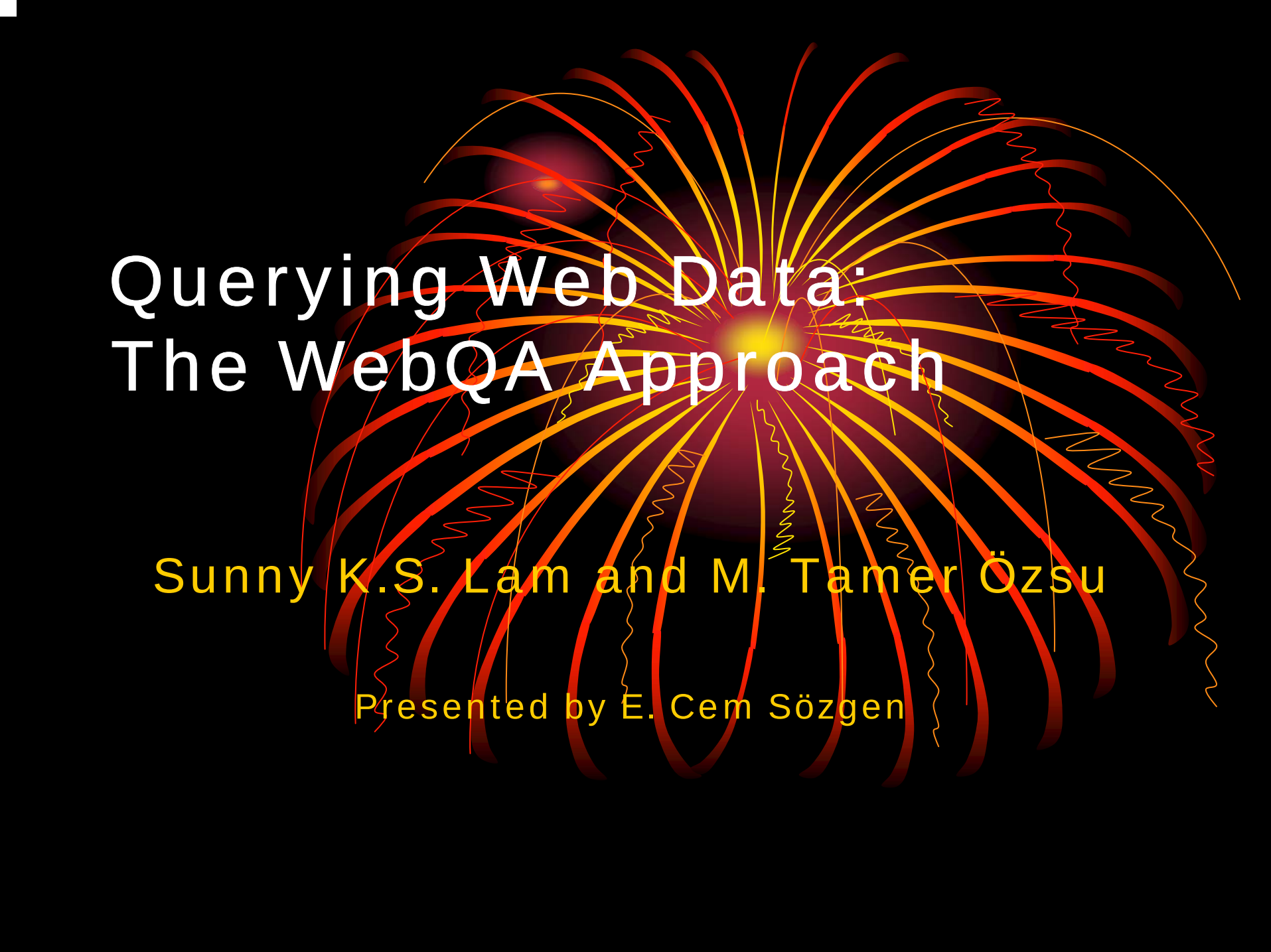


Querying Web Data: The WebQA Approach

Sunny K.S. Lam and M. Tamer Özsu

Presented by E. Cem Sözgen

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

What do people want from a web query system?

- The ideal system for querying the web:

  (from the author's point of view)
  - Accepts easy to pose query (possibly in natural language)
  - Searches all of the sources
  - Returns direct answers (not links)
- How about WebQA?

WebQA

- Factual query expressed in natural language
- Ranked list of short answers

e.g. Who invented the telephone?

1) Alexander Graham Bell (58.0)
2) Graham Bell (58.0)
3) Bell (58.0)
4) Alexander Graham (54.0)

Type of questions that WebQA do not deal?

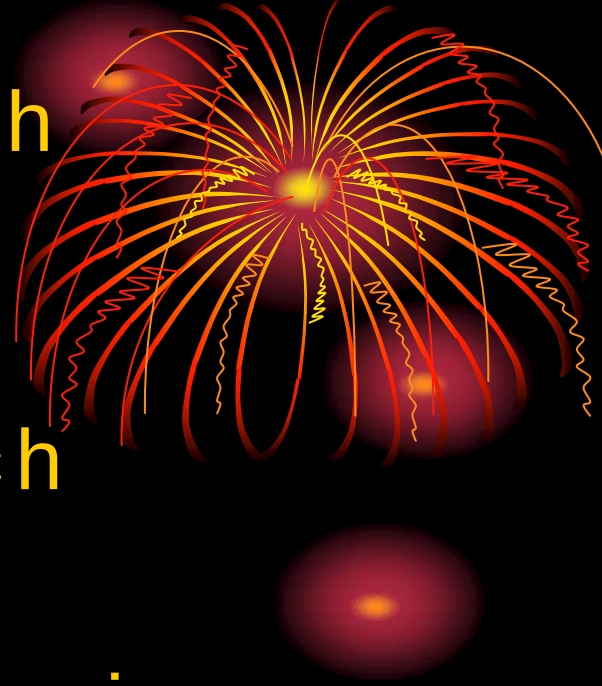
- Who are the players of Toronto Raptors? (multiple results)
- Notify me whenever the temperature of Waterloo drops below zero. (continuous query)
- How do I make pancakes? (procedural query)

Which areas are involved?

