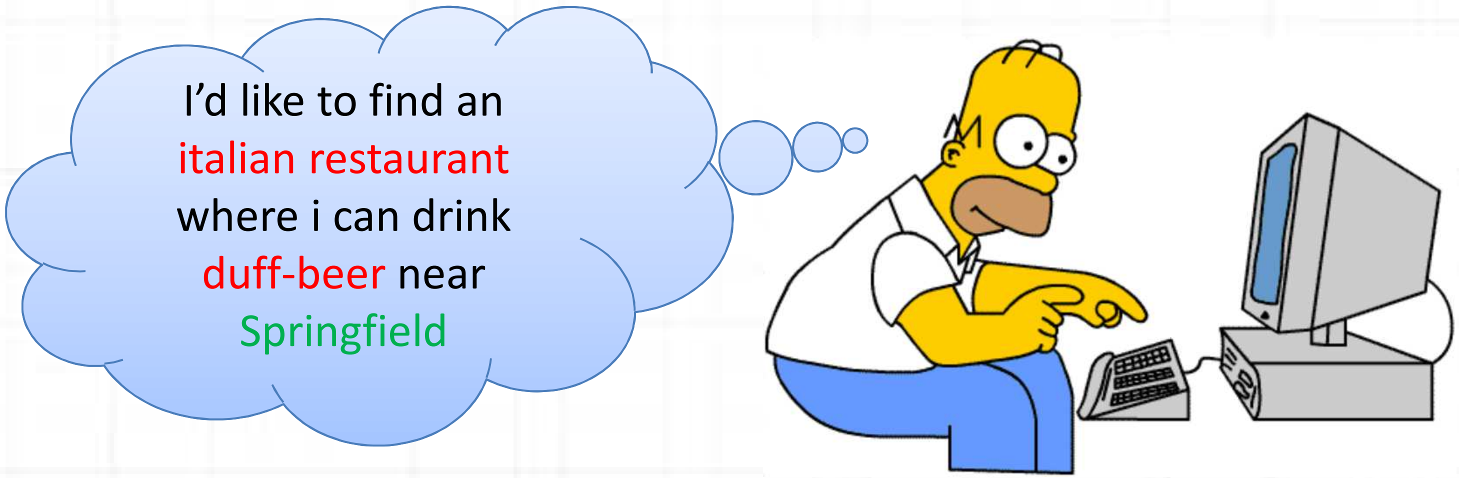
EFFICIENT RETRIEVAL OF THE TOP-K MOST RELEVANT SPATIAL WEB OBJECTS

GAO CONG CHRISTIAN S. JENSEN DINGMING WU

Manzoni Paolo
Menchicchi Fabio(Relatore)
(GRUPPO 1)

Motivations

- About one fifth of web search queries are textual and have local intent
- This imply that the user would like to find:
  - Something that satisfy his needs
  - Something near to him
  - … and know only best results …



Scenary

- The conventional internet is acquiring a geo-spatial dimension:
- Many points of interest are being associated with descriptive text
- Web documents are increasingly being geo-tagged
- Commercial search engines have started to provide location based services, such as map services, local search, and local advertisements.
- For example, Google Maps and Yellow Pages supports location-aware text retrieval queries

Example

Google maps Italia
piatti calabresi
Trattoria Verde Luna Di Esposito Mario E C. Sdf
Si mangia prevalentemente pesce e alcuni piatti della cucina calabrese come spaghetti bomba o spaghetti alla 'nduia, il caviale dei poveri, le sardelle, ...
baltazar.it
Trattoria Amedeo
Si mangia anche il pesce e la cucina è emiliana ma molti piatti sono di impronta calabrese. Porzioni abbondanti, servizio rapido, informale e ...
baltazar.it
Taverna Del Postiglione
I primi piatti emiliani (tortellini, tortelloni e tagliatelle) sono preparati con pasta fresca fatta in casa, così come i dolci che sono di produzione ...
paginegialle.it

Problems to face

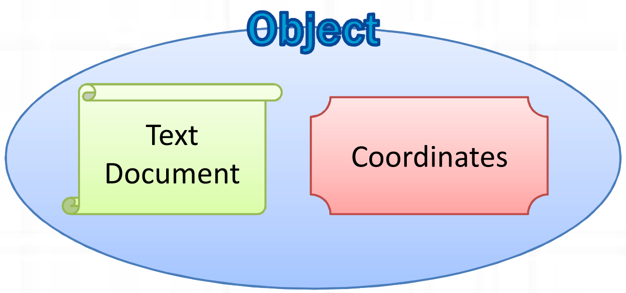
1. How to evaluate both spatial proximity and text similarity

   → Location aware top-K Text retrieval (LkT) query

1. How to join two different worlds:
   1. Inverted files for Text Searches
   2. R-tree for Geo-Searches

   → Ad hoc data structures (IR-tree , DIR-tree)

Object Evaluations



D= Spatial Database
O= Object in D → O=(O.loc, O.doc)

Q= Query → Q=(Q.loc, Q.keywords)

$$Dst(Q,O) = \alpha \frac{D_{\varepsilon}(Q.loc, O.loc)}{maxD} + (1-\alpha)\left(1 - \frac{P(Q.keywords|O.doc)}{maxP}\right)$$

