An Ontology-Based Information Retrieval Model

M. Fernández, D. Vallet, P. Castells

Universidad Autónoma de Madrid
Escuela Politécnica Superior

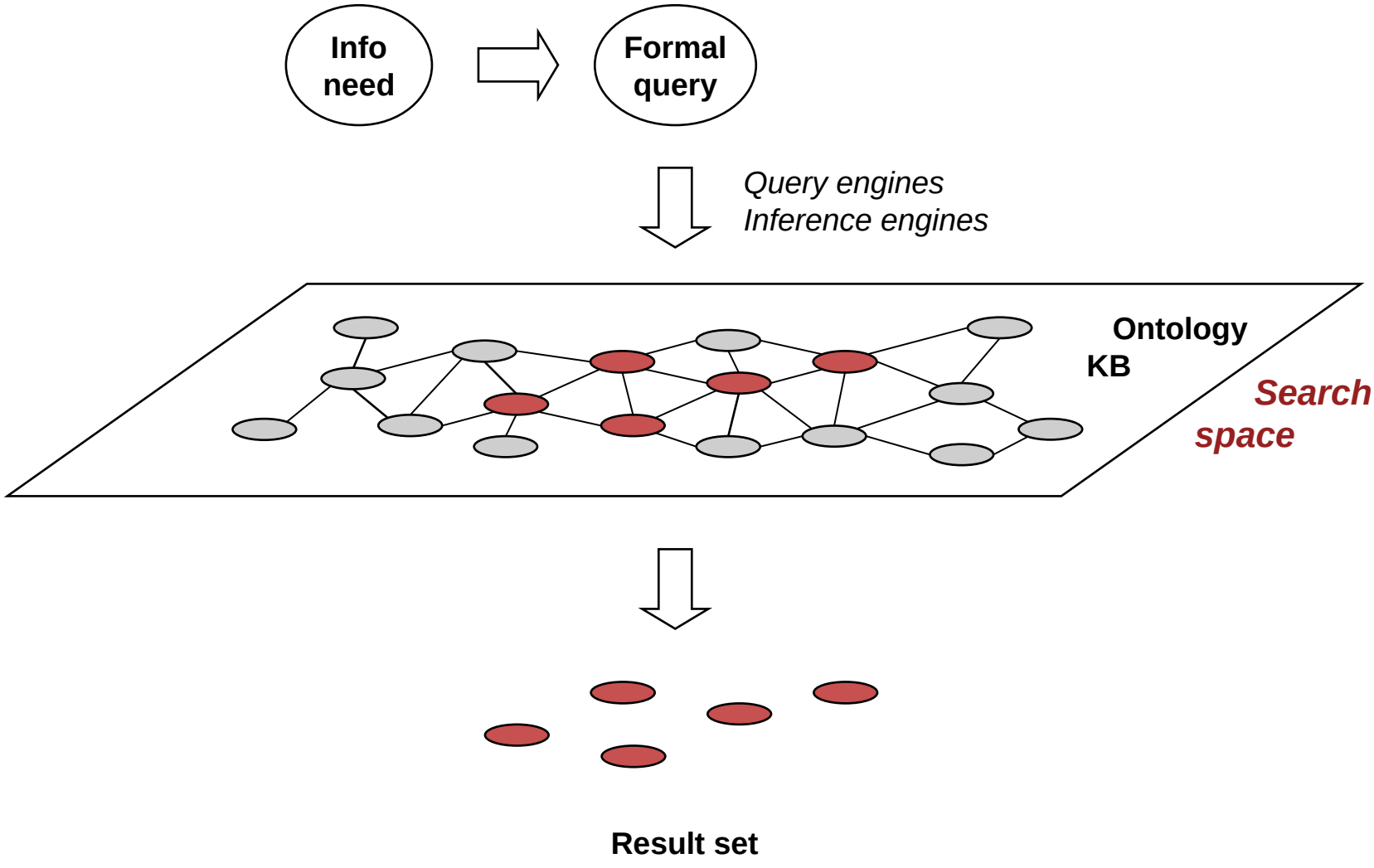
Ontology Workshop
Strasbourg, France, 25 October 2005

