Microsoft Research
微软亚洲研究院

Answer Extraction with Multiple Extraction Engines for Web-based Question Answering

Hong Sun[1], FuruWei[2], Ming Zhou[2]

Tianjin University[1]
Microsoft Research Asia[2]

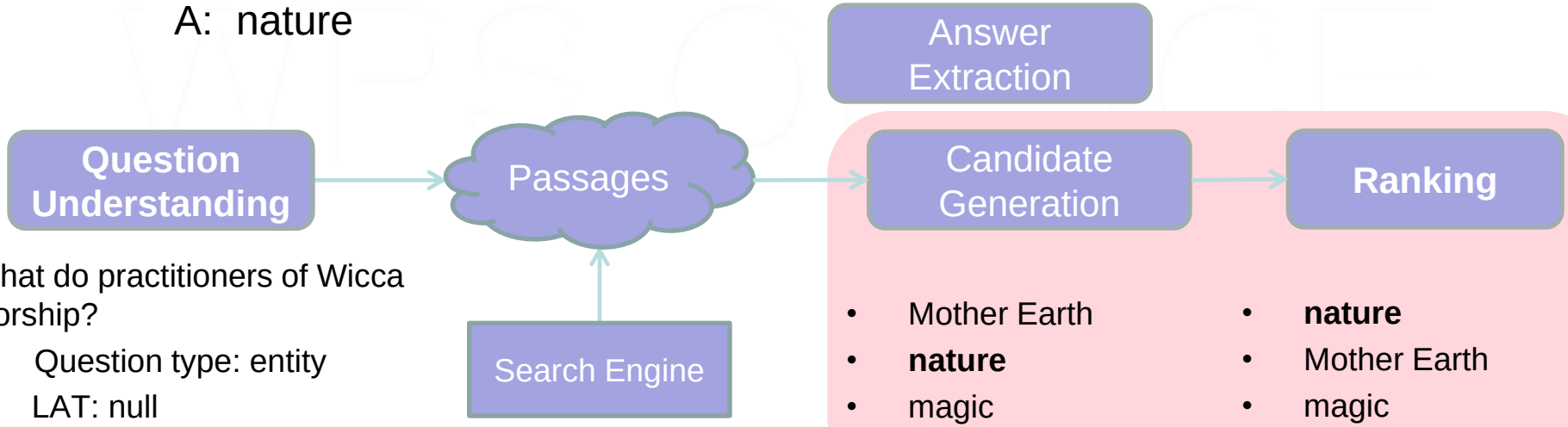
Contents

Contents

Introduction

- Problem Definition
  - Question Answering (QA)
    - Automatically give answers to questions described in natural language