Normalized euclidean distance between queriy and O

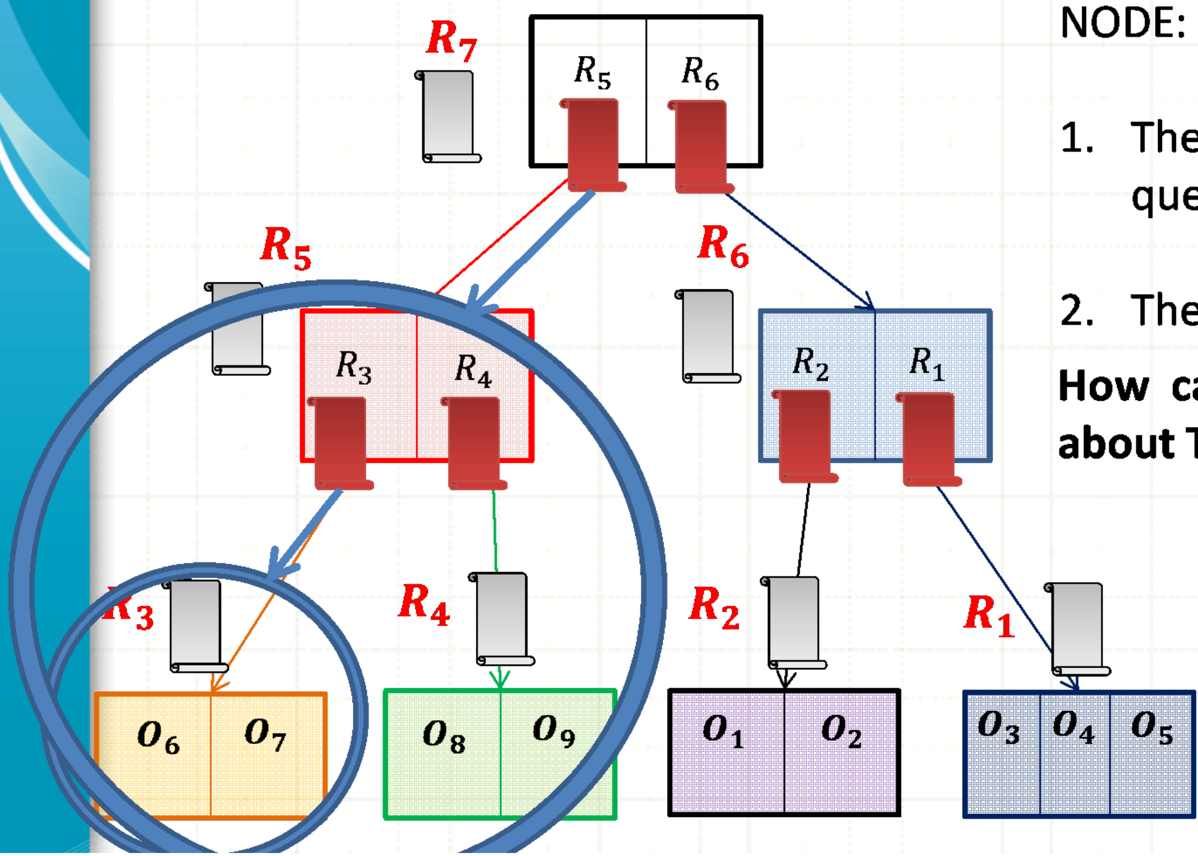
P → evaluates the normalized relevancy of keywords in the document related to O.

Lower values of **Dst** implys interesting objects for users

IR-tree

- Ad-hoc Data Structure created for LkT Query
- Basically an R-tree, with extra infos needed to evaluate Text Relevancy in each node ...

R-tree



We must be able to evaluate FOREACH NODE:

1. The SPATIAL DISTANCE from a query (R-tree help us)
2. The TEXT RELEVANCY for a query

How can we evaluate NON-LEAF NODES about TEXT RELEVANCY?

- **PSEUDO DOCUMENT**

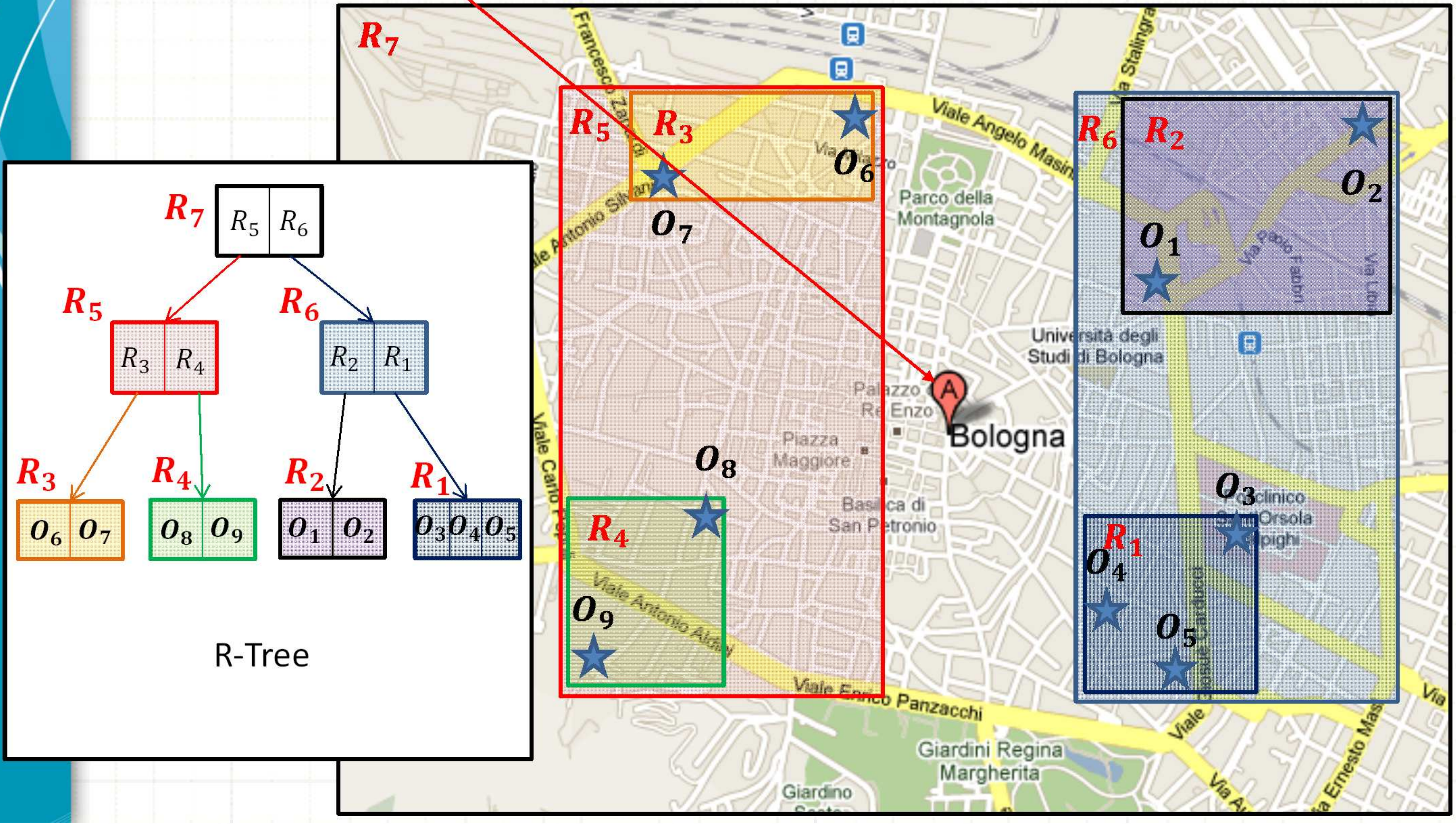
Represents all the documents into the entries of a child node.