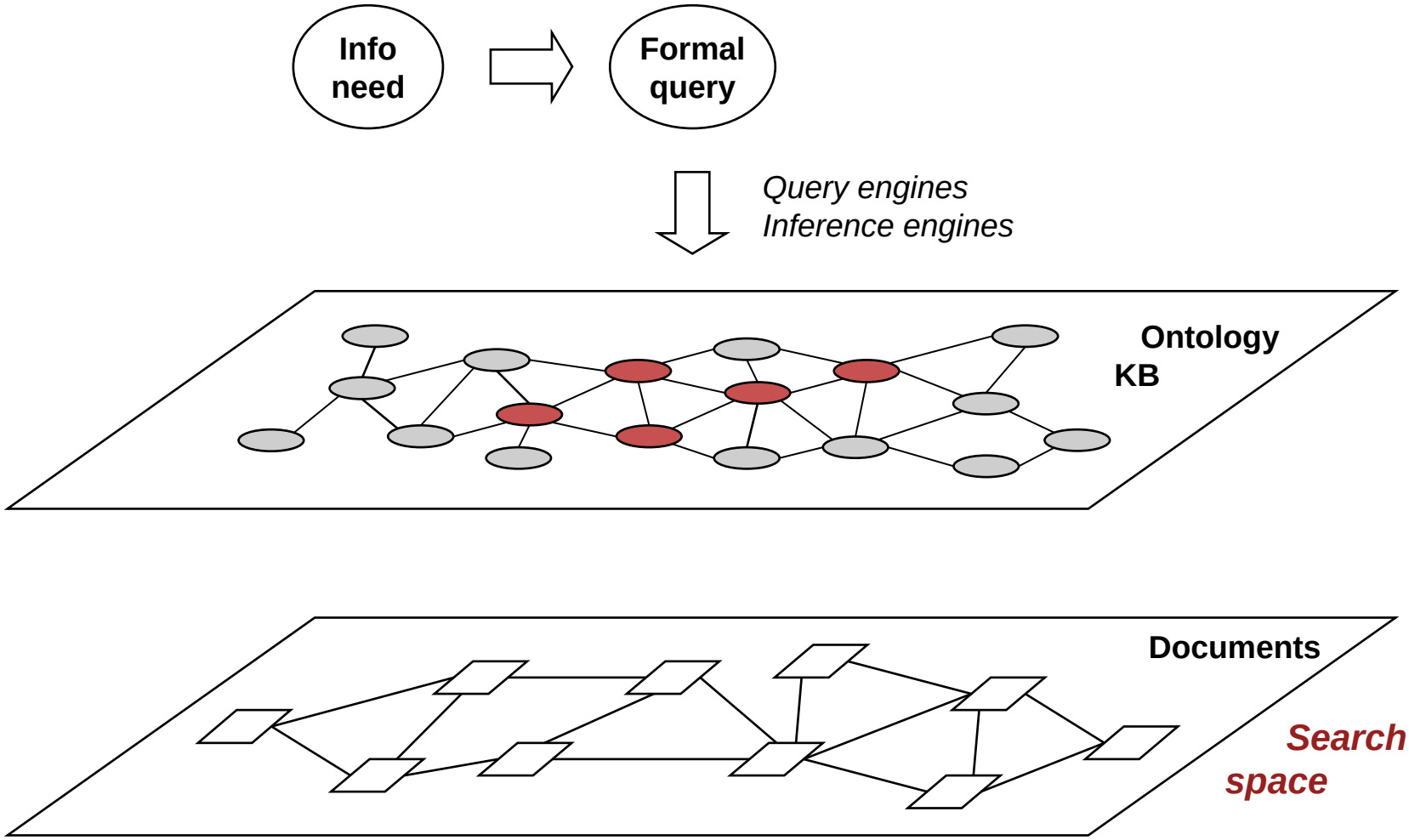
Contents

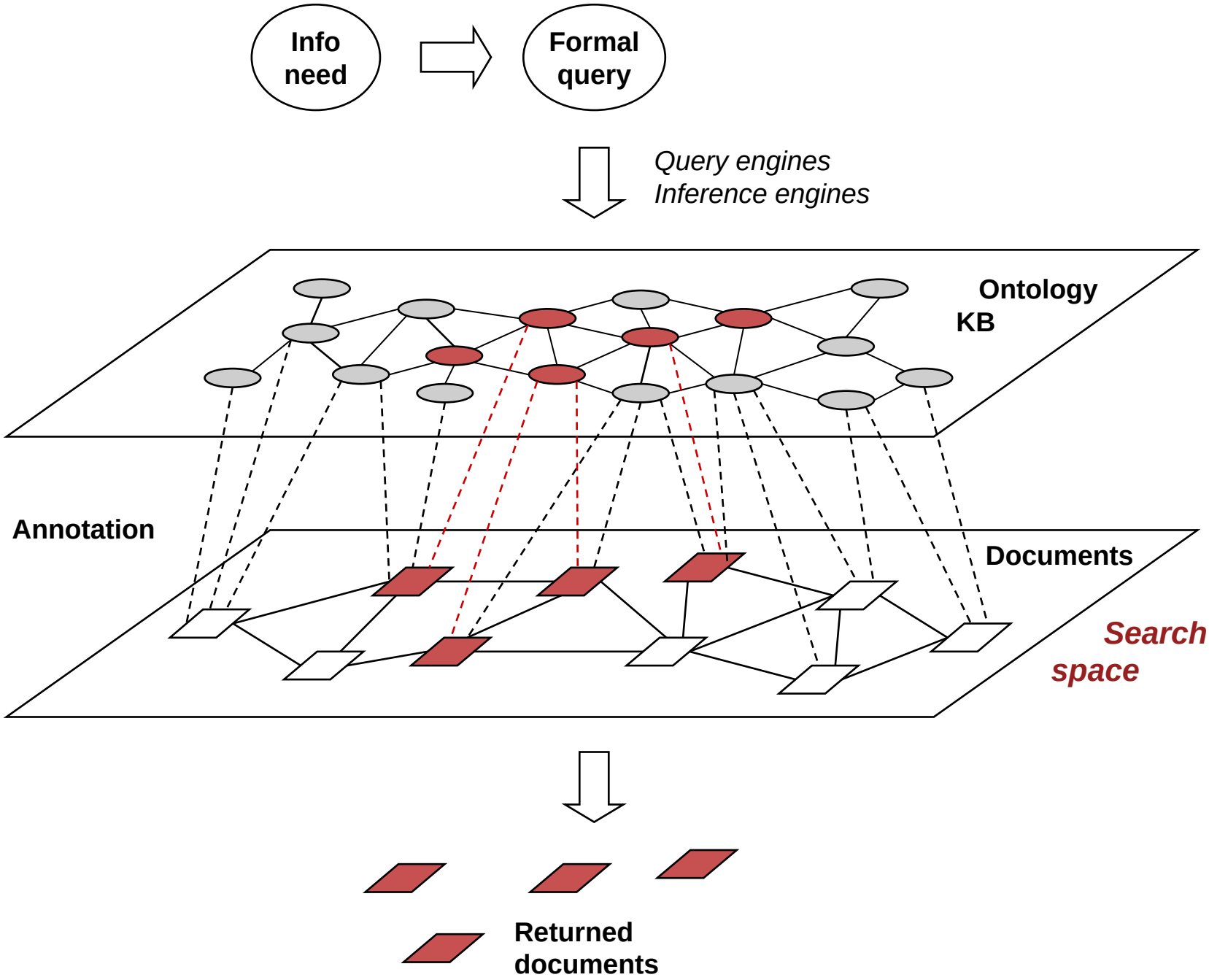
- The problem
- The model
- Experiments
- Conclusions

