06 – Web 1.0 (part 2)

Search engines - Indexing

- Documents are indexed according to the (potentially stemmed) terms they contain
- Term position allow to have a finer grain description of the documents (and allow phrase searches)
- An *inverted index* is used to get list of documents matching specific terms

tid	did	pos
my	1	1
care	1	2
is	1	3
⋮		
new	2	8
care	2	9
won	2	10

Searching the index – A naive approach

1. Give me all the documents containing the word “java”
2. Give me all the documents containing the word “java” but not the word “coffee”
3. Give me all the documents containing “java beans” or “api”

tid	did	pos
my	1	1
care	1	2
is	1	3
⋮		
new	2	8
care	2	9
won	2	10

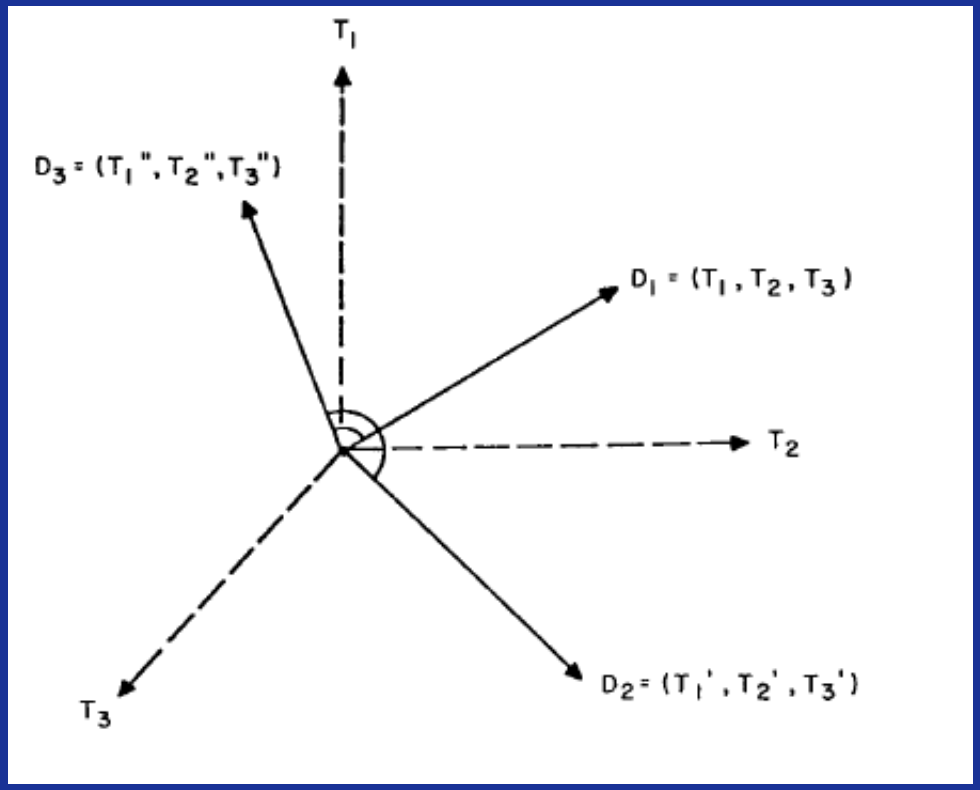
1.
```
select did from POSTING where tid = 'java'
```

2.
```
(select did from POSTING where tid = 'java') except (select did
from POSTING where tid = 'coffee')
```

3.
```
with
  D_JAVA (did, pos) as (select did, pos from POSTING where tid = 'java'),
  D_BEANS(did, pos) as (select did, pos from POSTING where tid = 'beans'),
  D_JAVABEANS(did) as
    (select D_JAVA.did from D_JAVA, D_BEANS
      where D_JAVA.did = D_BEANS.did
      and D_JAVA.pos + 1 = D_BEANS.pos),
  D_API(did) as (select did from POSTING where tid = 'api'),
(select did from D_JAVABEANS) union (select did from D_API)
```

"Real" search – the Vector Space Model

- Every document is represented by a *vector* in a *multidimensional space*
- Distances between documents (or the query and a document) are calculated in terms of *angles* between *vectors*