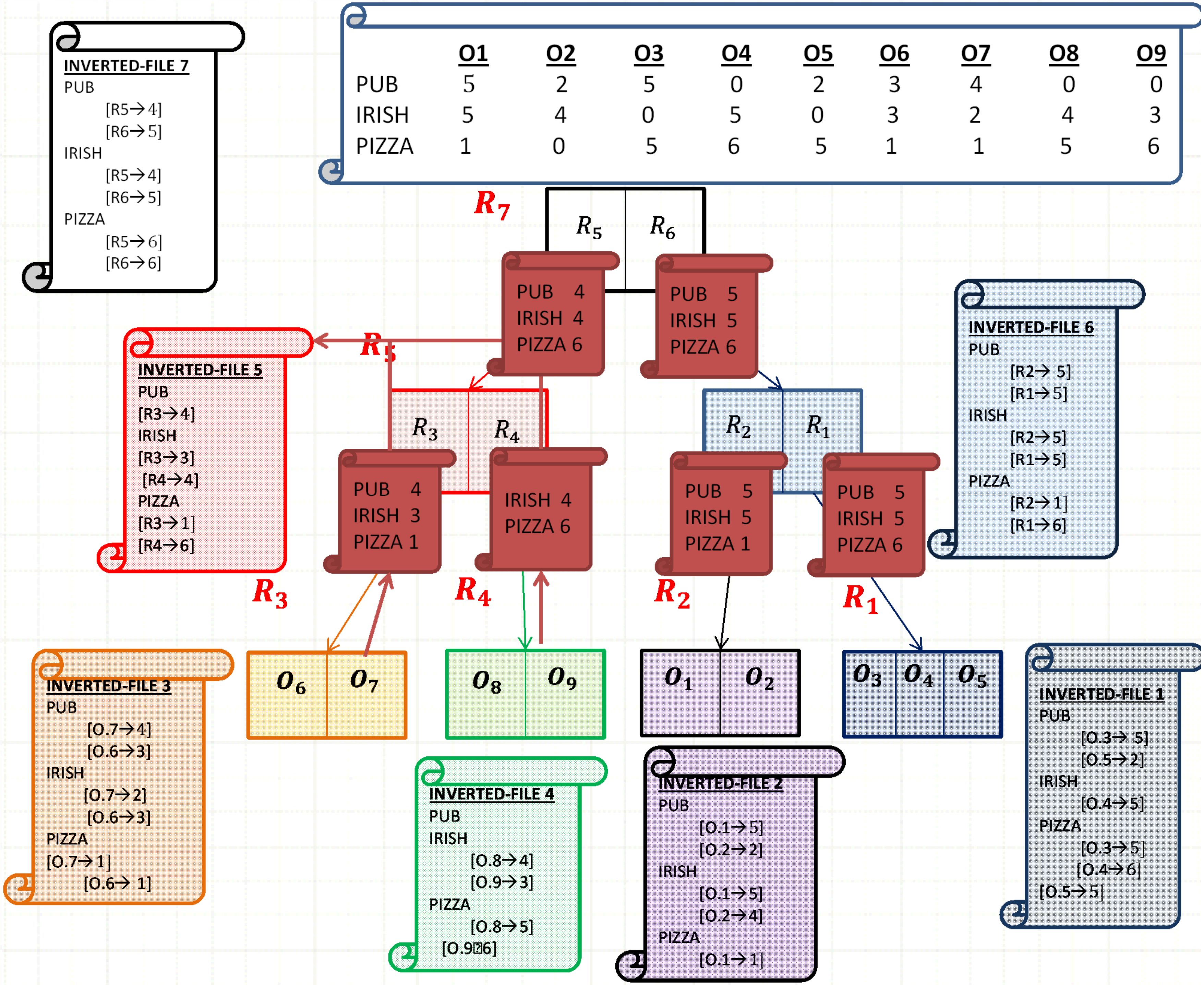
- INVERTED FILES

Example

Q=(A; ‘IRISH’, ‘PUB’)

	O1	O2	O3	O4	O5	O6	O7	O8	O9
PUB	5	2	5	0	2	3	4	0	0
IRISH	5	4	0	5	0	3	2	4	3
PIZZA	1	0	5	6	5	1	1	5	6





INVERTED-FILE 7
PUB
[R5→4]
[R6→5]
IRISH
[R5→4]
[R6→5]
PIZZA
[R5→6]
[R6→6]
O1 O2 O3 O4 O5 O6 O7 O8 O9
PUB 5 2 5 0 2 3 4 0 0
IRISH 5 4 0 5 0 3 2 4 3
PIZZA 1 0 5 6 5 1 1 5 6
R_7
R_5 R_6
PUB 4
IRISH 4
PIZZA 6
PUB 5
IRISH 5
PIZZA 6
R_5
INVERTED-FILE 5
PUB
[R3→4]
IRISH
[R3→3]
[R4→4]
PIZZA
[R3→1]
[R4→6]
R_3 R_4
PUB 4
IRISH 3
PIZZA 1
IRISH 4
PIZZA 6
R_3
R_4
R_2 R_1
PUB 5
IRISH 5
PIZZA 1
PUB 5
IRISH 5
PIZZA 6
R_2
R_1
INVERTED-FILE 6
PUB
[R2→ 5]
[R1→5]
IRISH
[R2→5]
[R1→5]
PIZZA
[R2→1]
[R1→6]
O_6 O_7
O_8 O_9
O_1 O_2
O_3 O_4 O_5
INVERTED-FILE 3
PUB
[O.7→4]
[O.6→3]
IRISH
[O.7→2]
[O.6→3]
PIZZA
[O.7→1]
[O.6→ 1]
INVERTED-FILE 4
PUB
IRISH
[O.8→4]
[O.9→3]
PIZZA
[O.8→5]
[O.9□6]
INVERTED-FILE 2
PUB
[O.1→5]
[O.2→2]
IRISH
[O.1→5]
[O.2→4]
PIZZA
[O.1→1]
INVERTED-FILE 1
PUB
[O.3→ 5]
[O.5→2]
IRISH
[O.4→5]
PIZZA
[O.3→5]
[O.4→6]
[O.5→5]

Searching in the tree

We can now evaluate each node of the tree both in text and distance, and explore it in a 'clever way'

