Understanding how search engines work,
an Information Retrieval perspective

Pedro Contreras
pedro@cs.rhul.ac.uk

Department of Computer Science
Royal Holloway, University of London

26 February 2008

Outline

The enterprise search problem

Data preparation - Parsing

- Tokenisation
- Stop words removal
- Stemming
- Entity detection
- Part of speech

Data storage - Indexing

- Index construction

Exploitation - Querying

- Exploiting data repositories

Clustering - Baire, or longest common prefix

Questions

Enterprise Search

Enterprise search characteristics:

- Diversity of content sources and formats, and not necessarily HTTP based
- Secure access
- Combined structured and unstructured search
- Dedicated search (e.g. email search)
- Intranet search
- Ranking and categorisation problem
- Social forces behind the creation of Internet and Intranet content are quite different
- Deployment environments for these domains also differs

Enterprise Search comparison with Web search

<u>Similarities</u>

- Crawling
- Indexing and ranking
- User interface

<u>Differences</u>

- Diverse repositories
- Fewer documents
- Access control
- Diverse doc type
- Different user needs
- Less filtering

Data preparation and data source

Different document types requiered different access methods. PDF, MS Word, HTML, Open formats, email, etc.

Good Formatted file

```
docno="lists-046-11826843"
name="Ron Whitney"
email="RFW@math.ams.org"
sent="Sat, 20 Apr 1996 17:44:20 -0400 (EDT)"
inreplyto="199604201303.6479@uvea.wolfram.com"
* Mail-System-Version:
id="830036660.25042.RFW@MATH.AMS.ORG"
subject="Re: conference call Mon 22 April"
To: w3c-math-erb@w3.org


I was unable to attend last week, but am available this
Monday. If the group plans to meet, I'll attend, but I
have no specificitems for the agenda now.

-Ron
```

Bad formatted file

```
name=""
email="mail137@163.com"
sent="Wed, 8 May 2002 14:39:51 -0400"
subject="øÏÀŸÕΔπ„ƒ˙µƒÕ¯æº∞≤˙∆∑"
To: web-access@w3.org

ª•¡™Õ¯∑¢'πµΩΩÒÃÏ“—æ±ª¥Ûð‡µƒ…Ã º“À¯¿˚”√£¯, «ÕΔπ„…
Ã º“≤˙∆∑≥…±æ◊ÓµÕ”÷”—βµƒ”™œ¯
≤fl¬'°£µ´”–µƒ»ÀÀ‰»ª Ω®¡¢µƒÕ¯æ£¯µ´»¥√ª∞—Õ¯æÕΔπ„≥ˆ•
£ª”–µƒ»ÀÃÏÃÏð°…œÕ¯Σ¯µ´»¥√ª∞—◊
'º∫
µƒ≤˙Δ∑∑¢≤ºµΩÕ¯…œ»•°£œ÷'/£¯Œ“√«Õ®π'ª ª–©◊®“µµƒ»|º¸
£¯ƒ‹—⁄ÀŸµƒΩ´ƒ˙µƒÕ¯æº∞≤˙∆∑ÕΔπ„≥
ˆ»•£°
---Œ“√«µƒ∑ˇŒÒ---

1°¢√,∑—∑ˇŒÒ£∫
 √,∑—Ã·π©πÿ”ª•¡™Õ¯”™œ¯∑Ω∑®µƒ∞ ı◊…—Ø
```


Parsing, tokenisation

Tokenisation is the process of splitting a stream of words into units or "tokens". Normally this process does not included the following symbols:

- period (.)
- comma (,)
- semicolon (;)
- quotation marks (")
- colon (:)
- brackets [ ]
- braces { }
- parentheses ( )
- mathematical operators + - / * = < >
- special characters | & ~
- the at sign @
- underscores and other rare characters

e.g

The "brown" fox jumps, quickly over the lazy dog*.

The brown fox jumps quickly over the lazy dog

Parsing, stop words removal

Stop Word is the name given to a word that will be filtered and is not consider relevant by an information retrieval system. Some of the more frequently used stop words for English include: *"a", "of", "the", "I", "it", "you", and "and"*. These are generally regarded as functional words that do not carry meaning for the system.

