

III.5 Advanced Query Types

(MRS book, Chapters 9+10; Baeza-Yates, Chapters 5+13)

- *5.1 Query Expansion & Relevance Feedback*
- *5.2 Vague Search:*

Phrases, Proximity-based Ranking,

More Similarity Measures: Phonetic, Editex, Soundex

- *5.3 XML-IR*

III.5.1 Query Expansion & Relevance Feedback

Average length of a query (in any of the major search engines) is **about 2.6 keywords**.

(source: <http://www.keyworddiscovery.com/keyword-stats.html>)

May be sufficient for most everyday queries:

“steve jobs”

Navigational

→ find specific resource;
known information need

...but not for all:

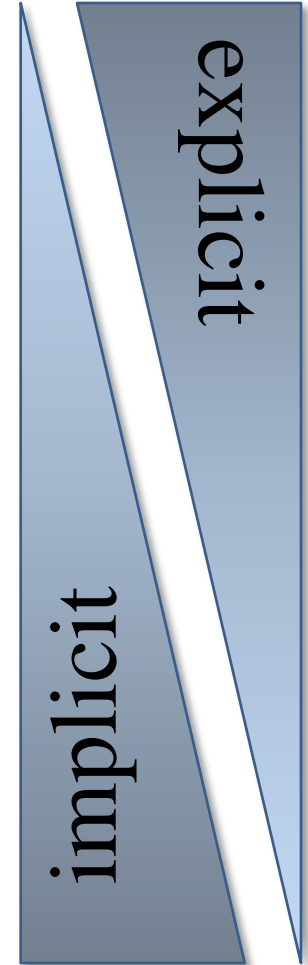
“transportation tunnel disasters”

Informational

→ learn about topic in general;
target not known; relevant
instances not captured
by keywords

Explicit vs. Implicit Relevance Feedback

- Manual document selection
- Query & click logs
- Eye tracking
- Pseudo relevance feedback



Relevance Feedback for the VSM

Given: a query q , a result set (or ranked list) D ,
a user's assessment $u: D \rightarrow \{+, -\}$
yielding positive docs $D^+ \subseteq D$ and negative docs $D^- \subseteq D$

Goal: derive query q' that better captures the user's intention,
by adapting term weights in the query or by query expansion

Classical approach: **Rocchio method** (for term vectors)

$$\vec{q}' = \alpha \vec{q} + \frac{\beta}{|D^+|} \sum_{d \in D^+} \vec{d} - \frac{\gamma}{|D^-|} \sum_{d \in D^-} \vec{d} \quad \text{with } \alpha, \beta, \gamma \in [0,1] \\ \text{and typically } \alpha > \beta > \gamma$$

Modern approach: replace explicit feedback by **implicit feedback**
derived from **query & click logs** (pos. if clicked, neg. if skipped)

or rely on **pseudo-relevance feedback**:

assume that all top-k results are positive

Rocchio Example

Documents $d_1 \dots d_4$ with relevance feedback:

	tf_1	tf_2	tf_3	tf_4	tf_5	tf_6	R
d_1	1	0	1	1	0	0	1
d_2	1	1	0	1	1	0	1
d_3	0	0	0	1	1	0	0
d_4	0	0	1	0	0	0	0

} $|D^+|=2, |D^-|=2$

Given: $\vec{q} = \langle 1, 1, 1, 1, 1 \rangle$

$$\text{Then: } \vec{q}' = \left(\frac{1}{2} \cdot 1 + \frac{1}{3 \cdot 2} \cdot 2 - \frac{1}{4 \cdot 2} \cdot 0, \frac{1}{2} \cdot 1 + \frac{1}{3 \cdot 2} \cdot 1 - \frac{1}{4 \cdot 2} \cdot 0, \dots \right)$$

$$\text{Using } \vec{q}' = \alpha \vec{q} + \frac{\beta}{|D^+|} \sum_{d \in D^+} \vec{tf}_d - \frac{\gamma}{|D^-|} \sum_{d \in D^-} \vec{tf}_d \quad \text{with } \alpha=1/2, \beta=1/3 \text{ and } \gamma=1/4, tf_{ij} \in [0,1]$$

Multiple feedback iterations possible: set $q = q'$ for the next iteration.

Relevance Feedback for Probabilistic IR

Compare to **Robertson/Sparck-Jones** formula (see Chapter III.3):

$$\text{sim}(d, q) = \sum_{i \in q \cap d} \log \frac{r_i + 0.5}{R - r_i + 0.5} + \sum_{i \in q \cap d} \log \frac{N - n_i - R + r_i + 0.5}{n_i - r_i + 0.5}$$

Where

- N : #docs in sample
- R : # relevant docs in sample
- n_i : #docs in sample that contain term i
- r_i : #relevant docs in sample that contain term i

Advantage of RSJ over Rocchio:

- No tuning parameters for reweighting the query terms!

Disadvantages:

- Document term weights are not taken into account
- Weights of previous query formulations are not considered
- No actual query expansion is used (existing query terms are just reweighted)

TREC Query Format & Example Query

<num> Number: 363

<title> **transportation tunnel disasters**

<desc> Description: What disasters have occurred in tunnels used for transportation?

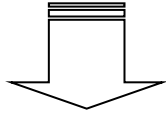
<narr> Narrative: A relevant document identifies a disaster in a tunnel used for trains, motor vehicles, or people. Wind tunnels and tunnels used for wiring, sewage, water, oil, etc. are not relevant. The cause of the problem may be fire, earthquake, flood, or explosion and can be accidental or planned. Documents that discuss tunnel disasters occurring during construction of a tunnel are relevant if lives were threatened.

- See also: TREC 2004/2005 Robust Track
<http://trec.nist.gov/data/robust.html>
- Specifically picks difficult queries (topics) from previous ad-hoc search tasks
- Relevance assessments by retired NIST staff

Query Expansion Example

Q: **transportation tunnel disasters** (from TREC 2004 Robust Track)

transportation 1.0



transit 0.9

highway 0.8

train 0.7

truck 0.6

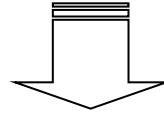
metro 0.6

"rail car" 0.5

car 0.1

...

tunnel 1.0



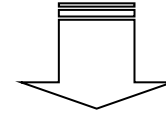
tube 0.9

underground 0.8

"Mont Blanc" 0.7

...

disasters 1.0



catastrophe 1.0

accident 0.9

fire 0.7

flood 0.6

earthquake 0.6

"land slide" 0.5

...

- **Expansion terms** from (pseudo-) relevance feedback, thesauri/gazetteers/ontologies, Google top-10 snippets, query & click logs, user's desktop data, etc.
- **Term similarities** pre-computed from corpus-wide correlation measures, analysis of co-occurrence matrix, etc.

Towards Robust Query Expansion

Threshold-based query expansion:

Substitute $\sim w$ by $\text{exp}(w) := \{c_1 \dots c_k\}$ for all c_i with $\text{sim}(w, c_i) \geq \delta$

Naive scoring:

$$s(q, d) = \sum_{w \in q} \sum_{c \in \text{exp}(w)} \text{sim}(w, c) * s_c(d)$$

*danger of
“topic dilution”/
“topic drift”*

Approach to careful expansion and scoring:

- Determine **phrases** from query or best initial query results (e.g., forming 3-grams and looking up ontology/thesaurus entries)
- If **uniquely mapped** to one concept then expand with synonyms and weighted hyponyms
- Avoid **undue score-mass accumulation** by expansion terms:

$$s(q, d) = \sum_{w \in q} \max_{c \in \text{exp}(w)} \{ \text{sim}(w, c) * s_c(d) \}$$

Query Expansion Example

From TREC 2004 Robust Track Benchmark:

Title: International Organized Crime

Description: Identify organizations that participate in international criminal activity, the activity, and collaborating organizations and the countries involved.

Search Word: <input type="text" value="organized crime"/>	<input type="button" value="Redisplay Overview"/>
Searches for organized crime: <input type="button" value="Noun"/>	Senses: <input type="text"/>
<p>1 sense of organized crime</p> <p>Sense 1</p> <p>organized crime, gangland, gangdom -- (underworld organizations)</p> <ul style="list-style-type: none">=> yakuza -- (organized crime in Japan; an alliance of criminal organizations and illegal enterprises)=> Mafia, Maffia, Sicilian Mafia -- (a secret terrorist group in Sicily; originally opposed tyranny but evolved into a criminal organization in the middle of the 19th century)=> Black Hand -- (a secret terrorist society in the United States early in the 20th century)=> Camorra -- (a secret society in Naples notorious for violence and blackmail)=> syndicate, crime syndicate, mob, family -- (a loose affiliation of gangsters in charge of organized criminal activities)	

Query Expansion Example

From TREC 2004 Robust Track Benchmark:

Title: International Organized Crime

Description: Identify organizations that participate in international criminal activity, the activity, and collaborating organizations and the countries involved.

Query = {international[0.145],
{gangdom[1.00], gangland[0.742], "organ[0.213] & crime[0.312]", camorra[0.254],
maffia[0.318], mafia[0.154], "sicilian[0.201] & mafia[0.154]",
"black[0.066] & hand[0.053]", mob[0.123], syndicate[0.093]},
organ[0.213], crime[0.312], collabor[0.415], columbian[0.686], cartel[0.466], ...}

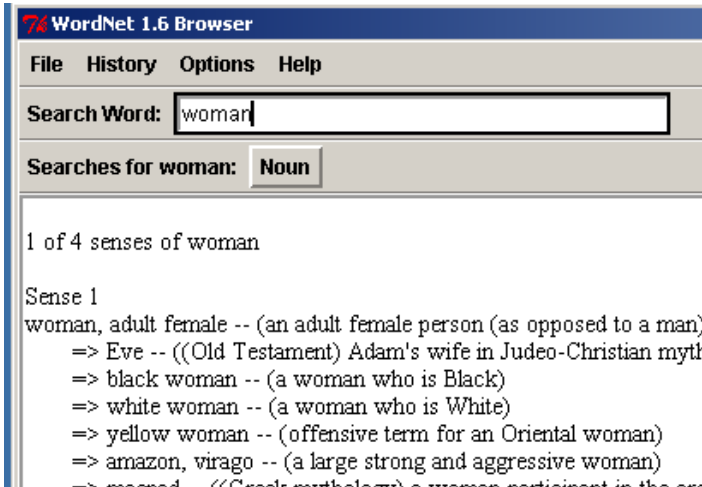
Top-5 Results (in TREC Aquaint News Collection)

1. Interpol Chief on Fight Against Narcotics
2. Economic Counterintelligence Tasks Viewed
3. Dresden Conference Views Growth of Organized Crime in Europe
4. Report on Drug, Weapons Seizures in Southwest Border Region
5. SWITZERLAND CALLED SOFT ON CRIME

...