For a Leaf node this is our evaluation :

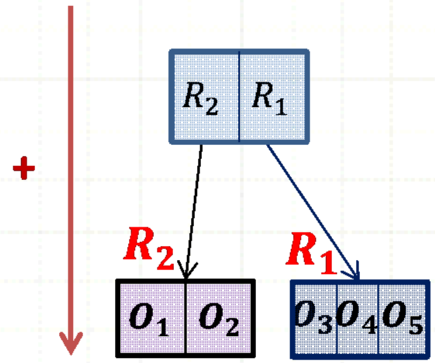
$$Dst(Q,O) = \alpha \frac{D_\varepsilon(Q.loc, O.loc)}{maxD} + (1-\alpha)\left(1 - \frac{P(Q.keywords|O.doc)}{maxP}\right)$$

For a NON Leaf node this is our evaluation :

$$MINDst(Q,N) = \alpha \frac{D_\varepsilon(Q.loc, N.rect)}{maxD} + (1-\alpha)\left(1 - \frac{P(Q.keywords|N.doc)}{maxP}\right)$$

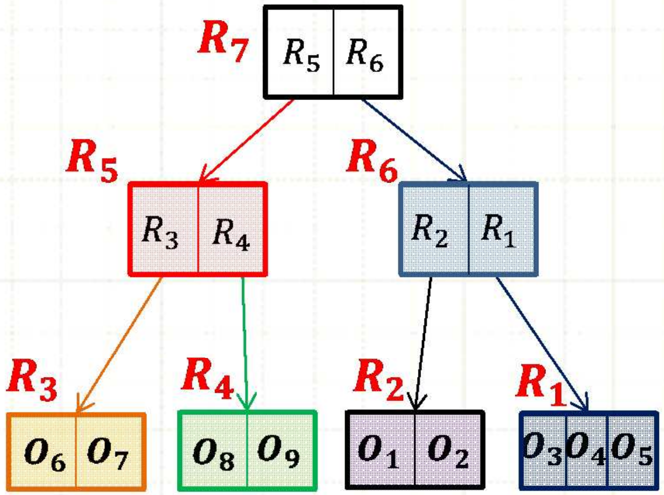
Because of the nature of Pseudo documents
Given a query point ***Q*** and a node ***N*** whose rectangle encloses a set of objects ***SO***

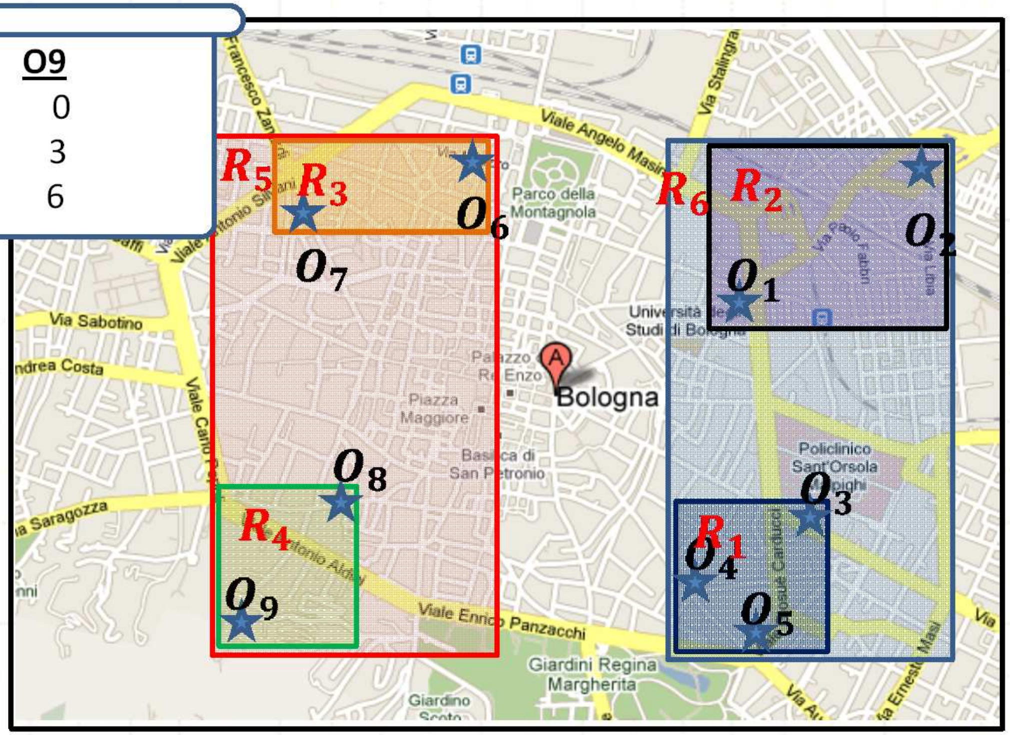
$$\forall O \in SO \rightarrow MINDst(Q,N) \leq Dst(Q,O)$$



	O1	O2	O3	O4	O5	O6	O7	O8	O9
PUB	5	2	5	0	2	3	4	0	0
IRISH	5	4	0	5	0	3	2	4	3
PIZZA	1	0	5	6	5	1	1	5	6

Q=(A; 'IRISH', 'PUB')





