Query Reformulation & Answer Extraction

Ling 573
NLP Systems and Applications
May 9, 2013

Comparing Question Reformulations

- "Exact Phrases in Information Retrieval for Question Answering", Stoyanchev et al, 2008
- Investigates
  - Role of 'exact phrases' in retrieval for QA
  - Optimal query construction through document retrieval
    - From Web or AQUAINT collection
  - Impact of query specificity on passage retrieval

Motivation

- Retrieval bottleneck in Question-Answering
  - Retrieval provides source for answer extraction
  - If retrieval fails to return answer-contained documents, downstream answer processing is guaranteed to fail
  - Focus on recall in information retrieval phase
  - Consistent relationship b/t quality of IR and of QA

Motivation

- Retrieval bottleneck in Question-Answering
  - Retrieval provides source for answer extraction
  - If retrieval fails to return answer-contained documents, downstream answer processing is guaranteed to fail
  - Focus on recall in information retrieval phase
  - Consistent relationship b/t quality of IR and of QA
- Main factor in retrieval: query
  - Approaches vary from simple to complex processing or expansion with external resources

Approach

- Focus on use of 'exact phrases' from a question

Approach

- Focus on use of ‘exact phrases’ from a question
- Analyze impact of diff’t linguistic components of Q
  - Relate to answer candidate sentences

Approach

- Focus on use of 'exact phrases' from a question
- Analyze impact of diff't linguistic components of Q
  - Relate to answer candidate sentences
- Evaluate query construction for Web, Trec retrieval
  - Optimize query construction

Approach

- Focus on use of ‘exact phrases’ from a question
- Analyze impact of diff’t linguistic components of Q
  - Relate to answer candidate sentences
- Evaluate query construction for Web, Trec retrieval
  - Optimize query construction
- Evaluate query construction for sentence retrieval
  - Analyze specificity

Data Sources & Resources

- Documents:
  - TREC QA AQUAINT corpus
  - Web

Data Sources & Resources

- Documents:
  - TREC QA AQUAINT corpus
  - Web
- Questions:
  - TREC2006, non-empty questions

Data Sources & Resources

- Documents:
  - TREC QA AQUAINT corpus
  - Web
- Questions:
  - TREC2006, non-empty questions
- Gold standard:
  - NIST-provided relevant docs, answer key: 3.5 docs/Q

Data Sources & Resources

- Documents:
  - TREC QA AQUAINT corpus
  - Web
- Questions:
  - TREC2006, non-empty questions
- Gold standard:
  - NIST-provided relevant docs, answer key: 3.5 docs/Q
- Resources:
  - IR: Lucene; NLTK, Lingpipe: phrase, NE annotation
    - Also hand-corrected

Query Processing Approach

- Exploit less-resource intensive methods
  - Chunking, NER
  - Applied only to questions, candidate sentences
    - Not applied to full collection

Query Processing Approach

- Exploit less-resource intensive methods
  - Chunking, NER
  - Applied only to questions, candidate sentences
    - Not applied to full collection → can use on Web
- Exact phrase motivation:
  - Phrases can improve retrieval
    - "In what year did the movie win academy awards?"

Query Processing Approach

- Exploit less-resource intensive methods
  - Chunking, NER
  - Applied only to questions, candidate sentences
    - Not applied to full collection → can use on Web
- Exact phrase motivation:
  - Phrases can improve retrieval
    - "In what year did the movie win **academy awards**?"
    - Phrase:

Query Processing Approach

- Exploit less-resource intensive methods
  - Chunking, NER
  - Applied only to questions, candidate sentences
    - Not applied to full collection → can use on Web
- Exact phrase motivation:
  - Phrases can improve retrieval
    - "In what year did the movie win **academy awards**?"
    - Phrase: Can rank documents higher
    - Disjunct:

Query Processing Approach

- Exploit less-resource intensive methods
  - Chunking, NER
  - Applied only to questions, candidate sentences
    - Not applied to full collection → can use on Web
- Exact phrase motivation:
  - Phrases can improve retrieval
    - "In what year did the movie win **academy awards**?"
    - Phrase: Can rank documents higher
    - Disjunct: Can dilute pool