Thesaurus/Ontology-based Query Expansion

General-purpose thesauri: **WordNet** family



The screenshot shows the WordNet 1.6 Browser interface. The search bar contains the word 'woman'. Below the search bar, it says 'Searches for woman: Noun'. The results show '1 of 4 senses of woman'. Sense 1 is 'woman, adult female -- (an adult female person (as opposed to a man))'. Below this, there are several related terms and their definitions: '=> Eve -- ((Old Testament) Adam's wife in Judeo-Christian myth)', '=> black woman -- (a woman who is Black)', '=> white woman -- (a woman who is White)', '=> yellow woman -- (offensive term for an Oriental woman)', '=> amazon, virago -- (a large strong and aggressive woman)', and '=> geisha -- ((Japanese) a woman participant in the artistic sites of Dionysus)'. A yellow box is overlaid on the right side of the screenshot, containing the text: '200,000 concepts and relations; can be cast into • description logics or • graph, with weights for relation strengths (derived from co-occurrence statistics)'.

woman, adult female – (an adult female person)

=> amazon, virago – (a large strong and aggressive woman)

=> donna -- (an Italian woman of rank)

=> geisha, geisha girl -- (...)

=> lady (a polite name for any woman)

...

=> wife – (a married woman, a man's partner in marriage)

=> witch – (a being, usually female, imagined to have special powers derived from the devil)

=> gold digger -- (a woman who associates with or marries a rich man in order to get valuables from him through gifts or a divorce settlement)

=> gravida -- (a pregnant woman)

=> heroine -- (a woman possessing heroic qualities)

=> jezebel -- (a shameless impudent scheming woman)

=> jilt -- (a woman who jilts a lover)

=> lady -- (a polite name for any woman)

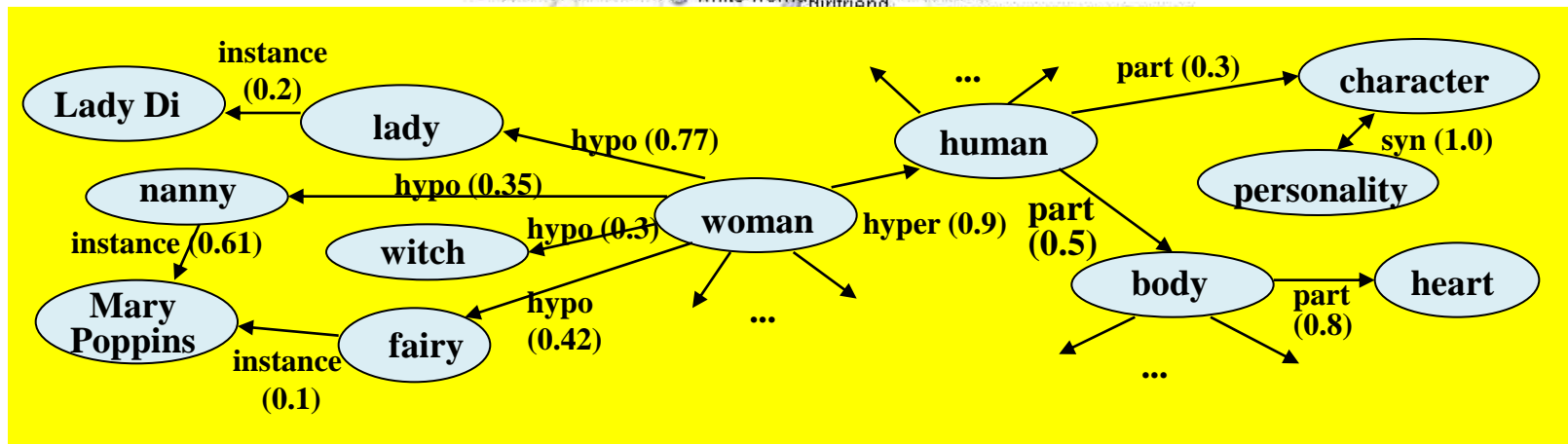
=> madam -- (an unnaturally frenzied or distraught woman)

=> matron, head nurse -- (a woman in charge of nursing in a medical institution)

Most Important Relations among Semantic Concepts

- **Synonymy** (different words with the same meaning)
e.g., “*emodiment*” \leftrightarrow “*archetype*”
- **Hyponymy** (more specific concept)
e.g., “*vehicle*” \rightarrow “*car*”
- **Hypernymy** (more general concept)
e.g., “*car*” \rightarrow “*vehicle*”
- **Meronymy** (part of something)
e.g., “*wheel*” \rightarrow “*vehicle*”
- **Antonymy** (opposite meaning)
e.g. “*hot*” \leftrightarrow “*cold*”
- Further issues include NLP techniques such as **Named Entity Recognition** (NER) (for noun phrases) and more general **Word Sense Disambiguation** (WSD) (incl. verbs, etc.) of words in context.

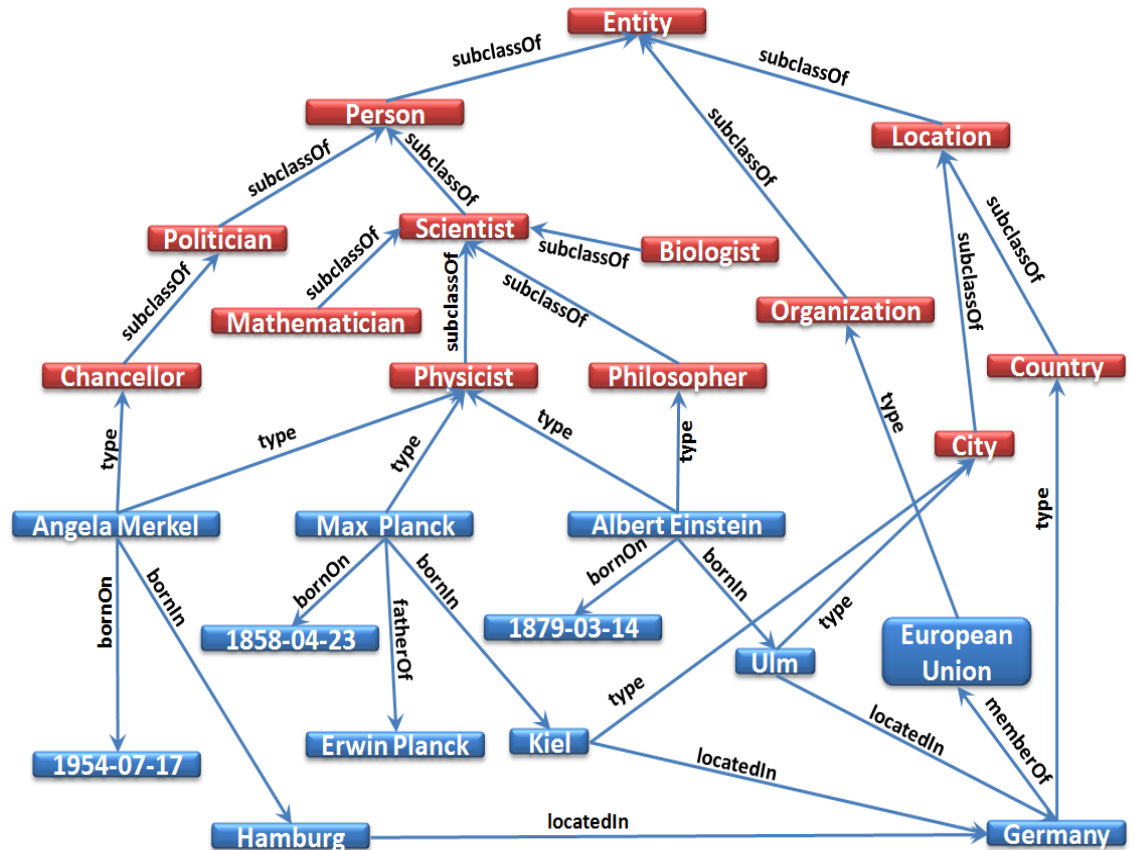
[Fellbaum:
Cambridge Press'98]



YAGO (Yet Another Great Ontology)

[Suchanek et al: WWW'07
Hoffart et al: WWW'11]

- Combine knowledge from **WordNet** & **Wikipedia**
- Additional **Gazetteers** (geonames.org)
- Part of the **Linked-Data** cloud