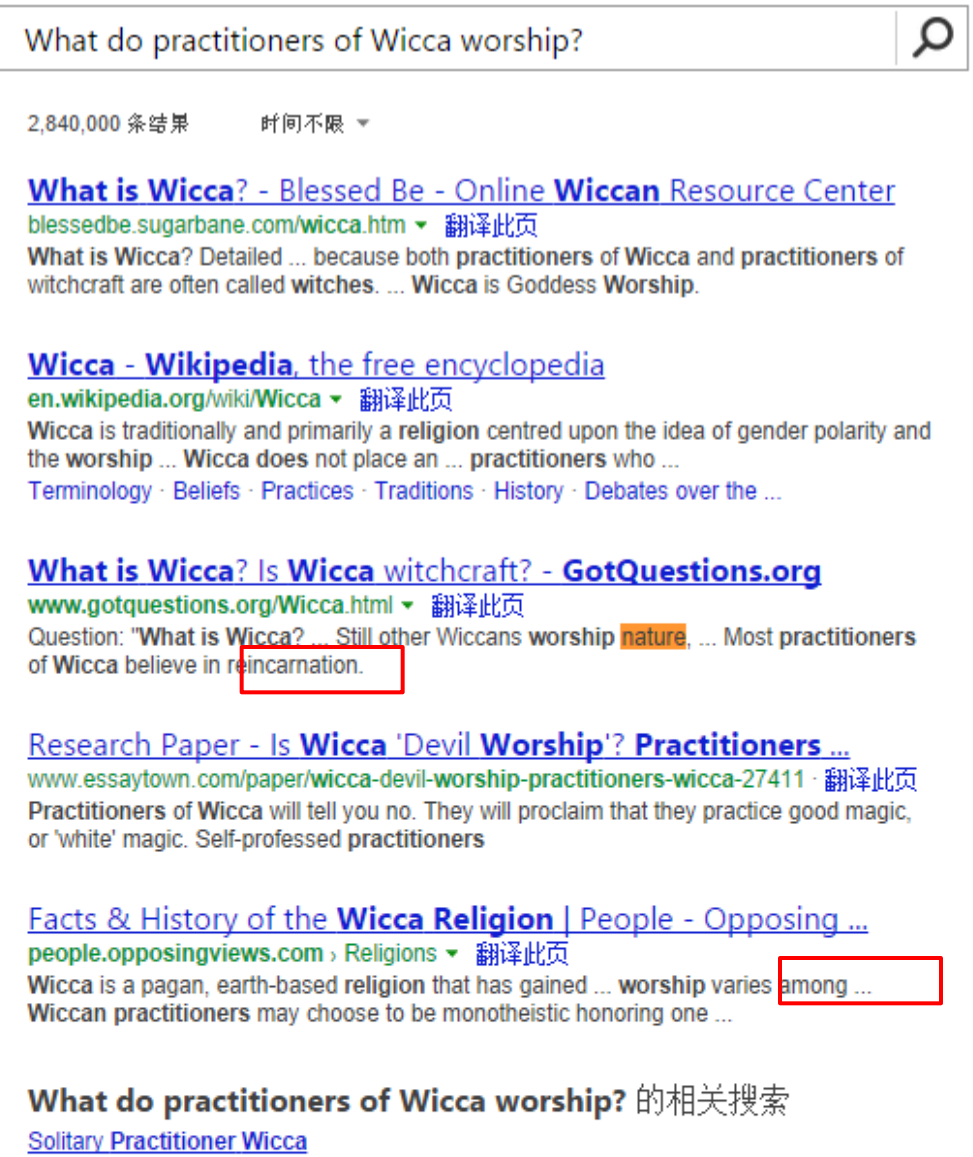
Q: What do practitioners of Wicca worship?
A: nature



Introduction

- Answer Extraction for Web-based Question Answering
  - Automatically pin-point exact answers from search snippets for the questions
  - Input
    - Search snippets retrieved by search engine
  - Output
    - Ranked candidates
      - Identify candidates from the snippets
      - Rank candidates based on certain ranking function, e.g. frequency
- Search engine snippets
  - Fast response; wide coverage; real-time contents; high-quality summarizations
  - Side-effect

Introduction



- Search snippets side-effect
  - Large amount of noises
    - 18% (traditional document set) vs 10% (search snippets) positive sentences rate
    - High pressure on ranking
  - Incomplete sentences
    - Affect syntactic analysis results
    - Affect methods relied on syntactic structures, e.g., NP, tree kernel

Introduction

- Answer Extraction with Multiple Extraction Engines for Web-based Question Answering
  - Different methods analyze text from different aspects
    - The chance of them all being wrong is small
  - Consensus information among them play important roles for
    - Pruning noisy candidates
    - Performing more accurate ranking

Contents

Related Work (Candidate Generation)

- Pattern based Method (Soubbbotin, 2001; Ravichandran and Hovy, 2002) Birthyear: “Capitalized words” (“**four-digits**” – “four-digits”)
  - High precision
  - Restricted by pre-defined question type
- NER based generation (Pasca and Harabagiu, 2001; Xu et al., 2003)
  - Most QA systems use NER
  - Extract entities and retain those ones matched with answer type
  - NER specially for QA is necessary
- N-grams (Brill et al., 2001)
  - For Web-based Question Answering
  - Collect high frequent n-grams
  - Huge amount of candidates, hard to rank
- Other units
  - NPs or dependency tree nodes (Sun et al., 2005; Shen and Klakow, 2006; Chrupala and Dinu, 2010)
  - Dictionary (Na et al., 2002; Echihabi et al., 2006; Chu-Carroll and Fan, 2011)