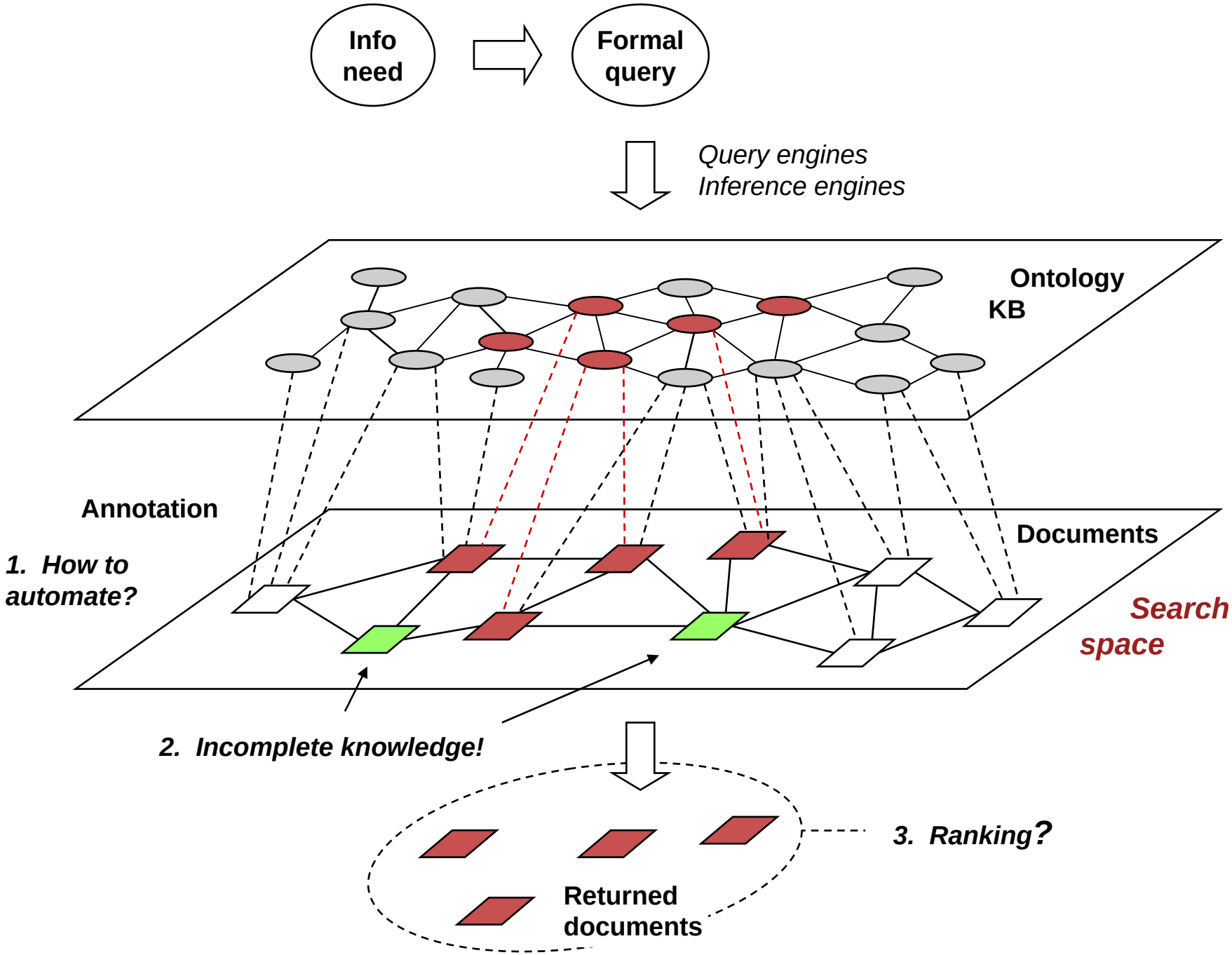
Definition of the Problem

- Use ontologies and KBs to improve keyword-based search
- Information need → formal ontology-based query (e.g. RDQL)
- Final search space = collection of documents
- Imperfect, approximate search: formal-semantics (*document*) ¬ full-semantics (*document*)
- Document ranking
- Assumption: incomplete ontologies, incomplete KBs


Info need
Formal query
Query engines
Inference engines
Ontology KB
Search space
Result set


Info need
Formal query
Query engines
Inference engines
Ontology KB
Documents
Search space


Info need
Formal query
Query engines
Inference engines
Ontology KB
Annotation
Documents
Search space
Returned documents

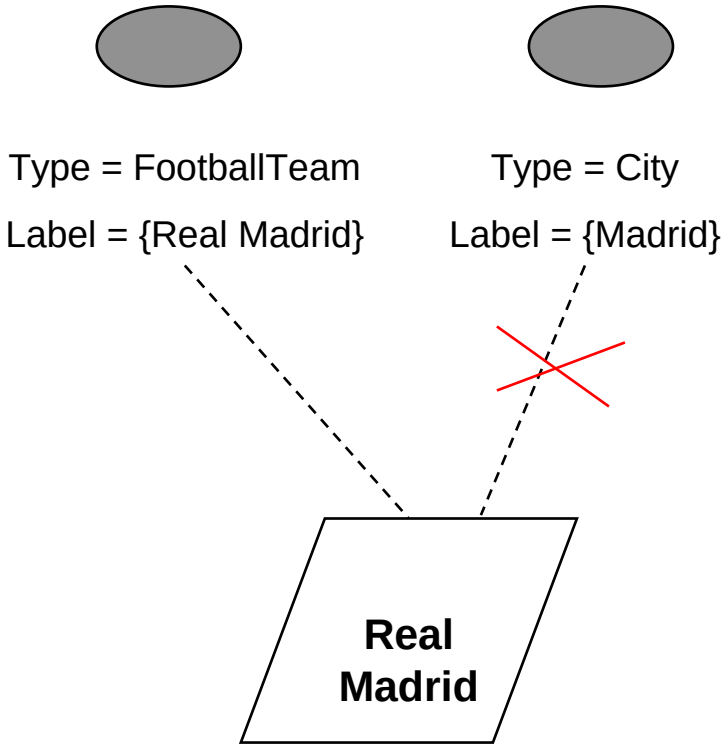

Info need
Formal query
Query engines
Inference engines
Ontology KB
Annotation
Documents
1. How to automate?
Search space
2. Incomplete knowledge!
3. Ranking?
Returned documents