YAGO-2 Numbers

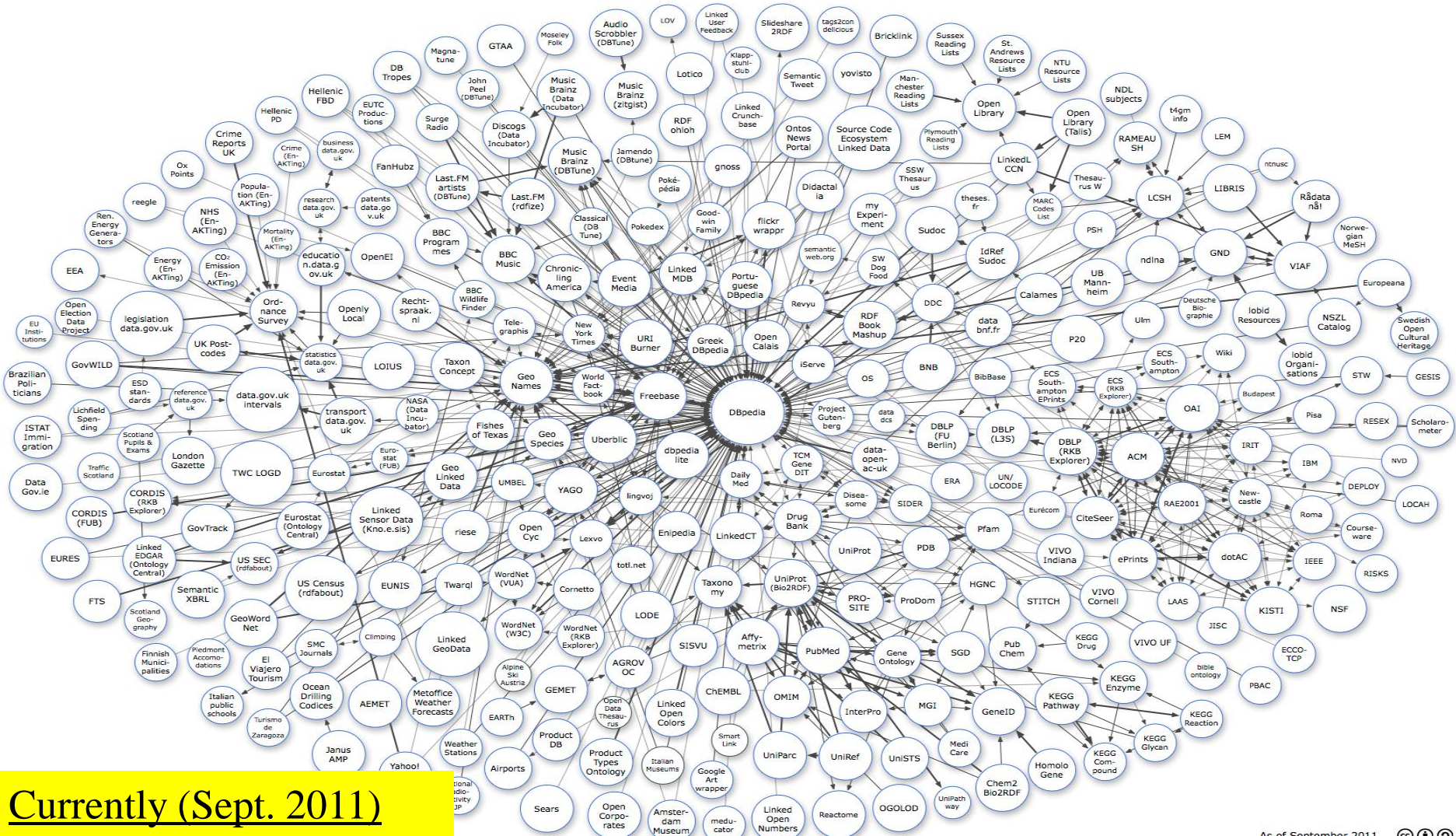
[Hoffart et al: WWW'11]


	Just Wikipedia	Incl. Gazetteer Data
#Relations	104	114
#Classes	364,740	364,740
#Entities	2,641,040	9,804,102
#Facts	120,056,073	461,893,127
- types & classes	8,649,652	15,716,697
- base relations	25,471,211	196,713,637
- space, time & proven.	85,935,210	249,462,793
Size (CSV format)	3.4 GB	8.7 GB

estimated **precision > 95%**
(for base relations excl. space, time & provenance)

www.mpi-inf.mpg.de/yago-naga/

Linked Data Cloud



As of September 2011 

Currently (Sept. 2011)

> 200 sources

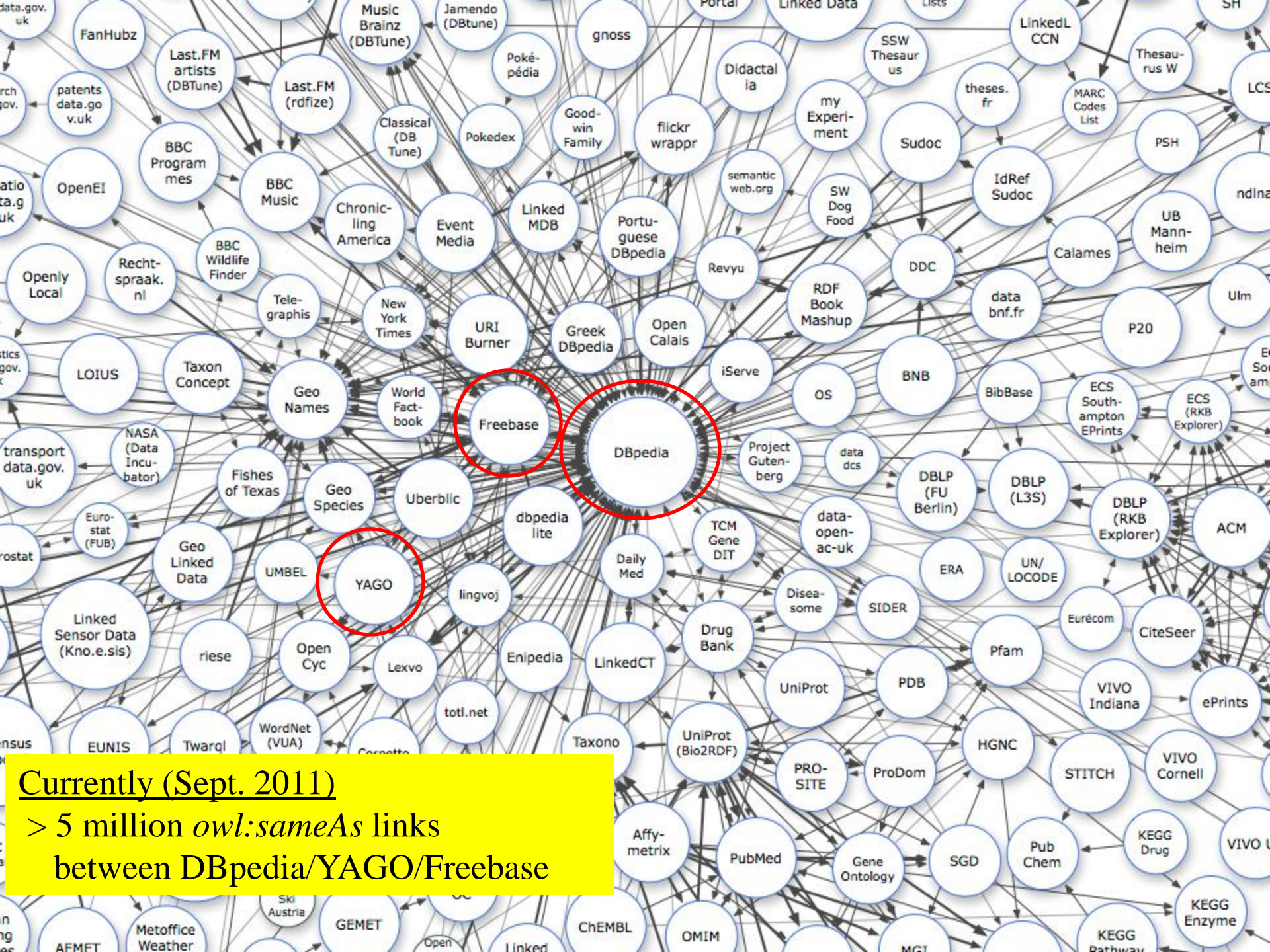
> 30 billion RDF triples

> 400 million links

<http://linkeddata.org/>

November 15, 2011

III.17



Currently (Sept. 2011)
> 5 million *owl:sameAs* links
between DBpedia/YAGO/Freebase

Common Similarity Measures for Ontological Relations

Dice coefficient:
$$\frac{2|\{docs\ with\ c_1\} \cap \{docs\ with\ c_2\}|}{|\{docs\ with\ c_1\}| + |\{docs\ with\ c_2\}|}$$

Jaccard coefficient:
$$\frac{|\{docs\ with\ c_1\} \cap \{docs\ with\ c_2\}|}{|\{docs\ with\ c_1\}| + |\{docs\ with\ c_2\}| - |\{docs\ with\ c_2\ and\ c_2\}|}$$

Conditional Probability:
$$P[doc\ has\ c_1\ |\ doc\ has\ c_2]$$

PMI (Pointwise Mutual Information):
$$\log \frac{freq(c_1 \wedge c_2)}{freq(c_1) \cdot freq(c_2)}$$

(With $freq(c)$ and $freq(c_1 \wedge c_2)$ usually estimated over large Web sample)

Graph-specific Similarity Measures

Compute (graph-based) similarity between *Philosopher* and *Chancellor* in an **IS-A ontology**

Leacock-Chodorow Measure:

$$\text{sim}(c_1, c_2) = -\log\left(\frac{\text{len}(c_1, c_2)}{2D}\right)$$

$\text{len}(c_1, c_2)$: length of shortest path between c_1, c_2

D : depth of the IS-A ontology

Lin Similarity:

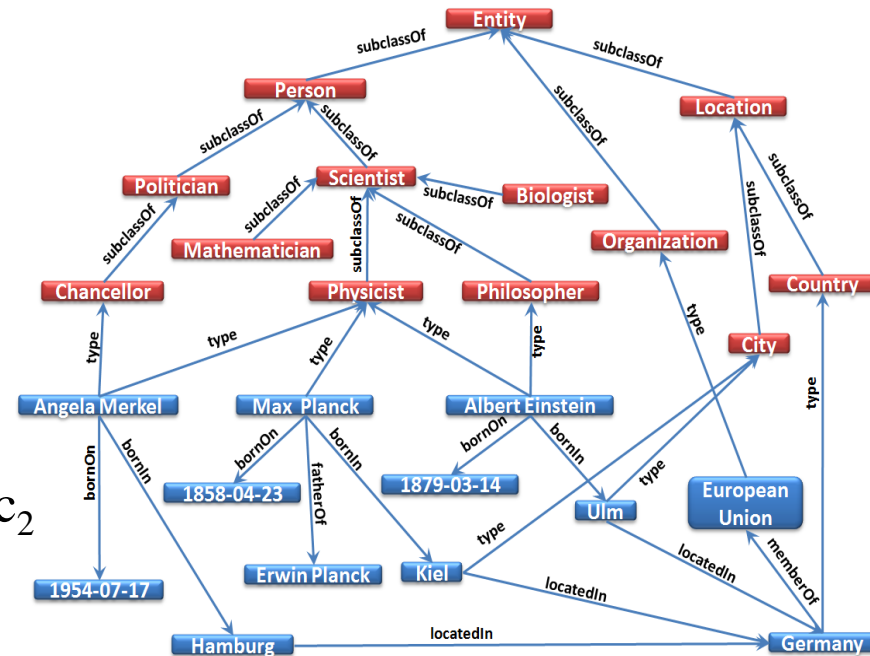
$$\text{sim}(c_1, c_2) = \frac{2 \cdot \text{IC}(\text{LCA}(c_1, c_2))}{\text{IC}(c_1) + \text{IC}(c_2)}$$