Stop List is the list or set of *stop words*, there are as many *stop lists* as there are languages. I.e. if a system processes text that includes English, French, German and Spanish it also should be a *stop list* for each of these languages.

Parsing, Stemming

Stemming is the process to obtain a word's root (also called normalisation) through eliminating suffixes.

running ran laughing laughs laughed languished seals	→	run ran laugh laugh laugh languish seal	connection connections connective connected connecting	→ connect

Benefits of Stemming

- Takes care of morphological variants
- Reduces index size

Known Limitations

- Impact on advanced syntax and exact match
- Accented characters are not supported
- Short words are not stemmed

more info see: the porter stemming algorithm, http://tartarus.org/~martin/PorterStemmer, http://snowball.tartarus.org/

Parsing, entity detection

The problem here is to find various structured data within unstructured documents, e.g.

- people's names
- project's names
- places
- amounts

Algorithms for entity detection normally are either rule or statistical based.

see: Special Interest Group on Natural Language Learning on the Association for Computational Linguistics (CoNLL). http://cnts.uia.ac.be/signll/conll.html

Parsing, part of speech

Part of speech is a categorisation process that take in consideration phrase function. Each part of speech explains not what the word is, but how the word is used. POS is very useful when dealing with Natural Language Processing in IR.

Parts of speech: the *verb*, the *noun*, the *pronoun*, the *adjective*, the *adverb*, the *preposition*, the *conjunction*, and the *article*.

e.g.

can	I think I can do it.	verb
can	Don't open that can of beans.	noun
only	This is my only pen.	adjective
only	He was only joking.	adverb
his	That book is his.	pronoun
his	That is his book.	adjective
English	Can you speak English?	noun
English	I am reading an English novel.	adjective

Data storage: indexing

After parsing

- Tokenisation
- Stop words removal
- Stemming
- Entity detection
- Part of speech

we can start the indexation process, which consist in storing the tokens in a DB, usually in a vector space fashion

Index design factors

- Merge factors
- Storage techniques
- Index size (compression)
- Lookup speed
- Maintenance
- Fault tolerance
- Scalability

Index construction, inverted index

T1: bab(y, ies, y's)
T2: child(ren's)
T3: guide
T4: health
T5: home
T6: infant
T7: safety
T8: toddler

D1: Infant & Toddler First Aid
D2: Babies & Children's Rooms (For Your Home)
D3: Child Safety at Home
D4: Your Baby's Health and Safety: From Infant to Toddler
D5: Baby Proofing Basics
D6: Your Guide to Easy Rust Proofing
D7: Beanie Babies Collector's Guide

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Vector Space Model (VSM)

$$\mathbf{t}_i^T = \begin{bmatrix} x_{i,1} & \dots & x_{i,n} \end{bmatrix} \qquad \mathbf{d}_j = \begin{bmatrix} x_{1,j} \\ \vdots \\ x_{m,j} \end{bmatrix}$$

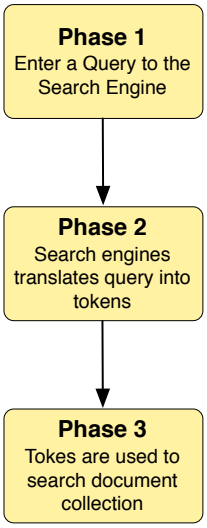
$$\mathbf{t}_i^T \rightarrow \begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \vdots & \ddots & \vdots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \quad \mathbf{d}_j \downarrow$$

Binary	$\chi(f_{ij})$
Logarithmic	$log(1 + f_{ij})$
Normal	$1/\sqrt{\sum_j f_{ij}^2}$
Inverse Document Frequency	$log(n/\sum_j \chi(f_{ij})$
etc.	.....