- Question answering (QA) techniques
- Metasearch techniques
- Mediator/Wrapper techniques
- Information Retrieval (IR) techniques
- Extraction techniques

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments



- Keyword Search approach
  - Search engines
  - Metasearchers
- Category Search approach
- Database view approach
- Semi-structured data querying approach
- Web Query Language approach
- Learning based approach
- Question answering approach

Mulder

- Very similar to WebQA
- Accepts short factual questions in NL
- Returns exact answers
- Similar main components
- Question types:
  - Nominal: Noun phrase
  - Numerical: Number
  - Temporal: Date
- Uses Google as a search engine

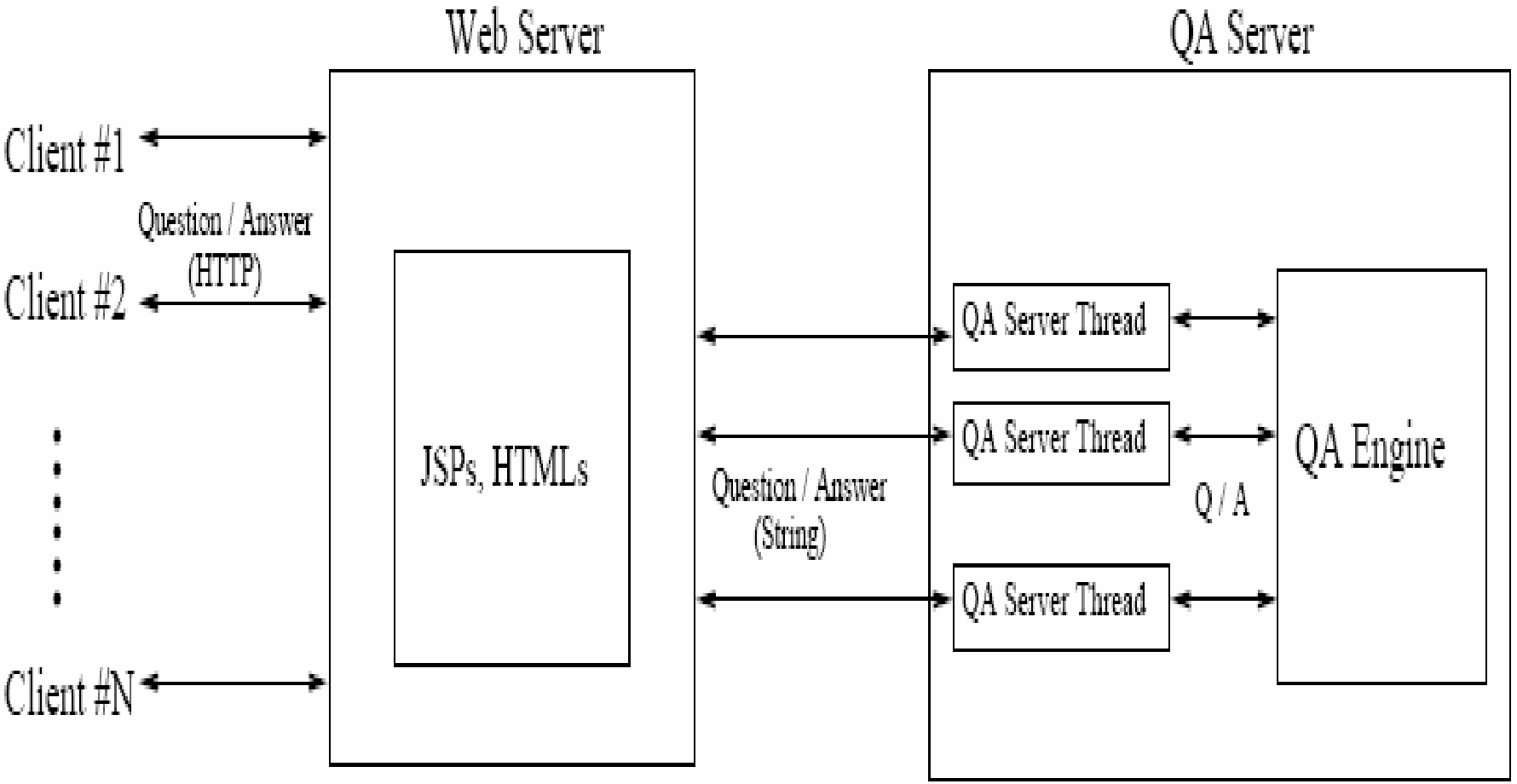
Differences

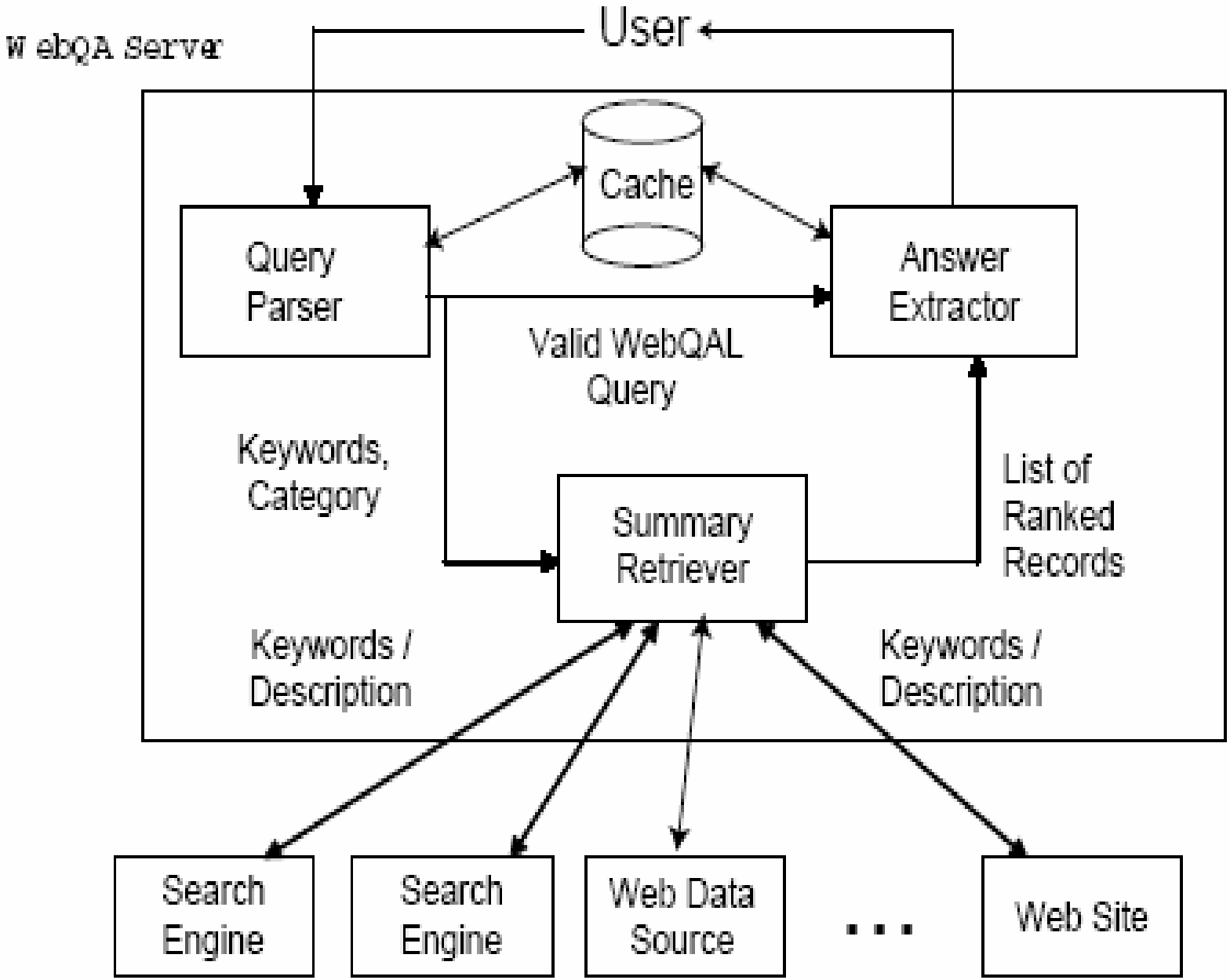
WebQA	Mulder
• Light NLP • 7 categories • Multiple sources • More fault tolerant • More flexible and scalable	• Heavy NLP • 3 categories • Single search engine

Outline

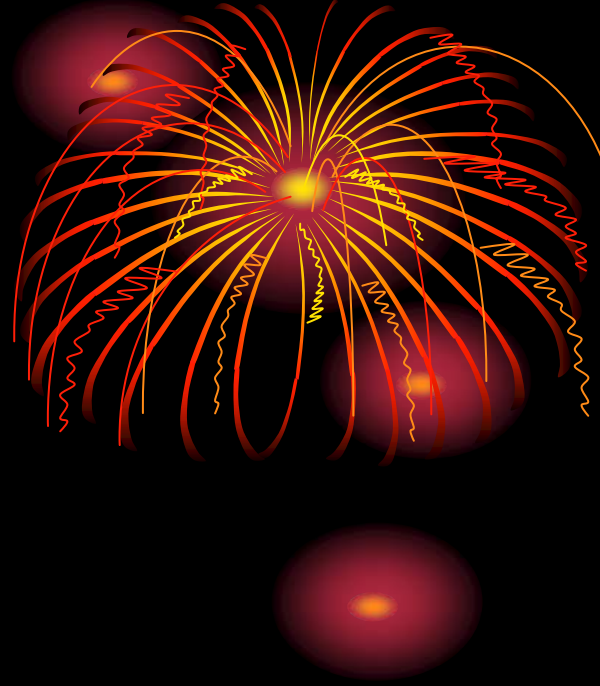
- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