$\text{LCA}(c_1, c_2)$: lowest common ancestor of c_1, c_2
 $\text{IC}(c)$: Information Content of c in the IS-A DAG
(including all sub-concepts/hyponyms)

Transitive path similarity:

$$\text{sim}^*(c_1, c_2) = \max\left\{ \prod_{i=1, \dots, n-1} \text{sim}(c_i, c_{i+1}) \mid \text{all paths from } c_1 \text{ to } c_n \right\}$$

(Computed by adaptation of Dijkstra's shortest-path algorithm)



Eye Tracking and Relevance Judgments

- Can **correctly detect the area** of the screen that is focused by the user in 60-90% of the cases
- Distinguish between
 - Pupil fixation
 - Saccades (abrupt stops)
 - Pupil dilation
 - Scan paths
- **Pupil fixations** mostly used to interpret the user's interest
- However **generally not appropriate to judge the quality** of search results (fixation strongly biased toward the top-ranked results in 60-70% of the cases → “trust bias”)

Eye tracking experiments
@ [University of Lübeck](#), 2007



@ [University of Tampere](#), 2007

Exploiting Query Logs for Query Expansion

Given: user sessions of the form (q, D^+) ,
and let “ $d \in D^+$ ” denote the event that d is clicked on

We are interested in the **correlation between words**
 w in a query and w' in a clicked-on document:

$$\begin{aligned} P[w' | w] &:= P[w' \in d \text{ for some } d \in D^+ | w \in q] \\ &= \sum_{d \in D^+} \underbrace{P[w' \in d | d \in D^+]}_{\text{relative frequency of } w' \text{ in } d} \cdot \underbrace{P[d \in D^+ | w \in q]}_{\text{relative frequency of } d \text{ being clicked on when } w \text{ appears in query}} \end{aligned}$$

Estimate
from query log:

Expand query by adding top m
words w' in descending order of $\prod_{w \in q} P[w' | w]$

Implicit Relevance Feedback [Xu, Croft: SIGIR'96]

→ Local Context Analysis

- Retrieve **top n ranked passages** by breaking the initial result documents into smaller passages (e.g., 300 words)

- For each **noun group c** (i.e., concept), compute the similarity **sim(q,c)** to the query q using a variant of TF*IDF

$$sim(q, c) = \prod_{t_i \in q} \left(\delta + \frac{\log(f(c, t_i) \cdot IDF_c)}{\log n} \right)^{IDF_i}$$

$\delta \in [0,1]$: tuning par.

- Expand q by the top r concepts** according to $sim(q,c)$ using
1- $(0.9 m/r)$ as expansion weight,
where m is the position of c in the ranked list of concepts

with $f(c, t_i) = \sum_{j=1}^n pf_{i,j} \cdot pf_{c,j}$

$pf_{i,j}$: frequency of term i in passage j

$$IDF_i = \max\left(1, \frac{\log_{10}(N / np_i)}{5}\right)$$

$$IDF_c = \max\left(1, \frac{\log_{10}(N / np_c)}{5}\right)$$

N: #passages in collection

np_c : #passages containing c

Implicit Relevance Feedback [Qiu, Frei: SIGIR'93]

→ Global Context Analysis

Idea: build global similarity thesaurus *automatically*!

- Consider **inverse term frequency** ITF_j of document d_j
- Compute weight vector k_i of term i
- TF*IDF-style **weights** $w_{i,j}$ for term i in document d_j
- Correlation matrix** $c_{u,v}$ between terms u, v

(Usually expand query with top r ranked terms v according to q)

$$sim(q, k_v) = \vec{q} \times \vec{k}_v = \sum_{k_i \in q} w_{i,q} \cdot c_{i,v}$$

$$ITF_j = \log\left(\frac{t}{t_j}\right)$$

t : #distinct terms in collection
 t_j : #distinct terms in d_j

$$\vec{k}_i = (w_{i,1}, w_{i,2}, \dots, w_{i,N})$$

$$w_{i,j} = \frac{(0.5 + 0.5 \cdot \frac{tf_{i,j}}{\max_j (tf_{i,j})}) \cdot ITF_j}{\sqrt{\sum_{l=1}^N (0.5 + 0.5 \cdot \frac{tf_{i,l}}{\max_k (tf_{i,k})})^2 \cdot ITF_l^2}}$$

$$c_{u,v} = \vec{k}_u \times \vec{k}_v = \sum_{\forall d_j} w_{u,j} \cdot w_{v,j}$$

$$sim(q, d_j) = \sum_{k_v \in d_j} \sum_{k_u \in q} w_{v,j} \cdot w_{u,q} \cdot c_{u,v}$$

III.5.2 Vague Search

<http://www.google.com/press/zeitgeist2010/regions/de.html>

Google.com 2008 (U.S.)

1. obama
2. facebook
3. att
4. iphone
5. youtube

Google news 2008 (U.S.)

1. sarah palin
2. american idol
3. mccain
4. olympics
5. ike (hurricane)

Google image 2008 (U.S.)

1. sarah palin
2. obama
3. twilight
4. miley cyrus
5. joker

Google translate 2008 (U.S.)

1. you
2. what
3. thank you
4. please
5. love

Google.de 2008

1. wer kennt wen
2. juegos
3. facebook
4. schüler vz
5. studi vz
6. jappy
7. youtube
- 8 yasni
9. obama
10. euro 2008

Search Engine Users: People who can't spell!
[Amit Singhal: SIGIR'05 Keynote]

Vague String Matching with Edit Distance

Idea:

Tolerate mis-spellings and other variations of search terms and score matches based on editing distance.

Examples:

1) *Query*: “Microsoft”

Vague Match: “Migrosoft”

Score ~ edit distance 3

2) *Query*: “Microsoft”

Vague Match: “Microsiphon”

Score ~ edit distance 5

3) *Query*: “Microsoft Corporation, Redmond, WA”

Vague match (at token level): “MS Corp., Readmond, USA”

But:

Requires substantial amount of **query rewriting/expansion** and/or expensive **string similarity comparisons** at query time!

Similarity Measures on Strings (1)

Hamming distance of strings $s_1, s_2 \in \Sigma^*$ with $|s_1|=|s_2|$:
number of different characters (cardinality of $\{i: s_1[i] \neq s_2[i]\}$)

Levenshtein distance (edit distance) of strings $s_1, s_2 \in \Sigma^*$:
minimal number of editing operations on s_1
(replacement, deletion, insertion of a character)
to change s_1 into s_2

For $\text{edit}(i, j)$: Levenshtein distance of $s_1[1..i]$ and $s_2[1..j]$ it holds:
 $\text{edit}(0, 0) = 0$, $\text{edit}(i, 0) = i$, $\text{edit}(0, j) = j$
 $\text{edit}(i, j) = \min \{ \text{edit}(i-1, j) + 1,$
 $\text{edit}(i, j-1) + 1,$
 $\text{edit}(i-1, j-1) + \text{diff}(i, j) \}$
with $\text{diff}(i, j) = 1$ if $s_1[i] \neq s_2[j]$, 0 otherwise

→ Efficient computation by **dynamic programming**

Dynamic Programming Example

for Levenshtein Edit Distance: $grate[1..i] \rightarrow great[1..j]$

	<i>g</i>	<i>r</i>	<i>e</i>	<i>a</i>	<i>t</i>
<i>g</i>	0	1	2	3	4
<i>r</i>	1	0	1	2	3
<i>a</i>	2	1	1	1	2
<i>t</i>	3	2	2	2	1
<i>e</i>	4	3	2	3	2

$$\text{edit}(s[1..i], t[1..j]) = \min \{$$

- \downarrow $\text{edit}(s[1..i-1], t[1..j]) + 1,$
- \rightarrow $\text{edit}(s[1..i], t[1..j-1]) + 1,$
- \swarrow $\text{edit}(s[1..i-1], t[1..j-1]) + \text{diff}(s[i], t[j]) \}$

Similarity Measures on Strings (2)

Damerau-Levenshtein distance of strings $s_1, s_2 \in \Sigma^*$:

minimal number of replacement, insertion, deletion, or *transposition* operations (exchanging two adjacent characters) for changing s_1 into s_2

For edit (i, j) : Damerau-Levenshtein distance of $s_1[1..i]$ and $s_2[1..j]$:

$\text{edit}(0, 0) = 0, \text{edit}(i, 0) = i, \text{edit}(0, j) = j$

$\text{edit}(i, j) = \min \{ \text{edit}(i-1, j) + 1,$
 $\text{edit}(i, j-1) + 1,$
 $\text{edit}(i-1, j-1) + \text{diff}(i, j),$
 $\text{edit}(i-2, j-2) + \text{diff}(i-1, j) + \text{diff}(i, j-1) + 1 \}$

with $\text{diff}(i, j) = 1$ if $s_1[i] \neq s_2[j]$, 0 otherwise

Similarity based on N-Grams

Determine for string s the set of its N-grams:

$$G(s) = \{\text{substrings of } s \text{ with length } N\}$$

(often tri-grams are used, i.e. $N=3$)

Distance of strings s_1 and s_2 :

$$|G(s_1)| + |G(s_2)| - 2|G(s_1) \cap G(s_2)|$$

Example:

$$G(\text{rodney}) = \{\text{rod}, \text{odn}, \text{dne}, \text{ney}\}$$

$$G(\text{rhodnee}) = \{\text{rho}, \text{hod}, \text{odn}, \text{dne}, \text{nee}\}$$

$$\text{distance}(\text{rodney}, \text{rhodnee}) = 4 + 5 - 2 \cdot 2 = 5$$

Alternative similarity measures:

Jaccard coefficient: $|G(s_1) \cap G(s_2)| / |G(s_1) \cup G(s_2)|$

Dice coefficient: $2 |G(s_1) \cap G(s_2)| / (|G(s_1)| + |G(s_2)|)$

N-Gram Indexing for Vague Search

Theorem (Jokinen and Ukkonen 1991):

For a query string s and a target string t , the Levenshtein edit distance is bounded by the N-gram-based bag-overlap:

$$\text{edit}(s, t) \leq d \Rightarrow |Ngrams(s) \cap Ngrams(t)| \geq |s| - (N - 1) - dN$$

→ For vague-match queries with edit-distance tolerance d , perform top- k query over N-grams, using counts of N-grams as score aggregation.