Where:

f_{ij} : number of times term *i* in document *j*

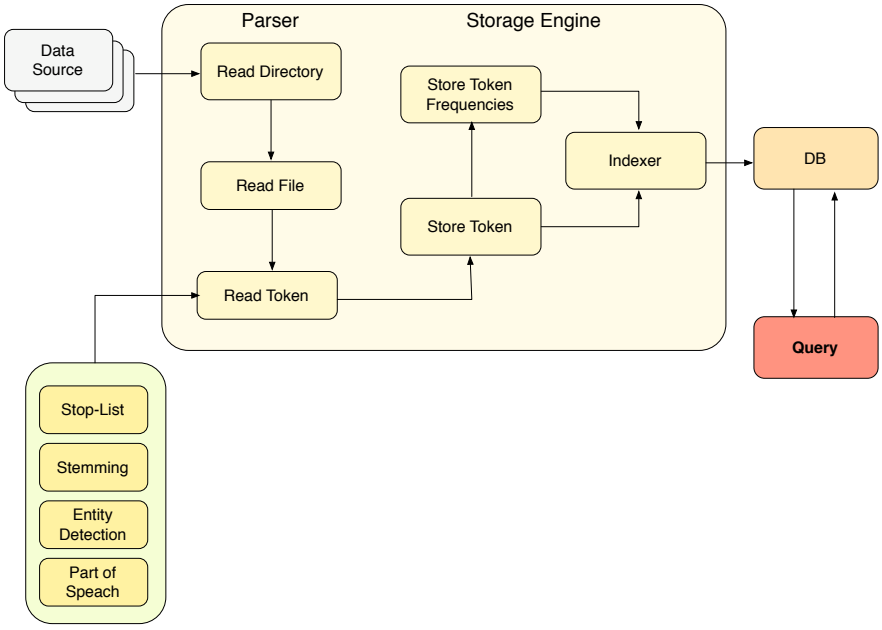
System exploitation, querying



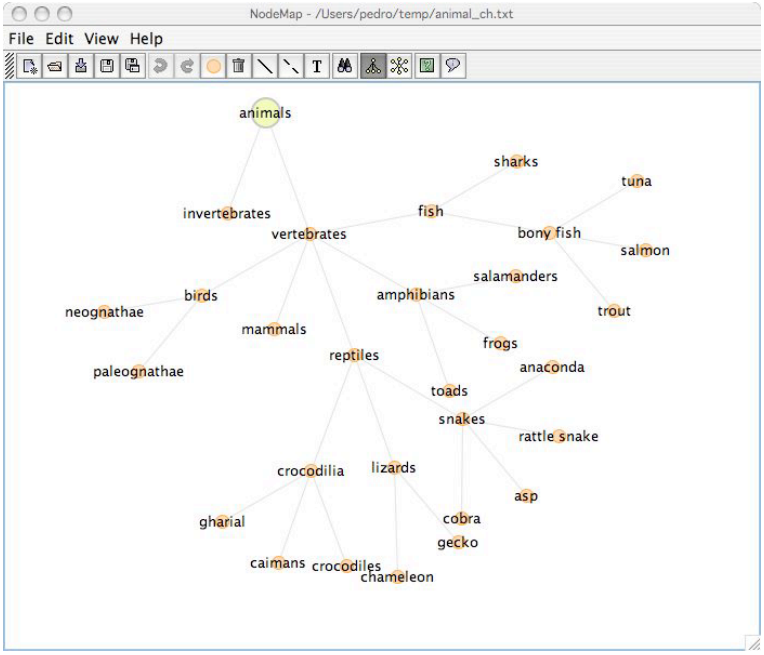
Types of Queries

- **Boolean**: *AND, OR, NOT*
- **Natural Language Queries**: query is formulated as a question or a statement
- **Thesaurus Queries**: the user select the term from a previous term-set provided by the IR system
- **Fuzzy Queries**: threshold of relevance is expanded to include additional documents
- **Term Searches:** based in a few words or phrases provided by the user
- **Probabilistic Queries**: IR systems based in a computed probability to retrieve documents

