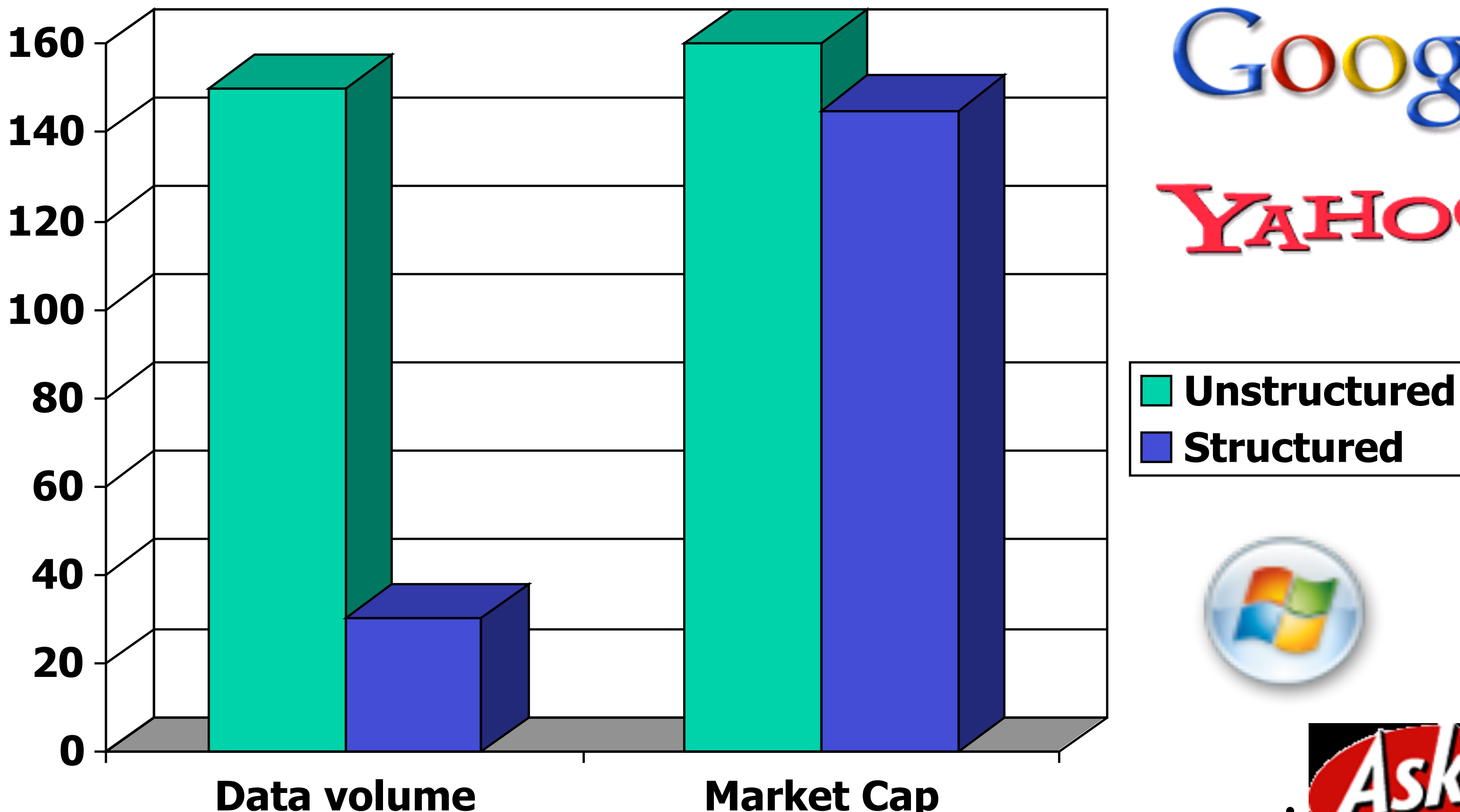
Today

- Basic Concept of Search Engines

Unstructured (text) vs. structured (database) data in 1996



Unstructured (text) vs. structured (database) data in 2006



Unstructured data in 1650

- Which plays of Shakespeare contain the words ***Brutus*** *AND* ***Caesar*** but *NOT* ***Calpurnia***?
- One could grep all of Shakespeare's plays for ***Brutus*** and ***Caesar,*** then strip out lines containing ***Calpurnia***?
  - Slow (for large corpora)
  - *NOT* ***Calpurnia*** is non-trivial
  - Other operations (e.g., find the word ***Romans*** near ***countrymen***) not feasible
  - Ranked retrieval (best documents to return)

Term-document incidence

	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

1 if play contains word, 0 otherwise

Brutus* AND *Caesar* but NOT *Calpurnia

Incidence vectors

- So we have a 0/1 vector for each term.
- To answer query: take the vectors for ***Brutus, Caesar*** and ***Calpurnia*** (complemented) ➨ bitwise *AND*.
- 110100 *AND* 110111 *AND* 101111 = 100100.