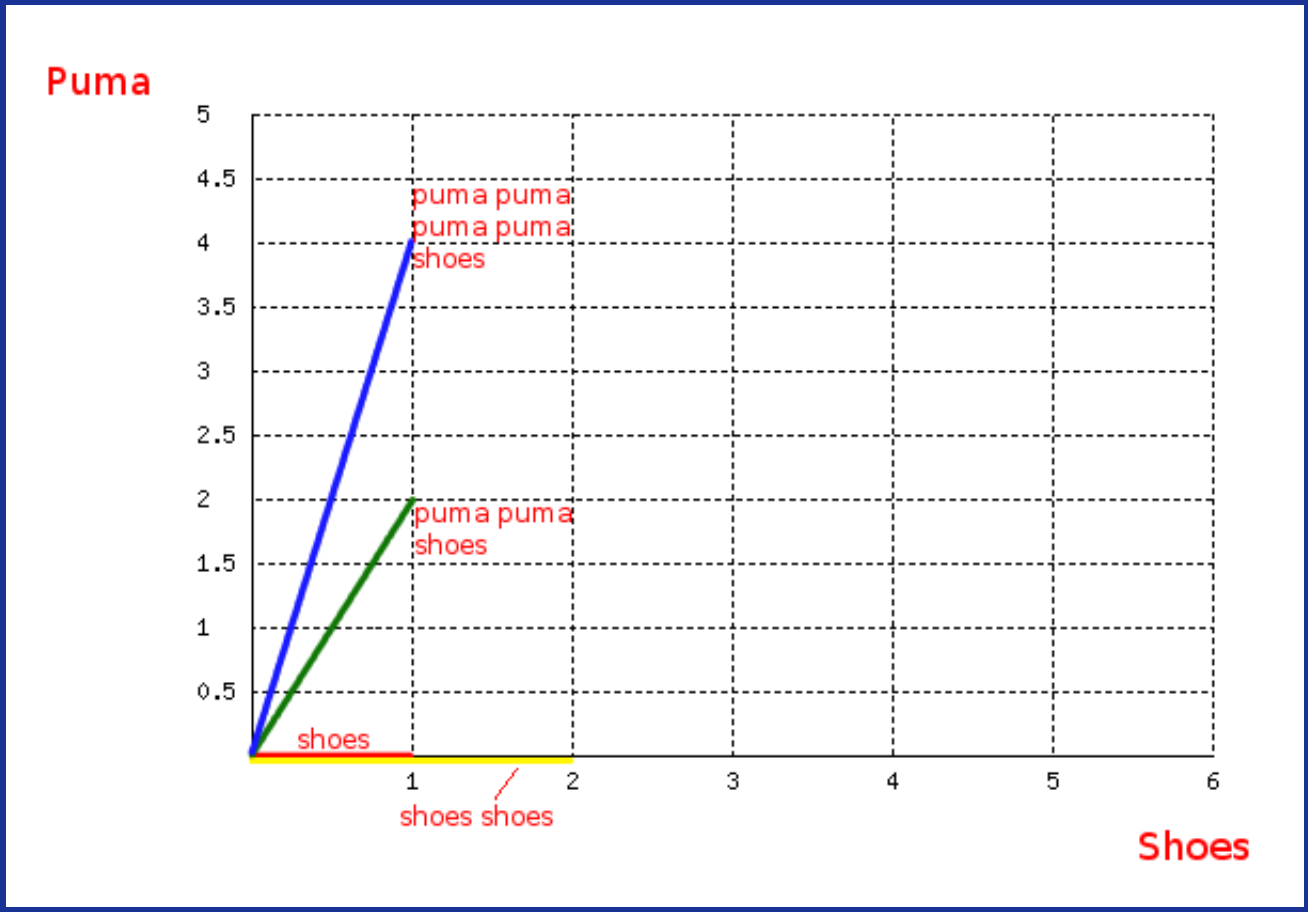


But... wait...

- What is a *vector*? What is a *MULTIDIMENSIONAL SPACE*???
- Def. Vector:
  - In computer science, it is equivalent to *array*
  - In mathematics, it is a geometric entity characterized by a *length* and a *direction*
    - A vector can be split into different components, one for each *dimension* of the space
- Def. Multidimensional (Euclidean) Space
  - A space with more than one dimension
  - … you already know 2d and 3d, I guess ;-)
- So... what?

