

A Method for Calculating Term Similarity on Large Document Collections

Wolfgang Bein

School of Computer Science
University of Nevada, Las Vegas

Jeffrey S. Coombs and **Kazem Taghva**

Information Science Research Institute
University of Nevada, Las Vegas

April 2003



Document Collections

- Set of documents is $\mathcal{D} = \{1, \dots, N\}$
- Set of keys is $\mathcal{K} = \{1, \dots, M\}$
- Both M and N large numbers, expect $M, N \gg 10^6$;

Question:

For a given key, say `car` what are terms that are are
“similar” in the collection?

“similar” means: occur together significantly
automobile, insurance, road, truck, SUV

Retrieval by General Logical Imaging (RbGLI)

Query:

Find all documents that discuss magnetic field intensity tests
from faults at Yucca Mountain

The aim of RbGLI is to expand the query so that documents which do *not* contain any of the query terms would still receive some weight if they contain terms which are similar to query terms.

Optical Character Recognition (OCR)

- For OCR texts, one hopes that documents which contain terms that are misrecognized by the OCR process might still be retrieved

- auto:

If appears as

a4to

in document d , then d might still be caught by

car

A Similarity Query

Given a constant $c \geq 1$, for a chosen $k \in \mathcal{K}$, provide pointers to keys k_1, \dots, k_c such that $k_1, \dots, k_c \in \mathcal{K}$ are the keys with highest EMIM to k .

EMIM: Expected Mutual Information Measure

... roughly means “co - occurrence”

Co-Occurrence:

$$n_{k,\ell} = |\{d \in \mathcal{D} \mid t_k(d) = 1 \wedge t_\ell(d) = 1 \}|$$

\mathcal{D}	...	t_k	t_ℓ	...
1	...	1	0	...
2	...	1	1	...
3	...	1	0	...
4	...	0	0	...

Two key-document incidence vectors t_k and t_ℓ :

EMIM:

$$f_k^i = \begin{cases} |\{d \in \mathcal{D} \mid k \text{ occurs in } d\}| & \text{for } i = 1 \\ |\{d \in \mathcal{D} \mid k \text{ does not occur in } d\}| & \text{for } i = 0 \end{cases}$$

$$n_{k,\ell}^{i,j} = |\{d \in \mathcal{D} \mid t_k(d) = i \wedge t_\ell(d) = j \}|.$$

$$EMIM(k, \ell) = \sum_{i=0}^1 \sum_{j=0}^1 n_{k,\ell}^{i,j} \log_2 \left(\frac{n_{k,\ell}^{i,j}}{f_k^i f_\ell^j} \right)$$

Run Time Considerations

Documents preprocessed such that the following two queries can be performed in $O(\log M + \log N)$

1. For any $k \in \mathcal{K}$, provide a pointer to a list of all distinct documents, which contain key k .
2. For any $d \in \mathcal{D}$, provide a pointer to a list of all distinct keys, which are contained in document d .

Preprocessing takes

$$O((M + N)(\log M + \log N))$$

Our Heuristic

Preprocess the data further to facilitate

$$O(\log M + \log N)$$

of the EMIM query.

→ Quadtree Heuristic

Our Heuristic takes

$$O((M + N)(\log M + \log N))$$

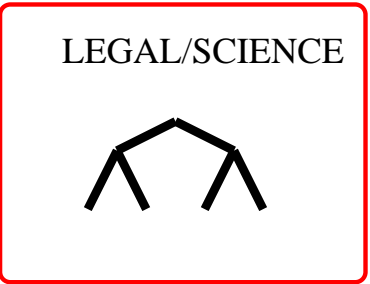
preprocessing

Reference Key Words

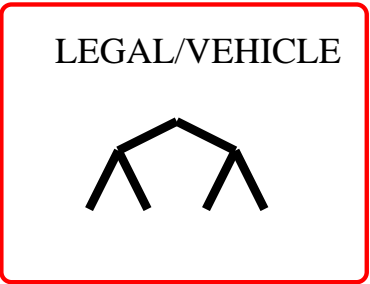
LEGAL MOUNTAIN SCIENCE VEHICLE



Quadtree



Quadtree



Quadtree

...