Answers to query

- Antony and Cleopatra, Act III, Scene ii

Agrippa [Aside to DOMITIUS ENOBARBUS]: Why, Enobarbus,

When Antony found Julius ***Caesar*** dead,

He cried almost to roaring; and he wept

When at Philippi he found ***Brutus*** slain.

- Hamlet, Act III, Scene ii

Lord Polonius: I did enact Julius ***Caesar*** I was killed i' the

Capitol; ***Brutus*** killed me.

Bigger corpora

- Consider *N* = 1M documents, each with about 1K terms.
- Avg 6 bytes/term incl spaces/punctuation
  - 6GB of data in the documents.
- Say there are *m* = 500K *distinct* terms among these.

Can't build the matrix

- 500K x 1M matrix has half-a-trillion 0's and 1's.
- But it has no more than one billion 1's.
  - matrix is extremely sparse.

Why?

- What's a better representation?
  - We only record the 1 positions.

Inverted index

For each term *T*, we must store a list of all documents that contain *T*.

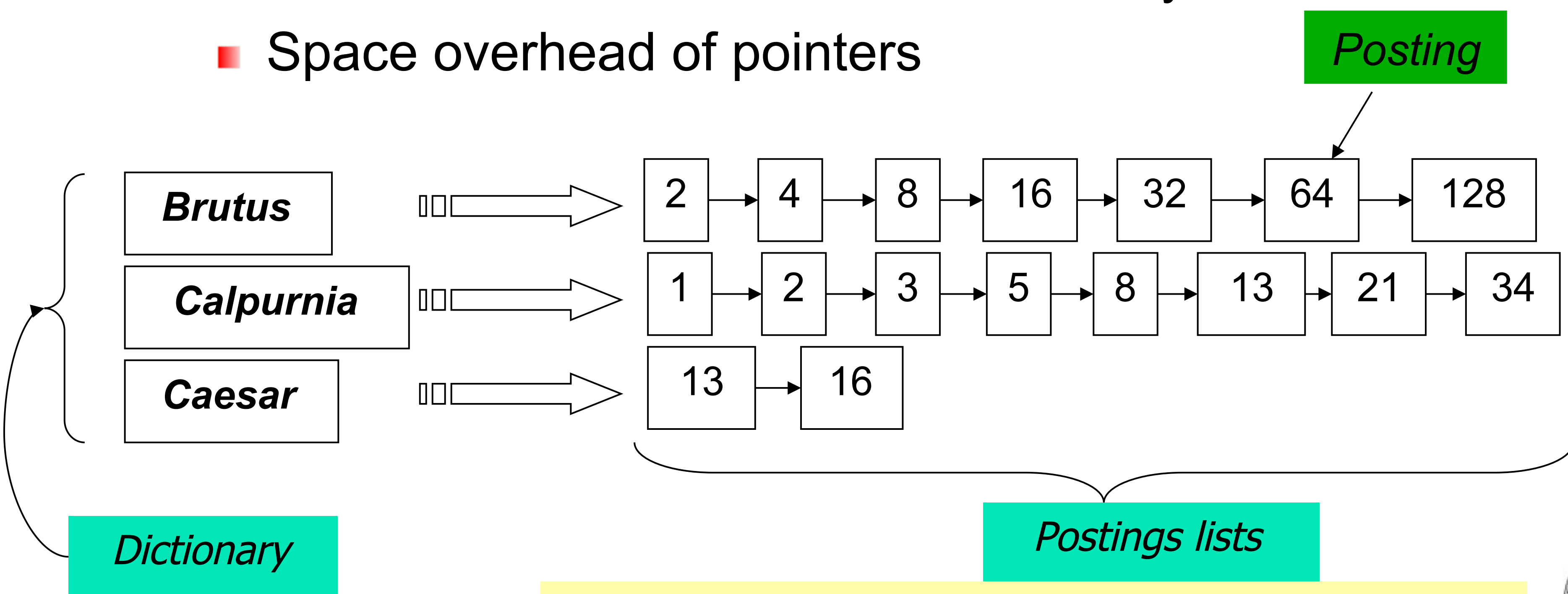
Do we use an array or a list for this?

