

# Learning to Rank Answers on Large Online QA Collections

Mihai Surdeanu, Massimiliano Ciaramita, Hugo Zaragoza

Barcelona Media Innovation Center, Yahoo! Research Barcelona  
mihai.surdeanu@barcelonamedia.org

April 4, 2008



## What is Question Answering?

Answer *natural language* questions with small fragments of *text*.

“What is the capital of Spain?” → “Madrid”

“What is the Future of Web Search Workshop?” → “The main objective of the workshop is to bring together key researchers in the area of Web search and Multimedia retrieval. This event will have a special focus on multimedia and specialized topics in Web search.”



# Motivation

- ▶ Most effort concentrated on factoid and definitional Question Answering (QA), e.g., TREC, CLEF evaluations.
- ▶ Little research and virtually no data available for non-factoid QA, such as manner or reason questions.
- ▶ Recent years have seen an explosion of user-generated content such as community-driven question-answering (Yahoo! Answers).
  - ▶ Advantages: large, open-domain, multilingual.
  - ▶ Disadvantages: high variance of quality.



# Examples

High Quality	Q: How do you quiet a squeaky door? A: Spray WD-40 directly onto the hinges of the door. Open and close the door several times. Remove hinges if the door still squeaks. Remove any rust, dirt or loose paint. Apply WD-40 to removed hinges. Put the hinges back, open and close door several times again.
High Quality	Q: How does a helicopter fly? A: A helicopter gets its power from rotors or blades. So as the rotors turn, air flows more quickly over the tops of the blades than it does below. This creates enough lift for flight.
Low Quality	Q: How to extract html tags from an html documents with c++? A: very carefully



## Goal

- ▶ Is it possible to learn an answer ranking model for complex questions from such noisy data?
- ▶ Which features/models are most useful in this scenario?