Document Annotation

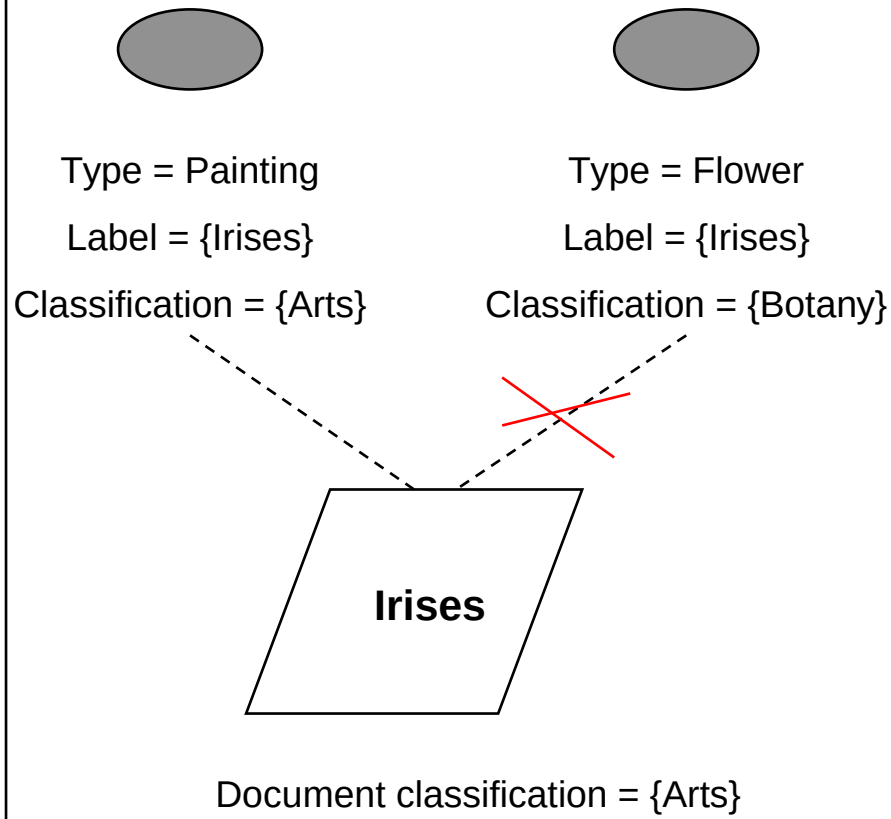
- Use non-embebed annotations for the semantic indexing of documents.
- Manual annotations vs. automatic annotations.
- Use a label property (multivaluated) to store the most usual text form(s) of the concept class or instances to find potential occurrences of those instances in text documents.
- Use of heuristics to cope with polysemia.
- Use of classification taxonomies as a source of semantic scope for disambiguation.