PriorityQueue for increasing values of $Dst(Q,O)$ or $MINDst(Q,N)$

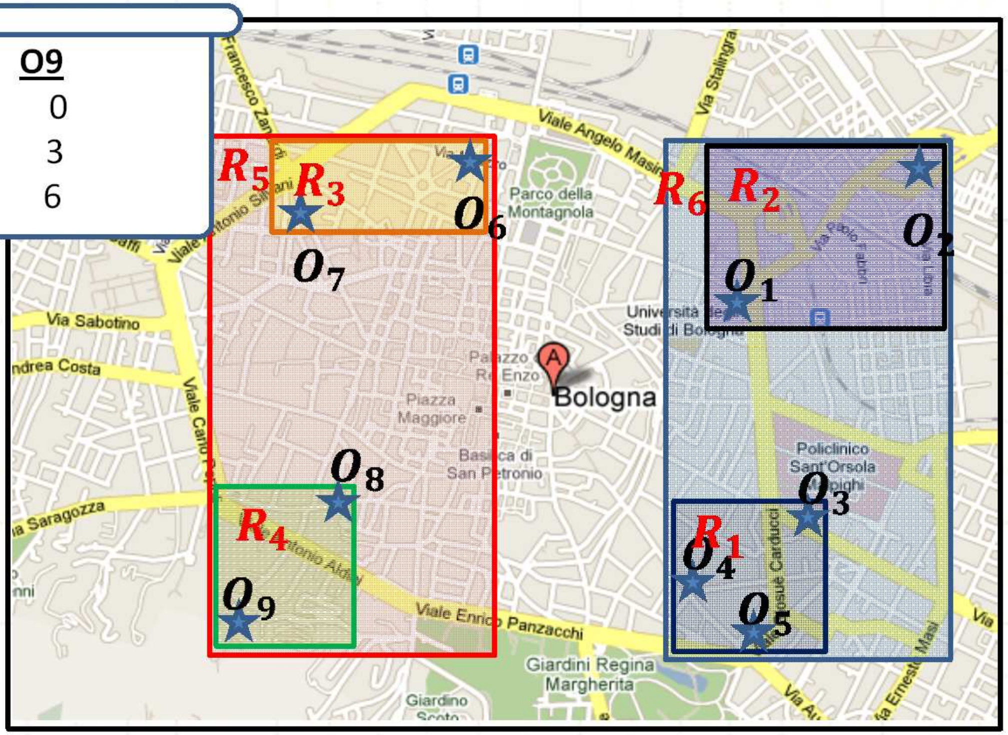
1) | R7 |
   0

2) | R6 | R5 |
   0,1   0,16

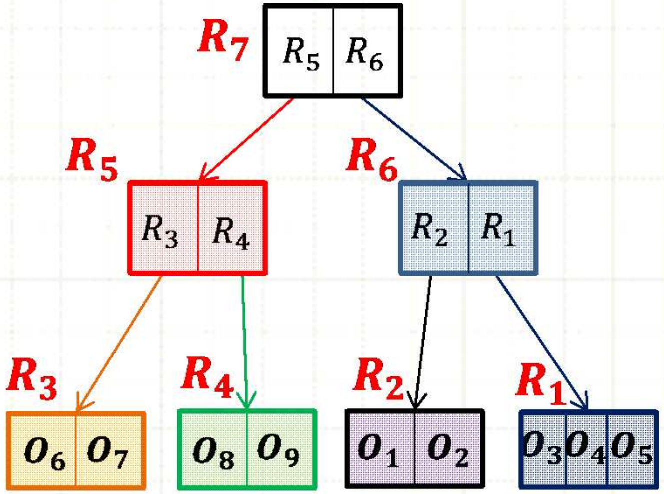
3) | R1 | R5 | R2 |
   0,15   0,16   0,19

```
Queue ← NewPriorityQueue();
Queue.Enqueue(Index.RootNode, 0);
while not Queue.IsEmpty() do
    Element ← Queue.Dequeue();
    if Element is an object then
        if not Queue.IsEmpty() and D_ST(Query, Object) >
              Queue.First().Key then
              Queue.Enqueue(Object, D_ST(Query, Object));
        else
              Report Element as the next nearest object;
              if k nearest objects have been found then
                  break;
    else if Element is a leaf node then
        for each entry(Object) in leaf node Element do
              Queue.Enqueue(Object, D_ST(Query, Object));
    else
        for each entry(Node) in node Element do
              Queue.Enqueue(Node, MIND_ST(Query, Node));
```

	O1	O2	O3	O4	O5	O6	O7	O8	O9
PUB	5	2	5	0	2	3	4	0	0
IRISH	5	4	0	5	0	3	2	4	3
PIZZA	1	0	5	6	5	1	1	5	6



Q=(A; 'IRISH', 'PUB')



PriorityQueue for increasing values of $Dst(Q,O)$ or $MINDst(Q,N)$

4)

R5	R2	O3	O4	O5
0,16	0,19	0,46	0,58	0,66

5)

R2	R3	O3	R4	O4	O5
0,19	0,25	0,46	0,48	0,58	0,66

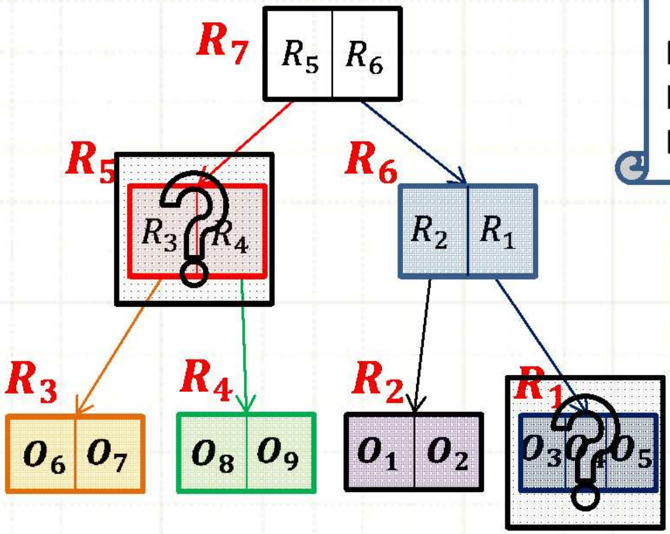
6)