## Outline

Introduction

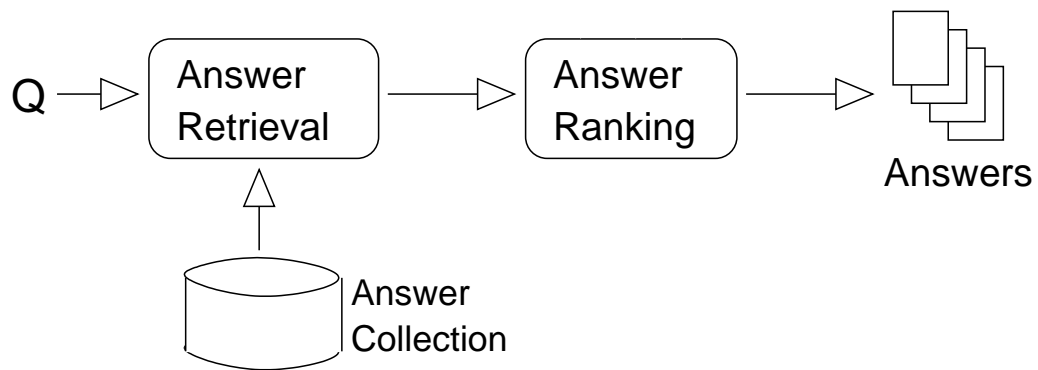
Approach

Experiments

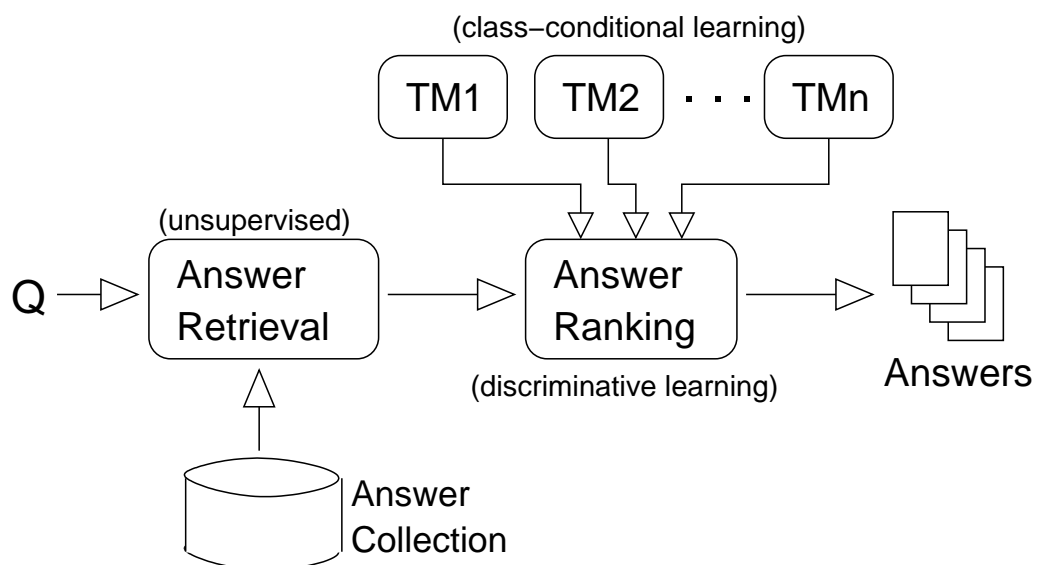
Conclusions



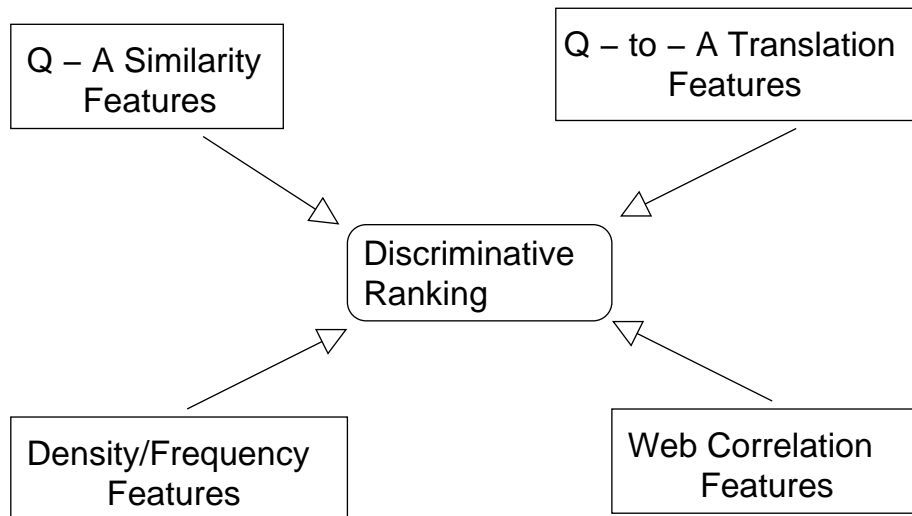
## Approach: System Architecture



## Approach: System Architecture



## Approach: Learning Framework



- ▶ Positive samples: Answers marked as best in Yahoo! Answers.
- ▶ Negative samples: All other answers retrieved by IR.



## Features

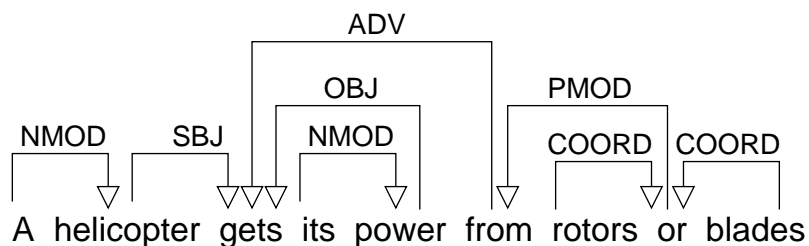
1. FG1: Similarity Features
  - ▶ BM25 and  $tf \cdot idf$  between Q and A.
2. FG2: Translation Features
  - ▶  $P(Q|A)$  given by IBM Model 1.
3. FG3: Density and Frequency Features
  - ▶ Same word sequence - Q terms recognized in the same order in A.
  - ▶ Answer span - largest distance between two Q terms in A.
  - ▶ Same sentence match - number of Q terms matched in a single sentence in A.
  - ▶ Overall match - number of Q terms matched in A.
  - ▶ Informativeness - number of NN, VB, JJ in A that are not found in Q.
4. FG4: Web Correlation Features
  - ▶ Web correlation - CCP using search engine hits.
  - ▶ Query-log correlation - PMI and  $\chi^2$  between (Q, A) words and a large query log.



