

# Evaluation of an Information Retrieval System for the Semantic Desktop using Standard Measures from Information Retrieval

---

*aposdle – New ways ...*

*... to work, learn and collaborate!*

Peter Scheir, Michael Granitzer, Stefanie N. Lindstaedt

## Starting point

- **We built an information retrieval system for the Semantic Desktop [Scheir et al. 2007]**
  - Semantic Web technology in desktop environment
- **How evaluate it?**

## Current situation

- **System for information retrieval in the Semantic Web / Desktop exist (see talk on Wednesday! ;)**
- **Not many are evaluated using classical measures**
- **No standard test corpus is available**

## Our work

- **Evaluation of an information retrieval system for the semantic desktop**
- **Using:**
  - Precision at ranks 10, 20 and 30
  - Inferred average precession (infAP)

## Basic search approach

- Retrieval of documents based on concepts stemming from a knowledge representation
- Information retrieval, list ranked by relevance
- Network model

## Extended search approach

- Query and result expansion
- Relate similar concepts by semantic similarity
- Relate similar documents by textual similarity
- Model associations in network and consider them during search
  
- See if approach provides better results as without associations

## Our corpus

- **Knowledge and document base from first release of APOSDLE system**
- **Requirements Engineering as application domain**
  - Requirements Engineering ontology
  - Requirements Engineering document base
- **Statistics**
  - 70 concepts in ontology
  - 1016 documents in document base
  - 21 concepts used for annotation
  - 496 documents annotated

# Evaluation approach

## ■ Configurations

- 8 different system configurations (with semantic and / or textual similarity)

## ■ Queries

- 79 queries

## ■ Judgments

- First 30 results
- Results overlapping in result sets of different configurations only judged once
- All judgments by the same person

## ■ Measures

- P(10), P(20), P(30)
- infAP [Yilmaz and Aslam, 2006]
- calculated using trec\_eval

## Evaluation results

Conf.	P(10)	P(20)	P(30)	infAP
conf_1	0.2418	0.2051	0.1700	0.1484
conf_2	0.3089	0.2778	0.2502	0.2487
conf_3	0.3165	0.2608	0.2131	0.2114
conf_4	0.3114	0.2582	0.2097	0.2001
conf_5	0.3114	0.2582	0.2097	0.2000
conf_6	0.3848	0.3405	0.3046	0.3253
conf_7	0.3924	0.3494	0.3089	0.3326
conf_8	0.3911	0.3487	0.3080	0.3318

## Evaluation results – query expansion

Conf.	P(10)	P(20)	P(30)	infAP
conf_1	0.2418	0.2051	0.1700	0.1484
conf_2	0.3089	0.2778	0.2502	0.2487
conf_3	0.3165	0.2608	0.2131	0.2114
conf_4	0.3114	0.2582	0.2097	0.2001
conf_5	0.3114	0.2582	0.2097	0.2000
conf_6	0.3848	0.3405	0.3046	0.3253
conf_7	0.3924	0.3494	0.3089	0.3326
conf_8	0.3911	0.3487	0.3080	0.3318

## Evaluation results – result expansion

Conf.	P(10)	P(20)	P(30)	infAP
conf_1	0.2418	0.2051	0.1700	0.1484
conf_2	0.3089	0.2778	0.2502	0.2487
conf_3	0.3165	0.2608	0.2131	0.2114
conf_4	0.3114	0.2582	0.2097	0.2001
conf_5	0.3114	0.2582	0.2097	0.2000
conf_6	0.3848	0.3405	0.3046	0.3253
conf_7	0.3924	0.3494	0.3089	0.3326
conf_8	0.3911	0.3487	0.3080	0.3318

## Evaluation results – nearly identical

Conf.	P(10)	P(20)	P(30)	infAP
conf_1	0.2418	0.2051	0.1700	0.1484
conf_2	0.3089	0.2778	0.2502	0.2487
conf_3	0.3165	0.2608	0.2131	0.2114
conf_4	0.3114	0.2582	0.2097	0.2001
conf_5	0.3114	0.2582	0.2097	0.2000
conf_6	0.3848	0.3405	0.3046	0.3253
conf_7	0.3924	0.3494	0.3089	0.3326
conf_8	0.3911	0.3487	0.3080	0.3318

## Discussion of $P(n)$

- [Buckley and Voorhees, 2000] suggest that 50 queries should be used for  $P(30)$
- For  $P(n < 30)$  number of queries should be increased
- 100 queries are suggested for  $P(20)$
  
- We use 79 queries for  $P(10)$ ,  $P(20)$  and  $P(30)$
- Ranking of system configurations is identical for  $P(20)$ ,  $P(30)$  and infAP

## Discussion of infAP (1)

- TREC 8 Ad-Hoc collection
  - 528,155 documents \* 50 queries = 26,407,750 possible relevance judgments
  - 86,830 query documents pairs judged
    - Depth-100 pooling of 129 systems
  - 0.33% of all possible relevance judgments performed
- Our collection
  - 1026 documents \* 79 queries = 81,054 possible relevance judgments
  - 1938 query documents pair judged
    - Depth-30 pooling of 8 configurations
    - 498 additional judgments
  - 2.39% of all possible relevance judgments performed

