# Integrating Knowledge Sources for Textual Case Based Reasoning

# Roadmap

- Motivation : Domain Requirements
- Why Textual CBR ? -- some issues in Textual CBR
- Towards a conceptual architecture
- Implementation and Results
- Work on Automated Rectification
- Related and Future Work

# Motivation : Domain Requirements

- MediaSearch : target users -- media and investing
- domains : press releases, company profiles, financials, news reports

**Issues**

-combination of text and non text attributes : knowledge engineering overhead needs to be minimal

-text sizes typically large : n-gramming would be too crude

-Flexibility in precision/recall (exactness/tolerance)

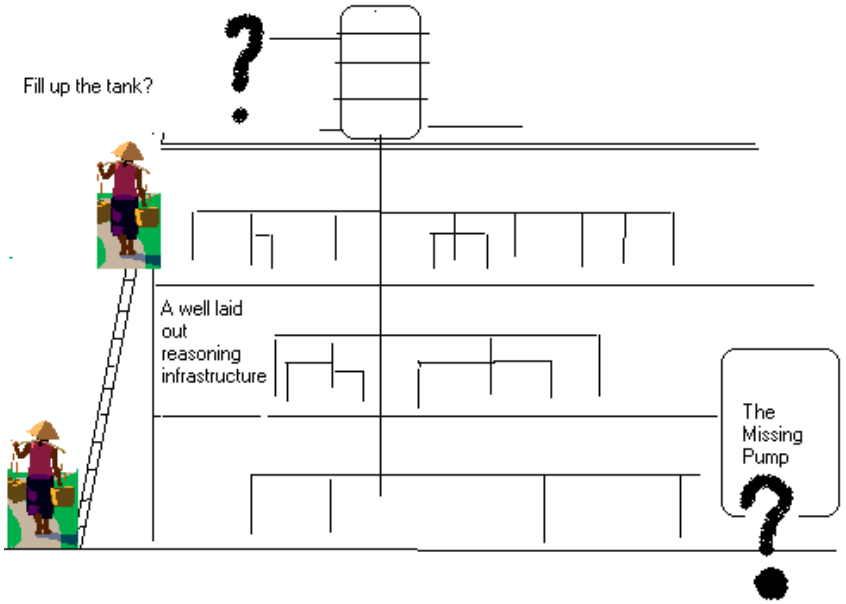
-Consult in conversational and structural CBR -- needed extensions to textual CBR as well

-Incremental domain vocabulary acquisition

-Easy maintenance and search

# Background : Some Challenges

Experiments with Structural and Conversational CBR : high lead times in deployment

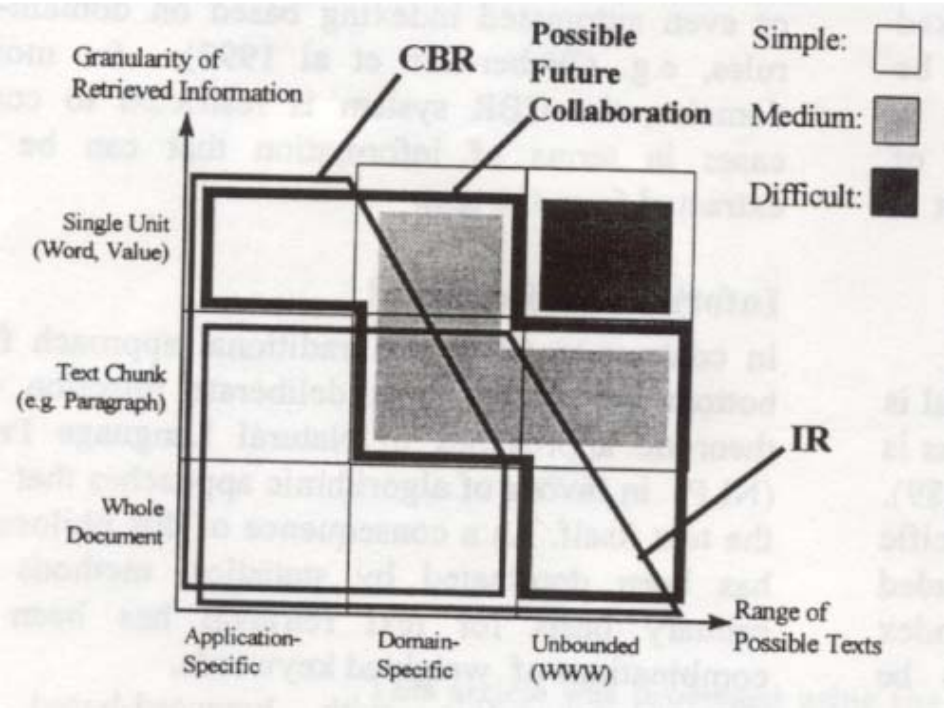


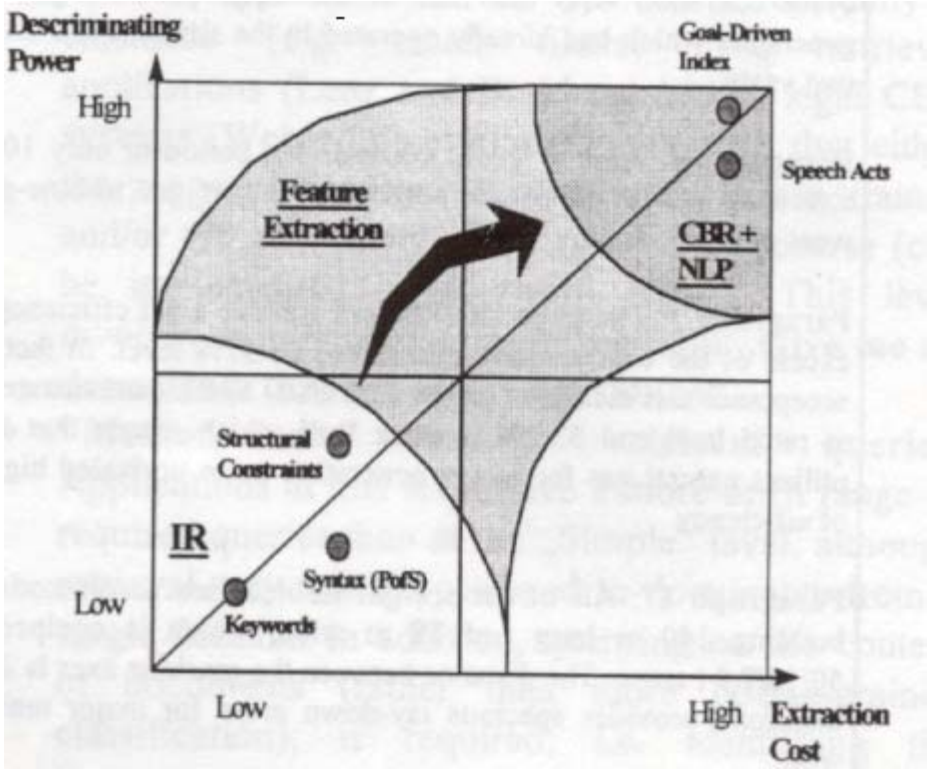

The Knowledge Engineering Bottleneck -- an analogy

# Positioning Textual CBR

- A middle ground between IR and NLP
- CBR and IR : differences and similarities
- CBR-IR possible architectures
- The tradeoff between discriminatory power and cost of extraction : feature extraction as in Neural Networks
- The spectrum of search