Related Work (Candidate Generation)

- Question-Biased Term Extraction (Sasaki, 2005)
  - Label each word in the passage with *{BIO}* labels
  - A Maximum Entropy classifier is trained
  - Features: n-gram overlapping; POS tag overlapping; context words/POS tags

- Answer Extraction as Sequence Tagging with Tree Edit Distance (Yao and Durme 2013)
  - Use CRF to give *{BIO}* labels to the words
  - Leverage tree edit distance features

- Answer Extraction with Graphic Model (Sun et.al, 2013)

Related Work (Ranking)

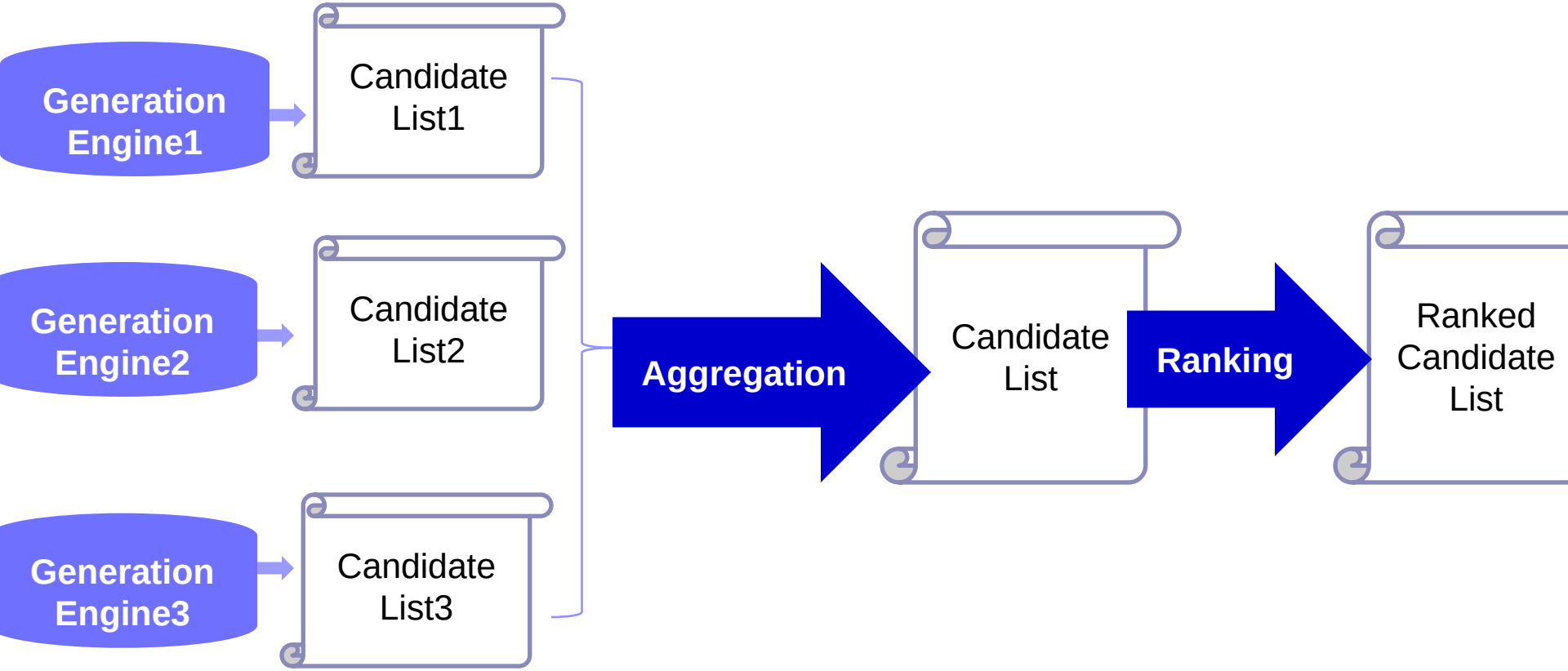
- Ranking Evidence
  - Similarity between question and candidates
    - Syntactic similarity
    - NP-based methods
  - Redundancy
    - NER, pattern, n-grams
  - Integration
    - Combine different evidences with a classification model
    - Tree edit distance; tree kernel (Severyn and Moschitti, 2013)