Query Processing

- NER on Question and target
  - target: 1991 eruption on Mt. Pinatubo vs Nirvana

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**
  - Uses LingPipe: ORG, LOC, PER
- Phrases (NLTK)
  - NP, VP, PP

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**
  - Uses LingPipe: ORG, LOC, PER
- Phrases (NLTK)
  - NP, VP, PP
- Converted Q-phrases:
  - Heuristic paraphrases on question as declarative
    - E.g. Who was|is NOUN|PRONOUN VBD → NOUN|PRONOUN was|is VBD

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**
  - Uses LingPipe: ORG, LOC, PER
- Phrases (NLTK)
  - NP, VP, PP
- Converted Q-phrases:
  - Heuristic paraphrases on question as declarative
    - E.g. Who was|is NOUN|PRONOUN VBD ➔ NOUN|PRONOUN was|is VBD
  - q-phrase: expected form of simple answer
    - E.g. When was Mozart born? ➔ Mozart was born
  - How likely are we to see a q-phrase?

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**
  - Uses LingPipe: ORG, LOC, PER
- Phrases (NLTK)
  - NP, VP, PP
- Converted Q-phrases:
  - Heuristic paraphrases on question as declarative
    - E.g. Who was|is NOUN|PRONOUN VBD → NOUN|PRONOUN was|is VBD
  - q-phrase: expected form of simple answer
    - E.g. When was Mozart born? → Mozart was born
  - How likely are we to see a q-phrase? Unlikely
  - How likely is it to be right if we do see it?

Query Processing

- NER on Question and target
  - target: 1991 eruption on **Mt. Pinatubo** vs **Nirvana**
  - Uses LingPipe: ORG, LOC, PER
- Phrases (NLTK)
  - NP, VP, PP
- Converted Q-phrases:
  - Heuristic paraphrases on question as declarative
    - E.g. Who was|is NOUN|PRONOUN VBD ➔ NOUN|PRONOUN was|is VBD
  - q-phrase: expected form of simple answer
    - E.g. When was Mozart born? ➔ Mozart was born
  - How likely are we to see a q-phrase? Unlikely
  - How likely is it to be right if we do see it? Very

Comparing Query Forms

- Baseline: words from question and target

Comparing Query Forms

- Baseline: words from question and target
- Experimental:
  - Words, quoted exact phrases, quoted names entities
  - Backoff: Lucene: weight based on type

Comparing Query Forms

- Baseline: words from question and target
- Experimental:
  - Words, quoted exact phrases, quoted names entities
  - Backoff: Lucene: weight based on type
  - Backoff: Web: 1) converted q-phrases;
    - 2) phrases; 3) w/o phrases -- until 20 retrieved
    - Combined with target in all cases

Comparing Query Forms

- Baseline: words from question and target
- Experimental:
  - Words, quoted exact phrases, quoted names entities
  - Backoff: Lucene: weight based on type
  - Backoff: Web: 1) converted q-phrases;
    - 2) phrases; 3) w/o phrases -- until 20 retrieved
    - Combined with target in all cases
- Max 20 documents: expensive downstream process
  - Sentences split, ranked

Query Components

Target Question	United Nations What was the number of member nations of the U.N. in 2000?
Named Entity	U.N., United Nations
Phrases	"member nations of the U.N."
Converted Q-phrase	"member nations of the U.N. in 2000"
Baseline Query	was the number of member nations of the U.N. in 2000 United Nations
Lucene Query with phrases and NE	was the number of member nations of the U.N. in 2000 "United Nations", "member nations of the u.n."
Cascaded web query	
query1	"member nations of the U.N. in 2000" AND (United Nations)
query2	"member nations of the u.n." AND (United Nations)
query3	(number of member nations of the U.N. in 2000) AND (United Nations)
query4	(United Nations)