Example for Jokinen/Ukkonen Theorem

$$\begin{aligned} \text{edit}(s,t) \leq d & \Rightarrow \text{overlap}(s,t) \geq |s| - (N-1) - dN \\ \text{overlap}(s,t) < |s| - (N-1) - dN & \Rightarrow \text{edit}(s,t) > d \end{aligned}$$

$s = \text{abababababa}, \quad |s|=11$

$N=2 \rightarrow N\text{-grams}(s) = \{\text{ab}(5), \text{ba}(5)\}$

$N=3 \rightarrow N\text{-grams}(s) = \{\text{aba}(5), \text{bab}(4)\}$

$N=4 \rightarrow N\text{-grams}(s) = \{\text{abab}(4), \text{baba}(4)\}$

$t_1 = \text{ababababab}, \quad |t_1|=10$

$t_2 = \text{abacdefaba}, \quad |t_2|=10$

$t_3 = \text{ababaaababa}, \quad |t_3|=11$

$t_4 = \text{abababb}, \quad |t_4|=7$

$t_5 = \text{ababaaabbbb}, \quad |t_5|=11$

task: find all t_i with $\text{edit}(s,t_i) \leq 2$

→ **prune** all t_i with $\text{edit}(s,t_i) > 2 = d$

→ **overlapBound** = $|s| - (N-1) - dN$
= 6 (for $N=2$)

→ **prune** all t_i with $\text{overlap}(s,t_i) < 6$

$N=2$:

$N\text{-grams}(t_1) = \{\text{ab}(5), \text{ba}(4)\}$

$N\text{-grams}(t_2)$

= $\{\text{ab}(2), \text{ba}(2), \text{ac}, \text{cd}, \text{de}, \text{ef}, \text{fa}\}$

$N\text{-grams}(t_3) =$

= $\{\text{ab}(4), \text{ba}(4), \text{aa}(2)\}$

$N\text{-grams}(t_4) = \{\text{ab}(3), \text{ba}(2), \text{bb}\}$

$N\text{-grams}(t_5)$

= $\{\text{ab}(3), \text{ba}(2), \text{aa}(2)\text{bb}(3)\}$

→ **prune** t_2, t_4, t_5 because $\text{overlap}(s,t_i) < 6$ for these t_i

Phrase Queries and Proximity Queries

Phrase queries such as:

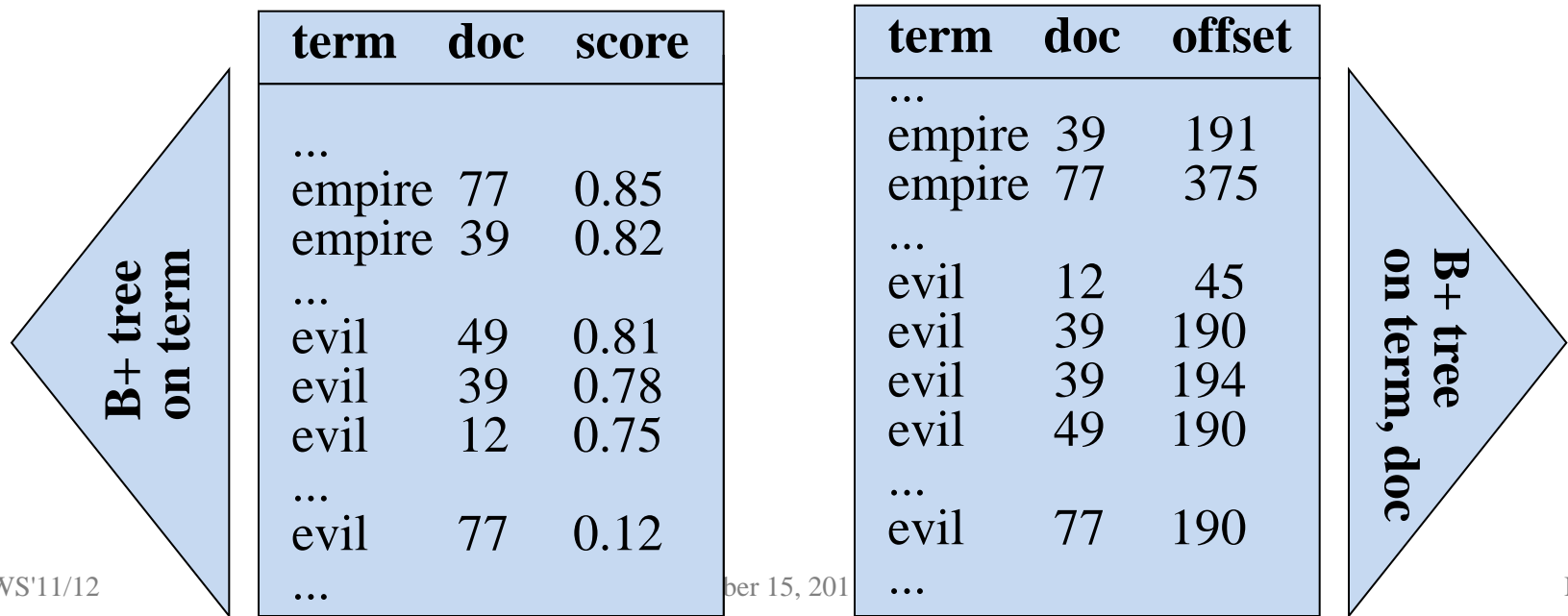
*“George W. Bush”, “President Bush”, “The Who”, “Evil Empire”,
“PhD admission”, “FC Schalke 04”, “native American music”,
“to be or not to be”, “The Lord of the Rings”, etc. etc.*

→ Difficult to anticipate and index all (meaningful) phrases

→ Sources would be thesauri (e.g. WordNet) or query logs

Standard approach:

Combine single-term index with separate position index



Biword and Phrase Indexing

Build index over all word pairs:

- index lists $(term_1, term_2, doc, score)$ or
- for each $term_1$ store nested list $(term_2, doc, score)$

Variations:

- treat nearest nouns as pairs,
or discount articles, prepositions, conjunctions
- index phrases from query logs, compute correlation statistics

Query processing:

- decompose even-numbered query phrases into biwords
- decompose odd-numbered query phrases into biwords
with low selectivity (as estimated by $df(term_1)$)
- may additionally use standard single-term index if necessary

Examples:

“to be or not to be” \rightarrow (to be) (or not) (to be)

“The Lord of the Rings” \rightarrow (The Lord) (Lord of) (the Rings)

N-Gram Indexing and Wildcard Queries

Queries with wildcards (simple regular expressions),
to capture mis-spellings, name variations, etc.

Examples:

Brit*ney, Sm*th*, Go*zilla, Marko*, reali*ation, *raklion

Approach:

- decompose words into N-grams of N successive letters
and index all N-grams as terms
- query processing computes AND of N-gram matches

Example (N=3):

Brit*ney → Bri AND rit AND ney

Generalization: decompose words into frequent fragments
(e.g., syllables, or fragments derived from mis-spelling statistics)

Proximity-based Ranking

Proximity Query Examples:

“root polynom three”, “high cholesterol measure”, “doctoral degree defense”, “statistical relational learning”

→ Particularly important for combinations of mostly frequent (and a few infrequent) keywords with otherwise different meaning.

Idea: Identify positions (pos) of all query-term occurrences in a document and reward short distances.

“Holistic” keyword proximity scores: [Büttcher/Clarke: SIGIR’06]

aggregation of per-term scores # + per-term-pair scores attributed to each term

$$\text{score}(t_1 \dots t_m) = \sum_{i=1..m} \text{core}(t_i) + \sum_{j \neq i} \left\{ \frac{\text{idf}(t_j)}{(\text{pos}(t_i) - \text{pos}(t_j))^2} \mid \neg \exists t_k (\text{pos}(t_i) < \text{pos}(t_k) < \text{pos}(t_j) \text{ or } \dots) \right\}$$

acc(t_j): cannot be pre-computed
→ expensive at query-time

count only pairs of query terms
with no other query term in between

Example: Proximity Score Computation

It¹ took² the³ **sea**⁴ a⁵ thousand⁶ **years,**⁷
A⁸ thousand⁹ **years**¹⁰ to¹¹ trace¹²
The¹³ granite¹⁴ features¹⁵ of¹⁶ this¹⁷ **cliff,**¹⁸
In¹⁹ crag²⁰ and²¹ scarp²² and²³ base.²⁴

E.J. Pratt
(1882-1964)

Query: < **sea**, **years**, **cliff** > (→ order of query terms matters!)

$$\text{acc}(d, \text{sea}) = \frac{\text{idf}(\text{years})}{(7-4)^2}$$

$$\text{acc}(d, \text{years}) = \frac{\text{idf}(\text{sea})}{(7-4)^2} + \frac{\text{idf}(\text{cliff})}{(18-10)^2}$$

$$\text{acc}(d, \text{cliff}) = \frac{\text{idf}(\text{years})}{(18-10)^2}$$

Efficient Proximity Search

Define aggregation function to be **distributive** [Broschart et al. 2007] rather than “holistic” [Büttcher/Clarke 2006]:

→ pre-compute term-pair distances at indexing time
and simply sum up at query-time!

$$score(t_1...t_m) = \sum_{i=1..m} \blacktriangleleft core(t_i) + \underbrace{\sum_{j \neq i} \left\{ \frac{idf(t_j)}{(pos(t_i) - pos(t_j))^2} \right\}}_{\text{count over all pairs of query terms}}$$

→ empirical result quality comparable to „holistic“ scores

Extensions: index all pairs within max. window size
(or nested list of nearby terms for each term),
with precomputed pair-score mass.