An IR system overview



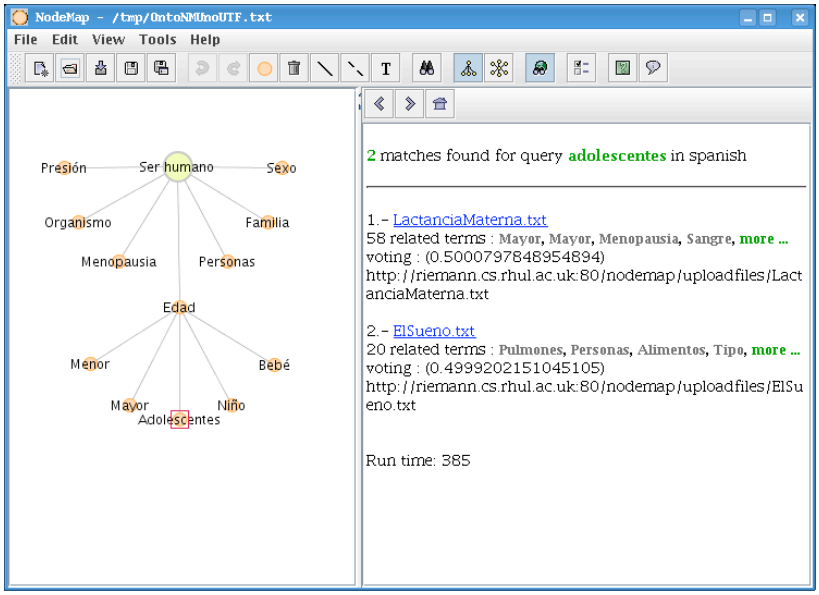
Demo 1: text classification



Here we have an ontology and we would like to assign or classify documents on a given category

http://thames.cs.rhul.ac.uk/wstalk

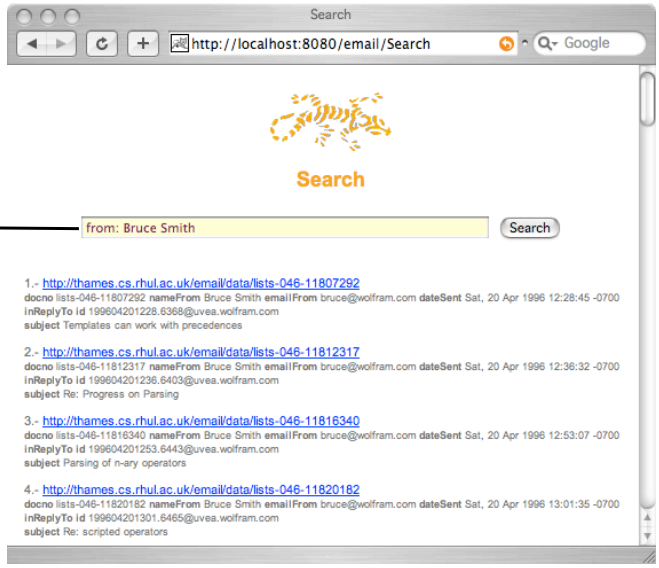
Demo 1: text classification



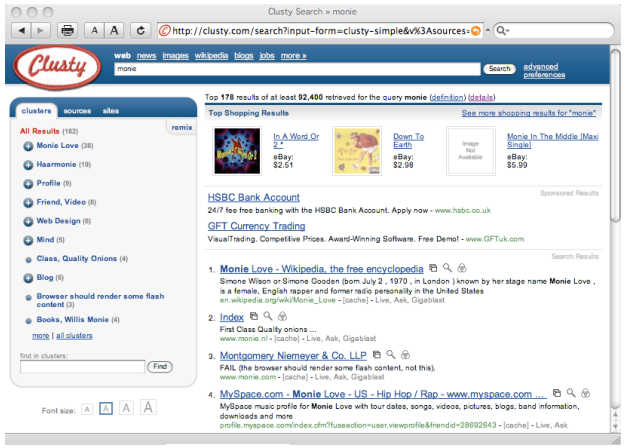
http://thames.cs.rhul.ac.uk/wstalk/prototype.html

Demo 2: email search

file:<keyword>
from: <name>
to: <email to>
cc: <email cc>
subject: <subject>
<keyword>



Search engines



Commercial

Google YAHOO! Ask alltheweb
HOTBOT WebCrawler Live Search altavista .....

Open source

Lucene Terrier Xapian Estraier and so on...

