



# Question Answering Combining Multiple Evidences

Jiangping Chen, He Ge, Yan Wu, Shikun Jiang

Information Retrieval Group  
School of Library and Information Sciences  
University of North Texas

[jpchen@unt.edu](mailto:jpchen@unt.edu)

November 2004

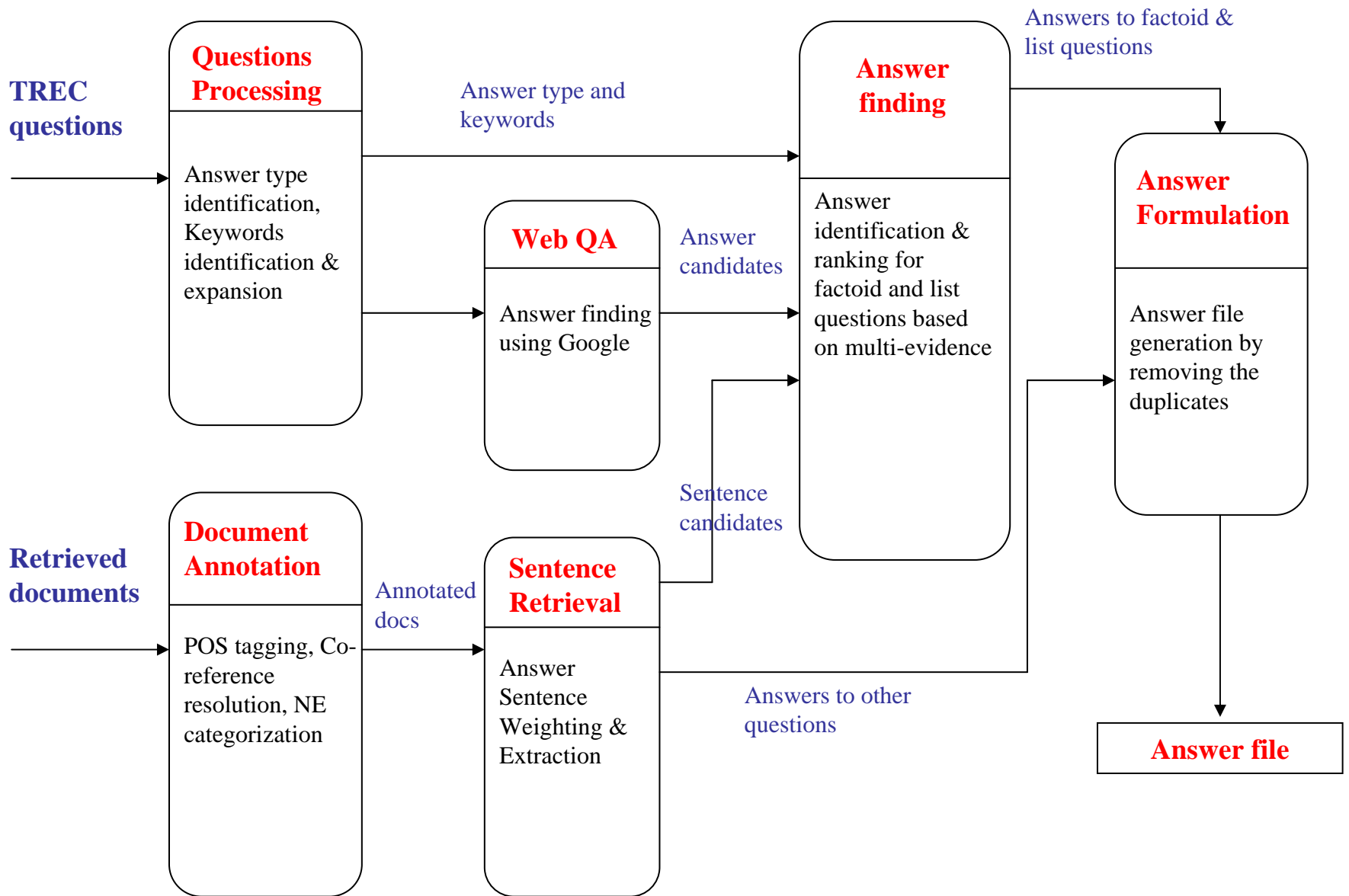
# General Information

- Purpose of QA participation
  - Fast building a QA system: [EagleQA](#)
  - Set foundation for multilingual QA
  - Explore QA strategy based on freely available lexical sources and NLP tools
- About our team
  - First-time participant of TREC QA
  - Very limited financial support
  - Very short system development period

# EagleQA System Overview

1. Question Processing
2. Document Annotation
3. Sentence Retrieval
4. Web QA
5. Answer Finding
6. Answer Formulation

# EagleQA Architecture



# NLP Tools and Knowledge Resources

1. **LingPipe** (<http://www.alias-i.com/lingpipe/>)
  - Sentence boundary detection, named entity detection, within-document co-reference resolution
  - In Java, source codes are available
2. **Minipar** (<http://www.cs.ualberta.ca/~lindek/minipar.htm>)
  - Developed by Dr. Dekang Lin at Uni. Of Alberta
  - Part-of-Speech tagging, phrase bracketing, named entity categorization. Parsing