Brutus	2	4	8	16	32	64	128	
Calpurnia	1	2	3	5	8	13	21	34
Caesar	13	16						

What happens if the word ***Caesar*** is added to document 14?

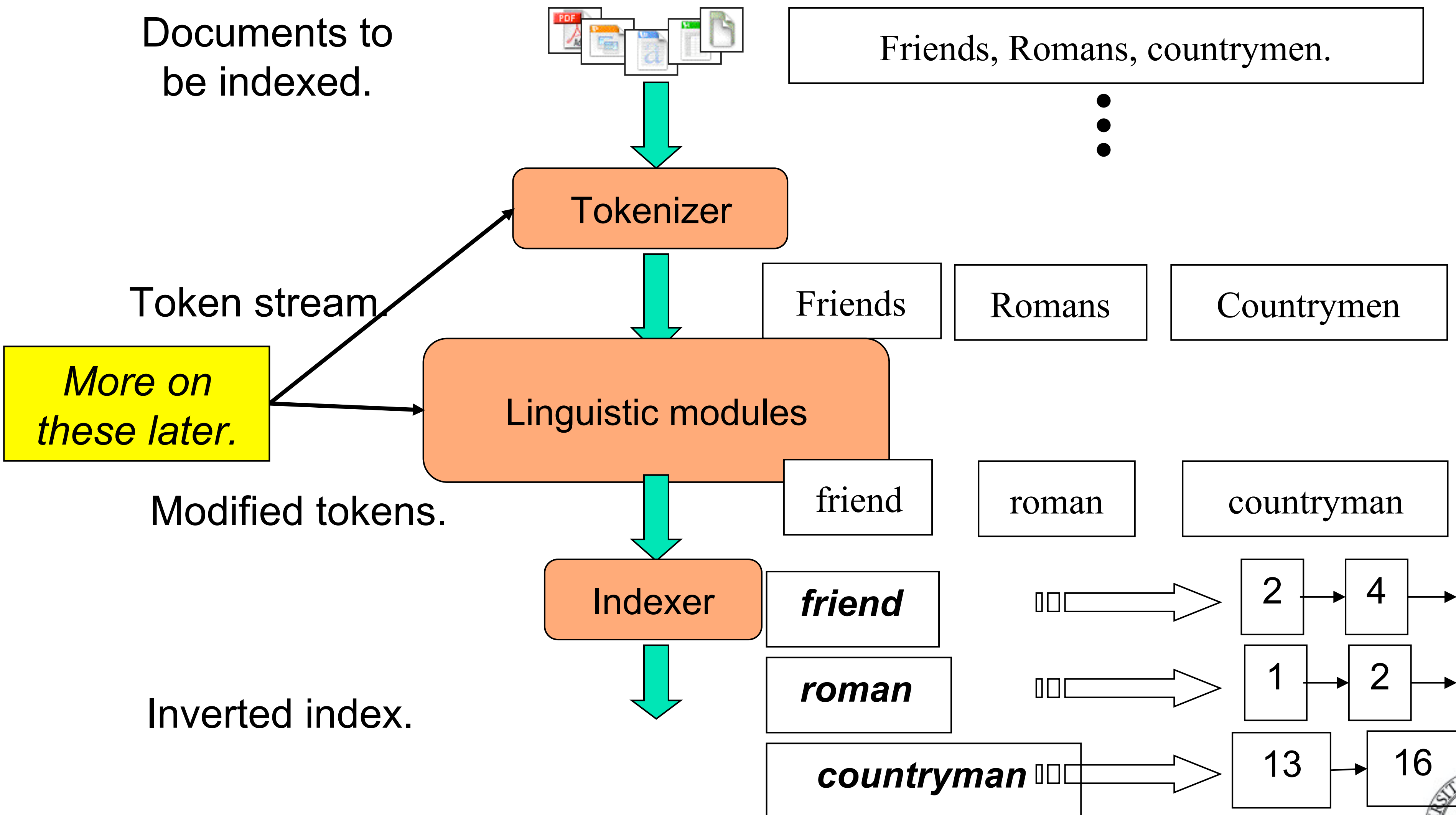
Inverted index

- Linked lists generally preferred to arrays
  - Dynamic space allocation
  - Insertion of terms into documents easy
  - Space overhead of pointers



Sorted by docID (more later on why).