## Discussion of infAP (2)

- Depth-100 pool would be 4138 query document pairs
- We judged 46.83% of potential depth-100 pool
- [Yilmaz and Aslam, 2006] find that with 25% of judgments ranking of systems is identical to 100% in TREC 8 Ad-Hoc
- Ranking of system configurations is identical for P(20), P(30) and infAP

# Thank you for your attention!

- Questions / Comments?
- [pscheir@know-center.at](mailto:pscheir@know-center.at)
- <http://www.aposdle.org/>



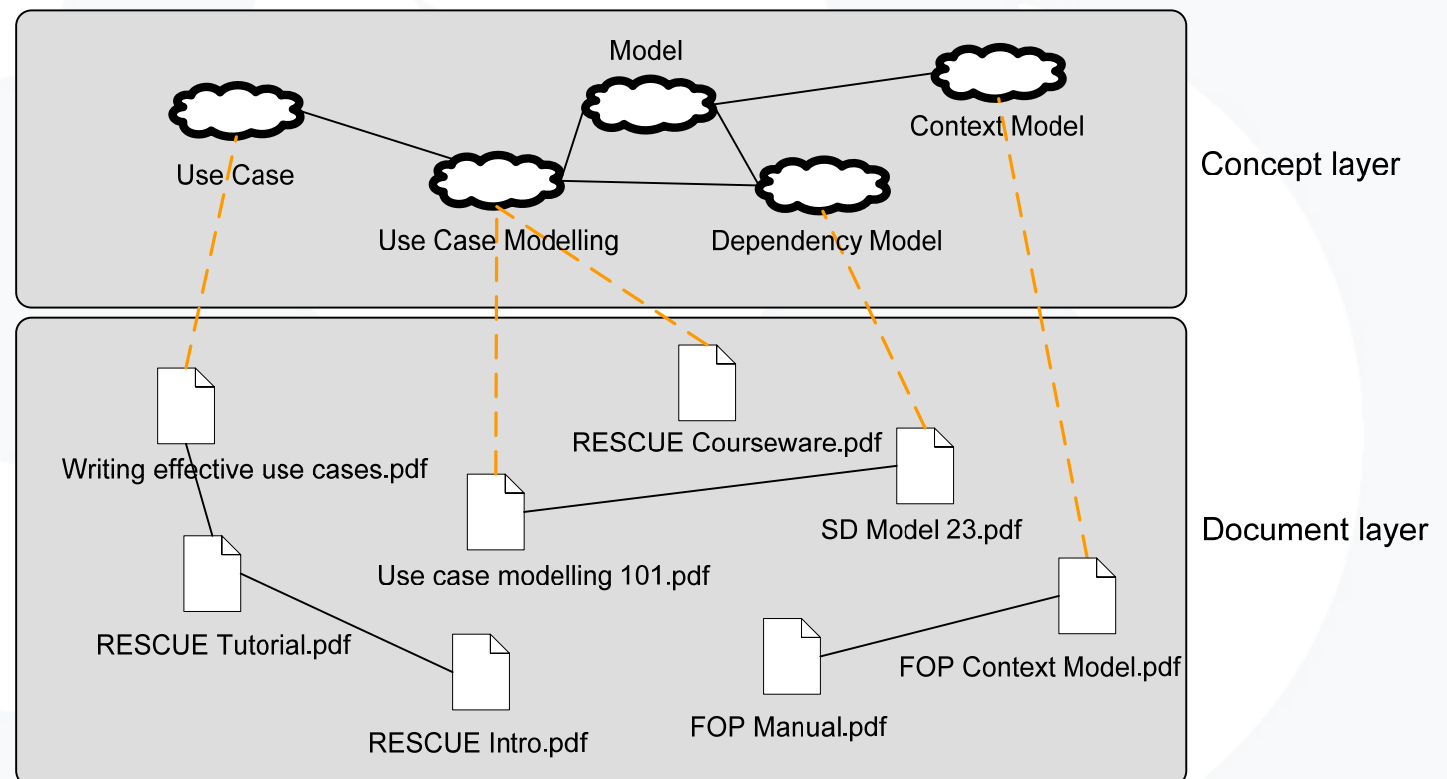
## Our questions

### ■ What do you think about

- the size of the collection?
- the amount of queries?
- the amount of judgments?
- the measures that were used?