# NLP Tools and Knowledge Resources

## 3. WordNet

<http://www.cogsci.princeton.edu/~wn/>

WordNet :: QueryData ( Author: Jason Rennie)

- Synonym, hypernym extraction
- Derived forms of words

## 4. Google ([www.google.com](http://www.google.com))

- Document Retrieval from the Web

# The Use of the NLP Tools and Knowledge Resources

- The resources and tools were utilized by certain modules of the QA system
  - WordNet + Minipar for Question Processing

Question: *Which was the first movie that James Dean was in?*  
Answer type: *movie*  
Keywords: *movie(synonym: movie, film, picture, moving picture, moving-picture show, motion picture, motion-picture show, picture show, pic, flick) / James Dean /*
  - LingPipe + Minipar for Document Annotation

```
<sent id="19">
  <TOK id="1" pos="U">Last</TOK>
  <TOK id="2" pos="N">week</TOK>
  <TOK id="3" pos="U">,</TOK>
  <TOK id="4" pos="DET">the</TOK>
  <TOK id="5" pos="N">literature</TOK>
  <TOK id="6" pos="N">prize</TOK>
  <TOK id="7" pos="V" base="go" subj="prize:TOK_6">went</TOK>
  <TOK id="8" pos="PREP">to</TOK>
  <TOK id="9" pos="N" mini_cat="LANG">Portuguese</TOK>
  <TOK id="10" pos="N">novelist</TOK>
<NP id="1" mini_cat="PERSON" ling_type="PERSON">
  <TOK id="11" pos="U">Jose</TOK>
  <TOK id="12" pos="N" mini_cat="PERSON">Saramago</TOK>
</NP>
  <TOK id="13" pos="U">.</TOK>
</sent>
```

Figure 1: Text Annotation Using LingPipe and Minipar

# Other Efforts

- An ontology derived from previous TREC questions
  - 16 top level categories: Animal, Person, Organization, Location, Currency, Code, Quote....
- Heuristic rules for answer type categorization

# Other Efforts (Continue)

- Expansion of information extraction
  - TREC questions cover broader categories such as movie, animal, plant, color that are not categorized by LingPipe and Minipar
  - Developed heuristic rules for certain categories
    - How was somebody die?
  - Collected name lists for certain categories
    - US. presidents/vice presidents
    - NBA teams

# Strategies for QA 2004

- **Factoid Questions**

- Shallow co-reference resolution to questions.  
Replacing “it, he she, their, the+noun ....” with the targets

*4.5 Which was the first movie that he was in?*

=>

*4.5 Which was the first movie that **James Dean** was in?*

# Factoid Questions (Continue)

- Question Processing
  - output expanded keyword lists and answer types
- Sentence Retrieval
  - number of question keywords
  - answer type occurrence in the categorization
  - 500 top ranked sentences were selected for answer finding

# Factoid Questions (Continue)

- **Answer Finding**
  - Answer candidate identification
    - Named entities identified by LingPipe and Minipar
    - WordNet hypernyms
    - Collected lists for certain categories
    - Additional named entity patterns: books, movies...
    - Special patterns for certain questions
  - Candidate ranking. Considering following factors
    - Answer sentence score
    - Candidate identification weight
    - Web QA results

# List Questions

- Same strategy as factoid questions, except
  - Answer fusion: decide the number of answers
    - Remove duplicate answers
    - Simple answer justification based on weights from
      - Answer type
      - Candidate selection strategy
      - Sentence score
- Answer justification needs further exploration

# Other Questions

- Least effort due to time constraints
- Current scoring favors recall. Therefore the key is to find the right sentences
- Strategy: Return the top ranked sentences as answers.
  - Similar to sentence retrieval, but added a weighting factor of keyword distance – the occurrence of the whole target were ranked higher

# Official Results

Run	Factoid (Accuracy)	List (Average F)	Other (Average F)	Final Score
UNTQA04M1	0.187	0.128	0.305	0.202
UNTQA04M2	0.196	0.123	0.305	0.205
UNTQA04M3	0.187	0.127	0.307	0.202
<i>median</i>	<i>0.170</i>	<i>0.094</i>	<i>0.184</i>	

**UNTQA04M1**: not use Google for answer finding

**UNTQA04M2**: Consider short summaries from Google as weighting factor

**UNTQA04M3**: Use a slightly different answer selection strategy for other question

# Analysis

- Analysis is still on-going
- Question processing results were satisfactory, accuracy is above 90%
- Document annotation combining LingPipe and Minipar was okay but needs expansion
  - Categories of named entities are not sufficient
  - False and missing annotation
- Answer finding module needs most work

# Preparing for Next Year

- Refine answer finding strategy
  - Thorough analysis of current answer finding problems
  - Fully testing the ranking algorithm
- Complete testing of Web QA module
  - Current Web QA is not effective
- Investigate better approach for confidence judgment
  - Is this answer trustable?
- Implement text retrieval module



Any Suggestions?  
Thank You Very Much!