## Representation of Content: Structures

To investigate the contribution of NLP, we replicate most features for five different representations of content:

- ▶ *Words (W)* - the text is seen as a bag of words.
- ▶ *N-grams (N)* - the text is represented as a bag of  $n$ -grams.
- ▶ *Dependencies (D)* - the text is represented as a bag of syntactic dependencies.



## Representation of Content: Structure Parameters

- ▶ Degree of lexicalization:
  - ▶ Fully lexicalized structures, e.g., “helicopter”  $\xrightarrow{\text{SBJ}}$  “get”.
  - ▶ Lexical elements replaced with coarse WordNet super senses (WNSS), e.g., n.artifact  $\xrightarrow{\text{SBJ}}$  v.possession.
  - ▶ Lexical elements replaced with WSJ NE tags, e.g., VEHICLE  $\xrightarrow{\text{SBJ}}$  “get”.
- ▶ Labels of relations: dependency relations can be labeled or unlabeled, e.g., “helicopter”  $\xrightarrow{\text{SBJ}}$  “get” vs. “helicopter”  $\rightarrow$  “get”.
- ▶ Structure size: controls the maximum number of elements in  $n$ -grams or dependency chains.



# Outline

Introduction

Approach

Experiments

Conclusions



## The Corpus

- ▶ Corpus build from a Nov. 2007 sample of Yahoo! Answers. Users ask questions and answer other users' questions. Best answers chosen by the asker or voted by participants.
- ▶ Focused on manner ("how to") questions. Corpus built using 2 filtering steps:
  1. Kept only questions that match the regular expression:  
`how (to|do|did|does|can|would|could|should)`  
and have an answer selected as best either by the asker or by the participants in the thread.
    - ▶ 364,419 (Q, best A) pairs.
  2. Removed the questions and answers of obvious low quality.
    - ▶ Heuristic: Both Q and A must have at least 4 words, out of which at least 1 noun and 1 verb.
    - ▶ 142,627 (Q, best A) pairs.
    - ▶ We index all As in this set as the collection **C**.
    - ▶ Partitioning of questions: 60% training, 20% development, 20% testing.



# Measures

- ▶ We evaluate results using two measures:
  1. Precision at rank 1 (P@1) - percentage of questions with correct answer on first position.
  2. Mean Reciprocal Rank (MRR) - score of a question is  $1/k$ , where  $k$  is position of correct answer.
- ▶ We are interested in the ranker's performance: we evaluate on the questions where the correct answer is retrieved from **C** in top  $N$  by Answer Retrieval.



## Overall Results

P@1	N = 10 (26.25% cov.)	N = 15 (29.04% cov.)	N = 25 (32.81% cov.)	N = 50 (38.09% cov.)
IR	45.94%	41.48%	36.74%	31.66%
Ranking	<b>53.48%</b> $\pm 0.01$	<b>49.65%</b> $\pm 0.03$	<b>43.52%</b> $\pm 0.09$	<b>37.51%</b> $\pm 0.09$
Relative Improvement	+16.41%	+19.69%	+18.45%	+18.47%