Inverted index construction



Indexer steps

- Sequence of (Modified token, Document ID) pairs.

Doc 1

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

Doc 2

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

- Sort by terms.

Core indexing step.

Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

→

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Indexer steps: Dictionary & Postings

- Multiple term entries in a single document are merged.
- Split into Dictionary and Postings
- Doc. frequency information is added.

Why frequency?

Will discuss later.

Term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

→

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Where do we pay in storage?

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Terms and counts

Lists of docIDs

Pointers

Later in the course:
How do we index efficiently?
How much storage do we need?

The index we just built

- How do we process a query? ← Today’s focus
  - Later - what kinds of queries can we process?

Query processing: AND

Consider processing the query:

Brutus *AND* ***Caesar***

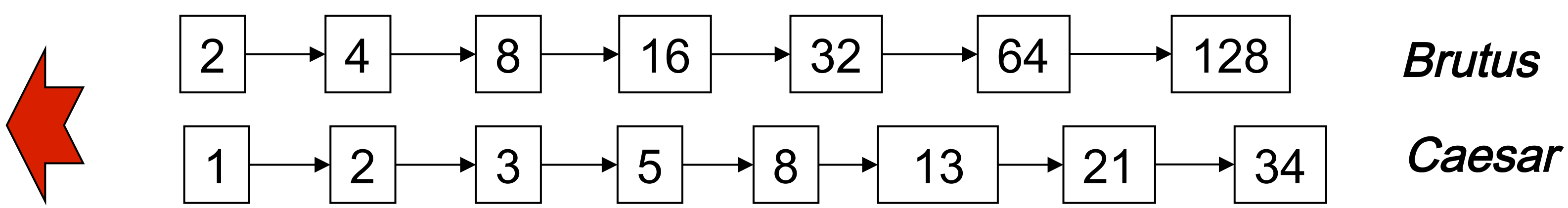
Locate ***Brutus*** in the Dictionary;

Retrieve its postings.

Locate *Caesar* in the Dictionary;

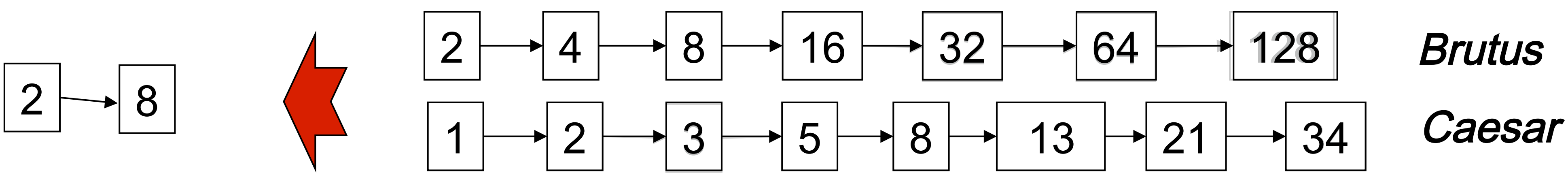
Retrieve its postings.

"Merge" the two postings:



The merge

Walk through the two postings simultaneously, in time linear in the total number of postings entries



If the list lengths are *x* and *y*, the merge takes O(*x*+*y*) operations.

Crucial: postings sorted by docID.

Intersecting two postings lists (a "merge" algorithm)

```
INTERSECT(p1, p2)
answer ← ⟨ ⟩
while p1 ≠ NIL and p2 ≠ NIL
do if docID(p1) = docID(p2)
     then ADD(answer, docID(p1))
          p1 ← next(p1)
          p2 ← next(p2)
     else if docID(p1) < docID(p2)
            then p1 ← next(p1)
            else p2 ← next(p2)
return answer
```

Boolean queries: Exact match

- The Boolean Retrieval model is being able to ask a query that is a Boolean expression:
  - Boolean Queries are queries using *AND, OR* and *NOT* to join query terms
    - Views each document as a set of words
    - Is precise: document matches condition or not.
- Primary commercial retrieval tool for 3 decades.
- Professional searchers (e.g., lawyers) still like Boolean queries:
  - You know exactly what you're getting.