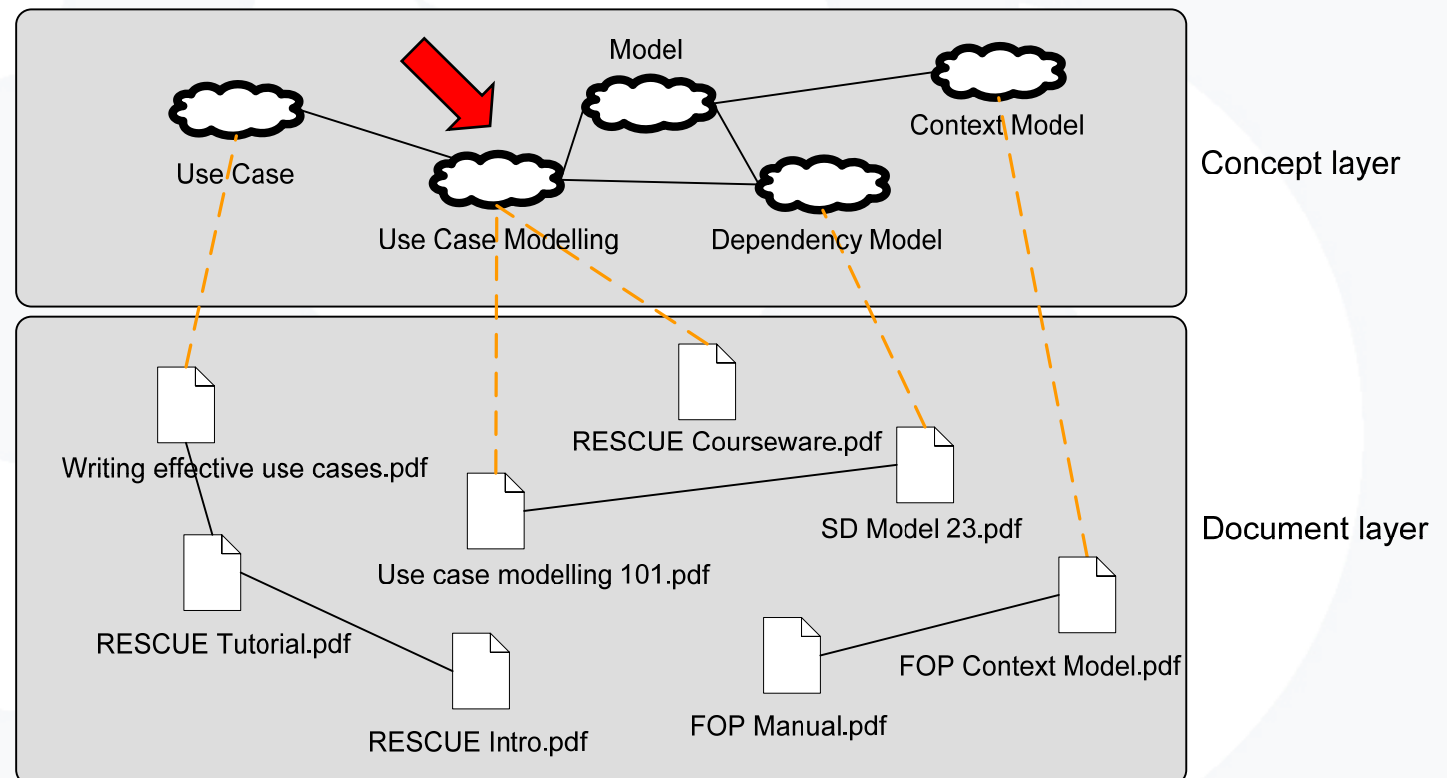
# Associative Network

## ■ Exact search



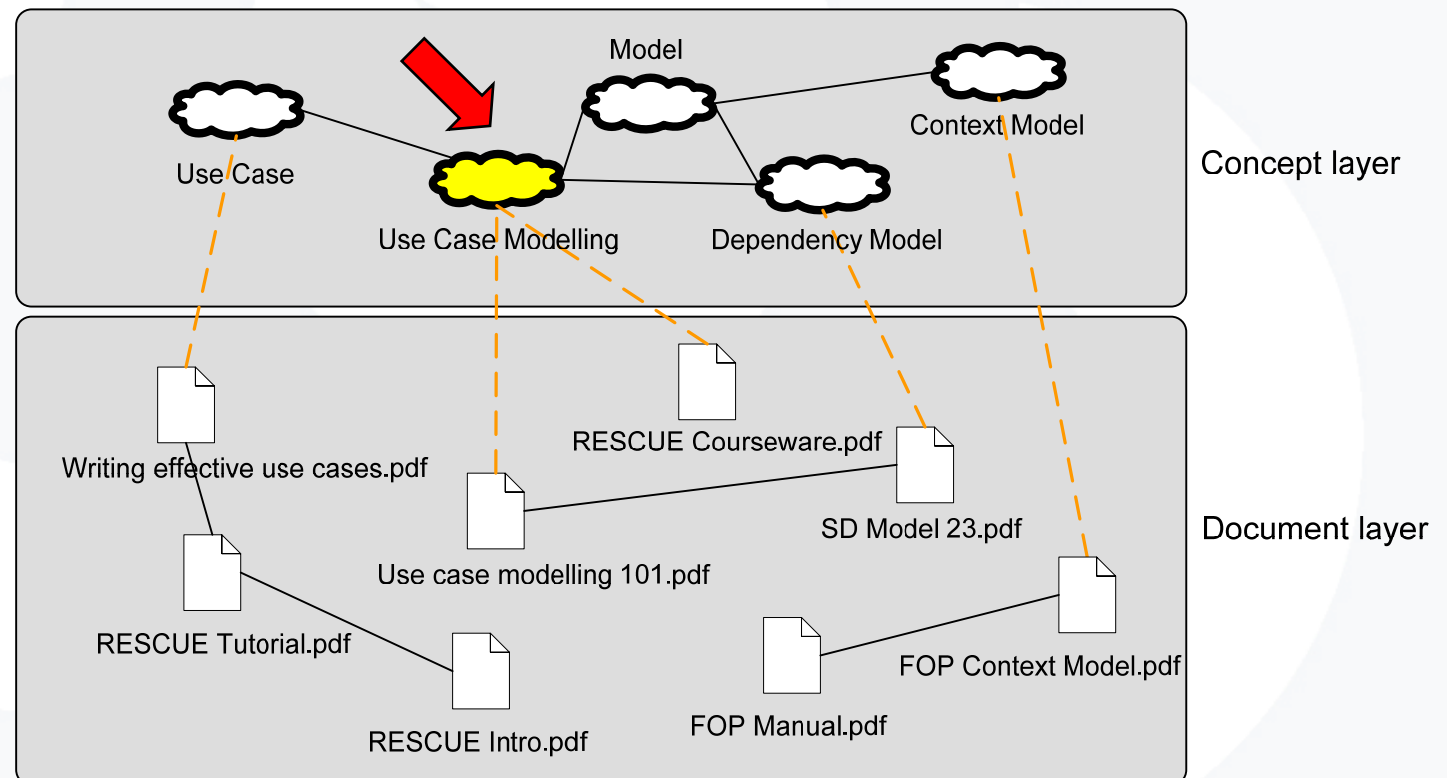
# Associative Network

## ■ Exact search



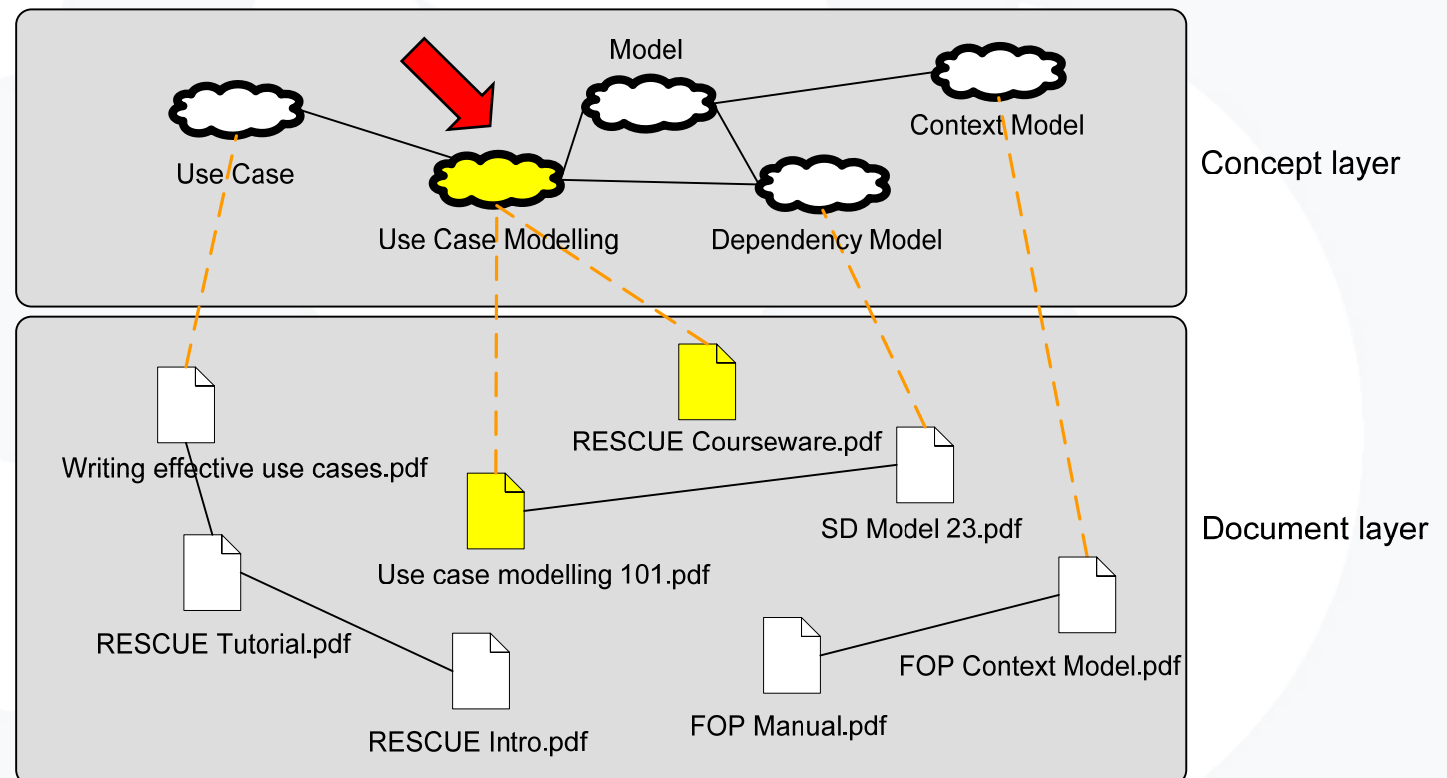
# Associative Network

## ■ Exact search



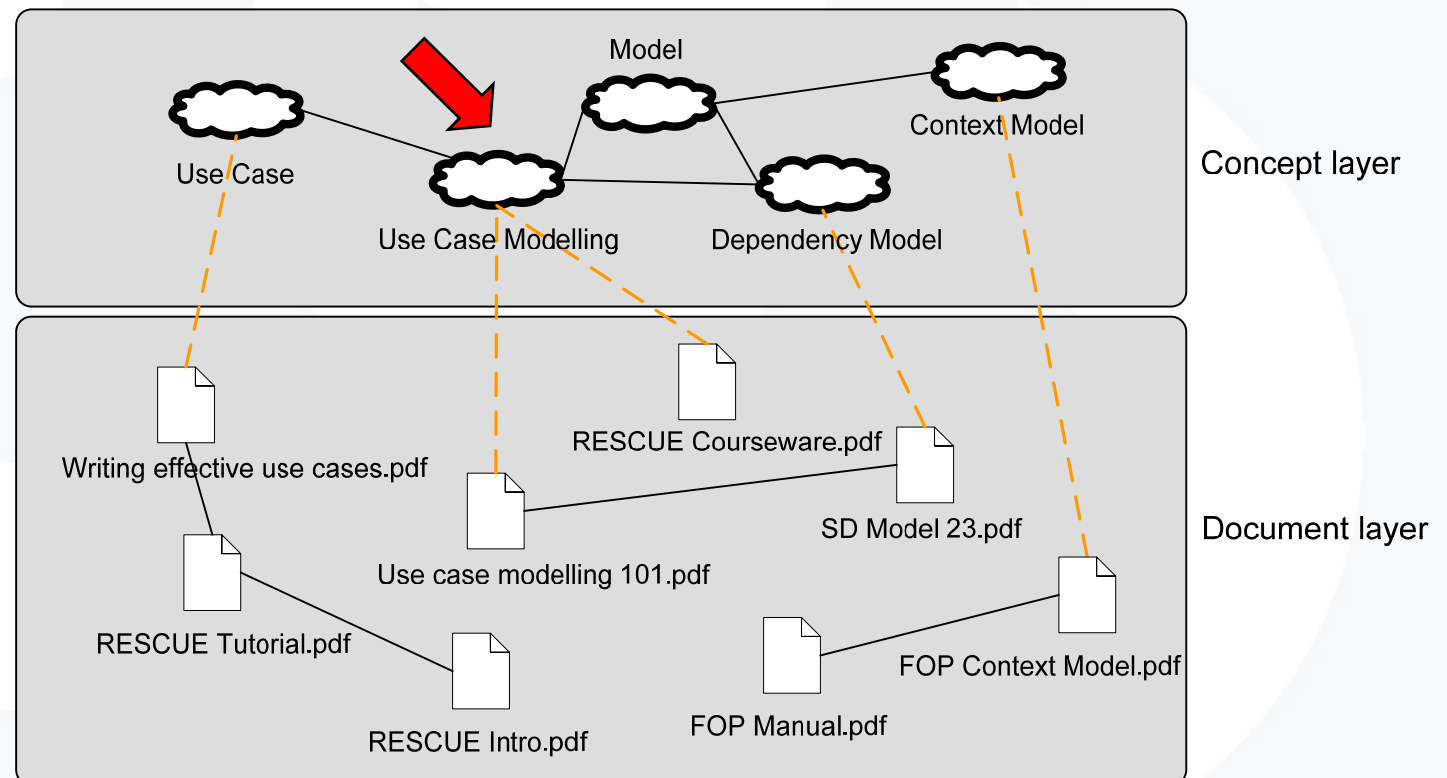