- So far
  - Single generation engine
  - Rely on syntactic structures
    - Search snippets' negative impact on that structure

Contents

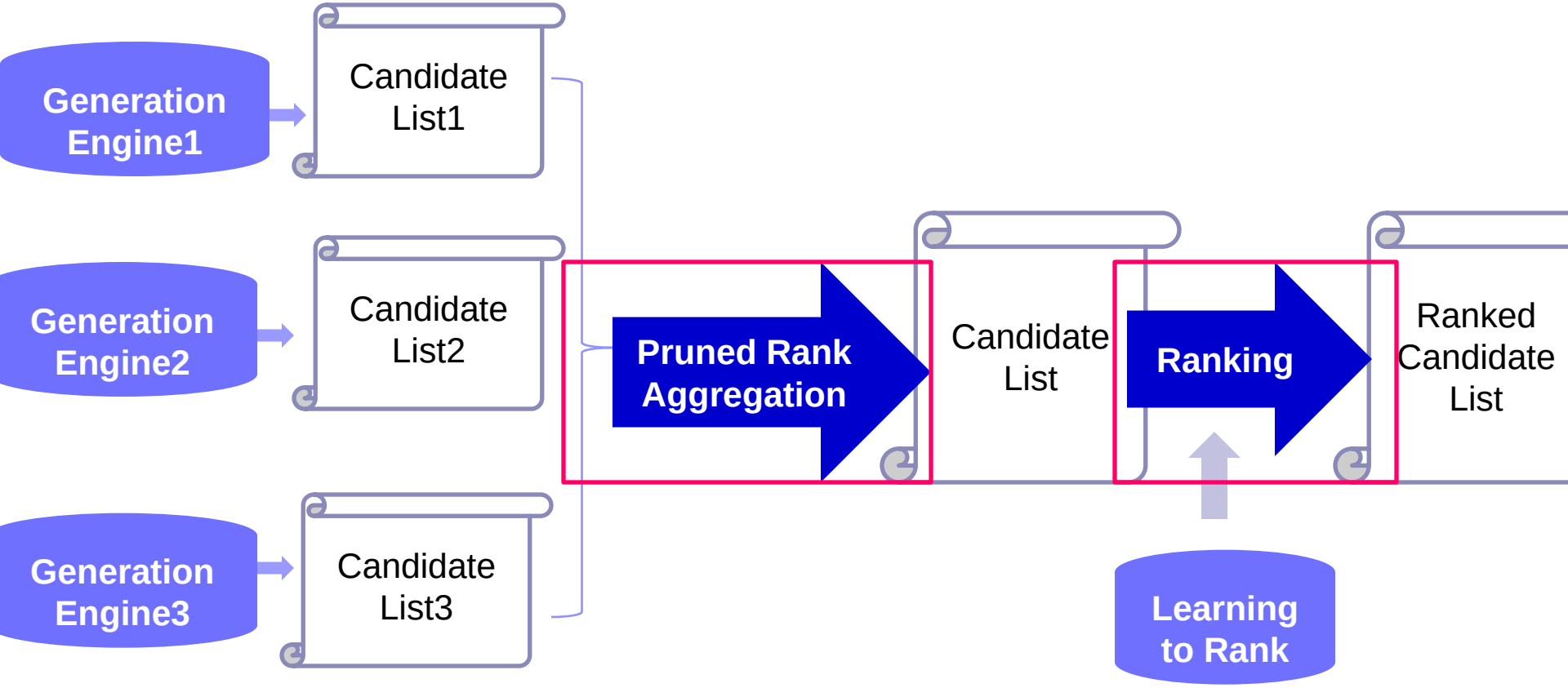
Method

- Answer Extraction with Multiple Extraction Engines



Method

- Answer Extraction with Multiple Extraction Engines



Pruned Rank Aggregation

Algorithm 1 Pruned Rank Aggregation with Multiple Extraction Engines

Input: $C_i = \{c_{i1}, ..., c_{il}\}$ as ordered candidate list from engine i, $l_i = |C_i|$, each list ordered by frequency of candidate $n_{c_{ij}}$; $\{w_1, ..., w_k\}$ are weights of each engine, $0 \leq w_i \leq 1, \sum_i w_i = 1$; $\{t_i, ...t_k\}$ are single engine pruning thresholds, $0 \leq t_i \leq 1$; p, m, n are pruning thresholds after aggregation.