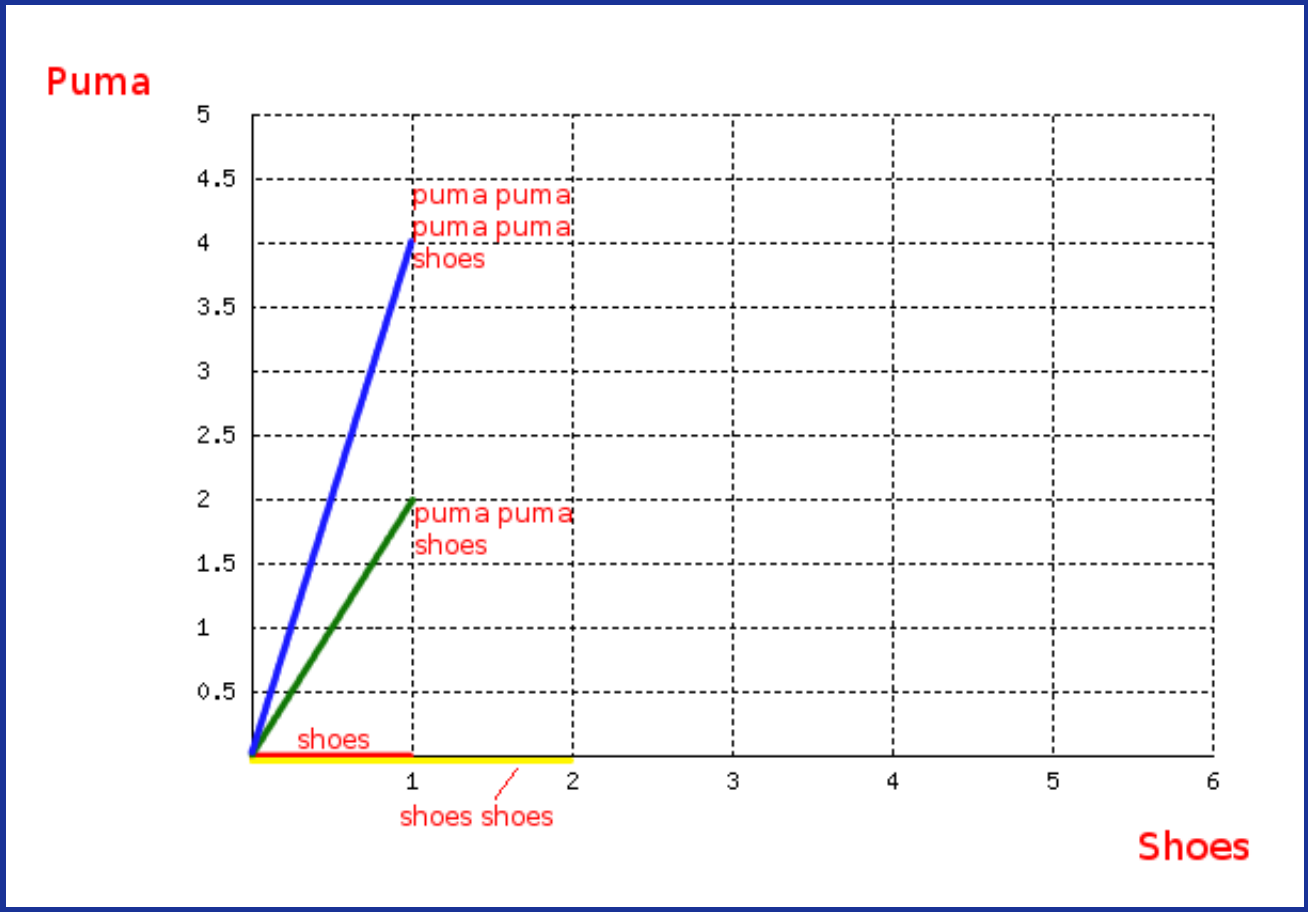
A bi-dimensional example

- Suppose you have four different documents, described according to how many times they contain the two words "shoes" and "puma"
- Note that all vectors start from the origin of the axes



A bi-dimensional example

- The more one document contains one of the two words, the more its vector "rotates" in the direction of the matching axis
- The distance between two documents is the *angle* formed by their vectors!



What about the query?

- The query itself can be considered like a (very short) document and transformed into a vector (i.e. “puma shoes” would be a vector starting at (0,0) and pointing at (1,1)).
- The distance (in terms of angle) between the query vector and the document vector matches how “good” a document is as a result for that query
- Documents can then be *ranked* according to their distances wrt the query
- Is this all?
  - Of course not (there are lots of other algorithms that are used for ranking)
  - … but it's a good start (and the vector space model will be used a lot for other applications – check the following lessons!)

Static vs. Dynamic Web

Static vs Dynamic Web

- Web pages can be classified in two main groups:
  - Static Web pages (delivered to the client exactly as they are stored on the server)
  - Dynamic Web pages (*generated* by a Web application)

- The applications that generate Web pages are *programs* whose *output* is an *HTML* document
  - As a comparison, think about the difference between a text file and a program printing a text file on the screen

How dynamic is “dynamic”?