Example with More Efficient Proximity Scoring Function

It¹ took² the³ **sea**⁴ a⁵ thousand⁶ **years,**⁷
A⁸ thousand⁹ **years**¹⁰ to¹¹ trace¹²
The¹³ granite¹⁴ features¹⁵ of¹⁶ this¹⁷ **cliff,**¹⁸
In¹⁹ crag²⁰ and²¹ scarp²² and²³ base.²⁴

E.J. Pratt
(1882-1964)

Query: {**sea**, **years**, **cliff**} (→ order of terms does not matter!)

$$\begin{aligned}\text{acc}(d, \text{cliff}, \text{sea}) &= \frac{1}{(18-4)^2} \\ \text{acc}(d, \text{cliff}, \text{years}) &= 0.0 + \frac{1}{(10-4)^2} \\ \text{acc}(d, \text{sea}, \text{years}) &= \frac{1}{(7-4)^2} + \frac{1}{(10-4)^2}\end{aligned}$$

Phonetic Similarity (1)

Soundex Code: (for English)

Mapping of words (especially last names) onto 4-letter codes such that words that are similarly pronounced have the same code

- first position of code = first letter of word
- vowels and “weak” consonants (a, e, i, o, u, y, h, w are ignored)
- code positions 2, 3, 4 :

b, p, f, v	→ 1	c, s, g, j, k, q, x, z	→ 2
d, t	→ 3	l	→ 4
m, n	→ 5	r	→ 6

- Successive identical code letters are combined into one letter (unless separated by the letter h)

Examples:

Powers → P620 , Perez → P620

Penny → P500, Pennee → P500

Tymczak → T522, Tanshik → T522

Phonetic Similarity (2)

Editex similarity:

edit distance with consideration of phonetic codes

For editex (i, j): Editex distance of $s_1[1..i]$ and $s_2[1..j]$ it holds:

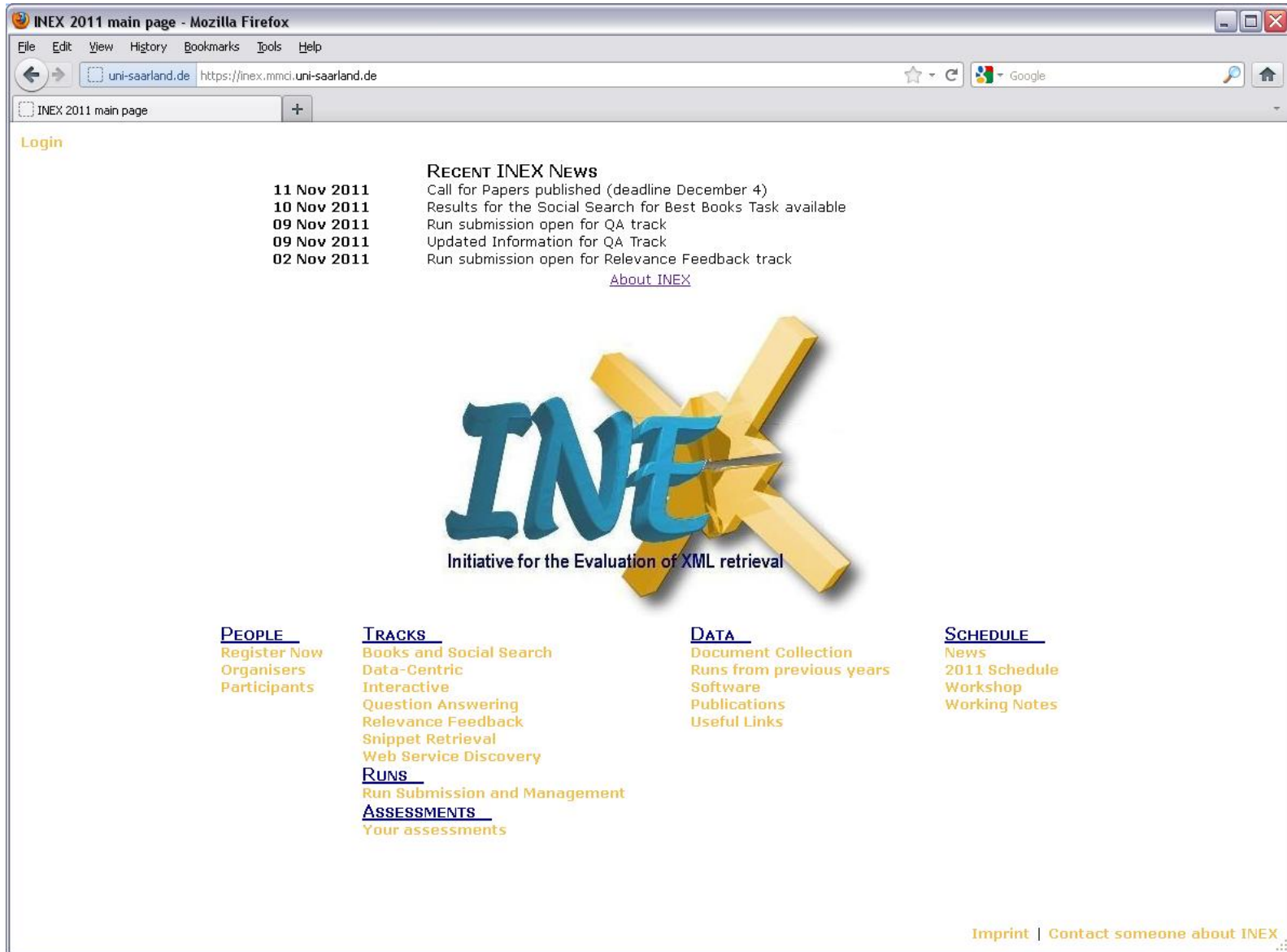
$$\begin{aligned} \text{editex}(0, 0) &= 0, \\ \text{editex}(i, 0) &= \text{editex}(i-1, 0) + d(s_1[i-1], s_1[i]), \\ \text{editex}(0, j) &= \text{editex}(0, j-1) + d(s_2[j-1], s_2[j]), \\ \text{editex}(i, j) &= \min \{ \text{editex}(i-1, j) + d(s_1[i-1], s_1[i]), \\ &\quad \text{editex}(i, j-1) + d(s_2[j-1], s_2[j]), \\ &\quad \text{edit}(i-1, j-1) + \text{diffcode}(i, j) \} \end{aligned}$$

with $\text{diffcode}(i, j) = 0$ if $s_1[i] = s_2[j]$
1 if $\text{group}(s_1[i]) = \text{group}(s_2[j])$, 2 otherwise
und $d(X, Y) = 1$ if $X \neq Y$ and X is h or w,
 $\text{diffcode}(X, Y)$ otherwise

with group:

{a e i o u y}, {b p}, {c k q}, {d t}, {l r},
{m n}, {g j}, {f p v}, {s x z}, {c s z}

III.5.3 XML-IR



History of INEX

- **2002-2011 (and beyond?)**
- Co-Initiative by the **University of Duisburg-Essen** (Norbert Fuhr) and **Queen Mary University London** (Mounia Lalmas)
- Funded by
 - DELOS Network of Excellence (EU)
 - IEEE Computer Society
- Combine two longstanding paradigms: **DB** and **IR**
- Many tracks over the years, including
 - Ad-hoc
 - Efficiency
 - Question Answering
 - Relevance Feedback
 - Interactive Track
 - Books & Social Search
 - Snippet Retrieval
 - Link-The-Wiki

INEX 2002-2006 Ad-Hoc Collection

```
<?xml version="1.0"?>
<!DOCTYPE article SYSTEM "../..//dtd/xmlarticle.dtd" PUBLIC "-//LBIN//DTD IEEE Mag//EN">
- <article>
  <fno>B1089</fno>
  <doi>10.1041/B1089s-2004</doi>
  - <fm>
    - <hdr>
      + <hdr1>
      + <hdr2>
    </hdr>
    - <edinfo>
      <obi>Editor: Mary Baker, Stanford University, mgbaker@cs.stanford.edu</obi>
    </edinfo>
    - <tig>
      <atl>Mobile Computing at the Beach</atl>
      <pn>pp. 89-92</pn>
    </tig>
    - <au sequence="first">
      <fnm>Justin Mazzola </fnm>
      - <snm>
        <ref type="prb" aid="b1089a1">Paluska</ref>
      </snm>
    </au>
    + <au sequence="additional">
    + <au sequence="additional">
    + <au sequence="additional">
    + <au sequence="additional">
    + <au sequence="additional">
    + <fig id="b1089x1">
  </fm>
  - <bdy>
    + <sec>
    - <sec>
```

```
    <st>KEYNOTE: EXPOSE YOUR INFRASTRUCTURE</st>
```

```
    <p ind="none" align="left">Tom Rodden from Nottingham University opened WMCSA 2003 with a rousing keynote about "bringing research into the real world." Specifically, he described the six-year-old Equator project. Equator tries to move research from the lab into the everyday digital world to learn how users actually use technology, rather than trying to create new bits of infrastructure to support what technologists want users to do.</p>
```

```
    <p ind="none" align="left">Interestingly, interfaces in the Equator project didn't completely hide the research systems' complexity or problems. Rather, the interfaces hid little and let users adapt. The extra bits of information from errors and complexity can add up to a great deal of information for the user. Much in the way we check the number of bars on our cell phone before making a call to ensure that we have enough signal strength, the Equator project found that users easily adjust to and master the information that their interfaces give them. As engineers, we often look at error and try to minimize it, much as interface designers look at complexity and try to hide it. However, users can calibrate themselves and their actions to the error information. Rodden's message was that you should expose aspects of the infrastructure and give users the information so that they can make their own decisions about their environment.</p>
```

```
    <p ind="none" align="left">Rodden showed videos and slides from three subprojects to drive home this point. In the first, Equator developed a system for the Glasgow City Architecture Museum involving physical museum attendees, virtual reality attendees, and Web site attendees. They were linked by audio, on-
```