Client-Server Architecture

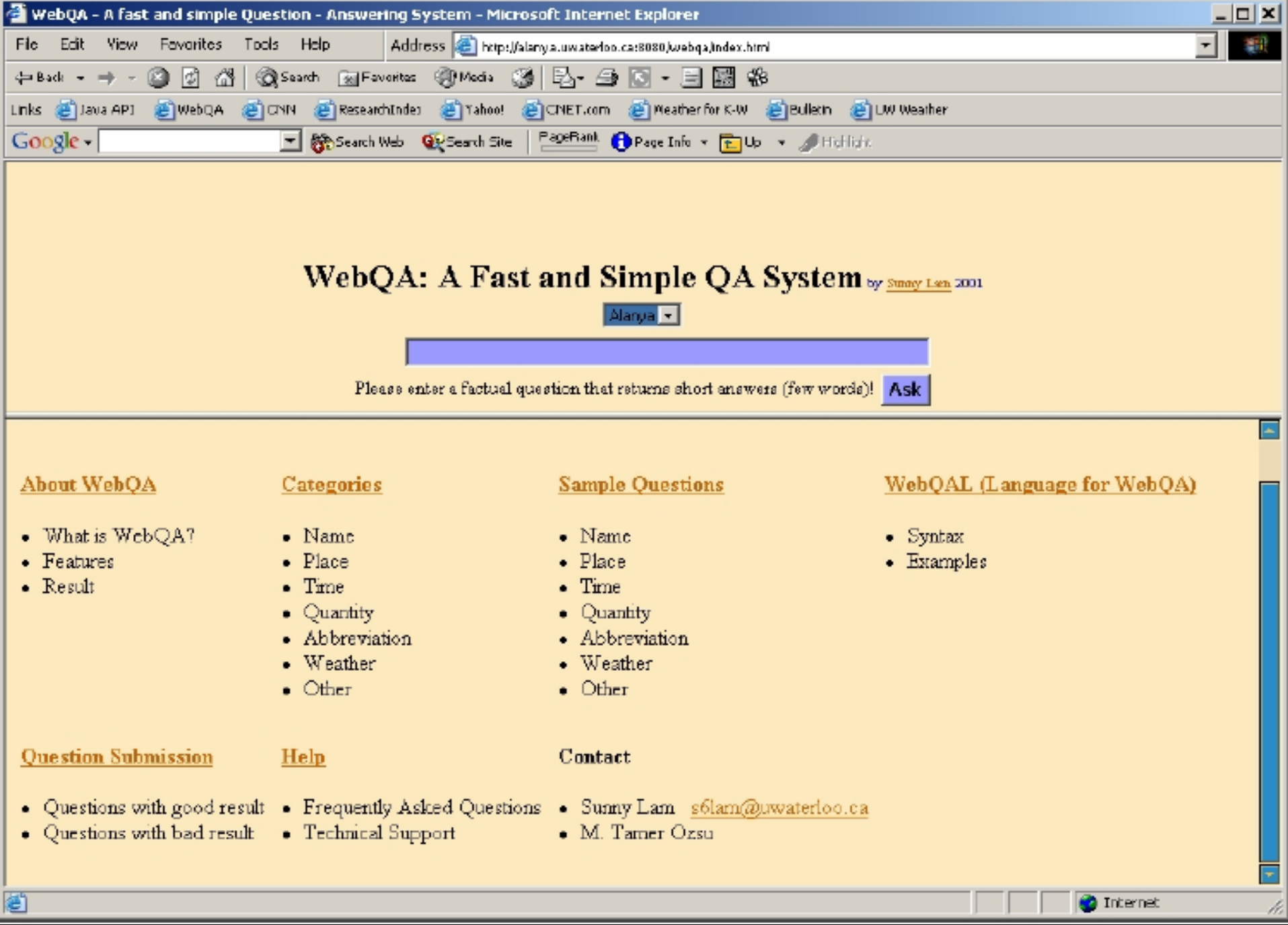



WebQA Server
User
Cache
Query Parser
Answer Extractor
Valid WebQAL Query
Keywords, Category
Summary Retriever
List of Ranked Records
Keywords / Description
Keywords / Description
Search Engine
Search Engine
Web Data Source
...
Web Site

Interface



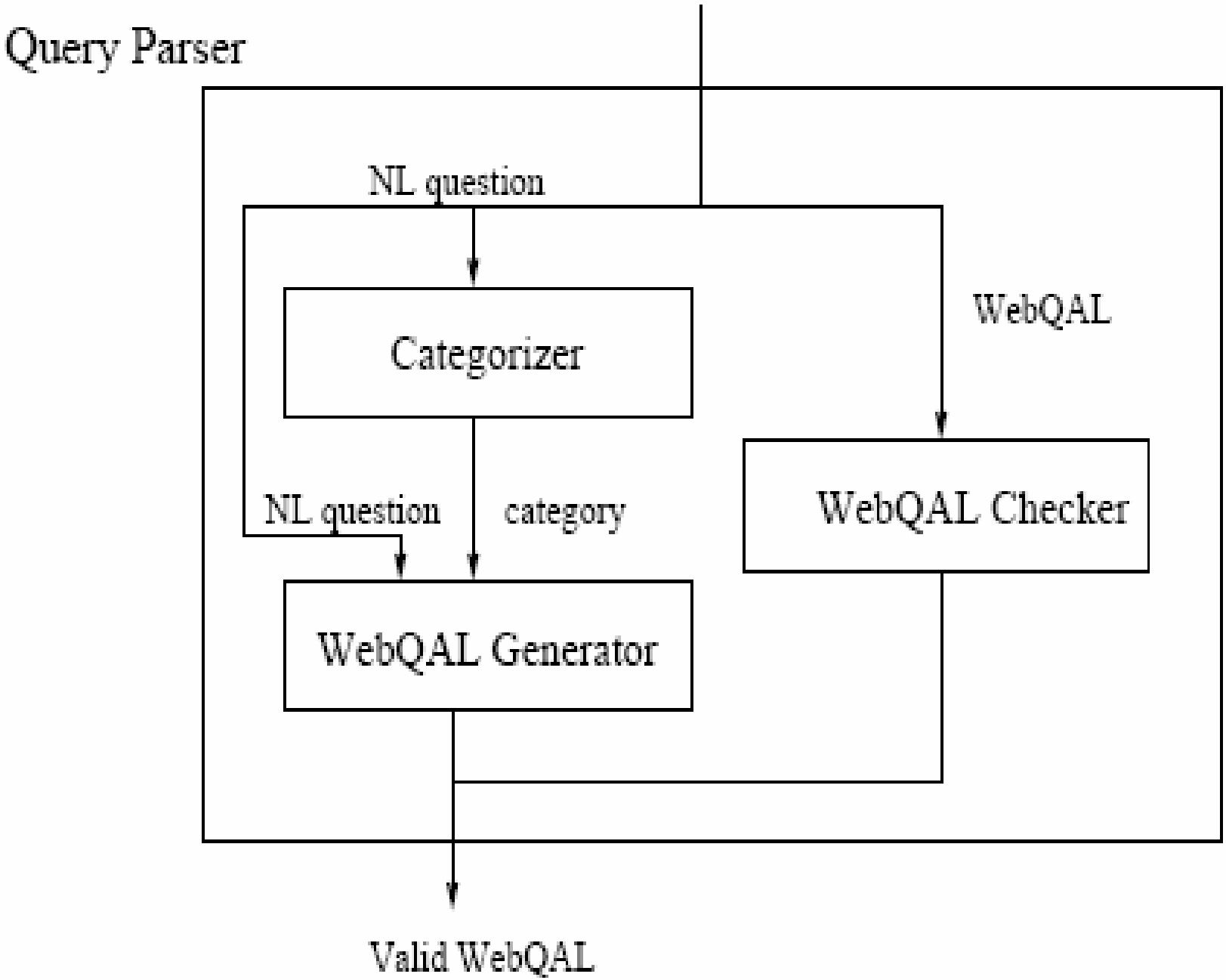
- Two types of interface
  - Textual Interface
    - Local access
    - Fast and provides debugging information
    - Need a copy of WebQA in local machine
- Graphical User Interface



Home page of WebQA

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments


Query Parser
NL question
Categorizer
WebQAL
NL question
category
WebQAL Checker
WebQAL Generator
Valid WebQAL