O1	R3	O3	R4	O4	O5	O2
0,23	0,25	0,46	0,48	0,58	0,66	0,7

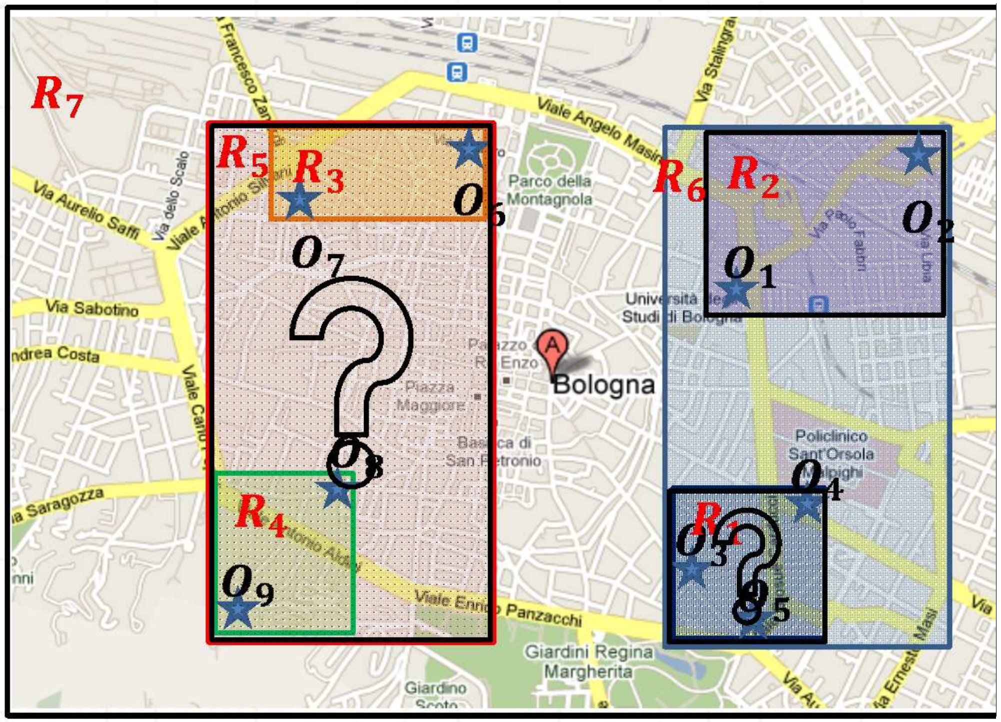
```
Queue ← NewPriorityQueue();
Queue.Enqueue(Index.RootNode, 0);
while not Queue.IsEmpty() do
    Element ← Queue.Dequeue();
    if Element is an object then
        if not Queue.IsEmpty() and D_ST(Query, Object) >
            Queue.First().Key then
            Queue.Enqueue(Object, D_ST(Query, Object));
        else
            Report Element as the next nearest object;
            if k nearest objects have been found then
                break;
    else if Element is a leaf node then
        for each entry(Object) in leaf node Element do
            Queue.Enqueue(Object, D_ST(Query, Object));
    else
        for each entry(Node) in node Element do
            Queue.Enqueue(Node, MIND_ST(Query, Node));
```

Considerations

Q=(A; 'IRISH', 'PUB')



	O1	O2	O3	O4	O5	O6	O7	O8	O9
PUB	5	2	5	0	2	3	4	0	0
IRISH	5	4	0	5	0	3	2	4	3
PIZZA	1	0	5	6	5	1	1	5	6



In the previous example we explore nodes R5 and R1, even if they do not contain really interesting objects ...

The guilty is the pseudo-document because inherits the best of all the documents inside its subtree,
and those documents may be very different between them

To avoid this we would like that inside a node there are:

- <u>SIMILAR DOCUMENTS</u>
- Near objects

DIR-tree

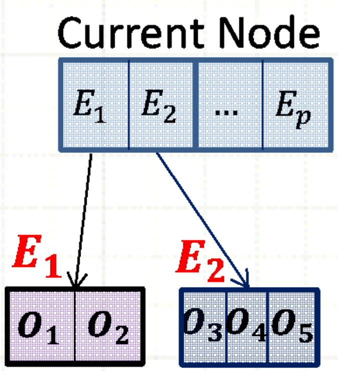
DIR-Tree

Unlike the IR-tree, the DIR-tree aims to take both location and text information into account during tree construction, by optimizing for a combination of minimizing the areas of the enclosing rectangles and maximizing the text similarities between the documents of the enclosing rectangles.

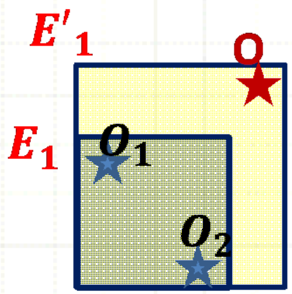
We need an heuristic function to evaluate in which leaf node insert an item during tree construction.

$$SimAreaCost(E_k, O) = (1-\beta)\frac{AreaCost(E_k)}{maxArea} + \beta(1 - cosine(E_k.Vector, O.vector))$$

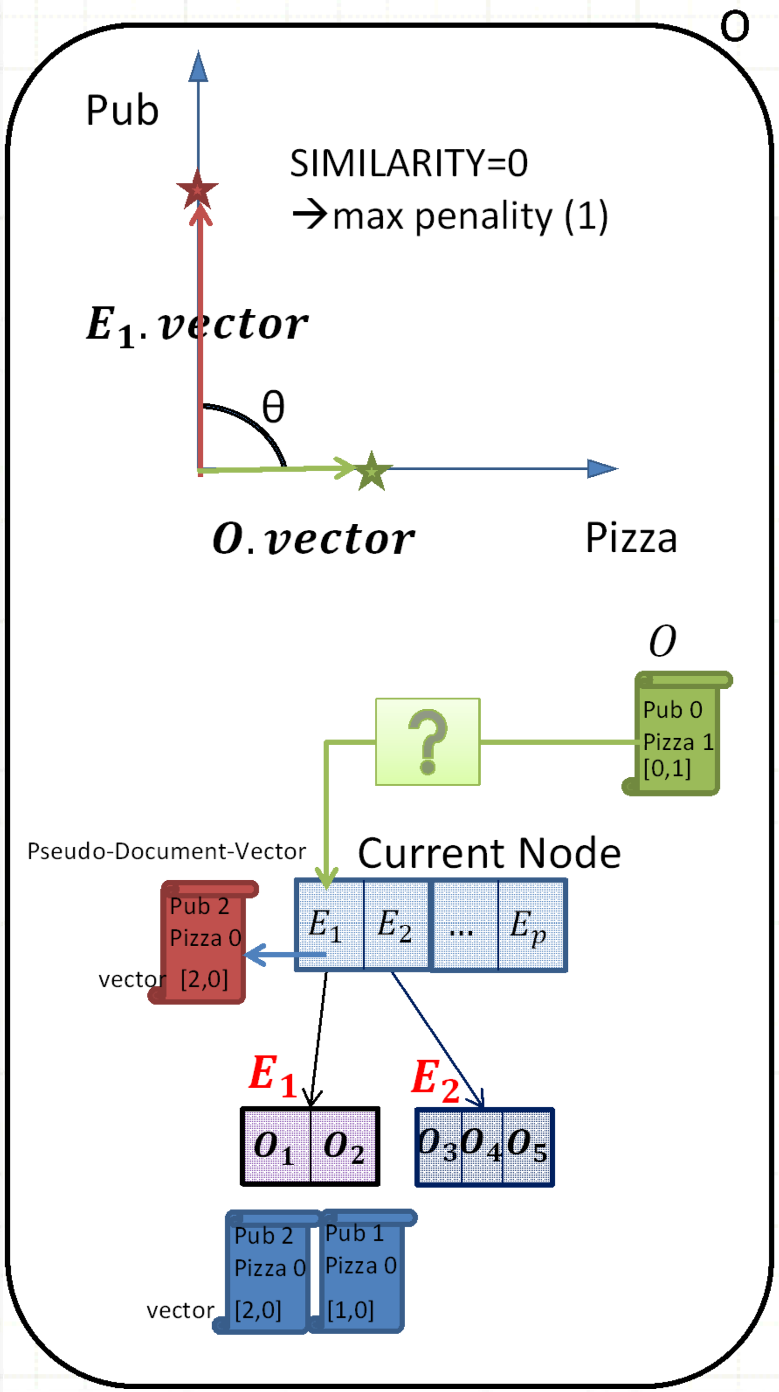
E_k = generic element of a non-leaf node
O = object to insert in the tree