Example: WestLaw http://www.westlaw.com/

- Largest commercial (paying subscribers) legal search service (started 1975; ranking added 1992)
- Tens of terabytes of data; 700,000 users
- Majority of users *still* use boolean queries
- Example query:
  - What is the statute of limitations in cases involving the federal tort claims act?
  - LIMIT! /3 STATUTE ACTION /S FEDERAL /2 TORT /3 CLAIM
- /3 = within 3 words, /S = in same sentence

Example: WestLaw http://www.westlaw.com/

- Another example query:
  - Requirements for disabled people to be able to access a workplace
  - disabl! /p access! /s work-site work-place (employment /3 place
- Note that SPACE is disjunction, not conjunction!
- Long, precise queries; proximity operators; incrementally developed; not like web search
- Professional searchers often like Boolean search:
  - Precision, transparency and control
- But that doesn't mean they actually work better….

Boolean queries: More general merges

- Exercise: Adapt the merge for the queries:

  Brutus *AND NOT* ***Caesar***

  Brutus *OR NOT* ***Caesar***

Can we still run through the merge in time O(x+y) or what can we achieve?

Merging

What about an arbitrary Boolean formula?

(Brutus *OR* ***Caesar)*** *AND NOT*

(Antony *OR* ***Cleopatra)***

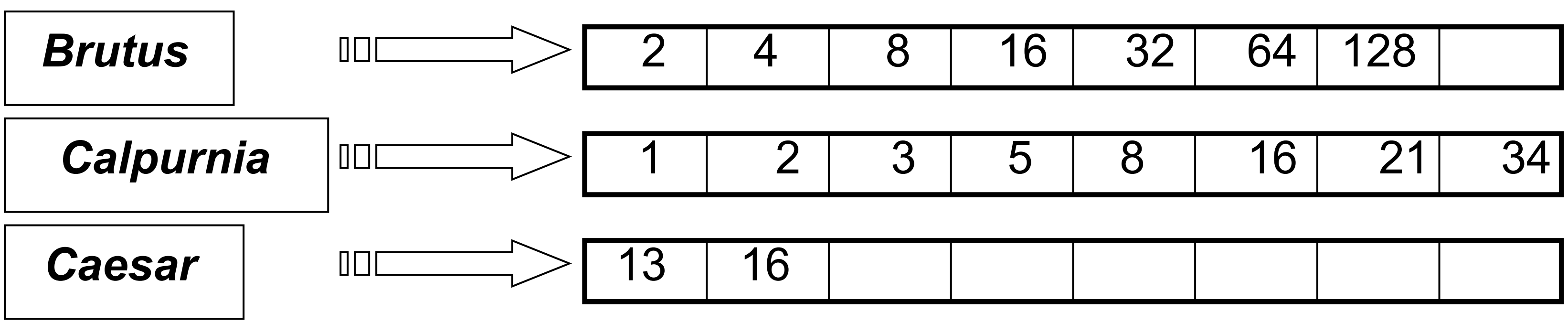
- Can we always merge in “linear” time?
  - Linear in what?
- Can we do better?

Query optimization

What is the best order for query processing?

Consider a query that is an *AND* of *t* terms.

For each of the *t* terms, get its postings, then *AND* them together.



Query: ***Brutus*** *AND* ***Calpurnia*** *AND* ***Caesar***

Query optimization example

- Process in order of increasing freq:
  - *start with smallest set, then keep cutting further.*

This is why we kept freq in dictionary

Brutus	2	4	8	16	32	64	128	
Calpurnia	1	2	3	5	8	13	21	34
Caesar	13	16						

Execute the query as (***Caesar*** *AND* ***Brutus)*** *AND* ***Calpurnia***.

More general optimization

- e.g., *(**madding** OR **crowd**) AND (**ignoble** OR **strife**)*
- Get freq's for all terms.
- Estimate the size of each *OR* by the sum of its freq's (conservative).
- Process in increasing order of *OR* sizes.

Exercise

Recommend a query processing order for

(tangerine OR trees) AND

(marmalade OR skies) AND

(kaleidoscope OR eyes)

Term	Freq
eyes	213312
kaleidoscope	87009
marmalade	107913
skies	271658
tangerine	46653
trees	316812

Query processing exercises

- If the query is ***friends*** *AND* ***romans*** *AND (NOT* ***countrymen****)*, how could we use the freq of ***countrymen***?
- Exercise: Extend the merge to an arbitrary Boolean query. Can we always guarantee execution in time linear in the total postings size?
- Hint: Begin with the case of a Boolean *formula* query: in this, each query term appears only once in the query.