Output: $A = a_1, ..., a_n$ is candidate list after aggregation and pruning

Initialize $A = \emptyset$
for all C_i **do**
  for all $c_{ij} \in C_i$ **do**
    if !ContainContentWord(c_{ij}) or ($j > t_i \cdot l_i$ and !Exists(c_{ij}, C'_i),$\forall i' \neq i$) **then** Remove c_{ij} from C_i
    end if
  end for
  Update $l_i = |C_i|$
end for
function MERGELISTWITHMODIFIEDSKR($\{C_i\}$)
  Initialize $M_{i,j} \Leftarrow 0$, $C' = \bigcup_i C_i$, update frequency $n_{c_k} = \Sigma_j n_{c_{ij}}$, if $c_k = c_{ij}$
  for all candidate list C_i **do**
    for all $j = 1$ to $l_i - 1$ **do**
      for all $l = j+1$ to l_i **do** $M_{c_{ij},c_{il}} \Leftarrow M_{c_{ij},c_{il}} + w_i \cdot \log(n_{c_{ij}} - n_{c_{il}} + 0.1)$
      end for
    end for
  end for
  Quick sort C' with $M_{C'_i,C'_j}$, candidate with larger value gets prior order
  return C'
end function
Compute word frequency f_{w_j} for all $w_j \in c_i$ and !IsStopword(w_j), where $c_i \in C'$
for all $c_i \in C'$ **do**
  if $i \leq |C'| \cdot p$ or $\Pi f_{w_i} \geq n, w_i \in c_i$ or $n_{c_i} \geq m$ **then** Add c_i to A
  end if
end for

- Leverage consensus information among different engines for more strict pruning
- Prune candidate exists in a single candidate list with low rank
- Rank aggregation with a modified score considering different engines' weights
- Postprocess after aggregation, further prune candidates

Method

- Ranking
  - Learning to Rank
    - Given Q, hypothesis space $h \in H(Q)$,compute candidate's score with $P(h|e) \times P(e|h,Q,R',K)$

      e, evidence; R' search results; K, Knowledge Base

      $$P(h|e) \times P(e|h,Q,R',K) \propto g(h) = \sum_{i=1}^{m} f_i(h,R',K,Q,H)\lambda_i$$

    - Select final answer with

      $$h^* = \arg\max_{h \in H(Q)} g(h)$$

    - Based on RankSVM (Joachims, 2002) algorithm

Method

- Features
  - Censuses features (9)
    - Measure different engines' agreements of a given candidate
    - Number of generation engines supporting the candidate, the rank, ect.
  - Redundancy features (7)
    - Measure the redundancy of a given candidate in the hypothesis
    - Frequency, n-gram based redundancy scores, etc.
  - Similarity features (41)
    - Measure text and semantic similarities between candidate and question
    - LCS, normalized LCS, match with answer type, etc.

Method

- Features
  - Candidate quality features (9)
    - Measure a candidate's quality
    - Number of stop word tokens, etc.
  - Search features (7)
    - Additional evidence given by search engine
    - Best rank given be search engine, extracted from title or snippet, etc.