Disambiguation Heuristics

Using the property label



Using Taxonomies

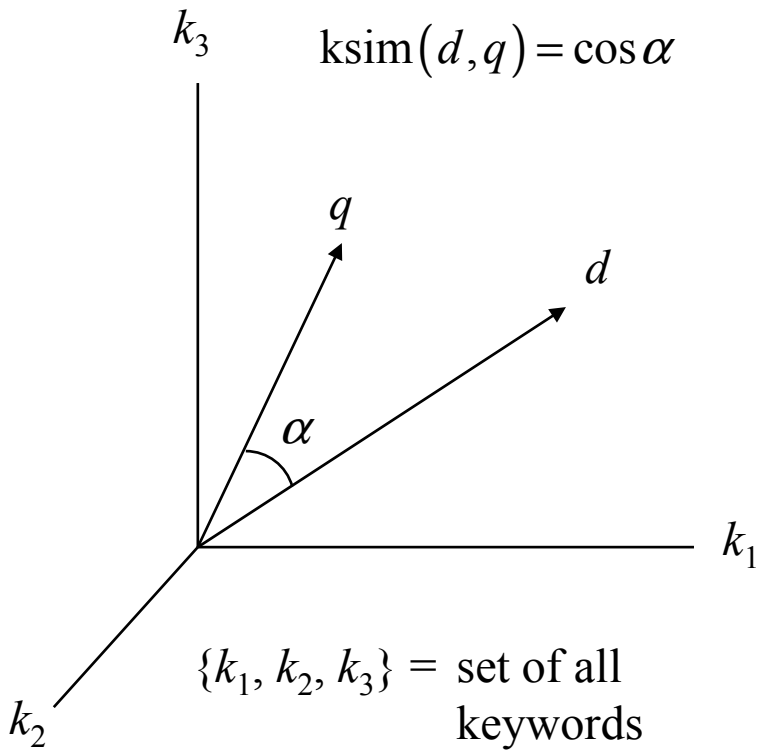


Adapting the Vector-Space IR Model

Keyword-Based IR Model

Query keyword-vector q

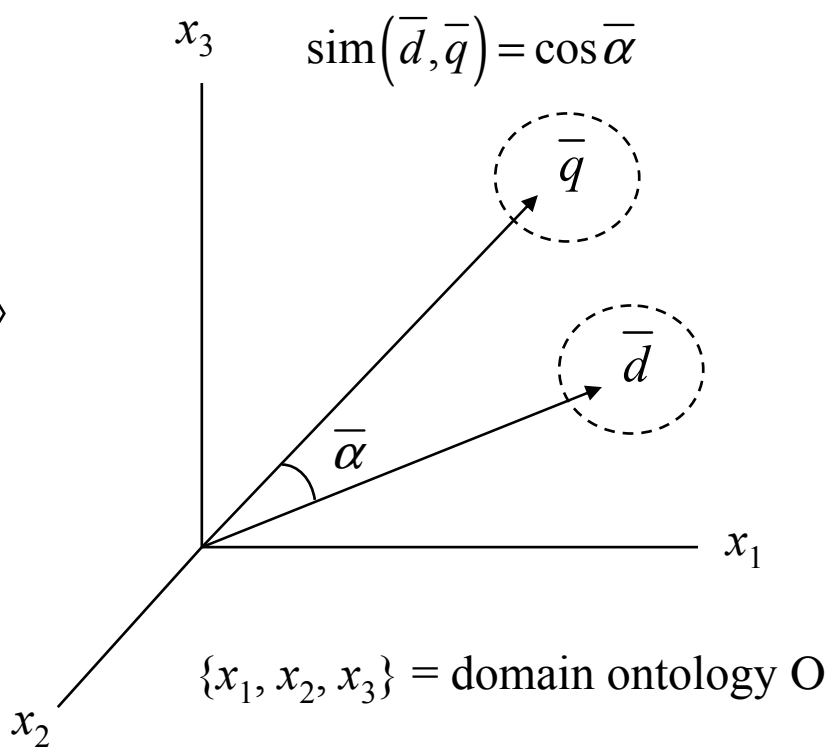
Document keyword-vector d



Semantic IR Model

Result-set concept-vector $\bar{q}$

Document concept-vector $\bar{d}$



Semantic Vector Space

- Building the query vector $\overline{q}$
  - Execute the query (e.g. RDQL) → Result set $R \subset \mathrm{O}^{|V|}$
  - Variable weighs: for each variable $v \in V$ in the query, $w_v \in [0,1]$
  - For each $x \in \mathrm{O}$,

$$\overline{q}_x = \begin{cases} w_v \text{ if } x \text{ instantiates } v \text{ in some tuple in } R \\ 0 \text{ otherwise} \end{cases}$$

- Building the document vector $\overline{d}$
  - Map concepts to keywords
  - Weight for an instance $x \in \mathrm{O}$ that annotates a document d: TF-IDF

$$\overline{d}_x = \frac{freq_{x,d}}{\max_{y \in O} freq_{y,d}} \cdot \log \frac{N}{n_x}$$

$freq_{x,d}$ = number of occurrences of keywords of x in d

n_x = number of documents annotated by x

N = total number of documents

Example: RDQL Query

"News articles about players from USA playing in basketball teams of Catalonia"

```
SELECT  ?player, ?team
WHERE   (?player <rdf:type> <SportsPlayer>)
        (?player <sports> <Basketball>)
        (?player <nationality> <USA>)
        (?player <playsIn> ?team)
        (?team <rdf:type> <SportsTeam>)
        (?team <locatedIn> <Catalonia>)
```

Query Vector

- Variable weights: e.g. $w_{player} = 1.0$, $w_{team} = 0.5$
- Result set:

player	*team*
Aaron Jordan Bramlet	*Caprabo Lleida*
Derrick Alston	*Caprabo Lleida*
Venson Hamilton	*DKV Joventut*
Jamie Arnold	*DKV Joventut*

- Query vector $\overline{q} = (0, 0, \ldots, 0, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{1}, \mathbf{0.5}, \mathbf{0.5}, 0, \ldots, 0, 0)$

Search Result

- Found documents: 66 news articles ranked from 0.1 to 0.89
- E.g. 1st document

*"Johnny Rogers and Berni Tamames went yesterday through the medical revision required at the beginning of each season, which consisted of a thorough exploration and several cardiovascular and stress tests, that their **team** mates had already passed the day before. Both **players** passed without major problems the examinations carried through by the medical **team** of the club, which is now awaiting the arrival of the Northamericans **Bramlett** and **Derrick Alston** to conclude the revisioning."*

Document vector $\overline{d} = (\ldots, \mathbf{1.73}, \ldots, \mathbf{1.73}, \ldots)$