CAR (3.1 , 0.4 , 1.2 , 9.8)

⋮

ROCK (2.1 . 9.4 . 7.5 . 2.1)

The Quadtree Heuristic

1. Find a small “representative” set of R reference keywords words r_1, \dots, r_R .
2. For each key $k \in \mathcal{K}$, create the reference incidence vector $a(k) = (a_1(k), \dots, a_R(k))$, where $a_i = EMIM(k, r_i)$, for $i = 1, \dots, R$.
3. For each (i, j) , $i, j = 1, \dots, R, i \neq j$: Initialize an empty two-dimensional quadtree $T_{i,j}$ with granularity $\epsilon_0 > 0$.
4. Select small integer α .
For each $k \in \mathcal{K}$, insert the α best pairs (best according to $a_i + a_j$ value) $(a_{i_1}(k), a_{j_1}(k)), \dots, (a_{i_\alpha}(k), a_{j_\alpha}(k))$, into their respective quadtrees $T_{i_1, j_1}, \dots, T_{i_\alpha, j_\alpha}$.

The co-occurrence query is implemented as follows:

1. For key k , find the two highest values in the reference incidence vector; let i_0 and j_0 be the corresponding indices.
2. In the quadtree T_{i_0, j_0} , find all keys $\tilde{\mathcal{K}}$ that are in the square of $a_{i_0}(k)$ and $a_{j_0}(k)$.
3. From $\tilde{\mathcal{K}}$ extract the c best keys. If fewer than c are found repeat with new pair i_0, j_0 .

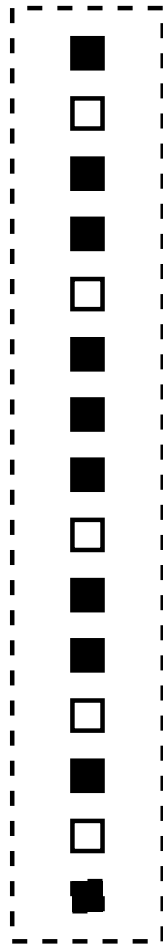
Experimentation

- 75,236 pages collection
- For any search term, select 5 extra terms with highest EMIM value.
- Retrieve documents using the expanded set of terms.
- Compare the results against “relevant” documents selected by human experts

Comparisons:

- Use Brute Force to get the best EMIM matches
- Alternatively use the Quadtree Heuristic to get the matches

Recall: 10 of 100 relevant documents



$$\text{Precision: } \frac{10}{15} = 66\%$$

Recall	Average Precision	Average Interpolated Precision
0	55.43	59.09
10	41.50	44.01
20	37.96	38.91
30	34.36	35.06
40	31.39	31.87
50	26.84	27.34
60	24.45	24.55
70	21.22	21.55
80	18.48	18.50
90	15.05	15.24
100	9.20	9.20
Average:	28.72	29.58

	11-Point Precision	3-Point Precision
	28.716	27.760
Interpolated	29.576	28.248

Average Precision and Recall for Brute Force

Recall	Average Precision	Average Interpolated Precision
0	55.55	59.05
10	41.51	43.95
20	37.97	38.86
30	34.29	35.00
40	31.44	31.93
50	26.86	27.35
60	24.42	24.53
70	21.17	21.50
80	18.49	18.52
90	15.05	15.25
100	9.19	9.19
Average:	28.72	29.56

	11-Point Precision	3-Point Precision
	28.723	27.777
Interpolated	29.557	28.243

Precision and Recall for Quadtree Heuristic

Conclusions

- Methods works very well on sample collection
- We expect the method to scale up favorably
- More work needs to be done to fine tune the method, α , ϵ_0 , choice of reference key words
- Experiments for real-world examples are under way with ISRI.