Query Components in Supporting Sentences

	sent w/ answer		all sentences		precision
	num	proportion	num	proportion	
Named Entity	907	0.320	4873	0.122	.18
Phrases	350	0.123	1072	0.027	.34
Verbs	396	0.140	1399	0.035	.28
Q-Phrases	11	0.004	15	0.00038	.73
Words	2573	0.907	29576	0.745	.086
Total Sentences	2836		39688		

Highest precision:

Query Components in Supporting Sentences

	sent w/ answer		all sentences		precision
	num	proportion	num	proportion	
Named Entity	907	0.320	4873	0.122	.18
Phrases	350	0.123	1072	0.027	.34
Verbs	396	0.140	1399	0.035	.28
Q-Phrases	11	0.004	15	0.00038	.73
Words	2573	0.907	29576	0.745	.086
Total Sentences	2836		39688		

Highest precision: Converted q-phrase, then phrase,..

Query Components in Supporting Sentences

	sent w/ answer		all sentences		precision
	num	proportion	num	proportion	
Named Entity	907	0.320	4873	0.122	.18
Phrases	350	0.123	1072	0.027	.34
Verbs	396	0.140	1399	0.035	.28
Q-Phrases	11	0.004	15	0.00038	.73
Words	2573	0.907	29576	0.745	.086
Total Sentences	2836		39688		

Highest precision: Converted q-phrase, then phrase,..
Words likely to appear, but don't discriminate

Results

- Document and sentence retrieval
- Metrics:
  - Document retrieval:
    - Average recall
    - MRR
    - Overall document recall: % of questions w/>=1 correct doc
  - Sentence retrieval
    - Sentence MRR
    - Overall sentence recall
    - Average prec of first sentence
    - # correct in top 10, top 50

Results

- Document and sentence retrieval
- Metrics:
  - Document retrieval:
    - Average recall
    - MRR
    - Overall document recall: % of questions w/>=1 correct doc

Results

	avg doc recall	avg doc MRR	overall doc recall	avg sent MRR	overall sent recall	avg corr sent in top 1	avg corr sent in top 10	avg corr sent in top 50
IR with Lucene on AQUAINT dataset								
baseline (words disjunction from target and question)	0.530	0.631	0.756	0.314	0.627	0.223	1.202	3.464
baseline + auto phrases	0.514	0.617	0.741	0.332	0.653	0.236	1.269	3.759
words + auto NEs & phrases	0.501	0.604	0.736	0.316	0.653	0.220	1.228	3.705
baseline + manual phrases	0.506	0.621	0.738	0.291	0.609	0.199	1.231	3.378
words + manual NEs & phrases	0.510	0.625	0.738	0.294	0.609	0.202	1.244	3.368
IR with Yahoo API on WEB								
baseline words disjunction	-	-	-	0.183	0.570	0.101	0.821	2.316
cascaded using auto phrases	-	-	-	0.220	0.604	0.140	0.956	2.725
cascaded using manual phrases	-	-	-	0.241	0.614	0.155	1.065	3.016

Discussion

- Document retrieval:
  - About half of the correct docs are retrieved, rank 1-2
- Sentence retrieval:
  - Lower, correct sentence ~ rank 3

Discussion

- Document retrieval:
  - About half of the correct docs are retrieved, rank 1-2
- Sentence retrieval:
  - Lower, correct sentence ~ rank 3
- Little difference for exact phrases in AQUAINT

Discussion

- Document retrieval:
  - About half of the correct docs are retrieved, rank 1-2
- Sentence retrieval:
  - Lower, correct sentence ~ rank 3
- Little difference for exact phrases in AQUAINT
- Web:
  - Retrieval improved by exact phrases
    - Manual more than auto (20-30%) relative
  - Precision affected by tagging errors

Answer Extraction

- Goal:
  - Given a passage, find the specific answer in passage
  - Go from ~1000 chars -> short answer span

Answer Extraction

- Goal:
  - Given a passage, find the specific answer in passage
  - Go from ~1000 chars -> short answer span
- Example:
  - Q: What is the current population of the United States?
  - Pass: The United States enters 2011 with a population of more than 310.5 million people, according to a U.S. Census Bureau estimate.