MRR	N = 10 (26.25% cov.)	N = 15 (29.04% cov.)	N = 25 (32.81% cov.)	N = 50 (38.09% cov.)
IR	61.33	56.12	50.31	43.74
Ranking	<b>67.77</b> $\pm 0.09$	<b>63.85</b> $\pm 0.01$	<b>56.90</b> $\pm 0.07$	<b>49.81</b> $\pm 0.08$
Relative Improvement	+10.50%	+13.77%	+13.09%	+13.87%





## Contribution of NL Analysis

	Individual representations					Combined representations			
	W	N	$N_{WN}$	D	$D_{WN}$	W +N	W +N $+N_{WN}$	W +N $+N_{WN}$ +D	W +N $+N_{WN}$ +D $+D_{WN}$
FG1	0	<b>+1.06</b>	-2.01	+0.84	-1.75	+1.06	+1.06	+1.06	+1.06
FG2	+4.95	+4.73	+5.06	+4.63	+4.66	+5.80	+6.01	<b>+6.36</b>	+6.36
FG3	+2.24	+2.33	+2.39	+2.27	+2.41	+3.56	+3.56	<b>+3.62</b>	+3.62

The NLP analysis provides *complementary* information to the bag-of-word models!



## Contribution of NL Analysis

	Individual representations					Combined representations			
	W	N	$N_{WN}$	D	$D_{WN}$	W +N	W +N $+N_{WN}$	W +N $+N_{WN}$ +D	W +N $+N_{WN}$ +D $+D_{WN}$
FG1	0	<b>+1.06</b>	-2.01	+0.84	-1.75	+1.06	+1.06	+1.06	+1.06
FG2	+4.95	+4.73	+5.06	+4.63	+4.66	+5.80	+6.01	<b>+6.36</b>	+6.36
FG3	+2.24	+2.33	+2.39	+2.27	+2.41	+3.56	+3.56	<b>+3.62</b>	+3.62

The NLP analysis provides *complementary* information to the bag-of-word models!



## Conclusions

- ▶ Answer ranking engine built using a community-generated question-answer collection:
  - ▶ Large-scale experimentation with various models/features.
  - ▶ Potential application: retrieval from social media.
  - ▶ Potential application: open-domain QA on the Web.
- ▶ Combination is key for improvement:
  - ▶ Combined several models: translation, similarity, frequency, density, web correlation.
  - ▶ Combined several representations of content: bag of words, n-grams, dependencies, word senses, NEs.
- ▶ NL analysis yields a small, yet statistically-significant improvement. OK considering that:
  - ▶ We use off-the-shelf NLP processors.
  - ▶ We evaluate on a large corpus with noisy and subjective information.



## Conclusions

- ▶ Answer ranking engine built using a community-generated question-answer collection:
  - ▶ Large-scale experimentation with various models/features.
  - ▶ Potential application: retrieval from social media.
  - ▶ Potential application: open-domain QA on the Web.
- ▶ Combination is key for improvement:
  - ▶ Combined several models: translation, similarity, frequency, density, web correlation.
  - ▶ Combined several representations of content: bag of words, n-grams, dependencies, word senses, NEs.
- ▶ NL analysis yields a small, yet statistically-significant improvement. OK considering that:
  - ▶ We use off-the-shelf NLP processors.
  - ▶ We evaluate on a large corpus with noisy and subjective information.



# Conclusions

- ▶ Answer ranking engine built using a community-generated question-answer collection:
  - ▶ Large-scale experimentation with various models/features.
  - ▶ Potential application: retrieval from social media.
  - ▶ Potential application: open-domain QA on the Web.
- ▶ Combination is key for improvement:
  - ▶ Combined several models: translation, similarity, frequency, density, web correlation.
  - ▶ Combined several representations of content: bag of words, n-grams, dependencies, word senses, NEs.
- ▶ NL analysis yields a small, yet statistically-significant improvement. OK considering that:
  - ▶ We use off-the-shelf NLP processors.
  - ▶ We evaluate on a large corpus with noisy and subjective information.



Thank you!