# Positioning Textual CBR



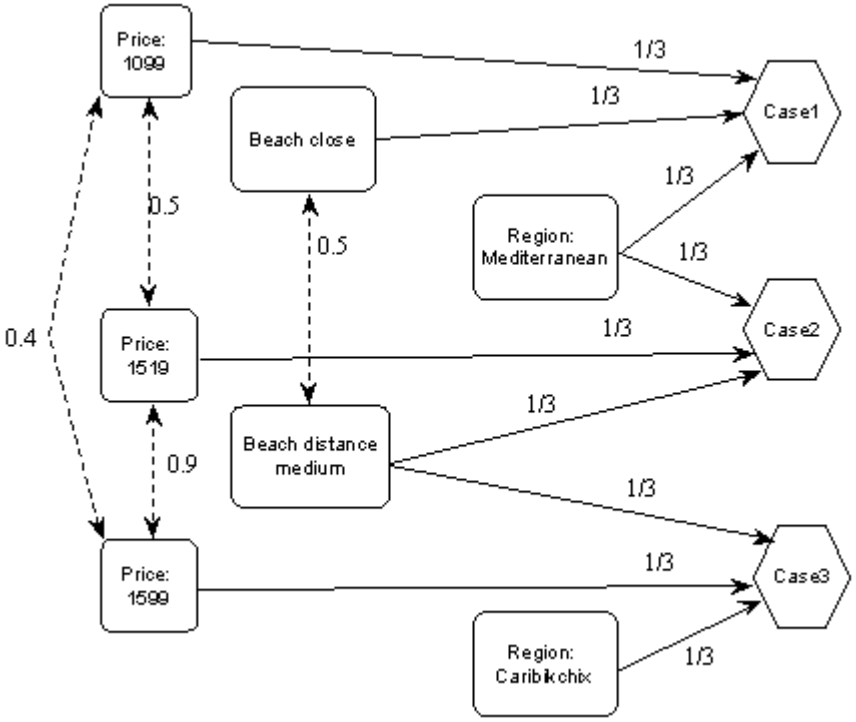



# Some challenges in Textual CBR

- Integration of multiple knowledge sources
- Vocabulary acquisition tools
- Information extraction

- In the current work :
  - we provide a conceptual framework for knowledge source integration
  - provide some new insights into the vocabulary acquisition problem

# Case Retrieval Networks

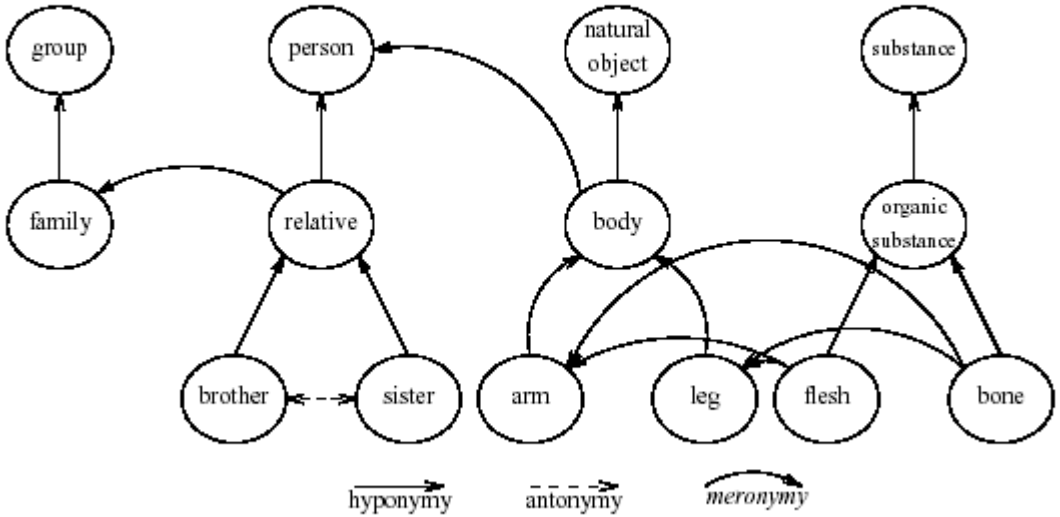



- Retrieval by spreading activation, computation nodes
- Efficiency issues
- Mapping CRNs to the world of Textual CBR

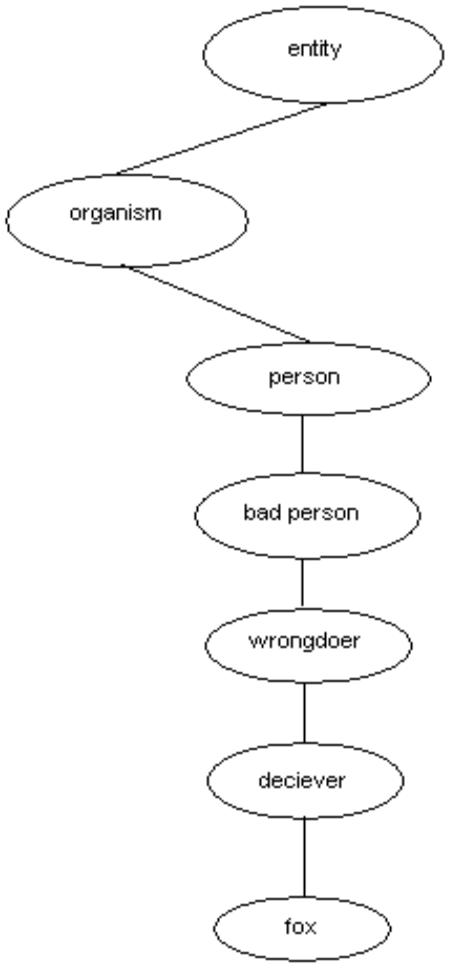
# Defining similarity using WordNet

- Sets of lexical concepts and a Semantic network of words
- Lexical relations: word forms
- Semantic relations: word meanings

Network representation of three semantic relations
among an illustrative variety of lexical concepts



# Using WordNet to arrive at numeric similarities between terms




The similarity between any two nodes in the tree is computed by the simple formula :