Method

- Extraction engines
  - CRF sequential labeling model
    - Label each word with a {B,I,O} label
    - Similar to (Sun et.al, 2013)
    - High precision, low recall
  - Wiki-title-based method
    - Use Wikipedia titles as dictionary to generate candidates
    - 7.8 million entries in total
    - High recall, but without considering context information
  - Noun Phrase-based method
    - Extract NPs from search snippets
    - High recall, but rely on syntactic analysis results

Contents

Experiments: Setting

- QA Components
  - Question analysis
    - Rule based question type and focus detection
    - Use rule-based method to generate queries
      - Named Entities; question string; verb, noun, adj, adv words; Noun phrases and verbs; verb and its dependent
  - Passage retrieval
    - Use search engine to retrieve passages
    - Top 20 snippets for each query
    - Select top 60 snippets most similar to question
      - Classified by several plain text similarity features and RankSVM model

Experiments: Data

- Public QA Data Set
  - TREC 1999-2007
  - Only retain questions whose answers are contained in the search snippets
  - Also employ previous data set(Yao et.al, 2013; Severyn and Moschitti, 2013)

Data	Question Number	Passage Type	Avg. Passage Per Q	Rate of Positive Passages Per Q(%)
Train	1200	Search	60	9.82
Test-1 (previous)	75	Search	60	10.33
	89	Document	17	18.72
Test-2	293	Search	60	10.11

Search snippets' positive passage rate is much lower than documents'

Experiments: Metrics

- Evaluation Metrics
  - Top 1/ 5 accuracy
    - Rate of questions whose answers are included in top 1/5 candidates
  - MRR (Mean Reciprocal Rank)

$$MRR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{rank(ans_i)}$$

N number of the testing questions

$rank(ans_i)$ rank of the correct answer for i^{th} question

Experiments: Baselines

- Baselines
  - Baseline1: Tree Kernel(Severyn and Moschitti, 2013)
    - QA state-of-the-art answer extraction method on formal text
  - Baseline2: N-grams (Brill et.al, 2002)
    - Commonly used method for Web-QA

Experiments: Results

Test Set	Passage	Method	Top 1 Acc.	Top 5 Acc.	MRR
Test-1	Document	Tree Kernel	70.79	82.02	73.91
		Ours	69.66	79.78	72.12
	Search	Tree Kernel	52.00	78.67	58.17
		Ours	**66.67**	84.00	70.71
Test-2	Search	Tree Kernel	51.19	72.35	59.81
		N-grams	50.85	72.70	60.78
		Ours	**66.55**	79.52	69.93

- Our method performs better than previous methods on search snippets
- Same method drops on performance when transfers to search snippets

Experiment: Analysis

- Pruned Rank Aggregation

	Combine (Single Prune)		Combine (Joint Prune)	
	Recall	N:P	Recall	N:P
No Pruning	94.54	59	94.54	59
With Pruning	85.67	29	**93.85**	31

•N:P, number of negative candidates with respect to positive ones
•Single prune: perform pruning without considering information from other extraction engines
•Joint Prune: Pruned Rank Aggregation method in this work

•Combination improves answer recall
•Pruning hurts answer recall
•Jointed prune helps to reduce the harm of pruning on recall

Experiments: Results

- Examples of different engines' results

Question	When did Jack Welch retire from GE?
Answer	2001
Different Methods	Top 5 Candidates
Tree kernel	general electric; chairman and ceo; his younger wife; oct 05, 2012;2001
NP	general electric; jack welch 's; 2001; one; chairman and ceo
CRF	2012; 2002; 2001; one; 1999
Wiki	general electric; retirement; 2012; 2001; ceo
Ours	2001;general electric; 2012; retirement; ceo

Experiment: Analysis

- Contributions of Different Feature Sets
  - Remove one feature set a time and test the performance

Feature Set	Top 1 Acc.	Top 5 Acc.	MRR
All	66.55	79.52	69.93
-Similarity	54.27	74.06	62.05
-Consensus	56.31	75.43	63.97
-Redundancy	57.34	76.79	64.74
-Quality	63.83	77.47	68.46
-Search	64.51	77.82	68.51

Contents

Conclusion

- Web-based Question Answering
  - High efficiency, wide coverage
  - Search snippets contain lots of noises and incomplete sentences
- Answer Extraction with Multiple Engines
  - Reduce negative impacts from search snippets
  - Leverage the consensus information for pruning and ranking

Future Work

- Extend the consensus information by candidates similarities

- Adopt more sophisticated similarity measures between question and search texts

Thank You