Answer Extraction

- Goal:
  - Given a passage, find the specific answer in passage
  - Go from ~1000 chars -> short answer span

- Example:
  - Q: What is the current population of the United States?
  - Pass: The United States enters 2011 with a population of more than 310.5 million people, according to a U.S. Census Bureau estimate.
  - Answer: 310.5 million

Challenges

- ISI’s answer extraction experiment:
  - Given:
    - Question: 413 TREC-2002 factoid questions
    - Known answer type
    - All correct answer passages

Challenges

- ISI's answer extraction experiment:
  - Given:
    - Question: 413 TREC-2002 factoid questions
    - Known answer type
    - All correct answer passages
  - Task: Pin-point specific answer string
  - Accuracy:
    - Systems: 68.2%, 63.4%, 56.7%
      - Still missing 30%+ answers

Challenges

- ISI's answer extraction experiment:
  - Given:
    - Question: 413 TREC-2002 factoid questions
    - Known answer type
    - All correct answer passages
  - Task: Pin-point specific answer string

  - Accuracy:
    - Systems: 68.2%, 63.4%, 56.7%
      - Still missing 30%+ answers
    - Oracle (any of 3 right): 78.9% (20% miss)

Basic Strategies

- Answer-type matching:
  - Build patterns for answer locations
    - Restrict by answer type

Basic Strategies

- Answer-type matching:
  - Build patterns for answer locations
    - Restrict by answer type
  - Information for pattern types:

Basic Strategies

- Answer-type matching:
  - Build patterns for answer locations
    - Restrict by answer type

  - Information for pattern types:
    - Lexical: word patterns
    - Syntactic/structural:
      - Syntactic relations b/t question and answer
    - Semantic:
      - Semantic/argument relations b/t question and answer

Basic Strategies

- Answer-type matching:
  - Build patterns for answer locations
    - Restrict by answer type
  - Information for pattern types:
    - Lexical: word patterns
    - Syntactic/structural:
      - Syntactic relations b/t question and answer
    - Semantic:
      - Semantic/argument relations b/t question and answer
  - Combine with machine learning to select

Pattern Matching Example

- Answer type: Definition

Pattern Matching Example

- Answer type: Definition

Pattern	Question	Answer
<AP> such as <QP>	What is autism?	", developmental disorders such as autism"
<QP>, a <AP>	What is a caldera?	"the Long Valley caldera, a volcanic crater 19 miles long"

Pattern Matching Example

- Answer type: Definition

Pattern	Question	Answer
<AP> such as <QP>	What is autism?	", developmental disorders such as autism"
<QP>, a <AP>	What is a caldera?	"the Long Valley caldera, a volcanic crater 19 miles long"

- Answer type: Birthdate

Pattern Matching Example

- Answer type: Definition

Pattern	Question	Answer
<AP> such as <QP>	What is autism?	“, developmental disorders such as autism”
<QP>, a <AP>	What is a caldera?	“the Long Valley caldera, a volcanic crater 19 miles long”

- Answer type: Birthdate
  - Question: When was Mozart born?
  - Answer: Mozart was born on ….