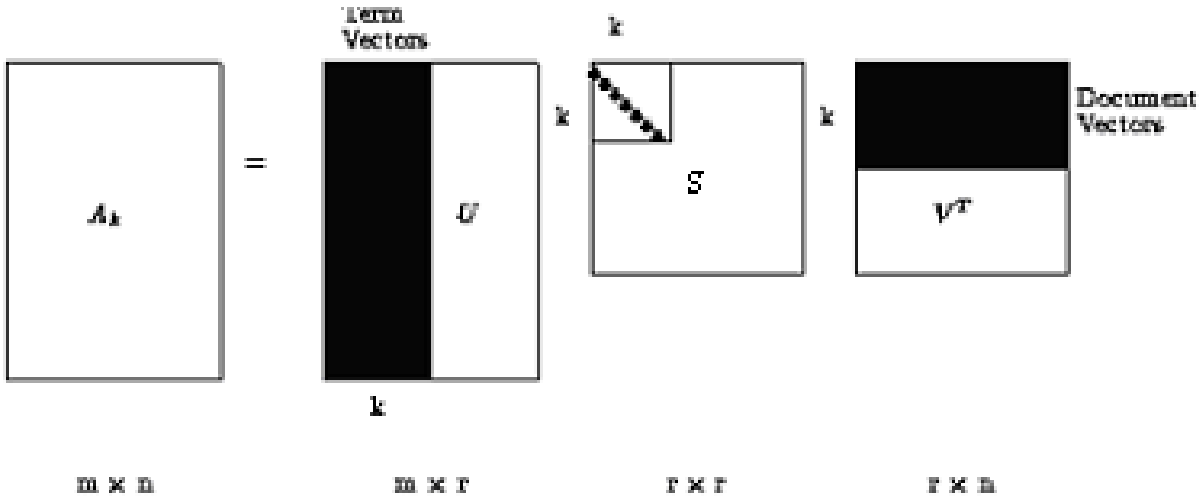
Similarity(node1, node2) = 100 - 100 * (Minimum distance / Maximum Distance)

Where

Maximum distance = Sum of levels to the root for the feature and

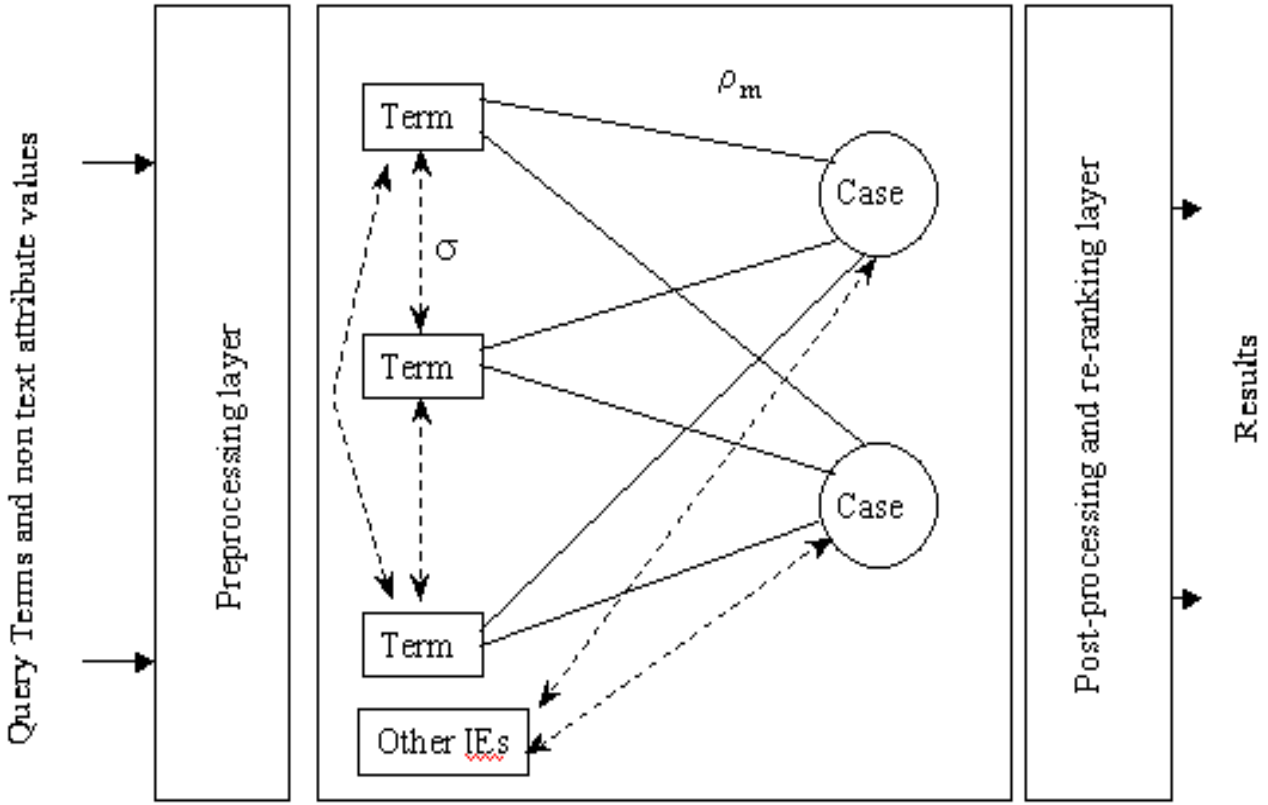
Minimum Distance = Sum of distances to the nearest parent

# Using Latent Semantic Indexing for defining relevance of terms to docs.




- Singular Values
- “Mushing Index”

# Our Conceptual Architecture



# The retrieval process

**Step 1**: Query is preprocessed and a set of IEs are activated. Initial activation $\alpha_0$ for all activated nodes is taken as unity.

**Step 2:** A set of computation nodes are created around each textual token (IE) after invoking WordNet. The similarity arcs are established with the newly created nodes.

**Step 3**: An activation $\alpha_0$ is propagated to all IE nodes from query nodes. Let the similarity function (WordNet similarity for text tokens, domain specific for non-text attributes) be $\sigma$. Let an aggregation function $\pi_e$ be defined at each IE node which aggregates the effects of incoming activations. Then the next level of activation at IE node e is

$$\alpha_1(e) = \pi_e(\sigma(e_1, e).\ \alpha_0(e_1),\ \ldots,\ \sigma(e_s, e).\ \alpha_0(e_s))$$ where s is the total number of IE nodes

