A Multi-lingual Meaning Based Search Engine

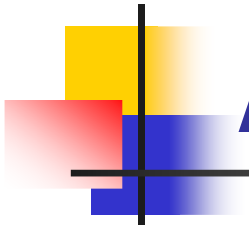
Under the guidance of

Prof Pushpak Bhattacharyya

By

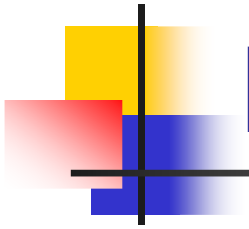
Sarvjeet Singh

99005029



Agenda

- Motivation
- Universal Networking Language
- Search Engine Model
- Partial Matching
- Conclusion
- Future Work
- Demo of the Search Engine

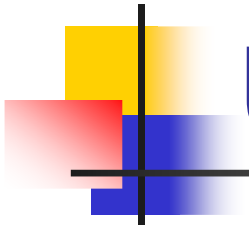


Motivation

- Amount of information on internet has grown exponentially
- Search Engines: Help in mining information
- Language Barrier
- Information Overload
- Lot of skill needed to form efficient queries

Universal Networking Language

- Electronic language for computers to express and exchange every kind of information
- UNL Expression
- Binary Relation
- Universal Words
  - Constrains
  - Attributes

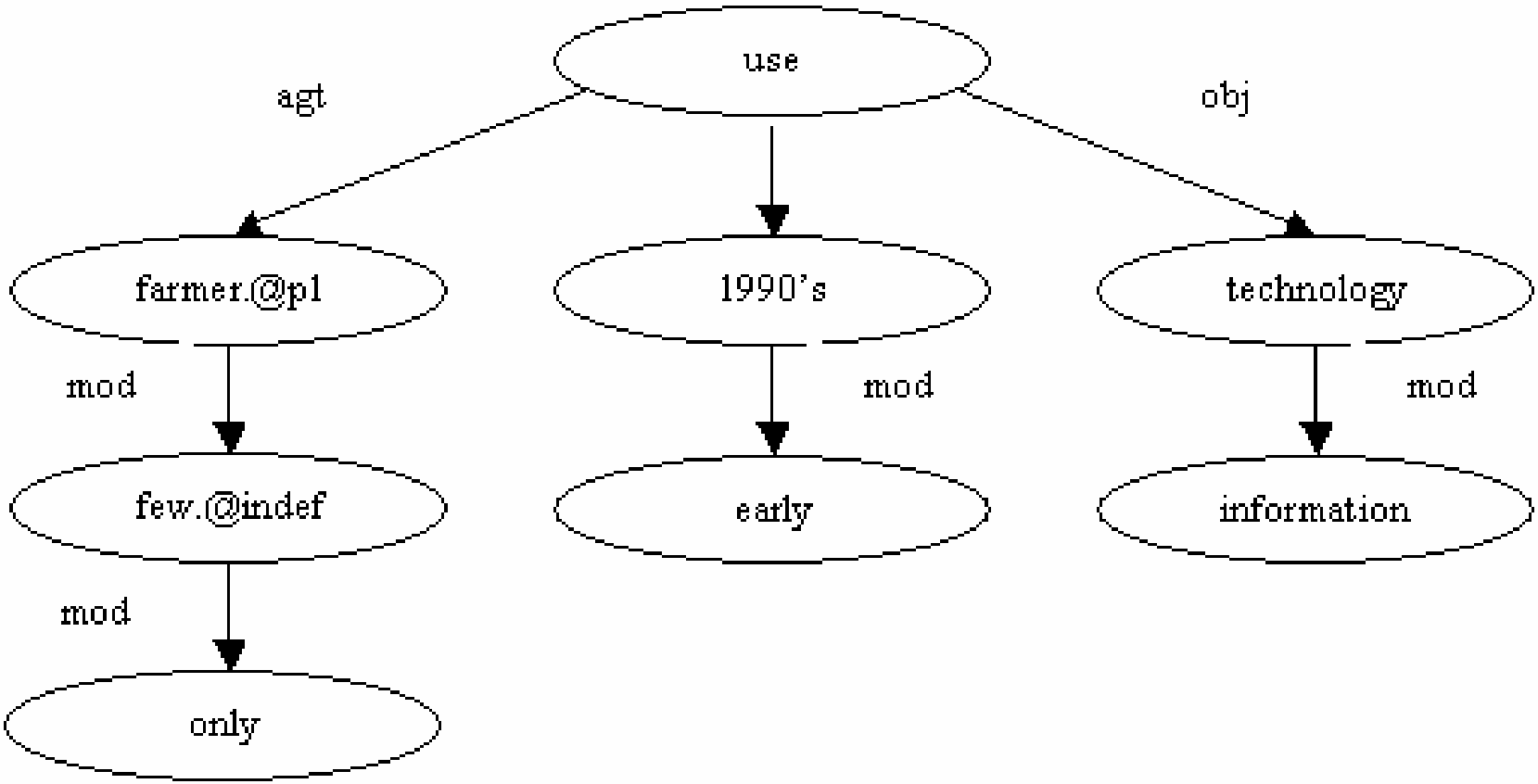


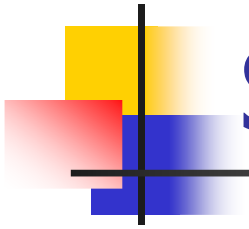
UNL: Example

Only a few farmers could use information technology in the early 1990's.