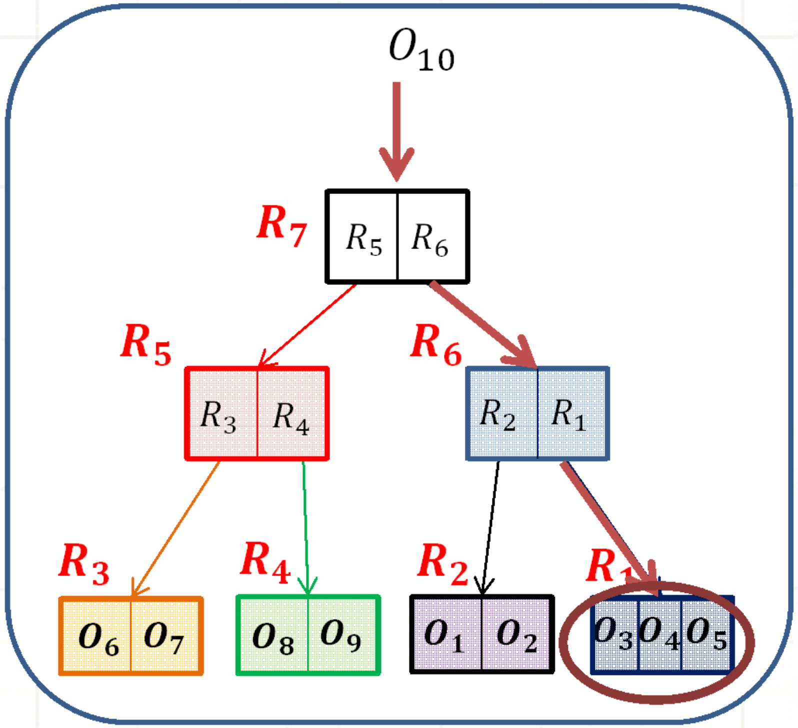


$$AreaCost(E_k) = area(E'_k.rectangle) - area(E_k.rectangle)$$



$$E'_k = area((E_k \cup O).rectangle)$$





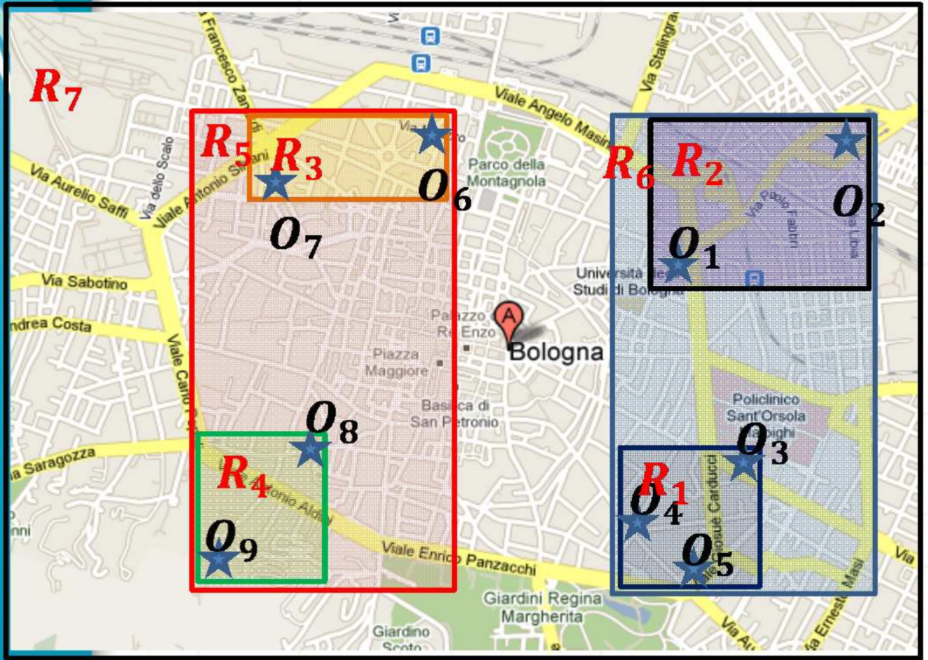
O.Vector = term frequency vector of the document O