Categories

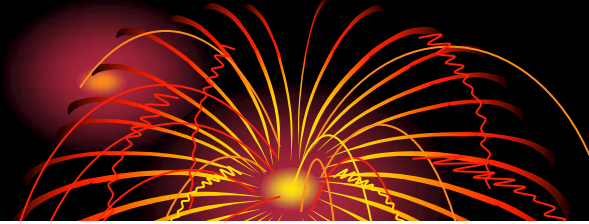
- Defined to improve system accuracy

- Name
- Place
- Time
- Quantity
- Abbreviation
- Weather
- Other

- Who invented the telephone? (Name)
- Who was George Washington? (Other)

```
function categorize(String nlQuestion) : Category
boolean hasWhat ⇐ false
for each word X in nlQuestion do
   if X = "what" or "which" then
      hasWhat ⇐ true
   else if X = "where" then
      return "Place"
   else if X = "when" then
      return "Time"
   else if X = "how" then
      if the word after X is in howQuantityTermList then
         return "Quantity"
      end if
   else if X = "who" or "whom" then
      newQuestion ⇐ nlQuestion without stopwords
      if every word in newQuestion starts with a upper letter then
         return "Other"
      else
         return "Name"
      end if
   else if X is in nameTermList then
      if hasWhat then
         return "Name"
      end if
   else if X is in placeTermList then
      if hasWhat then
         return "Place"
      end if
   else if X is in timeTermList then
      if hasWhat then
         return "Time"
      end if
   else if X is in quantityTermList then
      if hasWhat then
         return "Quantity"
      end if
   else if X is in abbreviationTermList then
      if hasWhat then
         return "Abbreviation"
      end if
   else if X is in weatherTermList then
      if hasWhat then
         return "Weather"
      end if
   else if X is in otherTermList then
      if hasWhat then
         return "Other"
      end if
   end if
end for
return "Other"
```

Output Options

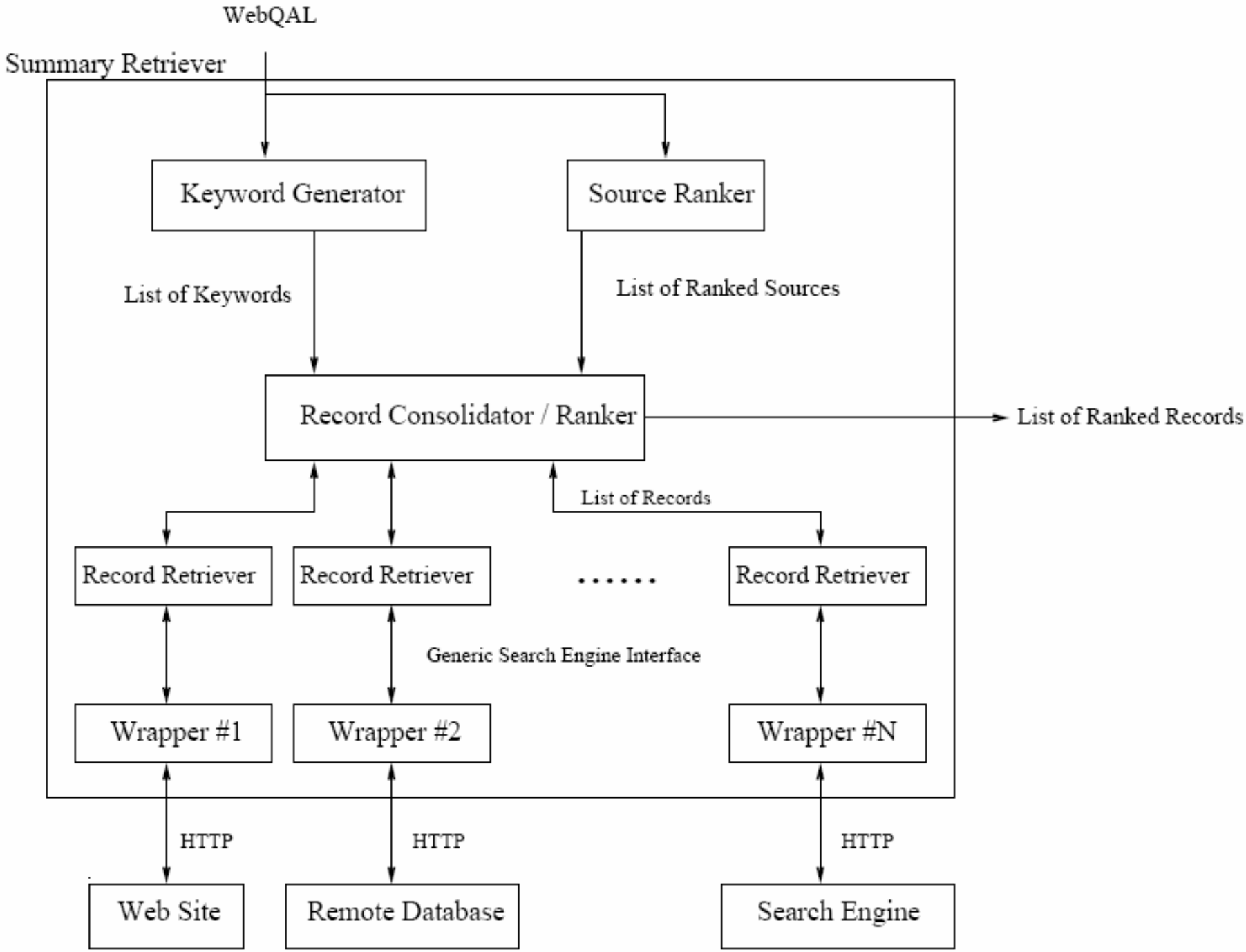


<Category>
[-output
<Output Option>]
-keywords
<Keyword List>

Category	Output Option
Name	N/A
Place	city division country continent unknown
Time	dd mm yyyy dd/mm mm/dd dd/mm/yyyy mm/dd/yyyy
Quantity	*any measurement unit*
Abbreviation	short long
Weather	all conditions temperature barometer wind dewpoint humidity visibility sunrise sunset moonrise moonset
Other	N/A