- A dynamic page might change according to
  - Context
    - i.e. Current time
  - Input data
    - i.e. User profile, configuration parameters, DB contents, user interactions
- The dynamic part can be
  - On the client side
    - Changes happen *within* the loaded page
    - i.e. Javascript+DHTML /Actionscript+Flash
  - On the server side
    - Changes happen *between* one page and another
    - i.e. Perl, PHP, ASP, Python, Ruby
- Of course, client and server side can be used simultaneously

Client- vs. server-side dynamic pages

- See attached ODS file

The deep Web

- The Deep (hidden) Web is the part of Internet which is not indexed by search engines
- The Deep Web is **much bigger** than the so-called "surface Web"

Deep Web resources

- There are different reasons why some pages cannot be indexed by a search engine
  - They are not linked by any other page
  - They are dynamically generated (according to context or parameters)
  - Their access is limited
    - robot exclusions or captchas
    - private Web (requiring user authentication)
  - Page content or links to other pages are generated on-the-fly by scripts that run on the client
  - Some file formats are not handled by search engines

Robots.txt and Sitemaps

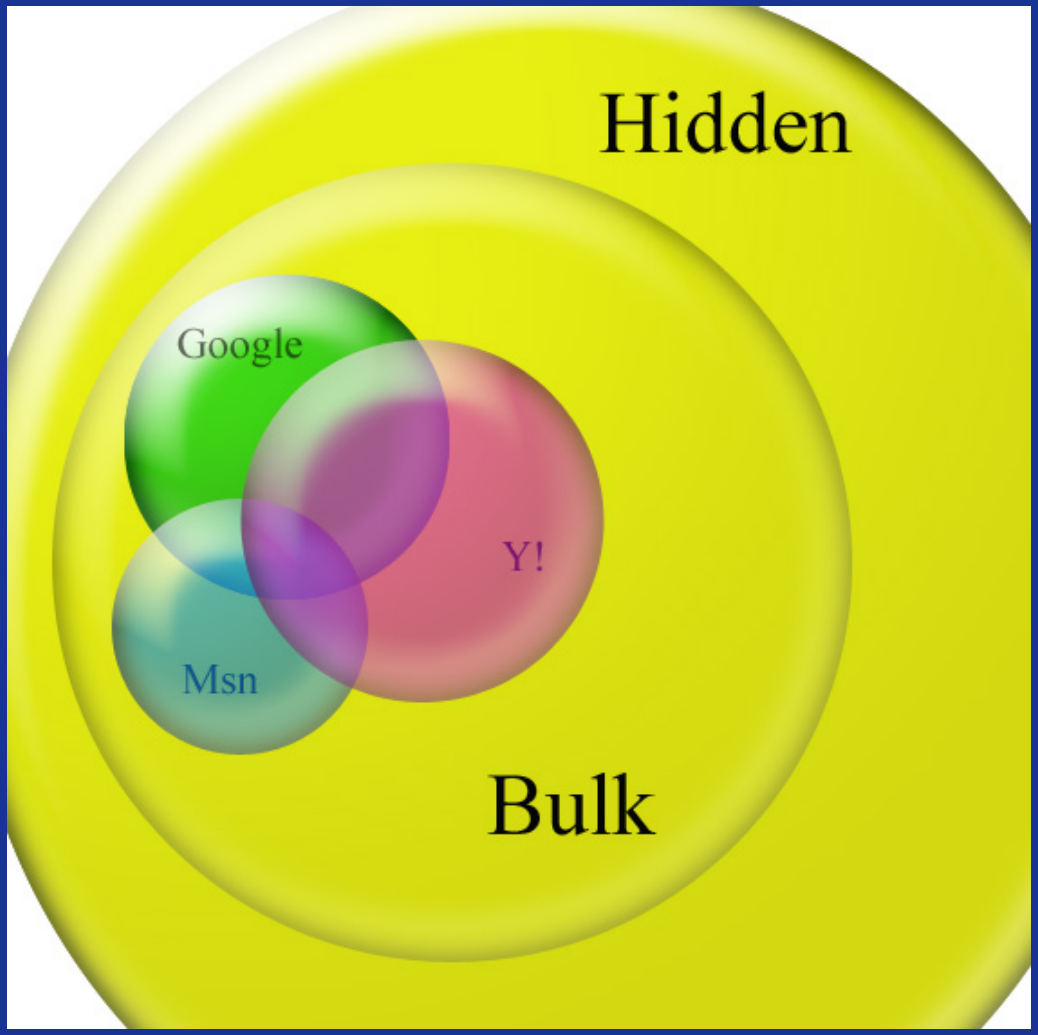
- Both are files that are used to instruct s.e. robots about which URLs they should index or not
  - robots.txt is a text file specifying URLs to *exclude*

```
User-agent: Google
Disallow:

User-agent: *
Disallow: /
```

  - sitemaps are xml files specifying URLs to *include* (check here for an example)

Search Engines coverage



References

- Some Web references:
  -