Clustering

Given a set of data points, each having a set of attributes, and a similarity measure among them, find clusters such that:

– Data points in one cluster are more similar to one another
– Data points in separate clusters are less similar to one another

Unlike the classification problem here we do not know the labels or data categories, for this reason this is also called unsupervised learning.

Notion of clusters can be ambiguous

How many clusters?

Two Clusters

Four Clusters

Six Clusters

High dimensionality

Motivation: traditional clustering method do not scale well in very high dimension spaces.

Additionally dimension reduction techniques (e.g. PCA, SVG) are expensive computationally.

Baire, or longest common prefix

Definition: a Baire space consists of countably infinite sequence with a metric defined in terms of the longest common prefix

Case of vectors x and y, with 1 attribute. Precision: digits 1, 2, ..., |K|

$$d_B(x_K, y_K) = \begin{cases} 1 & \text{if } x_1 \neq y_1 \\ \inf 2^{-n} & x_n = y_n \quad 1 \leq n \leq |K| \end{cases}$$

- each coordinate is normalised, so is a floating point value.
- then: we will define $d_B(x,y)$ based on sharing common prefix in all coordinates.

Baire, or longest common prefix

An example of Baire distance for two numbers (x and y) using a precision of 4

$x = 0.4256$

$y = 0.4278$

Baire distance between x and y:

$d_B(x_4, y_4) = 2^{-3} = |K| = 3$

That is:

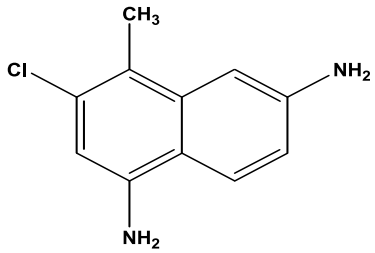
k=1 -> $x_k = y_k$ -> 4

k=2 -> $x_k = y_k$ -> 2

k=3 -> $x_k \neq y_k$ -> 5≠7

Baire, or longest common prefix

Binary Fingerprints: Chemical compounds



Encode

1 0 0 0 1 0 0 0 1 ...1

Clustering of compounds based on chemical descriptors or chemical representations, in the pharmaceutical industry

Baire metric, an example

Simple clustering hierarchy

Random projection schematically

A =

1	0	1	1	0
2	0	0	1	0
3	0	0	1	0
4	1	1	0	1

matrix **normalised** by column sums

N =

0	0.5	0.33	0
0	0	0.33	0
0	0	0.33	0
1	0.5	0	1

Random vector; **k = 2**

R =

0.13	0.45	0.76	0.49

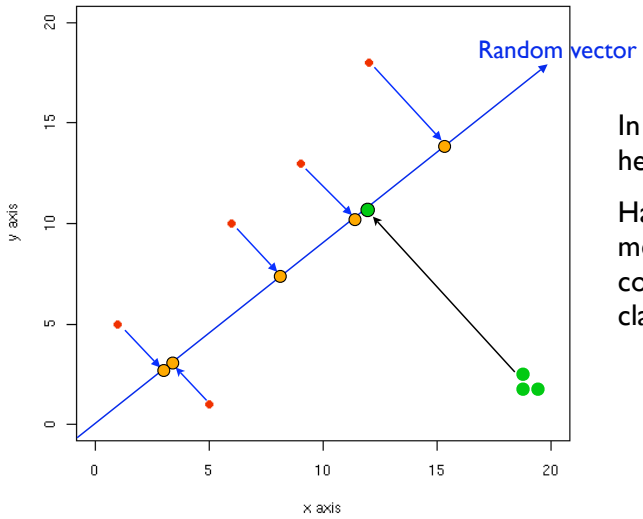
N x R = Random projected vector

1	0.47
2	0.25
3	0.25
4	0.84

Sorting ....

2	0.25	} 1st cluster
3	0.25	
1	0.47	} 2nd cluster
4	0.84	} 3rd cluster

Random projection and hashing



In fact random projection
here works as a class of hashing function.