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments


WebQAL
Summary Retriever
Keyword Generator
Source Ranker
List of Keywords
List of Ranked Sources
Record Consolidator / Ranker
List of Ranked Records
List of Records
Record Retriever
Record Retriever
.....
Record Retriever
Generic Search Engine Interface
Wrapper #1
Wrapper #2
Wrapper #N
HTTP
HTTP
HTTP
Web Site
Remote Database
Search Engine

- List used by Source Ranker

Category	Rank	Sources	Number of Records Needed
Name	1	Yahoo	50
Place	1	World Factbook	5
	2	Yahoo	50
	3	Excite	50
	4	Overture	50
Time	1	World Factbook	5
	2	Yahoo	50
	3	All The Web	50
Quantity	1	World Factbook	5
	2	Yahoo	50
Abbreviation	1	World Factbook	5
	2	Yahoo	50
	3	All The Web	50
Other	1	World Factbook	5
	2	Yahoo	50

- The structure of a record

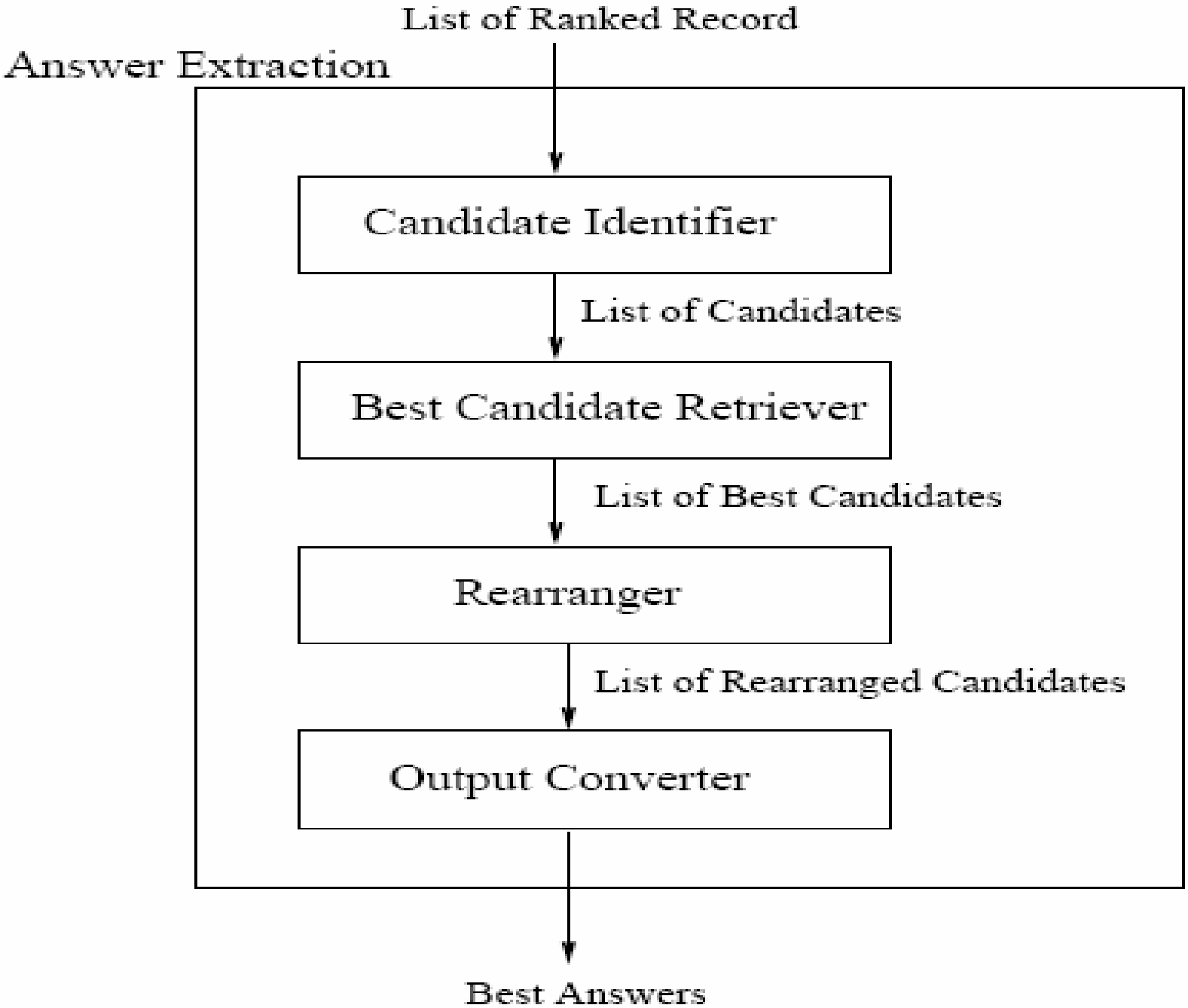
Attribute	Data Type
Source Name	String
Snippet	String
Local Rank	Integer

Mediator/Wrapper

- For information integration
- One wrapper for each data source
- Same Wrapper API
- One centralized mediator
- Different from data warehouse: integrated data is not materialized

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments


List of Ranked Record
Answer Extraction
Candidate Identifier
List of Candidates
Best Candidate Retriever
List of Best Candidates
Rearranger
List of Rearranged Candidates
Output Converter
Best Answers

Candidate Identifier

Attribute	Data Type
Name	String
Score	double

Structure of a Candidate

- Candidate list: list of candidates
- Four sub-identifiers
  - Country sub-identifier
  - Abbreviation sub-identifier
  - Weather sub-identifier
  - Search engine sub-identifier

Rearranger

1) Bell (58.0)
2) Alexander Graham Bell (50.0)

1) Alexander Graham Bell (58.0)
2) Bell (58.0)

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

Experiment 1

- To see the performance of categorizing questions
- TREC 9: 686/693 -> 98.99%
- TREC 10: 461/500 -> 92.2%