# Associative Network

## ■ Exact search



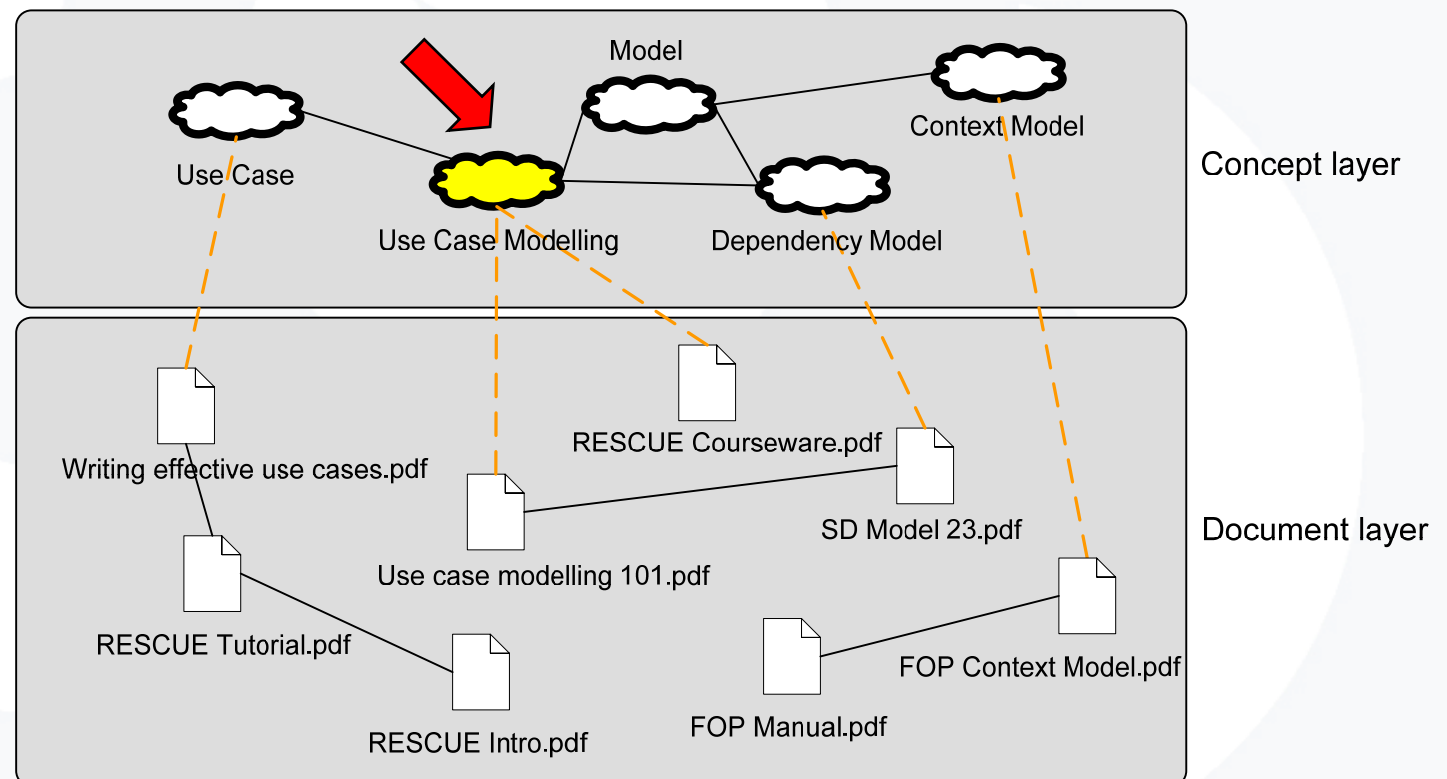
# Associative Network

## ■ Associative search



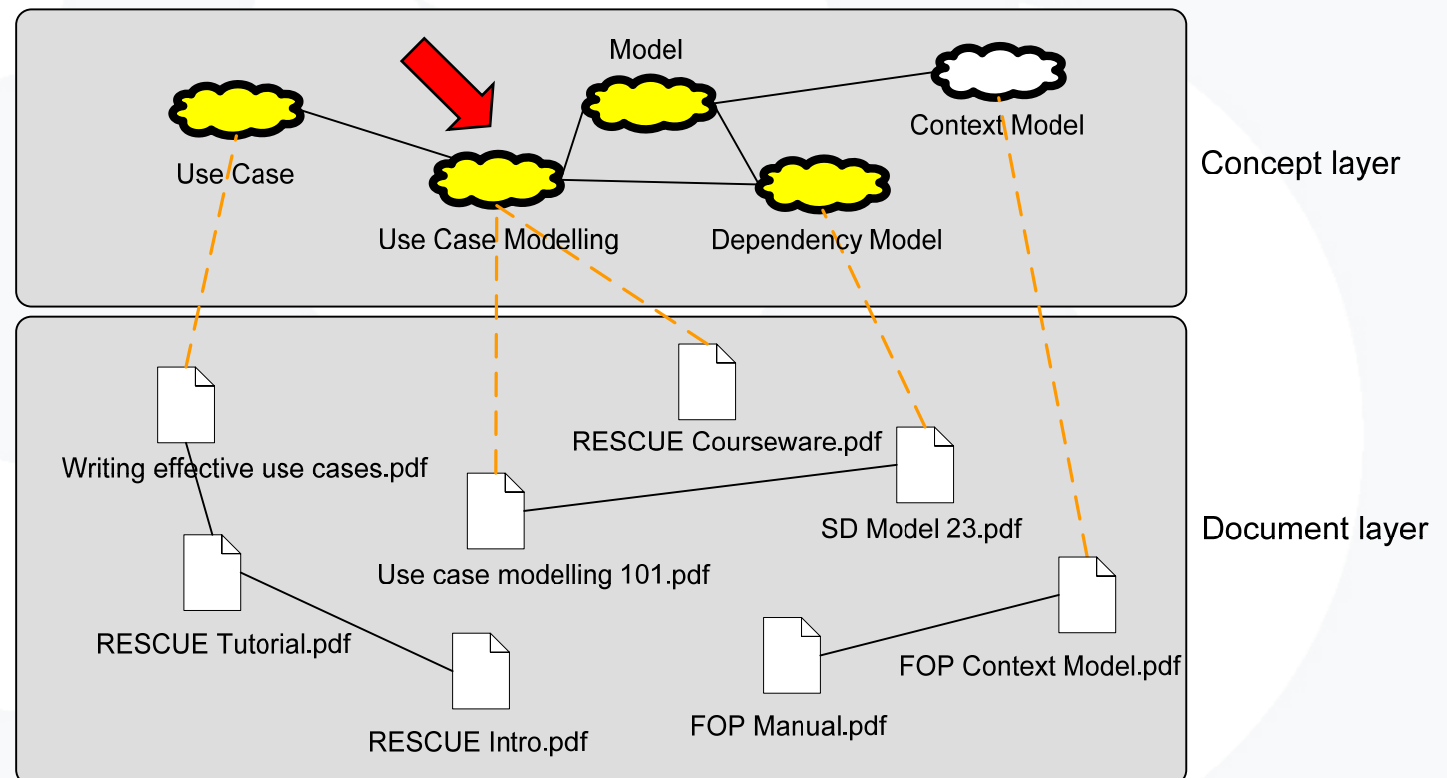
# Associative Network

## ■ Associative search



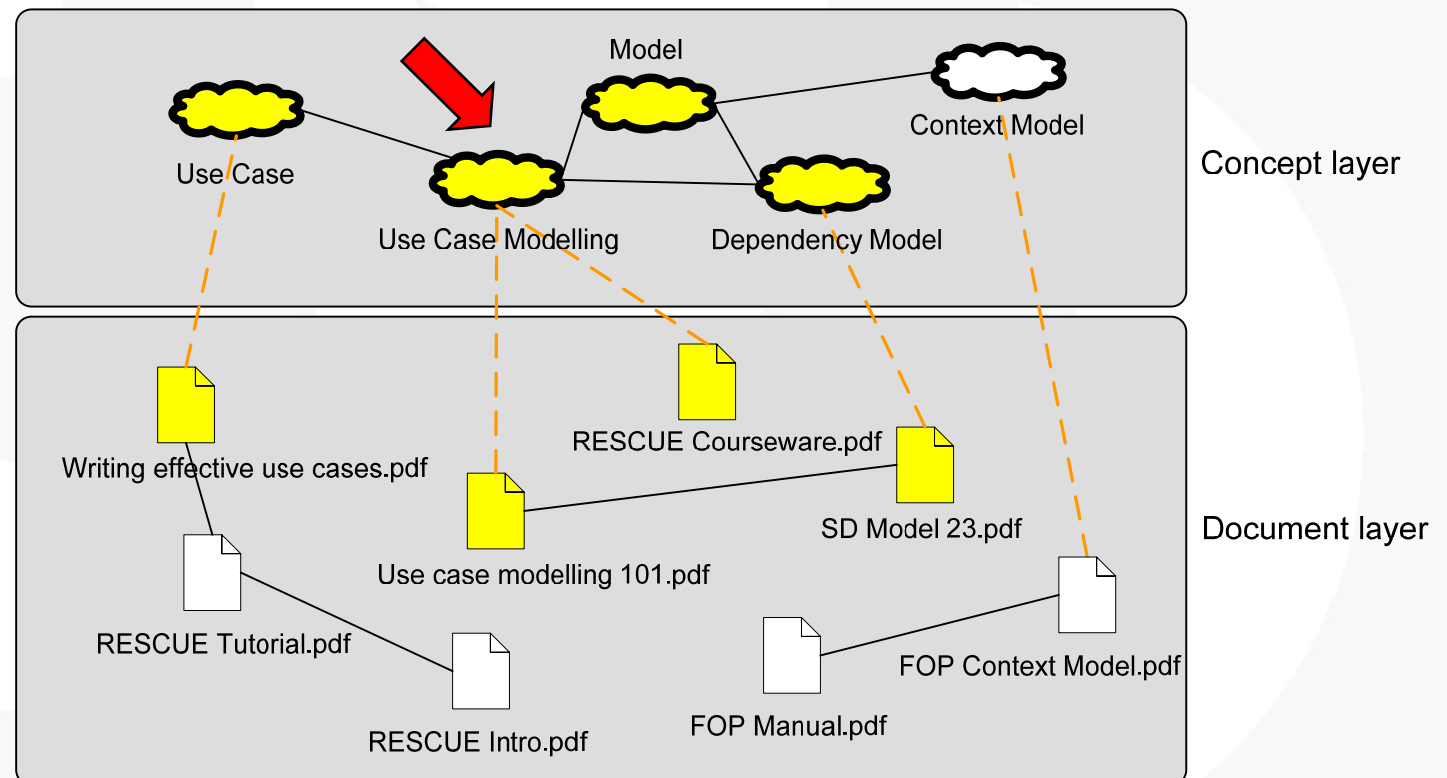