Pattern Matching Example

- Answer type: Definition

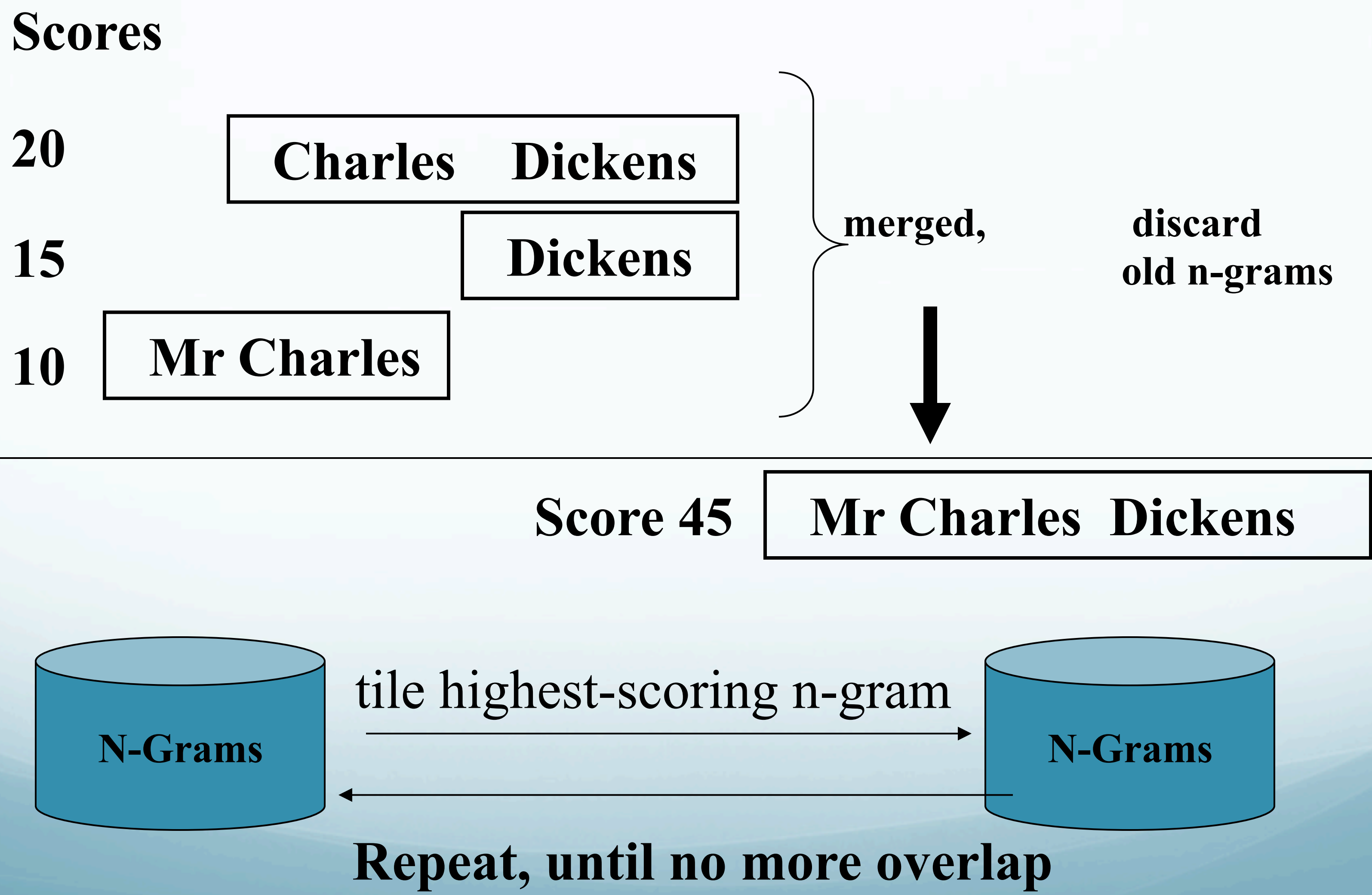
Pattern	Question	Answer
<AP> such as <QP>	What is autism?	", developmental disorders such as autism"
<QP>, a <AP>	What is a caldera?	"the Long Valley caldera, a volcanic crater 19 miles long"

- Answer type: Birthdate
  - Question: When was Mozart born?
  - Answer: Mozart was born on ....
  - Pattern: <QP> was born on <AP>
  - Pattern: <QP> (<AP> - .....)