Experiment 2

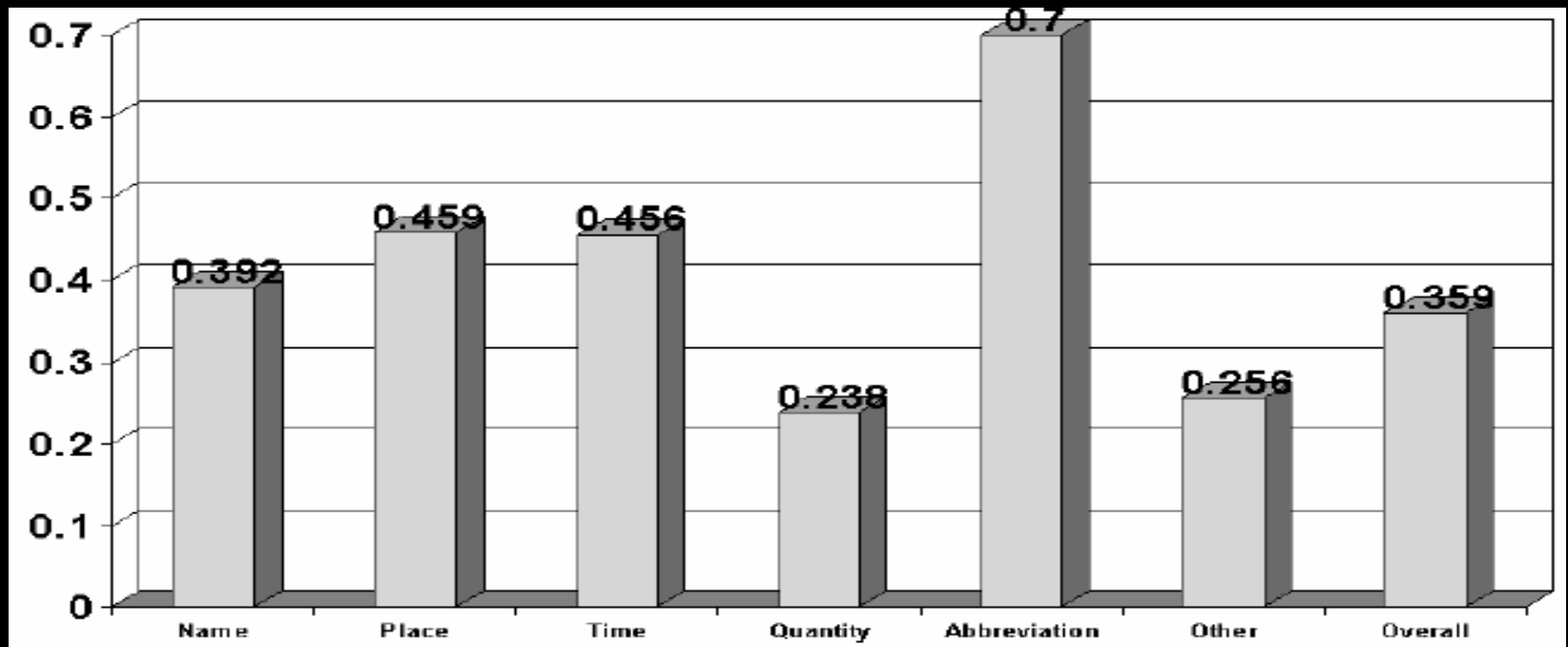
- To determine the best source ranking for each category

Category	Rank	Sources	Number of Records Needed
Name	1	Yahoo	50
Place	1	Yahoo	50
	2	Excite	50
	3	Overture	50
Time	1	Yahoo	50
	2	All The Web	50
Quantity	1	Yahoo	50
Abbreviation	1	Yahoo	50
	2	All The Web	50
Other	1	Yahoo	50

Experiment 3

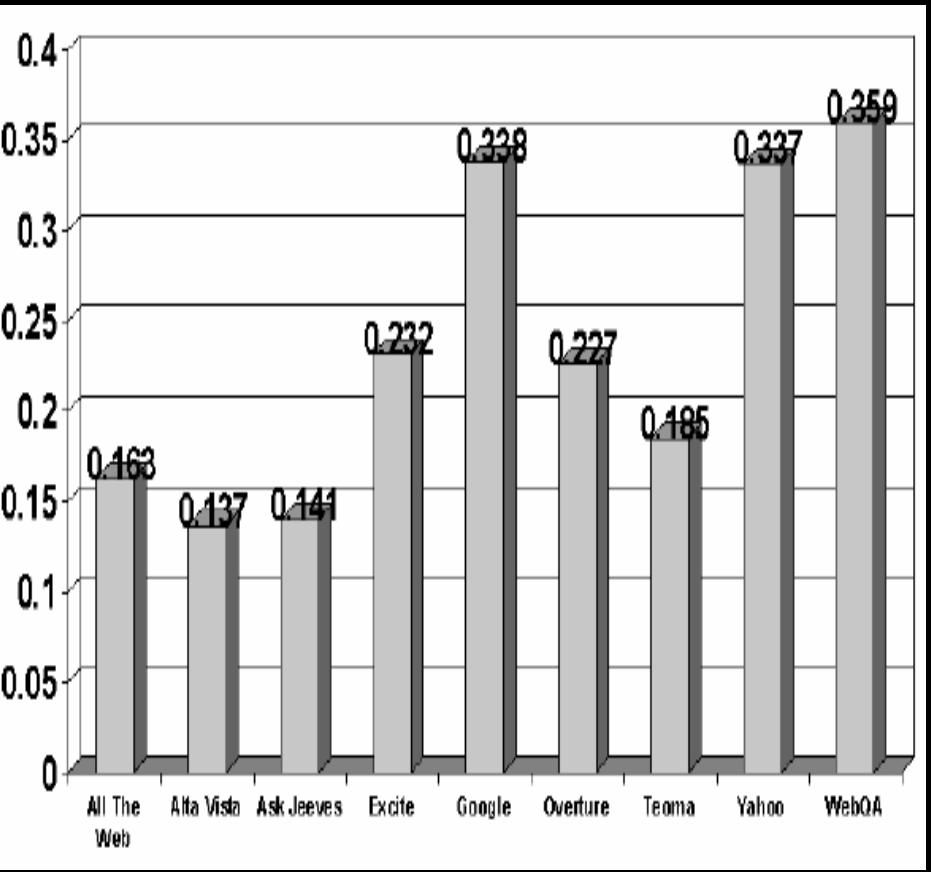
- To see how using secondary sources affects the results