# Associative Network

## ■ Associative search



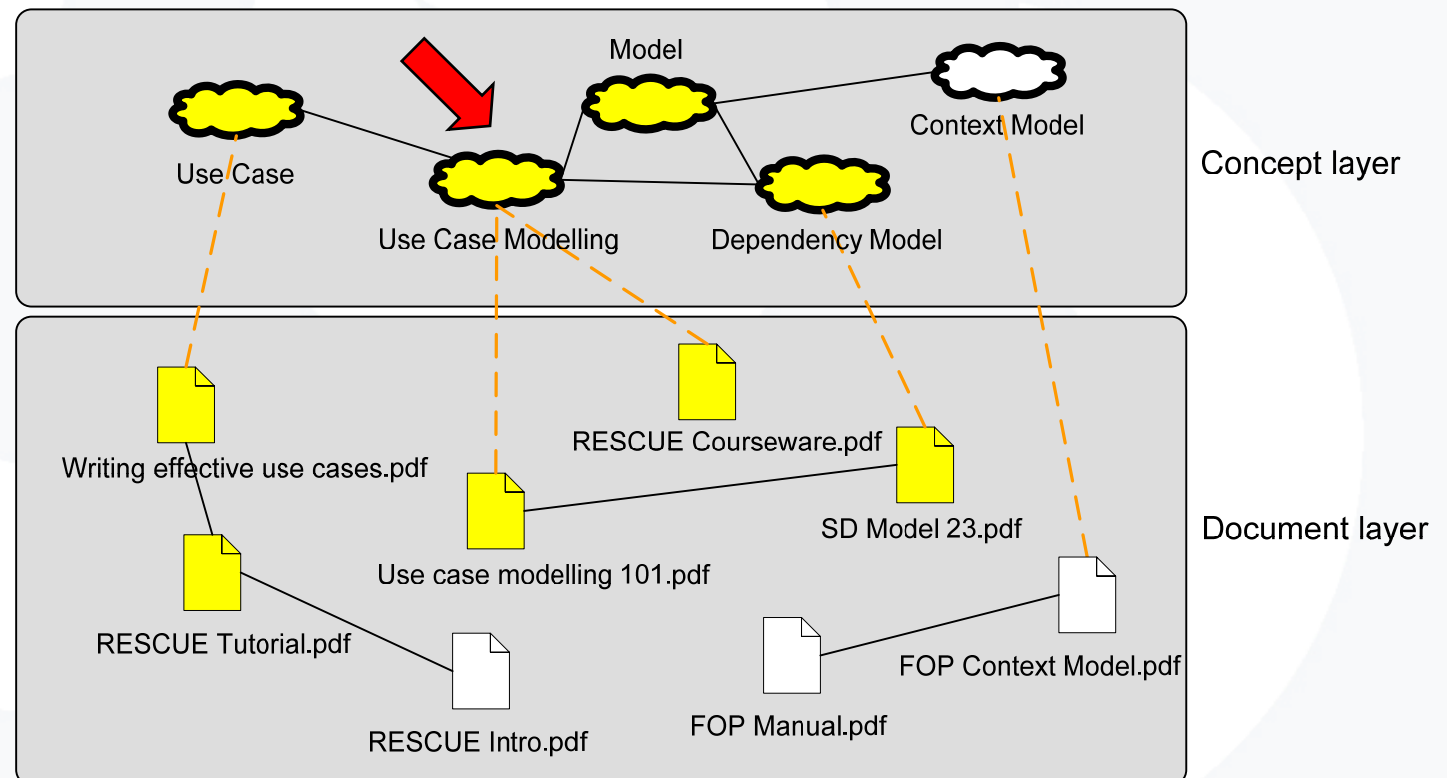
# Associative Network

## ■ Associative search



# Associative Network

## ■ Associative search



## Associative Information Retrieval

- Find relevant information by retrieving information that is by some means *associated* with information already known to be relevant
- **Not only more but *more relevant***

## Semantic Similarity

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

(Wu & Palmer, 1994)

$c_1$  ... first concept

$c_2$  ... second concept

$\text{lcs}$  ... least common subsumer of two concepts

$\text{depth}$  ... depth of concept in the class hierarchy

## Textual Similarity

$$\text{sim}(d1, d2) = \text{score}(d1_{25}, d2)$$

$d1$  ... document vector of the first document

$d2$  ... document vector of the second document

$d1_{25}$  ... document vector of the first document with all term weights removed except the 25 highest terms weights

$$\begin{aligned} \text{score}(q, d) &= \text{coord}(q, d) \cdot \text{queryNorm}(q) \\ &\cdot \sum_{t\_in\_q} (\text{tf}(t\_in\_d) \cdot \text{idf}(t)^2 \cdot t.\text{getBoost}() \cdot \text{norm}(t, d)) \end{aligned}$$

More details: Javadoc of `org.apache.lucene.search.Similarity`