Exercise

- Try the search feature at http://www.rhymezone.com/shakespeare/
- Write down five search features you think it could do better

What's ahead in IR? Beyond term search

- What about phrases?
  - ***Stanford University***
- Proximity: Find ***Gates*** *NEAR* ***Microsoft***.
  - Need index to capture position information in docs.
- Zones in documents: Find documents with (*author* = ***Ullman***) *AND* (text contains ***automata***).

Evidence accumulation

- 1 vs. 0 occurrence of a search term
  - 2 vs. 1 occurrence
  - 3 vs. 2 occurrences, etc.
  - Usually more seems better
- Need term frequency information in docs

Ranking search results

- Boolean queries give inclusion or exclusion of docs.
- Often we want to rank/group results
  - Need to measure proximity from query to each doc.
  - Need to decide whether docs presented to user are singletons, or a group of docs covering various aspects of the query.

IR vs. databases: Structured vs unstructured data

- Structured data tends to refer to information in "tables"

Employee	Manager	Salary
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

Typically allows numerical range and exact match

(for text) queries, e.g.,

Salary < 60000 AND Manager = Smith.

Unstructured data

- Typically refers to free-form text
- Allows
  - Keyword queries including operators
  - More sophisticated “concept” queries, e.g.,
    - find all web pages dealing with *drug abuse*
- Classic model for searching text documents

Semi-structured data

- In fact almost no data is "unstructured"
- E.g., this slide has distinctly identified zones such as the *Title* and *Bullets*
- Facilitates "semi-structured" search such as
  - *Title* contains data AND *Bullets* contain search

… to say nothing of linguistic structure

More sophisticated semi-structured search

- *Title* is about Object Oriented Programming AND *Author* something like stro*rup
- where * is the wild-card operator
- Issues:
  - how do you process "about"?
  - how do you rank results?
- The focus of XML search (*IIR* chapter 10)

Clustering, classification and ranking

- Clustering: Given a set of docs, group them into clusters based on their contents.
- Classification: Given a set of topics, plus a new doc *D*, decide which topic(s) *D* belongs to.

- Ranking: Can we learn how to best order a set of documents, e.g., a set of search results

The web and its challenges

- Unusual and diverse documents
- Unusual and diverse users, queries, information needs
- Beyond terms, exploit ideas from social networks
  - link analysis, clickstreams ...

- How do search engines work?
  And how can we make them better?

More sophisticated *information* retrieval

- Cross-language information retrieval
- Question answering
- Summarization
- Text mining
- …

Vector Spaces

Definition (1)

- A set V is a **vector space** over a field F (for example, the field of real or of complex numbers) if, given
- an operation *vector **addition*** defined in V, denoted $\mathbf{v} + \mathbf{w}$ (where $\mathbf{v}, \mathbf{w} \in V$), and
- an operation, *scalar **multiplication*** in V, denoted $a * \mathbf{v}$ (where $\mathbf{v} \in V$ and $a \in F$),
- the following properties hold for all $a, b \in F$ and $\mathbf{u}$, $\mathbf{v}$, and $\mathbf{w} \in V$:
- $\mathbf{v} + \mathbf{w}$ belongs to V.
  (Closure of V under vector addition)
- $\mathbf{u} + (\mathbf{v} + \mathbf{w}) = (\mathbf{u} + \mathbf{v}) + \mathbf{w}$
  (Associativity of vector addition in V)
- There exists a neutral element $\mathbf{0}$ in V, such that for all elements $\mathbf{v}$ in V, $\mathbf{v} + \mathbf{0} = \mathbf{v}$
  (Existence of an additive identity element in V)

Definition (2)

- For all **v** in V, there exists an element **w** in V, such that **v** + **w** = **0**
(Existence of additive inverses in V)
- **v** + **w** = **w** + **v**
(Commutativity of vector addition in V)
- *a* * **v** belongs to V
(Closure of V under scalar multiplication)
- *a* * (*b* * **v**) = (*ab*) * **v**
(Associativity of scalar multiplication in V)
- If 1 denotes the multiplicative identity of the field F, then 1 * **v** = **v**
(Neutrality of one)
- *a* * (**v** + **w**) = *a* * **v** + *a* * **w**
(Distributivity with respect to vector addition.)
- (*a* + *b*) * **v** = *a* * **v** + *b* * **v**
(Distributivity with respect to field addition.)