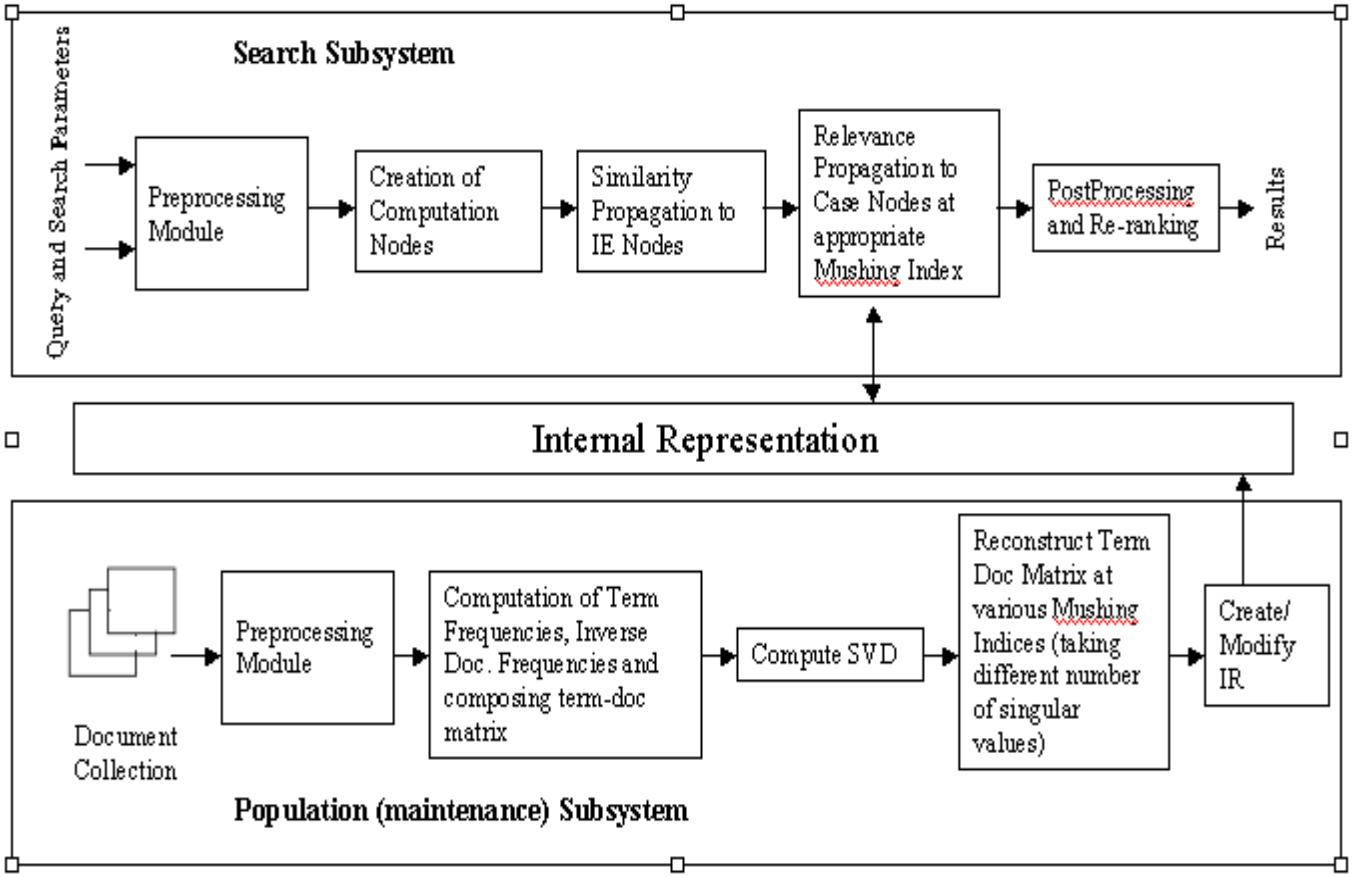
If the WordNet distance threshold is set at $\Gamma$ then for all e where $\alpha_1(e) > \Gamma$, $\alpha_1(e)$ is reset to 0.

**Step 4:** The resulting activations are propagated to all case nodes. Activation at case node c with aggregation function $\pi_c$ at Mushing Index m is

$$\alpha_2(c) = \pi_c(\rho_m(e_1, c).\ \alpha_1(e_1),\ \ldots,\rho_m(e_s, c).\ \alpha_1(e_s))$$