- 16,000 IEEE articles (scientific journal publications)
- XML-ified bibtex + document meta data, ~750 MB XML

INEX 2007-2009 Ad-Hoc Collection

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- generated by CLIX/Wiki2XML [MPI-Inf, MMCI@UdS] $LastChangedRevision: 92 $ on 16.04.2009 16:54:42[mclao0828] -->
<!DOCTYPE article SYSTEM "../article.dtd">
- <article xmlns:xlink="http://www.w3.org/1999/xlink">
  - <skyscraper wordnetid="104233124" confidence="0.9511911446218017">
    - <entity wordnetid="100001740" confidence="0.9511911446218017">
      - <plaza wordnetid="103965456" confidence="0.9508927676800064">
        - <header>
          <title>Toronto Eaton Centre</title>
          <id>354995</id>
        - <revision>
          <id>240664034</id>
          <timestamp>2008-09-24T11:58:13Z</timestamp>
        - <contributor>
          <username>Skeezix1000</username>
          <id>455783</id>
        </contributor>
        </revision>
      - <categories>
        <category>1981 architecture</category>
        <category>Skyscrapers in Toronto</category>
        <category>Shopping malls in Toronto</category>
        <category>Skyscrapers between 150 and 199 meters</category>
        <category>PATH (Toronto)</category>
        <category>Skyscrapers between 100 and 149 meters</category>
        <category>1992 architecture</category>
        <category>Eaton's</category>
      </categories>
    </header>
  - <body>
    - <template>
      <name>infobox shopping mall</name>
    - <parameters>
      - <image src="Toronto_Eaton_Centre_Logo.gif" width="150px">
        <caption>The Toronto Eaton Centre logo. </caption>
      </image>
      - <parking>
        Yonge Parkade, operated by Cadillac Fairview Express (785 spaces)
        <ref xlink:href="#xpointer(/reflist/entry[@id=%228%22])" xlink:type="simple">8</ref>
      </parking>
      <manager> Cadillac Fairview</manager>
    - <website>
      <weblink xlink:href="http://www.torontoeatoncentre.com/" xlink:type="simple"> www.torontoeatoncentre.com</weblink>
    </website>
    - <location>
      - <village wordnetid="108672738" confidence="0.9508927676800064">
```

- 2.6 Mio Wikipedia articles wrapped into XML
- Wiki-Markup + semantic annotations, ~50 GB XML data

INEX 2010-2011 Data-Centric Collection

```
<?xml version="1.0" encoding="UTF-8"?>
- <movie xmlns:xlink="http://www.w3.org/1999/xlink">
  <title>Punch-Drunk Love (2002) </title>
  <url>http://www.imdb.com/Title?Punch-Drunk Love (2002) </url>
  - <overview>
    <rating>7.4 48008votes </rating>
    - <directors>
      <director>Anderson, Paul Thomas </director>
    </directors>
    - <writers>
      <writer>Anderson, Paul Thomas </writer>
    </writers>
    + <releasedates>
    - <genres>
      <genre>Comedy </genre>
      <genre>Drama </genre>
      <genre>Romance </genre>
    </genres>
    <plot>Barry Egan is a small business owner with seven sisters whose abuse has kept him alone and unable to fall in love. When a harmonium and a mysterious woman enter his life, his romantic journey begins. <djblau@aol.com> Barry Egan runs his own company, is continually hounded by his seven sisters, and every now and then gets a tiny bit violent. One odd morning a harmonium first appears in the street then a striking young lady asks for his help with her car. She re-appears a few days later and there seems to be a spark between them, but can they possibly cut through his seemingly over-complicated life and his somewhat unusual interpersonal skills? Jeremy Perkins {jwp@aber.ac.uk} Barry Egan is a wreck, driven to breakdown by the henpecking of his seven sisters. He steals his heart and manhood away from the curbside. Slowly he learns how to direct them toward love, for the sake of and with the help of another troubled soul. Jeff Smith </plot>
    + <keywords>
  </overview>
  - <cast>
    - <actors>
      - <actor>
        <name>Andrews, Jason (I) </name>
        <character>(voice) Operator Carter <2> </character>
      </actor>
      - <actor>
        <name>Barahona, Jorge </name>
        <character>Jorge <13> </character>
      </actor>
      - <actor>
        <name>Beck, John E. </name>
        <character>Member of After Eden Band <27> </character>
      </actor>
      - <actor>
        <name>Bluehouse, Bobby </name>
        <character>After Eden Sound Man <30> </character>
      </actor>
      - <actor>
```

- **4.5 Mio IMDB files about movies/actors/directors**
- **Highly structured content + large textual fields (plots, etc.), ~4.5 GB**

NEXI Query Language [Trotman, Sigurbjörnsson: INEX'04]

Narrowed Extended XPath I

- Proposes a simple query language for both unstructured and structured IR queries against XML documents
- Content-only (CO) queries
“punch drunk love” + “seven sisters”
- Content-And-Structure (CAS) queries
`//article[about(./title, “punch drunk love”)]`
`//sec[about(./, “seven sisters”)]`

XML-IR and the W3C

- <http://www.w3.org/TR/xpath-full-text-10/>

```
doc("http://example.com/full-text.xml")  
/books/book[count(./content ftcontains "tests")>0]
```

- <http://www.w3.org/TR/2005/WD-xmlquery-full-text-use-cases-20051103/>

```
for $book in doc("http://example.com/full-  
text.xml")/books/book  
let $cont := $book/content[. ftcontains "tests"]  
where count($cont)>0  
return $book
```


Query Evaluation (Sub-Tasks)

Article

- Retrieve entire XML articles

Thorough

- Retrieve individual XML elements (including overlapping ones)

Focused

- Retrieve individual XML elements (non-overlapping)
- With a plethora of evaluation metrics, including *precision*, *recall*, $MA(i)P$, $NDC(i)G$, etc.

BM25 with Multiple Weighted Fields

Idea:

[Robertson,Zaragossa,Taylor: CIKM'04]

Extend BM25 to handle the impact of different document fields

(HTML: Punch Drunk Love <P>Punch Drunk Love</P>

$$w_{i,j} := \frac{(k_1 + 1)tf_{i,j}}{k_1((1-b) + b \frac{\text{len}(d_j)}{\text{avglen}}) + tf_{i,j}} \cdot \log \frac{N - df_i + 0.5}{df_i + 0.5}$$

$$\rightarrow tf'_{i,j} := \sum_{f=1}^K v_f \cdot tf_{i,j}[f]$$

- With field-specific weights v_1, \dots, v_K
- Preserves the non-linearity of the tf component

But:

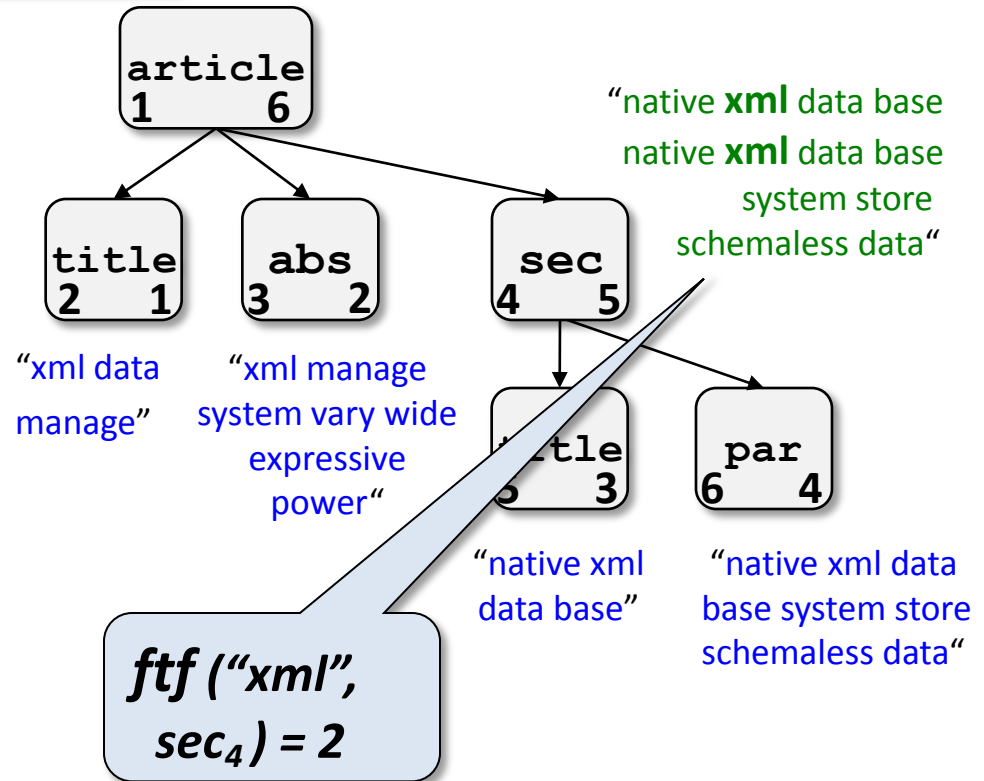
- Requires **adjustment of $\text{len}(d_j)$** to match weighted tf components
- Involves **new tuning parameters v_f**

TopX Data Model