Basic Strategies

- N-gram tiling:
  - Typically as part of answer validation/verification
  - Integrated with web-based retrieval
  - Based on retrieval of search 'snippets'
  - Identifies frequently occurring, overlapping n-grams
    - Of correct type

N-gram Tiling



Automatic Pattern Learning

- Ravichandran and Hovy 2002; Echihabi et al, 2005

- Inspiration (Soubottin and Soubottin '01)
  - Best TREC 2001 system:

Automatic Pattern Learning

- Ravichandran and Hovy 2002; Echihabi et al, 2005

- Inspiration (Soubottin and Soubottin ’01)
  - Best TREC 2001 system:
    - Based on extensive list of surface patterns
      - Mostly manually created

Automatic Pattern Learning

- Ravichandran and Hovy 2002; Echihabi et al, 2005

- Inspiration (Soubottin and Soubottin ’01)
  - Best TREC 2001 system:
    - Based on extensive list of surface patterns
      - Mostly manually created

  - Many patterns strongly associated with answer types
    - E.g. <NAME> (<DATE>-<DATE>)
      - Person’s birth and death

Pattern Learning

- S & S ‘01 worked well, but

Pattern Learning

- S & S '01 worked well, but
  - Manual pattern creation is a hassle, impractical

Pattern Learning

- S & S ‘01 worked well, but
  - Manual pattern creation is a hassle, impractical
- Can we learn patterns?
  - Supervised approaches:

Pattern Learning

- S & S ‘01 worked well, but
  - Manual pattern creation is a hassle, impractical

- Can we learn patterns?
  - Supervised approaches:
    - Not much better,
      - Have to tag training samples, need training samples

Pattern Learning

- S & S ‘01 worked well, but
  - Manual pattern creation is a hassle, impractical

- Can we learn patterns?
  - Supervised approaches:
    - Not much better,
      - Have to tag training samples, need training samples
  - Bootstrapping approaches:
    - Promising:

Pattern Learning

- S & S '01 worked well, but
  - Manual pattern creation is a hassle, impractical

- Can we learn patterns?
  - Supervised approaches:
    - Not much better,
      - Have to tag training samples, need training samples
  - Bootstrapping approaches:
    - Promising:
      - Guidance from small number of seed samples
      - Can use answer data from web

Finding Candidate Patterns

- For a given question type
  - Identify an example with qterm and aterm

Finding Candidate Patterns

- For a given question type
  - Identify an example with qterm and aterm
  - Submit to a search engine
  - Download top N web docs (N=1000)
  - Select only sentences w/qterm and aterm

Finding Candidate Patterns

- For a given question type
  - Identify an example with qterm and aterm
  - Submit to a search engine
  - Download top N web docs (N=1000)
  - Select only sentences w/qterm and aterm
  - Identify all substrings and their counts
    - Implemented using suffix trees for efficiency
  - Select only phrases with qterm AND aterm
  - Replace qterm and aterm instances w/generics

Example

- Q: When was Mozart born?
- A: Mozart (1756-....

Example

- Q: When was Mozart born?
- A: Mozart (1756 –
- Qterm: Mozart; Aterm: 1756
  - The great composer Mozart (1756–1791) achieved fame
  - Mozart (1756–1791) was a genius
  - Indebted to the great music of Mozart (1756–1791)

Example

- Q: When was Mozart born?
- A: Mozart (1756 –
- Qterm: Mozart; Aterm: 1756
  - The great composer Mozart (1756–1791) achieved fame
  - Mozart (1756–1791) was a genius
  - Indebted to the great music of Mozart (1756–1791)
- Phrase: Mozart (1756-1791); count =3

Example

- Q: When was Mozart born?
- A: Mozart (1756 –
- Qterm: Mozart; Aterm: 1756
  - The great composer Mozart (1756–1791) achieved fame
  - Mozart (1756–1791) was a genius
  - Indebted to the great music of Mozart (1756–1791)
- Phrase: Mozart (1756-1791); count =3
- Convert to : <Name> (<ANSWER>

Patterns

- Typically repeat with a few more examples

Patterns

- Typically repeat with a few more examples
- Collect more patterns:
  - E.g. for Birthdate
    - a. born in <ANSWER> , <NAME>
    - b. <NAME> was born on <ANSWER> ,
    - c. <NAME> (<ANSWER> -
    - d. <NAME> (<ANSWER> -)
- Is this enough?