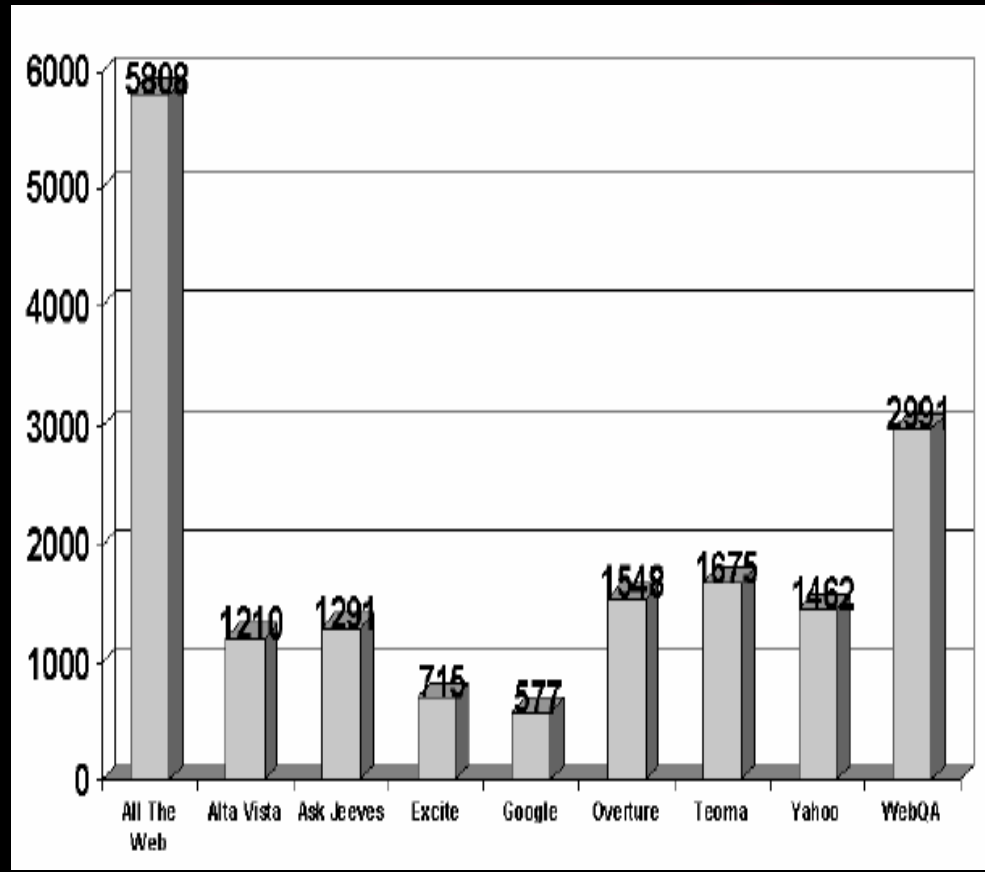
	With Secondary Source	Without Secondary Source
TREC-9 Score	0.35	0.359
Response Time	2981	2991



Experiment 4

- Comparison of WebQA with other systems





Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

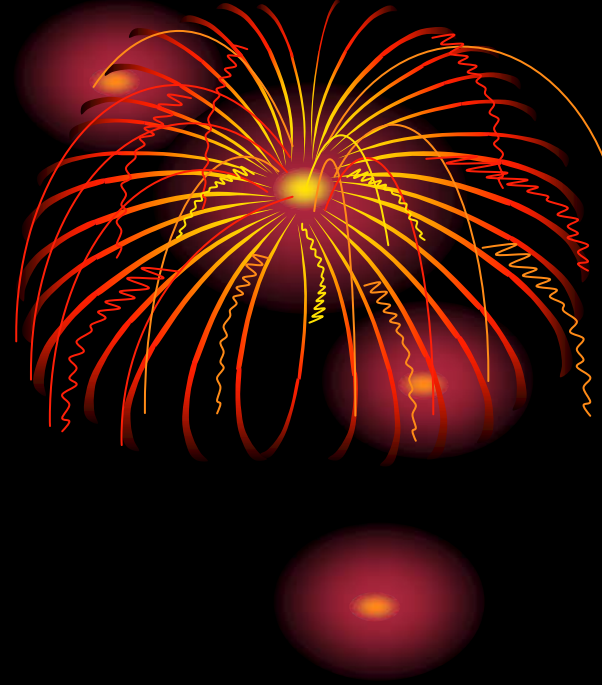
References

1) M.T. Özsu and P. Valduriez, *Principles of Distributed Database Systems, 2nd edition*, Prentice-Hall, Inc., 1999; ISBN 0-13-659707-6
2) S.K.S. Lam and M. T. Özsu. “Querying Web Data – The WebQA Approach,” In *Proc. 3rd International Conference on Web Information Systems Engineering*, Singapore, December 2002, pages 139-148.
3) S. K. S. Lam. *WebQA: A web querying system using the QA approach*. Master's thesis, University of Waterloo, School of Computer Science, Waterloo, Canada, Spring 2002.
4) http://www.viz.co.nz/internet-facts.htm
5) C. C. T. Kwok, O. Etzioni, and D. S.Weld. Scaling question answering to the Web. In *Proceedings of 10th International World Wide Web Conference*, pages 150–161, 2001.

Outline

- Introduction
- Background and Literature
- WebQA Architecture
- Query Parser
- Summary Retriever
- Answer Extractor
- Evaluation
- References
- Comments

Comments...



The followings shows a verb-to-noun conversion table.

create	creator
created	creator
creates	creator
invent	inventor
invented	inventor
invents	inventor
locate	location
located	location
locates	location
own	owner
owned	owner
owns	owner
sang	singer
skate	skater
skated	skater
skates	skater
sing	singer
sings	singer
write	author
writes	author
wrote	author