Hashing is much faster than alternative methods because avoid the pair-wise comparisons required for partition and classification

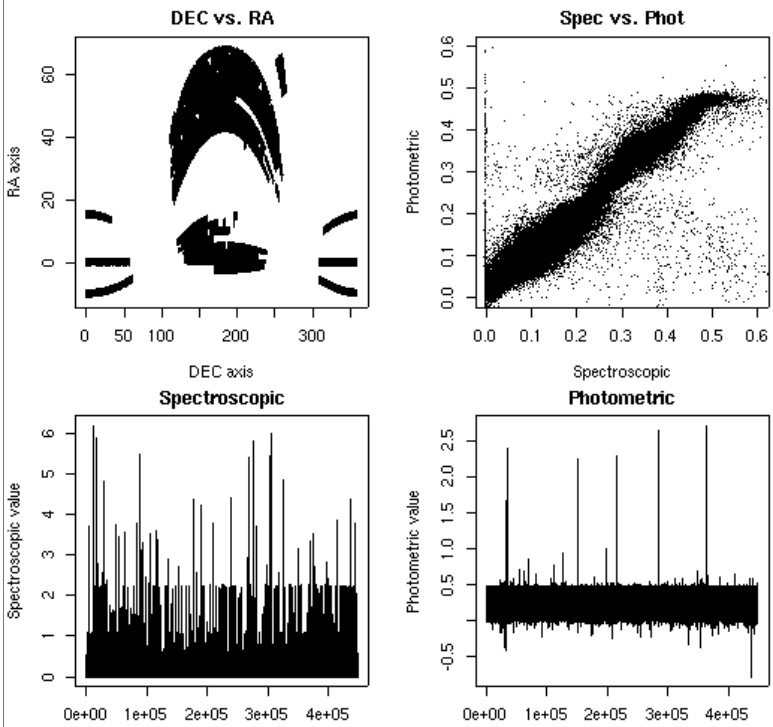
If two points (p , q) are close, they will have a very small |p-q| (Euclidean metric) value; and they will hash to the same value with high probability; if they are distant, they should collide with small probability

A note in data sparsity and Random Projection

– Usually text data matrices are sparse

– Very high dimensional data sets have points mostly lying at the vertices of a regular simplex or polygon.

Current work



Currently we are applying the Baire metric to spectrometric and photometric data from the Sloan Digital Sky Survey

Questions

References

[1] Michael Berry and Murray Browne, Understanding Search Engines. Mathematical Modeling and Text Retrieval. SIAM. 2nd Ed. 2005.

[2] Michael Berry, Zlatko Drma and Elizabeth Jessup. Matrices, Vector Spaces, and Information Retrieval. SIAM review. Vol. 41, No 2, pp 335-362. 1999.

[3] Ian Witten, Alistair Moffat and Timothy Bell. Managing Gigabytes. Compressing and Indexing Documents and Images. 2nd. Ed. M. Kaufman. 1999.

[4] Christopher Bishop. Pattern Recognition and Machine Learning. Springer, 2007.

[5] Fionn Murtagh, Geoff Downs and Pedro Contreras. Hierarchical Clustering of Massive, High Dimensional Data Sets by Exploiting Ultrametric Embedding. SIAM Journal on Scientific Computing, in press.

[6] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction:
Applications to image and text data. Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-01). 2001.

[7] Survey of Text Mining, Clustering, Classification, and Retrieval. Michael Berry, editor. Springer. 2004

[8] Sholom Weiss, Nitin Indurkhya, Tong Zhang, and Fred Damerau. Text Mining, Predictive Methods for Analysing unstructured Information. Springer. 2005.

[9] David Grossman and Ophir Frieder. Information Retrieval, Algorithms and Heuristics. 2nd. Ed. Springer. 2005.

[10] Ronen Feldman and James Sanger. The Text Mining Handbook, Advances Approaches in Analyzing Unstructured Data. Cambridge press. 2007.

[11] Clustering and Information Retrieval. Weili Wu, Hui Xiong, and Shashi Shekhar Ed. Kluwer. 2004.

[12] Boris Mirkin. Clustering for Data Mining, A Data Recovery Approach. Chapman and Hall. 2005.

[13] Keith van Rijsbergen. The Geometry of Information Retrieval. Cambridge press. 2004.