agt(use(icl>do).@ability-past, farmer(icl>person).@pl)
obj(use(icl>do), technology(icl>thing))
mod:01(farmer(icl>person), few(icl>number).@indef)
mod(:01, only)
mod(technology(icl>thing), information)
mod:02(1990's(icl>time), early)
tim(use(icl>do), :02)

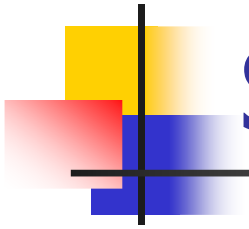
UNL Graph


use
agt
obj
farmer.@pl
1990's
technology
mod
mod
mod
few.@indef
early
information
mod
only



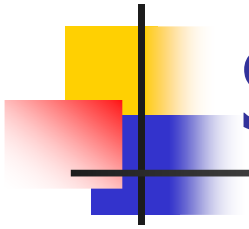
Search Engine Model

- Focused Crawler
  - Presenting a domain – Yahoo categories
  - Training a binary classifier
  - Focused Crawling using classifier score as the guide
  - Details of document crawled kept in a table called *docindex*



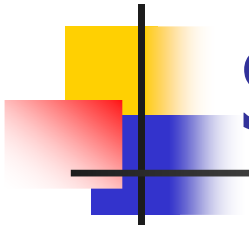
Search Engine model (cont)

- Enconverter and Deconverter modules
  - ENCO and DECO software
  - Implemented on windows platform
  - Implemented as PHP script running on a apache web server on a Windows 2000 machine
  - Currently English and Hindi ENCO integrated
  - Caching



Search Engine Model (cont)

- HTML Parser
  - Parses the HTML documents – separates the formatting from the sentences
  - Document design template – HTML tags with placeholders for sentences
  - Can get the original document by merging the sentences with the document design template