Semantic rank value: $\cos(\overline{d}, \overline{q}) = 0.89$

Keyword rank value: $\cos(d, q) = 0.02$

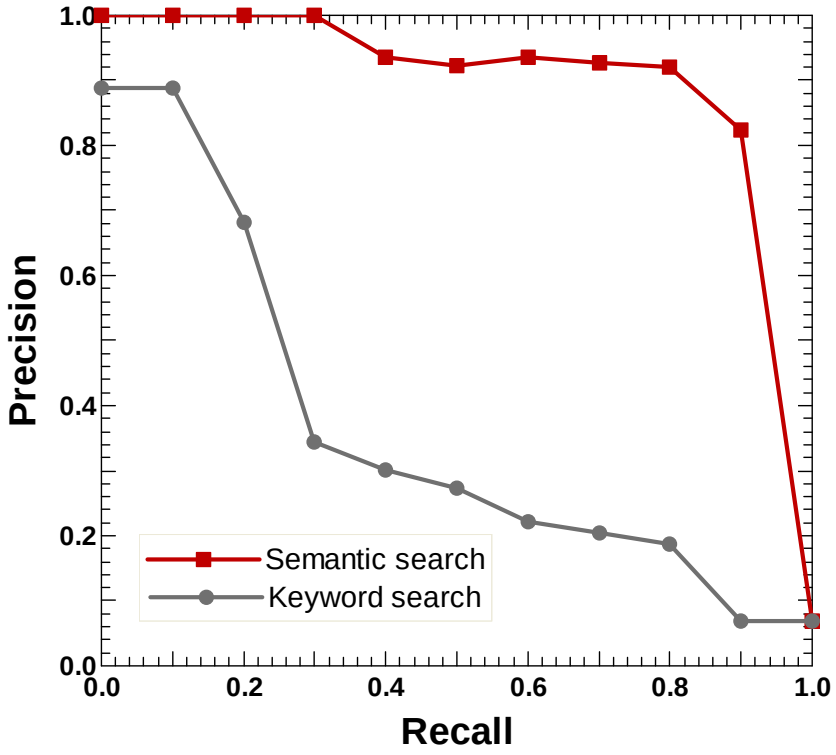
} Combined rank value: 0.89

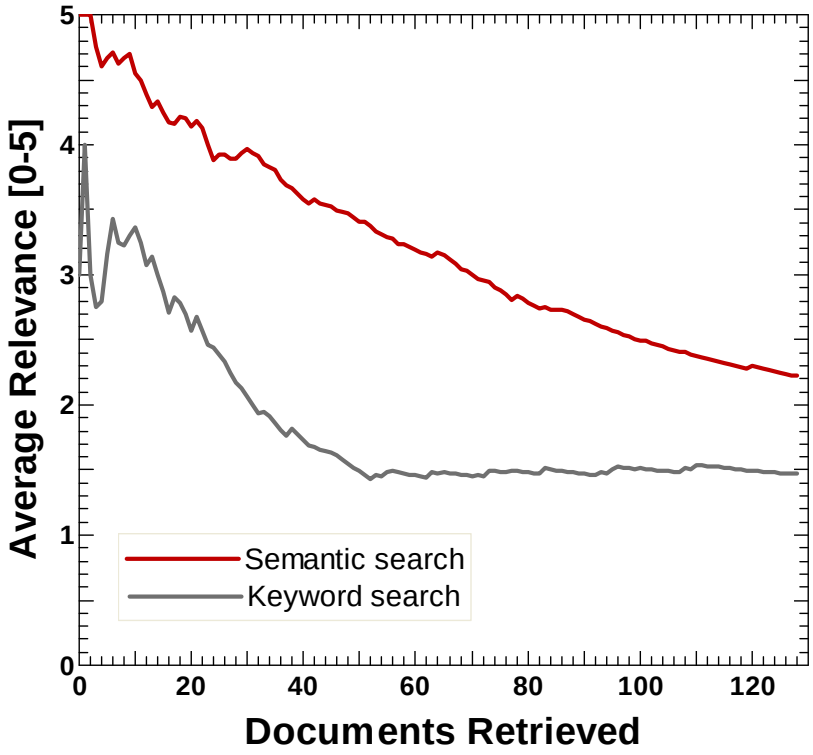
Experiments

- Document collection: news articles from the CNN web site
  - 145,316 news articles (445 MB)
- KIM domain ontology and KB developed by Ontotext Lab
  - Some minor extensions and adjustments
  - 281 domain classes, 138 properties,
  - 35,689 instances, 465,848 sentences.
  - 71 MB in RDF text format
- Annotation
  - Over 3 · 10^6 automatic annotations (i.e. over 25 per document on average)
- Query, retrieval and ranking
  - Comparison of semantic ranking and keyword-based ranking (Jakarta Lucene)
  - Manual ranking of documents from 0 to 5
  - $w_v = 1$ for all v in the RDQL queries