Patterns

- Typically repeat with a few more examples
- Collect more patterns:
  - E.g. for Birthdate
    - a. born in <ANSWER> , <NAME>
    - b. <NAME> was born on <ANSWER> ,
    - c. <NAME> (<ANSWER> -
    - d. <NAME> (<ANSWER> -)
- Is this enough?
  - No – some good patterns, but
    - Probably lots of junk, too; need to filter

Computing Pattern Precision

- For question type:
  - Search only on qterm

Computing Pattern Precision

- For question type:
  - Search only on qterm
  - Download top N web docs (N=1000)
  - Select only sentences w/qterm

Computing Pattern Precision

- For question type:
  - Search only on qterm
  - Download top N web docs (N=1000)
  - Select only sentences w/qterm
  - For each pattern, check if
    - a) matches w/any aterm; C_o
    - b)matches/w right aterm: C_a

Computing Pattern Precision

- For question type:
  - Search only on qterm
  - Download top N web docs (N=1000)
  - Select only sentences w/qterm
  - For each pattern, check if
    - a) matches w/any aterm; C_o
    - b)matches/w right aterm: C_a
  - Compute precision $P = C_a/C_o$
  - Retain if match > 5 examples

Pattern Precision Example

- Qterm: Mozart
- Pattern: <NAME> was born in <ANSWER>

Pattern Precision Example

- Qterm: Mozart
- Pattern: <NAME> was born in <ANSWER>
- Near-Miss: Mozart was born in Salzburg
- Match: Mozart born in 1756.

Pattern Precision Example

- Qterm: Mozart
- Pattern: <NAME> was born in <ANSWER>
- Near-Miss: Mozart was born in Salzburg
- Match: Mozart born in 1756.
- Precisions:
  - 1.0 <NAME> (<ANSWER> -)
  - 0.6 <NAME> was born in <ANSWER>
  - ....

Nuances

- Alternative forms:
  - Need to allow for alternate forms of question or answer
    - E.g. dates in different formats, full names, etc
  - Use alternate forms in pattern search

Nuances

- Alternative forms:
  - Need to allow for alternate forms of question or answer
    - E.g. dates in different formats, full names, etc
  - Use alternate forms in pattern search
- Precision assessment:
  - Use other examples of same type to compute
  - Cross-checks patterns

Answer Selection by Pattern

- Identify question types and terms
- Filter retrieved passages, replace qterm by tag
- Try to match patterns and answer spans
- Discard duplicates and sort by pattern precision

Pattern Sets

- WHY-FAMOUS

  1.0 <ANSWER> <NAME> called

  1.0 laureate <ANSWER> <NAME>

  1.0 by the <ANSWER> , <NAME> ,
  1.0 <NAME> - the <ANSWER> of

  1.0 <NAME> was the <ANSWER> of

- BIRTHYEAR

  1.0 <NAME> (<ANSWER> -)

  0.85 <NAME> was born on <ANSWER> ,

  0.6 <NAME> was born in <ANSWER>

  0.59 <NAME> was born <ANSWER>

  0.53 <ANSWER> <NAME> was born

Results

- Improves, though better with web data

TREC Corpus

Question type	Number of questions	MRR on TREC docs
BIRTHYEAR	8	0.48
INVENTOR	6	0.17
DISCOVERER	4	0.13
DEFINITION	102	0.34
WHY-FAMOUS	3	0.33
LOCATION	16	0.75

Web

Question type	Number of questions	MRR on the Web
BIRTHYEAR	8	0.69
INVENTOR	6	0.58
DISCOVERER	4	0.88
DEFINITION	102	0.39
WHY-FAMOUS	3	0.00
LOCATION	16	0.86

Limitations & Extensions

- Where are the Rockies?
- ..with the Rockies in **the background**

Limitations & Extensions

- Should restrict to semantic / NE type
  - Where are the Rockies?
  - ..with the Rockies in **the background**