**Step 5:** The results obtained after Step 3 are reranked after postprocessing to produce the final ranking.

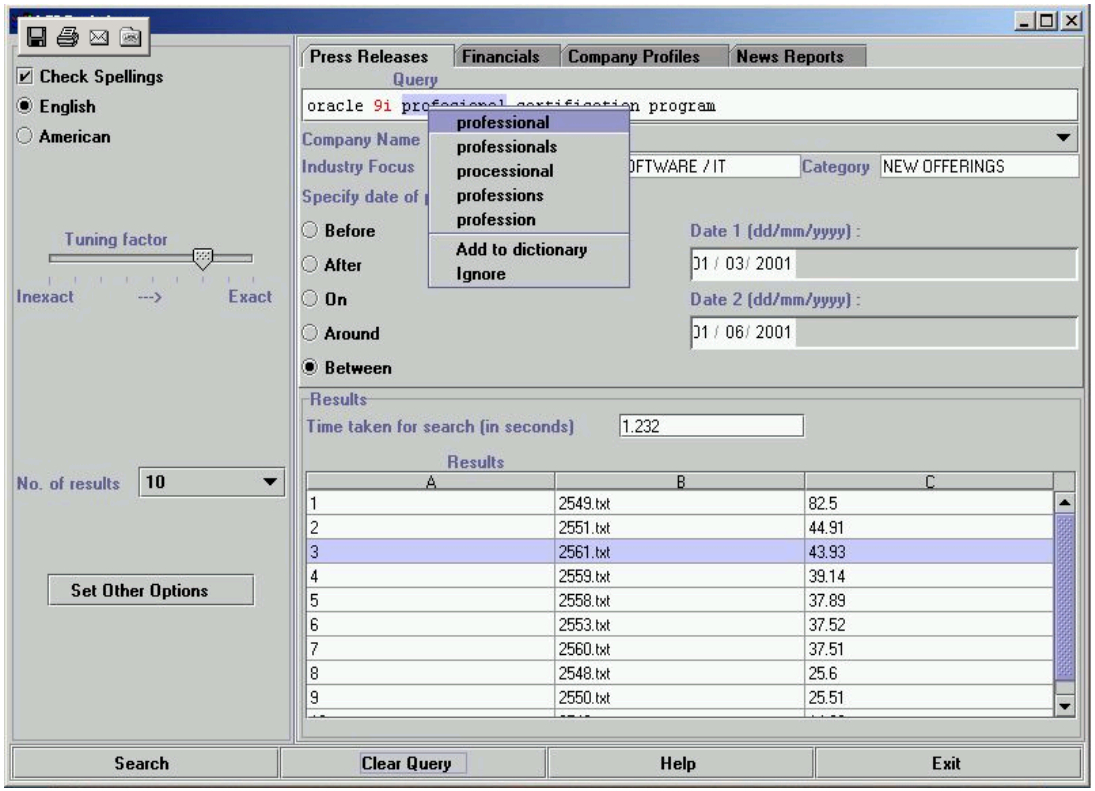
# The tool architecture



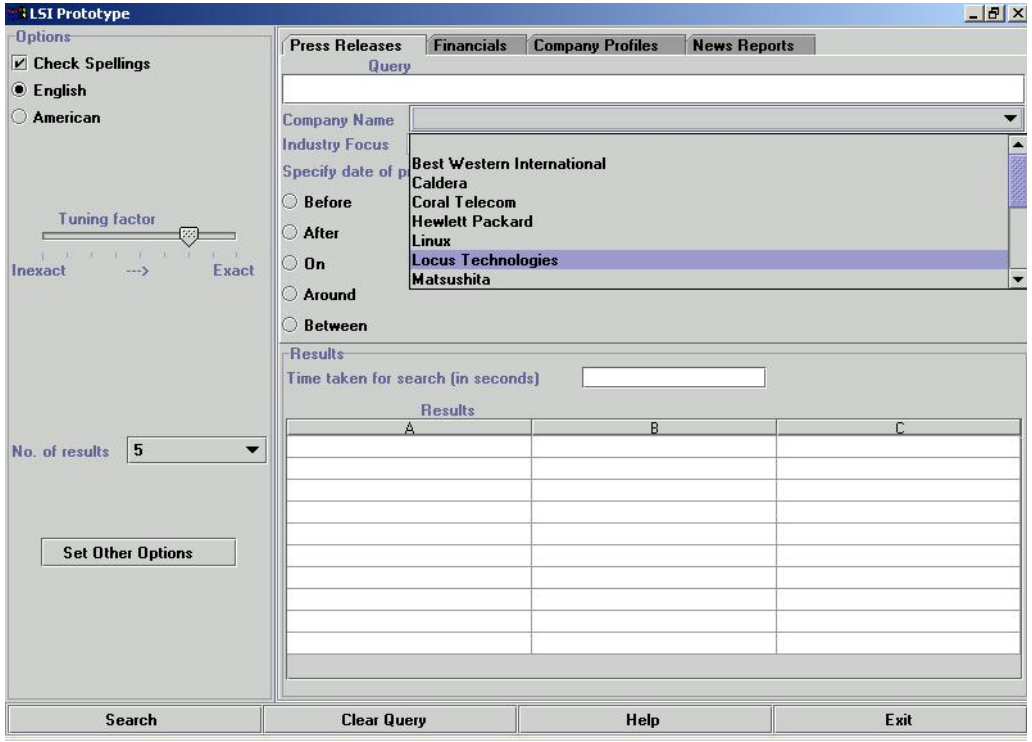
# Pre- and post-processing

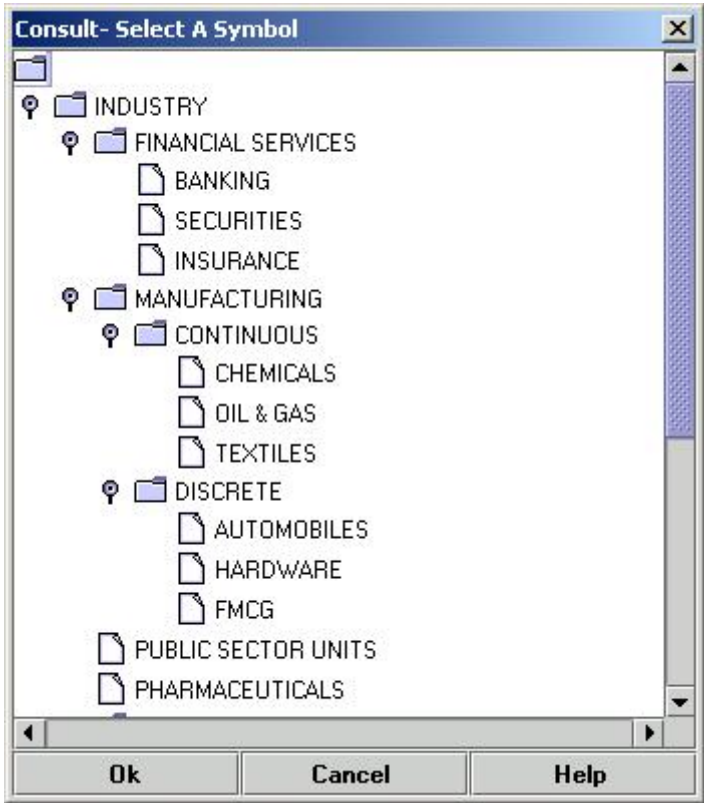
- The “spellcheck and suggest replacements”
  - an improvisation over the Leveinshtein Pattern matching algorithm
  - can suggest the following replacements which Word 97 cannot
    - “punctuation” for “puctuations”
    - “removes” for “revoves”
    - “visionary” for “visniory”
- Other pre-processing operations
- Post-processing and Re-ranking