An example of Vector Space

- For all *n*, $\mathbf{R}^n$ forms a vector space over **R**, with component-wise operations.
- Let **V** be the set of all n-tuples, $[v_1,v_2,v_3,...,v_n]$ where v_i is a member of **R**={real numbers}
- Let the field be **R**, as well
- Define Vector Addition:

  For all v, w, in **V**, define $v+w=[v_1+w_1,v_2+w_2,v_3+w_3,...,v_n+w_n]$
- Define Scalar Multiplication:

  For all a in **F** and v in **V**, $a*v=[a*v_1,a*v_2,a*v_3,...,a*v_n]$
- Then **V** is a Vector Space over **R**.

Linear dependency

- Linear combination:
- $\alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n = 0$ for some $\alpha_1 \ldots \alpha_n$ not all zero

  $\Rightarrow y = \alpha_1 \mathbf{v}_1 + \ldots + \alpha_n \mathbf{v}_n$ has a unique expression
- In case $\alpha_i > 0$ and the sum is 1 it is called convex combination

Normed Vector Spaces

- Given a vector space *V* over a field *K*, a norm on *V* is a function from *V* to **R**,
- it associates each vector $\mathbf{v}$ in *V* with a real number, $||\mathbf{v}||$
- The norm must satisfy the following conditions:
  - For all *a* in *K* and all $\mathbf{u}$ and $\mathbf{v}$ in *V*,

    1. $||\mathbf{v}|| \geq 0$ with equality if and only if $\mathbf{v} = \mathbf{0}$
    2. $||a\mathbf{v}|| = |a|\ ||\mathbf{v}||$
    3. $||\mathbf{u} + \mathbf{v}|| \leq ||\mathbf{u}|| + ||\mathbf{v}||$
- A useful consequence of the norm axioms is the inequality
  - $||\mathbf{u} \pm \mathbf{v}|| \geq |\ ||\mathbf{u}|| - ||\mathbf{v}||\ |$
- for all vectors $\mathbf{u}$ and $\mathbf{v}$

Inner Product Spaces

- Let V be a vector space and **u**, **v**, and **w** be vectors in V and c be a constant.
- Then, an *inner product* (,) on V is
  - a function with domain consisting of pairs of vectors and
  - range real numbers satisfying
  - the following properties:
    1. (**u**, **u**) $\geq$ 0 with equality if and only if **u** = **0**.
    2. (**u**, **v**) = (**v**, **u**)
    3. (**u** + **v**, **w**) = (**u**, **w**) + (**v**, **w**)
    4. (c**u**, **v**) = (**u**, c**v**) = c(**u**, **v**)

Example

- Let V be the vector space consisting of all continuous functions with the standard + and *. Then define an inner product by

$$(f,g)=\int_0^1 f(t)g(t)dt$$

- For example: $(x,x^2) = \int_0^1 (x)(x^2)dx = \frac{1}{4}$

- The four properties follow immediately from the analogous property of the definite integral:

$$(f+g,h)=\int_0^1 (f+g)(t)h(t)\ dt$$

$$=\int_0^1 \left(f(t)h(t)+g(t)h(t)\right)\ dt=\int_0^1 f(t)h(t)\ dt+\int_0^1 g(t)h(t)\ dt$$

$$=(f,h)+(g,h)$$

Inner Product Properties

- $(\mathbf{v}, \mathbf{0}) = 0$
- $\| v \| = \sqrt{(v, v)}$
- If $(\mathbf{v}, \mathbf{u}) = 0$, $\mathbf{v}$,$\mathbf{u}$ are called orthogonal
- Schwarz Inequality:
  - $[(\mathbf{v}, \mathbf{u})]^2 \leq (\mathbf{v}, \mathbf{v})\,(\mathbf{u}, \mathbf{u})$
- The classical scalar product is the component-wise product
- $(x_1, x_2, \ldots, x_n)\,(y_1, y_2, \ldots, y_n) = x_1 y_1 + x_2 y_2 + \ldots + x_n y_n$
- $$\cos(u, v) = \frac{(u, v)}{\| u \| \cdot \| v \|}$$

Projection

- From $\cos(\vec{x},\vec{w}) = \dfrac{\vec{x}\cdot\vec{w}}{\|\vec{x}\|\cdot\|\vec{w}\|}$

- It follows that

$$\|\vec{x}\|\cos(\vec{x},\vec{w}) = \frac{\vec{x}\cdot\vec{w}}{\|\vec{w}\|} = \vec{x}\cdot\frac{\vec{w}}{\|\vec{w}\|}$$

- Norm of $\vec{x}$ times the cosine between $\vec{x}$ and $\vec{w}$, i.e. the projection of $\vec{x}$ on $\vec{w}$