Limitations & Extensions

- Where are the Rockies?
- ..with the Rockies in **the background**

- Should restrict to semantic / NE type
  - London, which…., lies on the River Thames
  - <QTERM> word* lies on <ANSWER>
    - Wildcards impractical

Limitations & Extensions

- Where are the Rockies?
- ..with the Rockies in **the background**

- Should restrict to semantic / NE type
  - London, which…., lies on the River Thames
  - <QTERM> word* lies on <ANSWER>
    - Wildcards impractical

- Long-distance dependencies not practical

Limitations & Extensions

- Where are the Rockies?
- ..with the Rockies in **the background**

- Should restrict to semantic / NE type
  - London, which…., lies on the River Thames
  - <QTERM> word* lies on <ANSWER>
    - Wildcards impractical

- Long-distance dependencies not practical
  - Less of an issue in Web search
    - Web highly redundant, many local dependencies
    - Many systems (LCC) use web to **validate** answers

Limitations & Extensions

- When was LBJ born?
- Tower lost to Sen. LBJ, *who ran for both the*...

Limitations & Extensions

- When was LBJ born?
- Tower lost to Sen. LBJ, *who ran for both the*…

- Requires information about:
  - Answer length, type; logical distance (1-2 chunks)

Limitations & Extensions

- When was LBJ born?
- Tower lost to Sen. LBJ, *who ran for both the*…

- Requires information about:
  - Answer length, type; logical distance (1-2 chunks)

- Also,
  - Can only handle single continuous qterms
  - Ignores case
  - Needs handle canonicalization, e.g of names/dates

Integrating Patterns II

- Fundamental problem:

Integrating Patterns II

- Fundamental problem:
  - What if there's no pattern??

Integrating Patterns II

- Fundamental problem:
  - What if there's no pattern??
    - No pattern -> No answer!!!
- More robust solution:
  - Not JUST patterns

Integrating Patterns II

- Fundamental problem:
  - What if there's no pattern??
    - No pattern -> No answer!!!
- More robust solution:
  - Not JUST patterns
  - Integrate with machine learning
    - MAXENT!!!
    - Re-ranking approach

Answering w/Maxent

$$P(a \mid \{a_1, a_2, \ldots a_A\}, q) = \frac{\exp[\sum_{m=1}^{M} \lambda_m f_m(a, \{a_1, a_2, \ldots a_A\}, q)]}{\sum_{a'} \exp[\sum_{m=1}^{M} \lambda_m f_m(a', \{a_1, a_2, \ldots a_A\}, q)]}$$

$$\widehat{a} = \underset{a}{\operatorname{argmax}}[\sum_{m=1}^{M} \lambda_m f_m(a, \{a_1, a_2, \ldots a_A\}, q)]$$

Feature Functions

- Pattern fired:
  - Binary feature

Feature Functions

- Pattern fired:
  - Binary feature
- Answer frequency/Redundancy factor:
  - # times answer appears in retrieval results

Feature Functions

- Pattern fired:
  - Binary feature
- Answer frequency/Redundancy factor:
  - # times answer appears in retrieval results
- Answer type match (binary)

Feature Functions

- Pattern fired:
  - Binary feature
- Answer frequency/Redundancy factor:
  - # times answer appears in retrieval results
- Answer type match (binary)
- Question word absent (binary):
  - No question words in answer span

Feature Functions

- Pattern fired:
  - Binary feature
- Answer frequency/Redundancy factor:
  - # times answer appears in retrieval results
- Answer type match (binary)
- Question word absent (binary):
  - No question words in answer span
- Word match:
  - Sum of ITF of words matching b/t questions & sent

Training & Testing

- Trained on NIST QA questions
  - Train: TREC 8,9;
  - Cross-validation: TREC-10
- 5000 candidate answers/question
- Positive examples:
  - NIST pattern matches
- Negative examples:
  - NIST pattern doesn't match
- Test: TREC-2003: MRR: 28.6%; 35.6% exact top 5