E_k.**Vector** = term frequency vector of the pseudo-document related to E_k

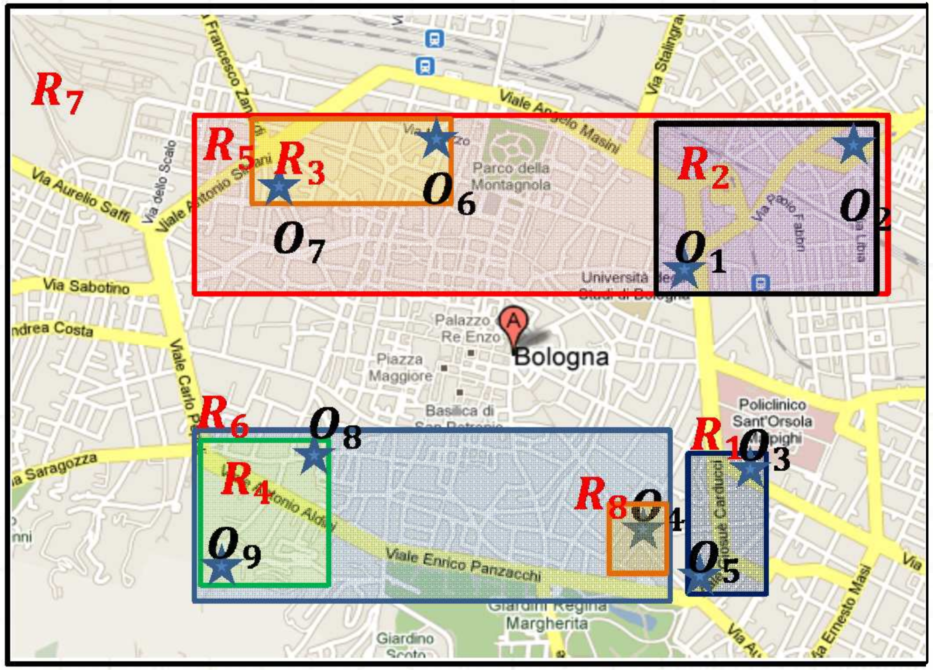
Here the benefits

	O1	O2	O3	O4	O5	O6	O7	O8	O9
PUB	5	2	5	0	2	3	4	0	0
IRISH	5	4	0	5	0	3	2	4	3
PIZZA	1	0	5	6	5	1	1	5	6

IR-Tree



DIR-Tree

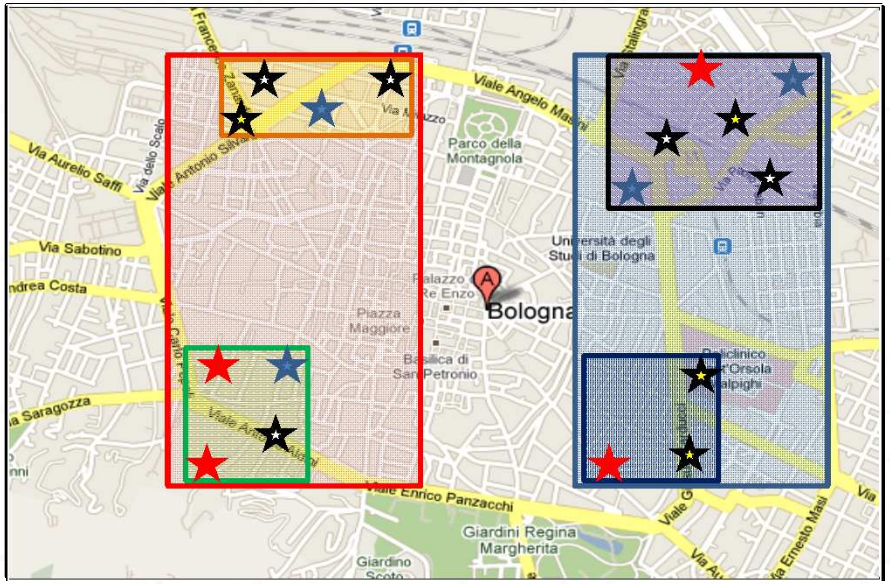


How to improve performances …

- Clustering :

Spatial web objects often belong to different categories, for example accommodations, restaurants, and tourist attractions.

The idea is to cluster objects into groups according to their corresponding documents.

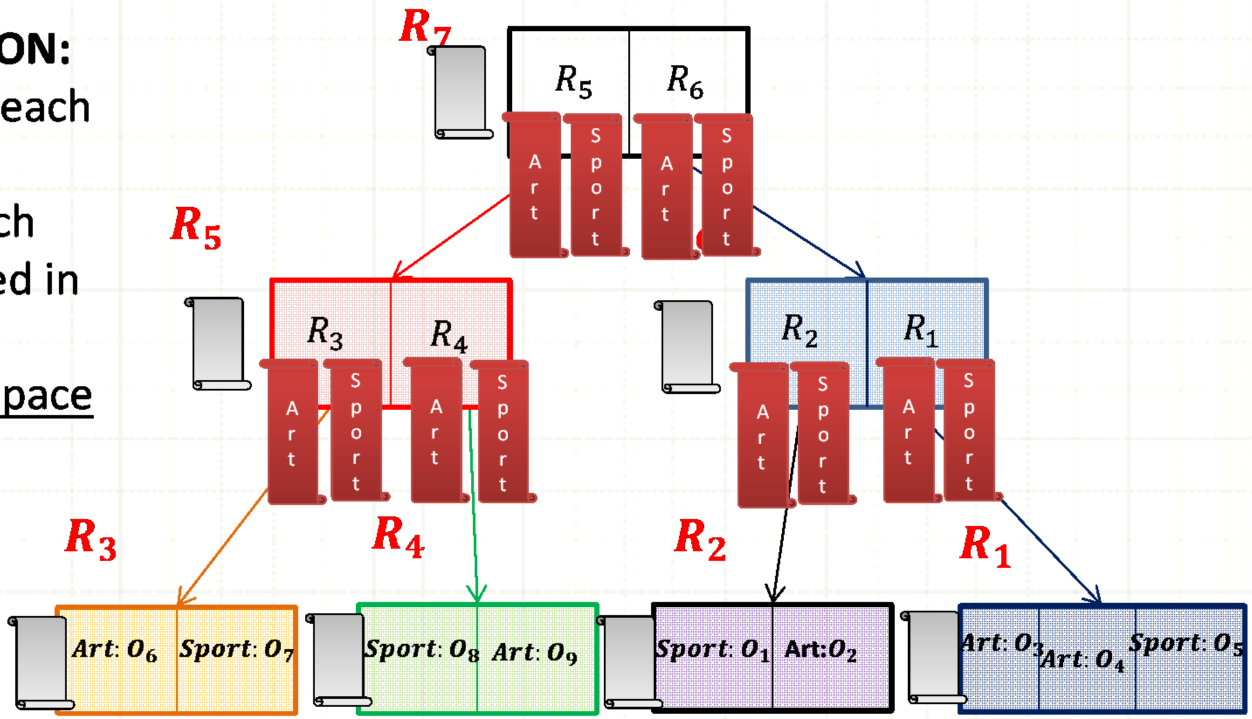


But what does it involve from the point of view of:

- Efficiency?
- Changes in the trees(IR or DIR) structure?

CLUSTER 1 → Art(O_6, O_9, O_2 , O_3,O_4)
CLUSTER 2 → Sport(O_7,O_8, O_1 , O_5)

STRUCTURE MODIFICATION:
We need to memorize in each entry of each node one pseudo-document for each kind of cluster represented in a subtree
This imply an additional space usage



ADVANTAGES
bounds estimated using clusters in a node will be tighter than those estimated for whole nodes

The problem of finding a clustering solution that minimizes ScanTime is NP-hard.

Performances results

Property	Dataset
Total number Of objects	131.461
Average number of unique words per objects	112
Total number of unique words in dataset	30.616
Total number of words in dataset	14.809.845

Baseline
IR-tree
CIR-tree
DIR-tree
CDIR-tree