# Implementation

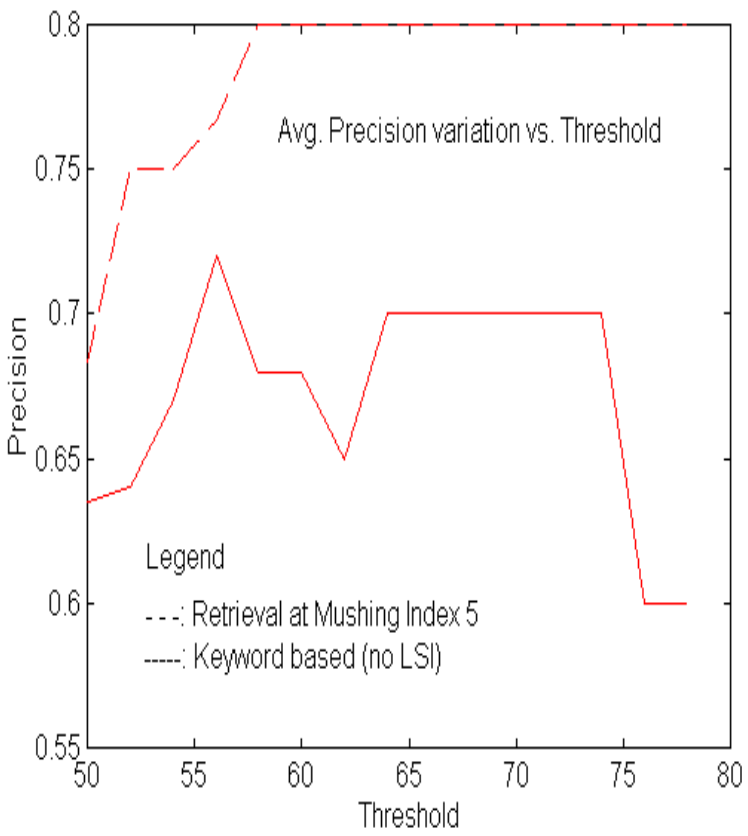


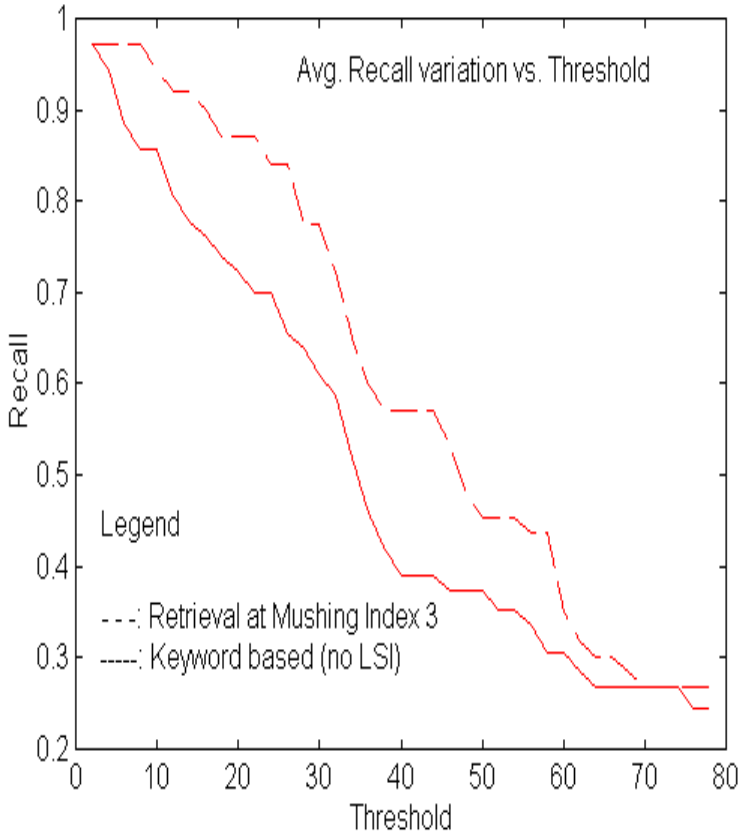

# Implementation



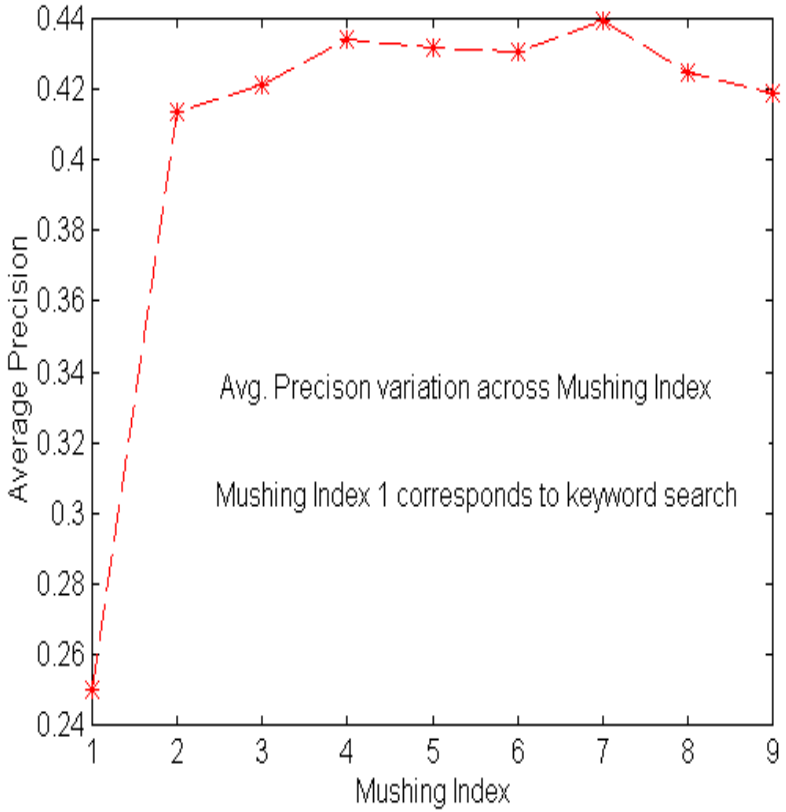



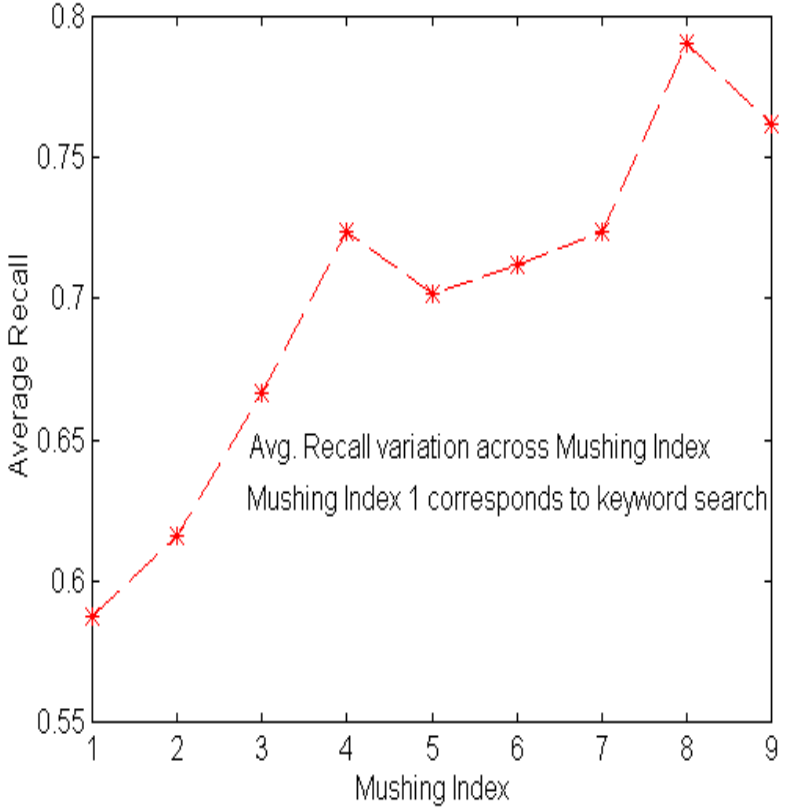
# Results


Avg. Precision variation vs. Threshold
Precision
Threshold
Legend
- - -: Retrieval at Mushing Index 5
-----: Keyword based (no LSI)
0.8
0.75
0.7
0.65
0.6
0.55
50
55
60
65
70
75
80


Avg. Recall variation vs. Threshold
Recall
Threshold
Legend
- - -: Retrieval at Mushing Index 3
-----: Keyword based (no LSI)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0
10
20
30
40
50
60
70
80

# Results (contd..)



# The idea of rectification

- LSI can lead to spurious associations, if
  - document collections are small
  - The collections are not representative of inherent content structure

experiments

- Initial experiments :
  - manual rectification
    - term x term matrix generated
    - highly boosted pairs identified
    - spurious ones manually selected
    - corrections mapped back to term x document matrix

# Some results with the Deerwester Matrix

For query ‘interface’ the documents retrieved are

| | **Before rectification** | | **After rectification** |
|---|---|---|---|
| | Doc no | | Doc no |
| (1) | 2 | (1) | 3 |
| (2) | 1 | (2) | 1 |
| (3) | 4 | (3) | 4 |
| (4) | 3 | (4) | 5 |
| (5) | 5 | (5) | 2 |