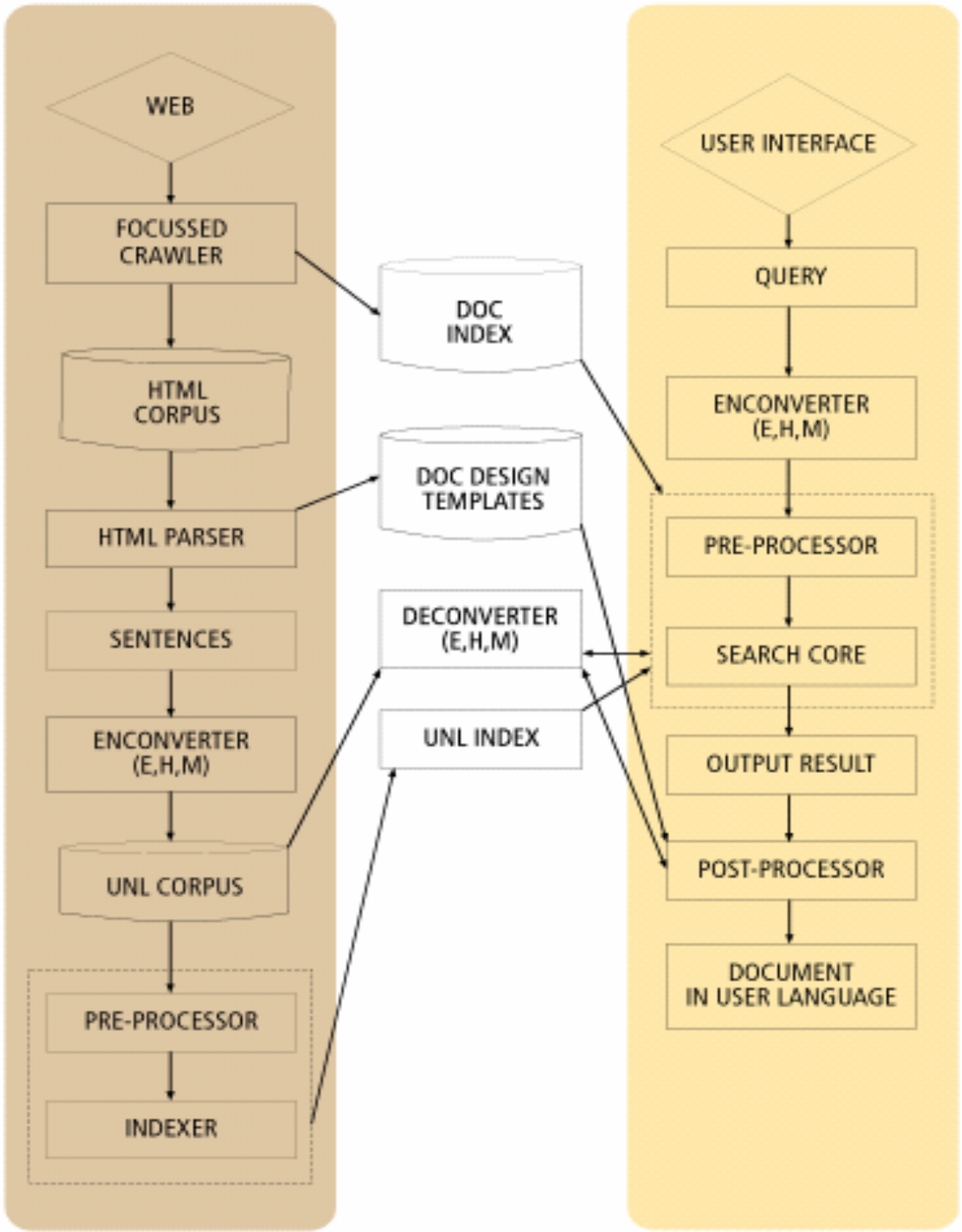


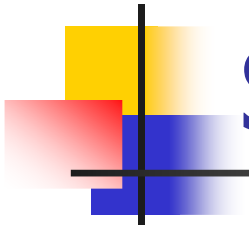
Search Engine Model (cont)

- Indexer Module
  - Preprocessor
    - Spaces and tabs removed
    - Attributes stripped off
    - UWs not having id are assigned a unique dummy id
    - Compound UW ids are replaced by the UNL expressions of all the relations in the subgraph, sorted alphabetically, separated by a “^”

Search Engine model (cont)

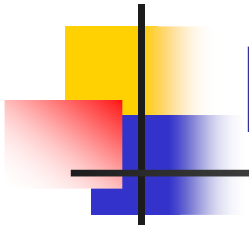
- Indexer
  - Separates resulting UNL expressions into REL, UW1, UW2, UW1ID, UW2ID
  - Updates unlindex – MYSQL table
  - Sentences consisting of single word handled separately
- Search Module
- Post Processor Module
  - Merges sentences with document design


WEB
FOCUSSED CRAWLER
HTML CORPUS
HTML PARSER
SENTENCES
ENCONVERTER (E,H,M)
UNL CORPUS
PRE-PROCESSOR
INDEXER
DOC INDEX
DOC DESIGN TEMPLATES
DECONVERTER (E,H,M)
UNL INDEX
USER INTERFACE
QUERY
ENCONVERTER (E,H,M)
PRE-PROCESSOR
SEARCH CORE
OUTPUT RESULT
POST-PROCESSOR
DOCUMENT IN USER LANGUAGE



Search Module

- Query Matching
  - Complete
  - Partial
- Indexing
  - Simply indexing on binary relations is not enough
  - Need to take into account the connections between binary relations



Partial matching

$$R_q(d) = \frac{\sum_{s \in S_d} r_q(s)}{|S_d|}$$

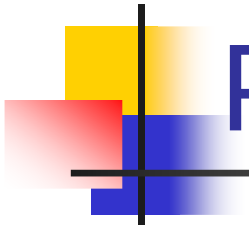
$$r_q(s) = \alpha \frac{n}{N} + \beta \frac{l}{L}$$

Partial Matching - Algorithm

- Indexing
- N, L - Independent of sentence

$$L = \sum_{v \in V} (degree(v) - 1)$$

- Need to find out n and l for each sentence to calculate relevance
- Can find all matching relation edges of the document with a single SQL query
- Sort by (document, sentence)



Partial Matching

- Input: All matching relation edges for a (d,s) pair
- Output: $r_q(s)$

- Algorithm to calculate n and l

  n = Number of relation edges in the input

  Initialize uidtable, l=0

  for each relation edge in query

  find the matching edge in input

Algorithm (cont)

if edge found

if $UWID1_q \in$ uidtable // got a link

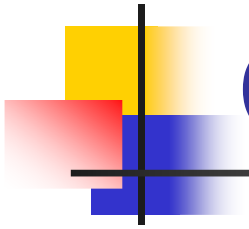
if UWID1 of current relation $\in$ uidtable($UWID1_q$)
//correct link

l++

else uidtable($UWID1_q$) = uidtable($UWID1_q$) $\cup$ UWID1

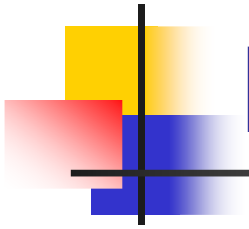
else uidtable($UWID1_q$) = { UWID1 of current relation }

// Repeat the same procedure for second UW also



Conclusion

- Language barrier eliminated
- Benefit of intermediate language representation
- Meaning based search



Future Work

- Testing on a bigger corpus
- Improving dictionary and rule base of ENCO/DECO software
- Improving the HTML parser
- Focused Crawler
- Incorporating global page rank