```
<article>
  <title>XML Data Management
</title>
<abs>XML management systems vary
  widely in their expressive power.
</abs>
<sec>
  <title>Native XML Data Bases.
</title>
  <par>Native XML data base systems
    can store schemaless data.
  </par>
</sec>
</article>
```

$ftf("xml", article_1) = 4$

"xml data manage xml manage system vary
wide expressive power native xml native
xml data base system store
schemaless data"



- XML trees with XML elements as inner nodes and **text nodes as leafs**
- Additionally associate inner nodes with redundant **full-content text nodes** for entire subtree

BM25 with Hierarchical Scores

[TopX @ INEX '05–'09]

Content Index (Tag-Term Pairs)

DocID	Tag	Term	Pre	Post	FTF
1	article	xml	1	6	4
1	sec	xml	4	5	2
1	title	xml	5	3	1
1	par	xml	6	4	1
...

Element Freq.

Tag	Term	EF
article	xml	863
sec	xml	947
title	xml	62
par	xml	674
...

Element Statistics

Tag	N	AvLen
article	659K	269.2
sec	1.6M	89.1
title	2.2M	2.8
par	2.8M	34.1
...

author["gates"]
vs.
section["gates"]

XML-specific variant of Okapi BM25

$$score(A//t_1, \dots, t_m, e) = \sum_{i=1}^m \frac{(k_1 + 1) ftf(t_i, e)}{K + ftf(t_i, e)} \log \frac{N_A - ef_A(t_i) + 0.5}{ef_A(t_i) + 0.5}$$

$$with K = k_1 \left((1 - b) + b \frac{length(e)}{avg_{e'} \{length(e') \mid e' \text{ with tag } A\}} \right)$$

with $k_1 = 2.0$, $b=0.75$, and

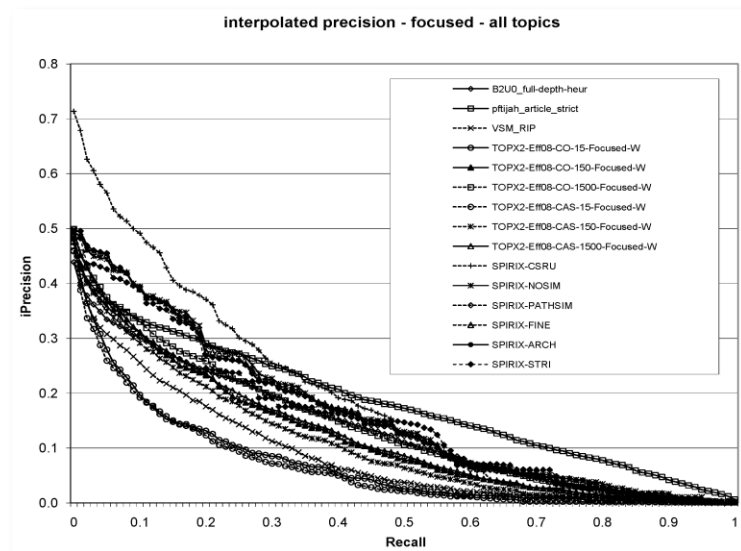
tag-specific element frequencies ef_A and full-text term frequencies ftf over XML subtrees

TopX 2008 Results

INEX Efficiency Track 2008:

Summary of 21 Runs by 5 Groups

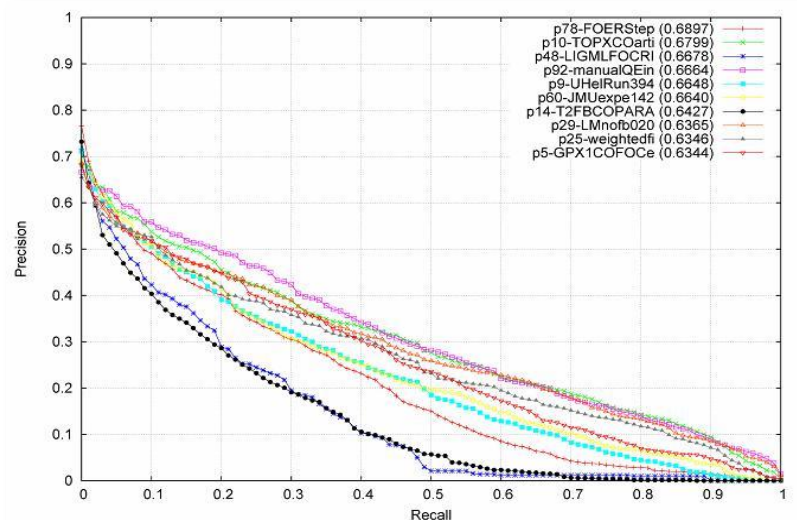
Part.ID	Run ID	iP[0.00]	iP[0.01]	iP[0.05]	iP[0.10]	MAiP	AVG MS.	SUM MS.	#Topics
Focused									
10	TOPX2-Eff08-CAS-15-Focused-W	0.4587	0.3878	0.2592	0.1918	0.0662	90.99	51,499	566
10	TOPX2-Eff08-CAS-150-Focused-W	0.4747	0.4282	0.3494	0.2915	0.1094	112.32	63,574	566
10	TOPX2-Eff08-CAS-1500-Focused-W	0.4824	0.4360	0.3572	0.3103	0.1241	253.42	143,436	566
10	TOPX2-Eff08-CO-15-Focused-W	0.4751	0.4123	0.2793	0.1971	0.0726	49.79	28,180	566
10	TOPX2-Eff08-CO-150-Focused-W	0.4955	0.4520	0.3674	0.3114	0.1225	85.96	48,653	566
10	TOPX2-Eff08-CO-1500-Focused-W	0.4994	0.4560	0.3749	0.3298	0.1409	239.73	135,688	566
16	SPIRIX-ARCH	0.4953	0.4950	0.4544	0.3892	0.1601	100.97	28,779	70
16	SPIRIX-CSRU	0.7134	0.6787	0.5648	0.4915	0.1890	4,723.80	1,346,284	70
16	SPIRIX-FINE	0.4888	0.4882	0.4528	0.3898	0.1628	101.78	29,010	70
16	SPIRIX-NOSIM	0.4943	0.4854	0.4443	0.3940	0.1651	103.23	29,421	70
16	SPIRIX-PATHSIM	0.4997	0.4957	0.4550	0.3885	0.1588	105.30	30,013	70
16	SPIRIX-STRI	0.4821	0.4821	0.4260	0.3942	0.1573	100.48	28,637	33
42	B2U0_full-depth-heur	0.4388	0.3964	0.3344	0.3013	0.1357	2,994.00	1,679,634	561
56	VSM_RIP	0.4836	0.4058	0.3077	0.2553	0.0895	4,807.55	2,730,687	568
Article									
53	pftijah_article_strict	0.4599	0.4272	0.3689	0.3346	0.1839	701.98	398,722	568
Thorough									
10	TOPX2-Eff08-CAS-15-Thorough-W	0.1811	0.0288	0.0069	0.0053		89.31	50,549	566
10	TOPX2-Eff08-CO-15-Thorough-W	0.1890	0.0357	0.0084	0.0065		70.91	40,133	566
42	B2U0_full-depth-sr	0.2196	0.0541	0.0077	0.0080		3,519.59	1,974,492	561
53	pftijah_asp_strict	0.2674	0.1008	0.0294	0.0136		2,306.08	1,309,854	568
53	pftijah_asp_vague	0.2653	0.1120	0.0357	0.0141		8,213.05	4,665,010	568
53	pftijah_star_strict	0.2415	0.1029	0.0471	0.0169		17,186.03	9,761,663	568



INEX Ad-Hoc Track 2008:

Top-15 out of 163 Runs by 23 Groups

#	iP[0.01]	Institute	Run
1	0.690	University of Waterloo	FOERStep
2	0.688	University of Waterloo	FOER
3	0.680	Max-Planck-Institut Informatik	TOPX-CO-articleOnly-Proximity
4	0.669	Max-Planck-Institut Informatik	TOPX-CO-articleOnly-baseline
5	0.668	LIG	LIG-ML-FOCRIC-4OUT-05-00
6	0.666	University of Lyon3	manualQE_indri03_focused
7	0.665	University of Helsinki	UHel-Run3-94
8	0.665	University of Helsinki	UHel-Run2-93
9	0.664	Saint Etienne University	JMU_expe_142
10	0.653	University of Helsinki	UHel-Run1-92
11	0.647	Max-Planck-Institut Informatik	TOPX-CO-all-Focused
12	0.643	University of California, Berkeley	T2FB_CO_PARA
13	0.642	University of Lyon3	manual_indri01_focused
14	0.641	Saint Etienne University	JMU_expe_136
15	0.636	INDIAN STATISTICAL INSTITUTE	LM-nofb-0.20



Summary of Section III.5

- **Difficult queries** cannot easily be solved with 2.6 keywords
- Relevance feedback and query expansion can **more accurately reflect the user's information need**
- Simple Rocchio weighting scheme vs. Probabilistic IR
→ lots of **heuristics** and **ad-hoc tuning parameters**
- Explicit **thesauri** and implicit **term correlations** for automatic query expansion with **phrases** and **proximity-based ranking**
- XML-IR combines ideas from DB and IR in a unified (semistructured) data model with **both text and semantic annotations**

Additional References

- Yonggang Qiu, Hans-Peter Frei: Concept Based Query Expansion. SIGIR 1993: 160-169
- Jinxi Xu, W. Bruce Croft: Query Expansion Using Local and Global Document Analysis. SIGIR 1996: 4-11
- Christiane Fellbaum: A Semantic Network of English: The Mother of All WordNets. Computers and the Humanities 32(2-3): 209-220 (1998)
- Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum: Yago: a core of semantic knowledge. WWW 2007: 697-706
- Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, Edwin Lewis-Kelham, Gerard de Melo, Gerhard Weikum: YAGO2: exploring and querying world knowledge in time, space, context, and many languages. WWW (Companion Volume) 2011: 229-232
- Martin Theobald, Mohammed AbuJarour, Ralf Schenkel: TopX 2.0 at the INEX 2008 Efficiency Track. INEX 2008: 224-236
- Martin Theobald, Ralf Schenkel, Gerhard Weikum: Efficient and self-tuning incremental query expansion for top-k query processing. SIGIR 2005: 242-249
- Andrew Trotman, Börkur Sigurbjörnsson: Narrowed Extended XPath I (NEXI). INEX 2004: 16-40
- Stephen E. Robertson, Hugo Zaragoza, Michael J. Taylor: Simple BM25 extension to multiple weighted fields. CIKM 2004: 42-49
- Wei Lu, Stephen E. Robertson, Andrew MacFarlane: Field-Weighted XML Retrieval Based on BM25. INEX 2005: 161-171