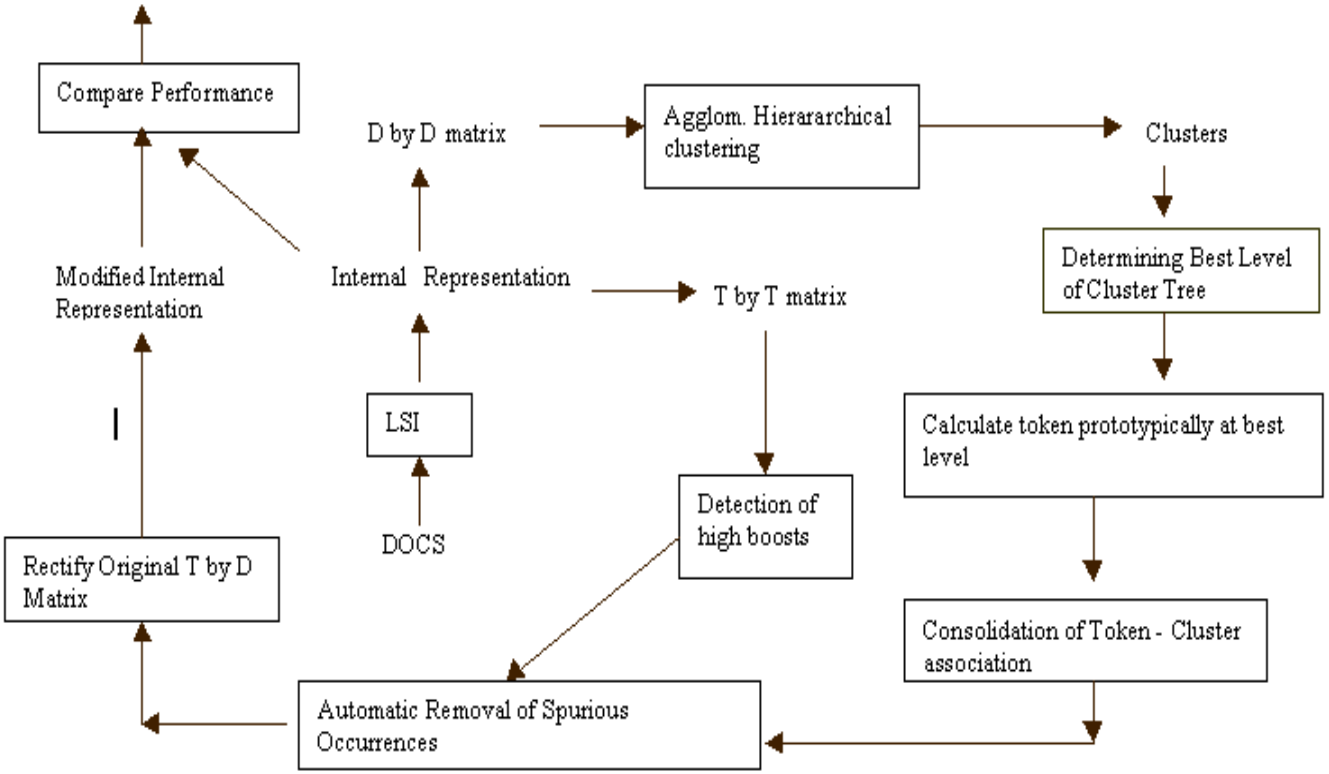
For query ‘human’ the documents retrieved are

| | **Before rectification** | | **After rectification** |
|---|---|---|---|
| | Doc no | | Doc no |
| (1) | 2 | (1) | 4 |
| (2) | 1 | (2) | 1 |
| (3) | 4 | (3) | 3 |
| (4) | 3 | (4) | 5 |
| (5) | 5 | (5) | 2 |

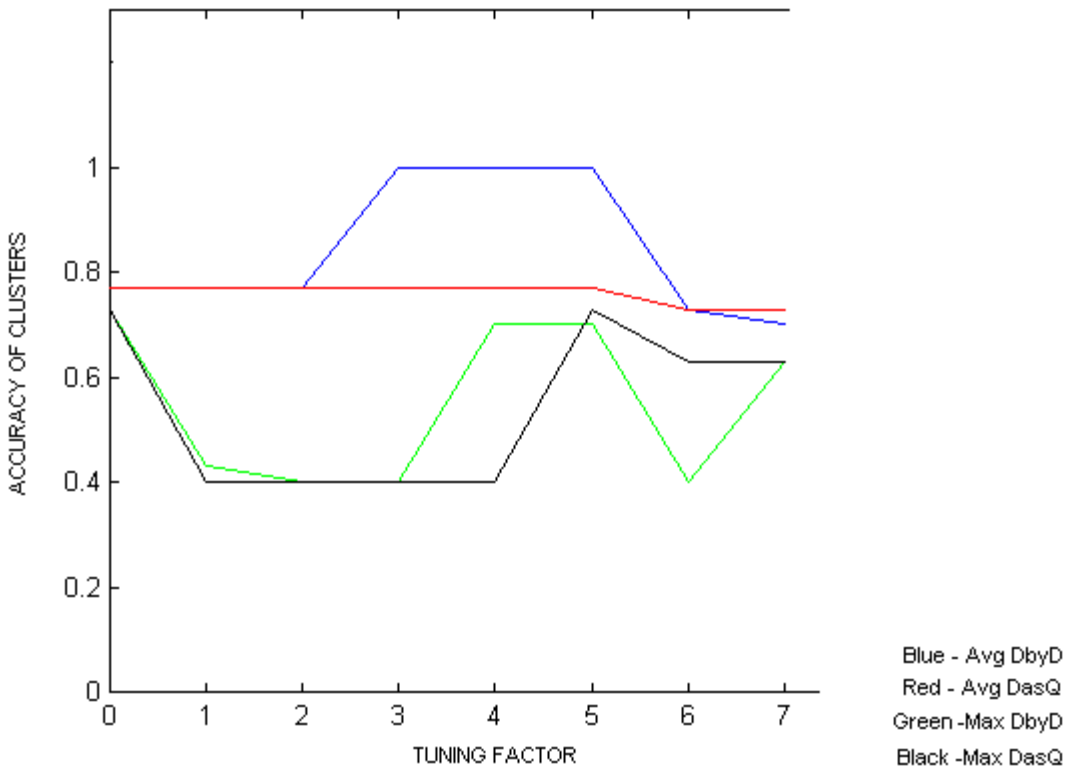
For query ‘user’ the documents retrieved are

| | **Before rectification** | | **After rectification** |
|---|---|---|---|
| | Doc no | | Doc no |
| (1) | 2 | (1) | 2 |
| (2) | 1 | (2) | 3 |
| (3) | 4 | (3) | 5 |
| (4) | 3 | (4) | 9 |
| (5) | 5 | (5) | 1 |

# Towards Automated Rectification



# Some experiments on text classification



# Related Work

- FAQFinder
- SIMATIC
- SMART
- INQUERY
- FallQ
- SPIRE

# To sum up, Our Contributions

(1) a conceptual architecture that provides a new approach to integrate explicit knowledge and latent knowledge – and allows for effective retrieval which can be flexibly tuned for precision and recall.

(2) the idea that while construction of explicit domain specific vocabulary may be troublesome, it may be easier to use principal component techniques like LSI to establish numeric associations between terms (though the explicit relation that causes this association may not be explicit as in concept hierarchies, the knowledge is readily usable) – and remove spurious associations. Thus a major part of vocabulary acquisition exercise (in terms of synonyms, noise words, ontologies) can be simplified.