Similarity Metrics

- The simplest distance for continuous *m*-dimensional instance space is *Euclidian distance*.
- The simplest distance for *m*-dimensional binary instance space is *Hamming distance* (number of feature values that differ).
- Cosine similarity is typically the most effective

The Vector Space Model (VSM)

d_1: Politic

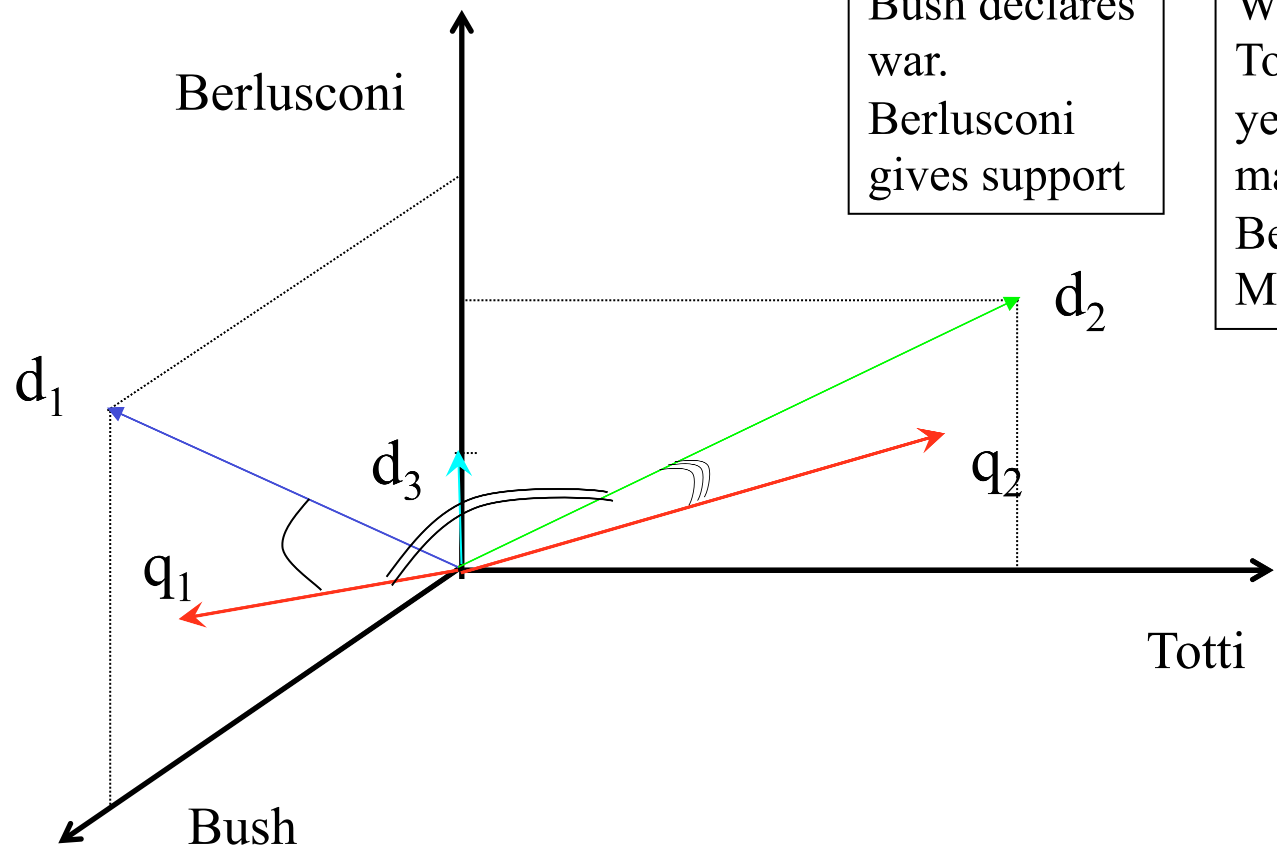
Bush declares war. Berlusconi gives support

d_2: Sport

Wonderful Totti in the yesterday match against Berlusconi's Milan

d_3:Economic

Berlusconi acquires Ibrahimović before elections



q_1: Berlusconi supports

q_2 : Totti match

Summary of VSM

- VSM (Salton89')
  - Features are dimensions of a Vector Space **Linear Kernel**
  - Documents and Queries are vectors of feature weights.
  - *d* is retrirved for q if $\vec{d} \cdot \vec{q} > th$