Experiments – Query a

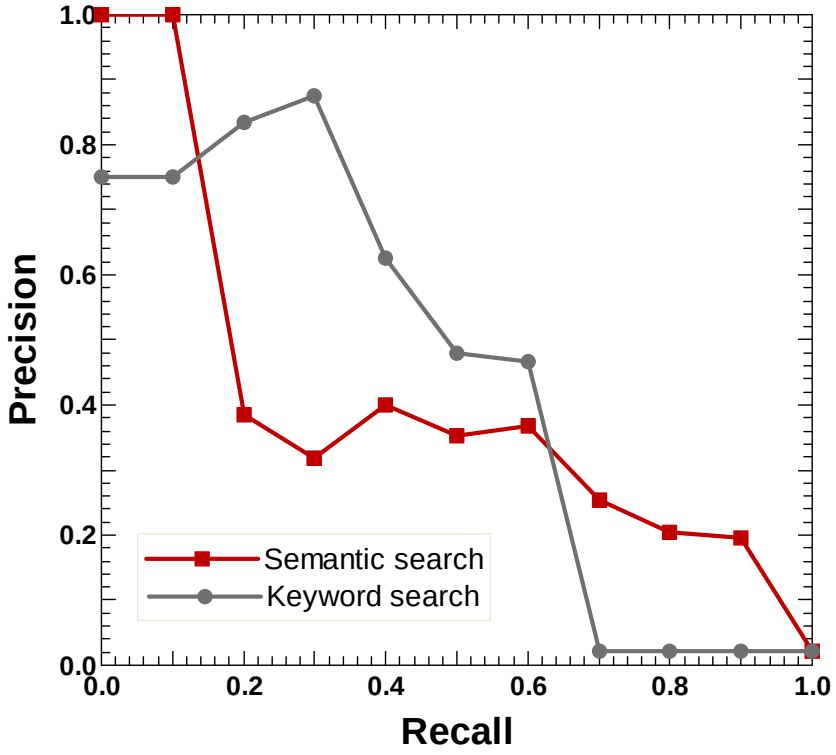
"News articles about players from USA playing in basketball teams of Catalonia"

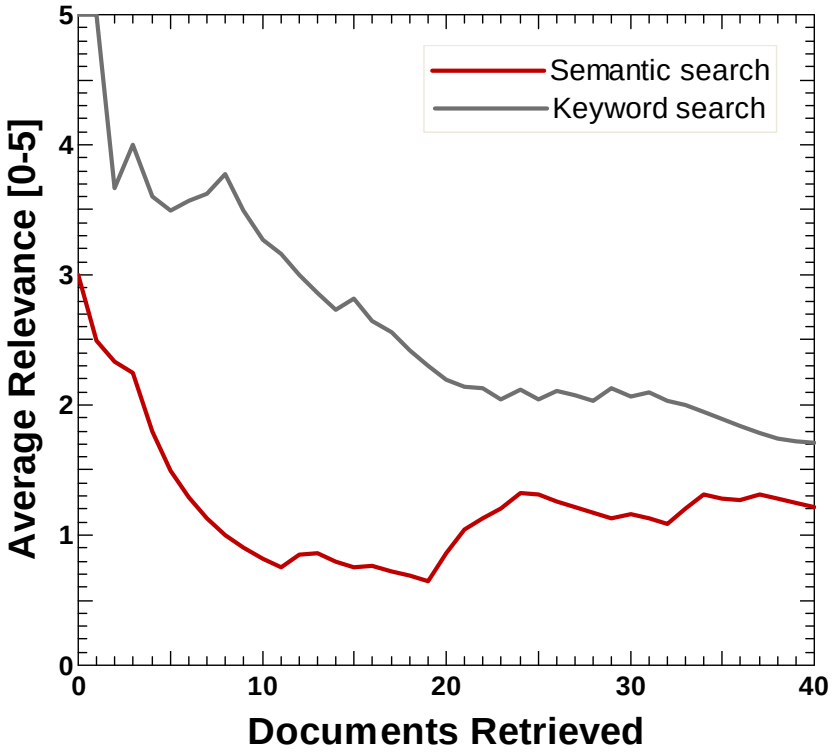




Experiments – Query b

"*News articles about the European Union*"





Improvements

- Better precision by using structured semantic queries (more precise information needs)
  - E.g. a football player **playing in** the Juventus vs. **playing against** the Juventus
- Better recall when querying for instances by class (query expansion)
  - E.g. "News about **companies** quoted on NASDAQ"
- Better recall by using inference
  - E.g. "Watersports in Spain" → ScubaDiving, Windsurf, etc. in Cádiz, Málaga, Almería, etc.
- Better precision by using query weights
  - E.g. new articles about car models released this year, where the **release date** is not necessarily mentioned
- Better precision by reducing polysemic ambiguities
  - Use of instances label and classification of concepts and documents
- Conditions on concepts and conditions on documents
  - E.g. **film review** published by "Le Monde" within the last 7 days about sci-fi **movi**

Conclusions

- A proposal for ontology-based information retrieval
  - Document retrieval vs. formal query answering
  - Ranking algorithm based on an adaptation of the vector-space model
  - Assume incomplete KB and ontologies
  - The improvements increase with the number of clausses in the formal query
- Limitations
  - KBs are hard and expensive to build, maintain, and difficult to share
  - Resort to keyword-based search when insufficient knowledge available
- Future work
  - Personalization
    - Contextual personalization

Thank you!