# Future Work

- Applicability of automated rectification to bigger document collections
- Information Extraction
- WSD, POS tagging, Anaphoric references, knowledge of syntax
- Relevance feedback
- Alternate spreading activation models
- Multiple formats
- Larger volumes
- Mushing over other data-types

# Publications based on this work

**Conference** :

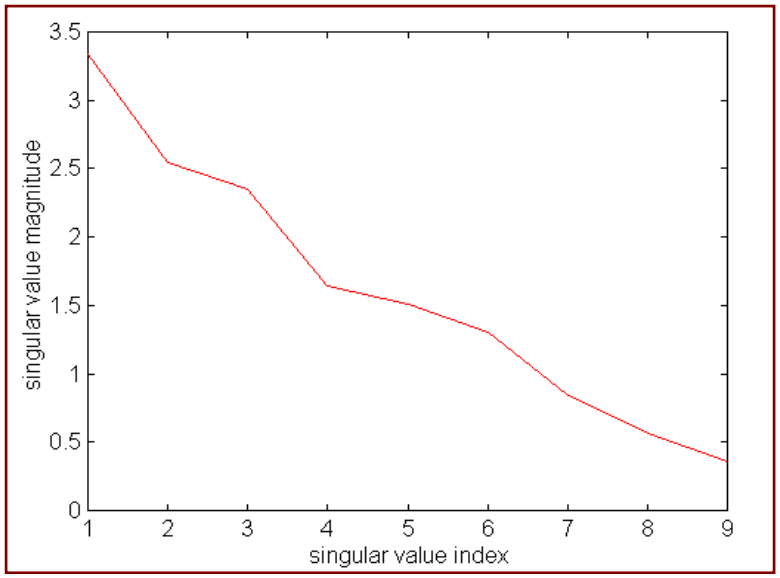
[1] Sutanu Chakraborti, Sailaja Ambati, Vivek Balaraman and Deepak Khemani. Integrating Knowledge Sources and Acquiring Vocabulary for Textual CBR. Proceedings of the 8th UK Workshop on CBR, Cambridge December 2003 , pp. 74-84

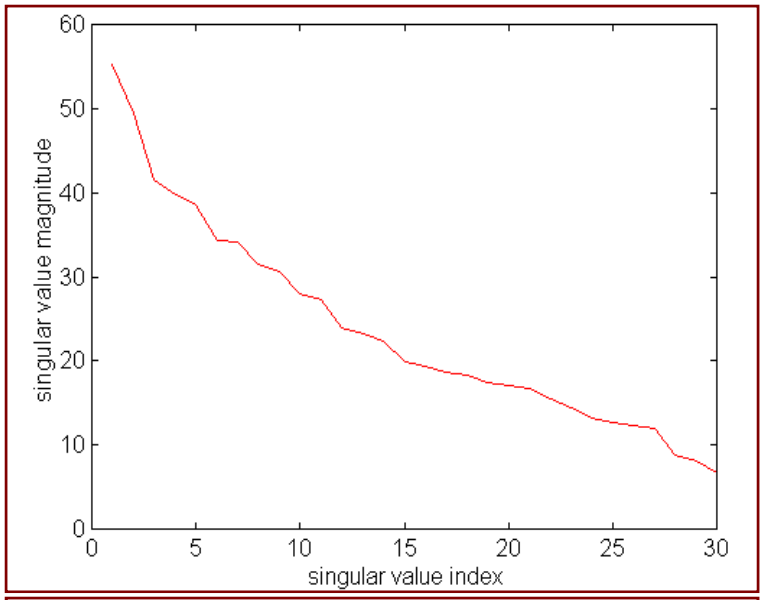
**Journal :**

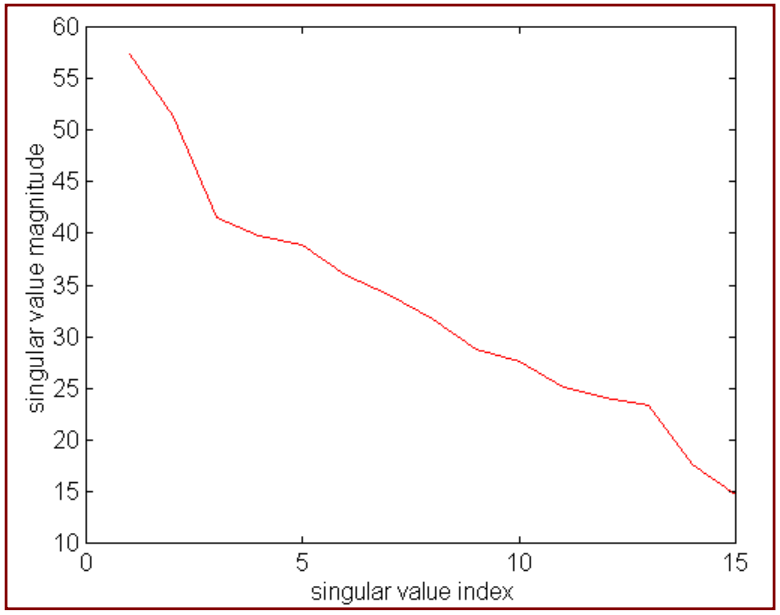
[2] Sutanu Chakraborti, Sailaja Ambati, Vivek Balaraman and Deepak Khemani. Integrating Knowledge Sources and Acquiring Vocabulary for Textual CBR. Accepted for a forthcoming edition of the BCS SGAI journal Expert Update. (volume details and page numbers not available)

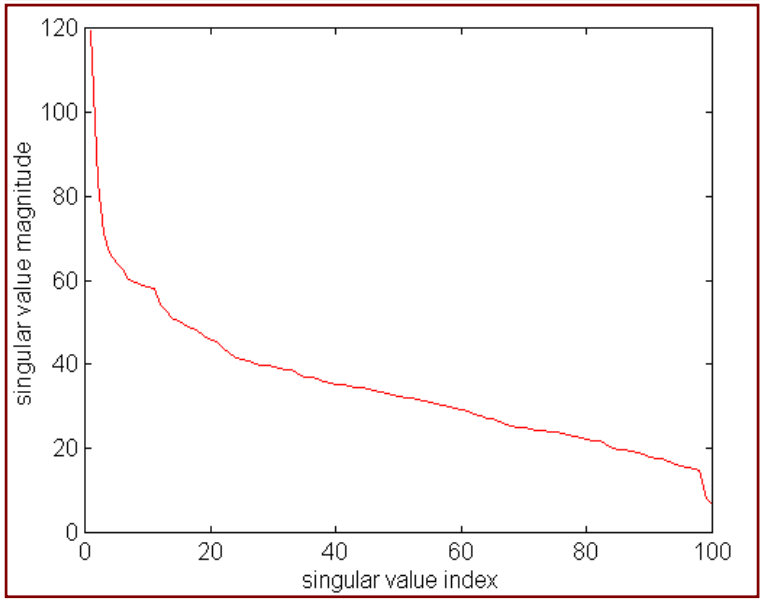
# Thank you.

I can be reached at : sutanuc@pune.tcs.co.in


singular value magnitude
singular value index
0
0.5
1
1.5
2
2.5
3
3.5
1
2
3
4
5
6
7
8
9


singular value magnitude
singular value index
0
10
20
30
40
50
60
0
5
10
15
20
25
30


singular value magnitude
singular value index
10
15
20
25
30
35
40
45
50
55
60
0
5
10
15


singular value magnitude
singular value index
0
20
40
60
80
100
120
0
20
40
60
80
100

back

Titles:

1 : Human computer interface for lab ABC computer applications

2 : A survey of user opinion of computer system response time .

3 : The ESP user interface management system .

4 :system and human system engineering testing of EPS.

5 :Relation of user perceived response time to error measurment .

6: The generation of random ,binary , unordered tree .

7 : The intersection graph of path in trees

8: Graph minor IV : width of trees and well- quasi - ordering .

9 : Graph minor: A survey

back