Answer Extraction

Ling573
NLP Systems and Applications
May 19, 2011

Roadmap

- Noisy-channel Question-Answering
- Answer selection by reranking
- Redundancy-based Answer Selection

Noisy Channel QA

- Employed for speech, POS tagging, MT, summ, etc
- Intuition:
  - Question is a noisy representation of the answer

Noisy Channel QA

- Employed for speech, POS tagging, MT, summ, etc
- Intuition:
  - Question is a noisy representation of the answer
- Basic approach:
  - Given a corpus of (Q,S_A) pairs
  - Train $P(Q|S_A)$
  - Find sentence with answer as
    - $S_{i,Aij}$ that maximize $P(Q|S_{i,Aij})$

QA Noisy Channel

- A: Presley died of heart disease at Graceland in 1977, and..
- Q: When did Elvis Presley die?

QA Noisy Channel

- A: Presley died of heart disease at Graceland in 1977, and..
- Q: When did Elvis Presley die?

- Goal:
  - Align parts of Ans parse tree to question
    - Mark candidate answers
    - Find highest probability answer

Approach

- Alignment issue:

Approach

- Alignment issue:
  - Answer sentences longer than questions
  - Minimize length gap
    - Represent answer as mix of words/syn/sem/NE units

Approach

- Alignment issue:
  - Answer sentences longer than questions
  - Minimize length gap
    - Represent answer as mix of words/syn/sem/NE units
  - Create ‘cut’ through parse tree
    - Every word –or an ancestor – in cut
    - Only one element on path from root to word

Approach

- Alignment issue:
  - Answer sentences longer than questions
  - Minimize length gap
    - Represent answer as mix of words/syn/sem/NE units
  - Create 'cut' through parse tree
    - Every word –or an ancestor – in cut
    - Only one element on path from root to word

Presley died of heart disease at Graceland in 1977, and..
Presley died PP PP in DATE, and..
When did Elvis Presley die?

Approach (Cont'd)

- Assign one element in cut to be 'Answer'
- Issue: Cut STILL may not be same length as Q

Approach (Cont'd)

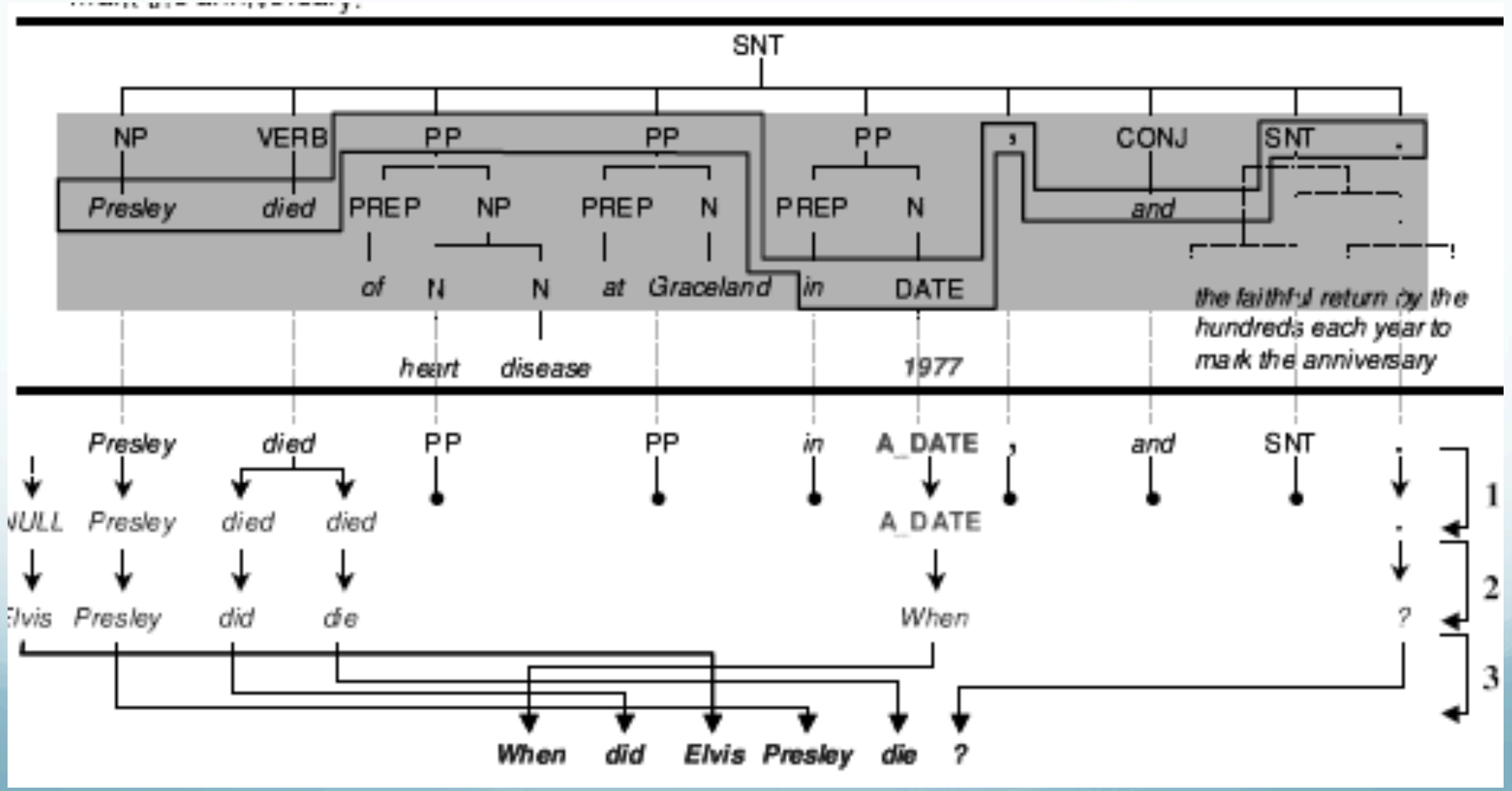
- Assign one element in cut to be 'Answer'
- Issue: Cut STILL may not be same length as Q
- Solution: (typical MT)
  - Assign each element a fertility
    - 0 – delete the word; > 1: repeat word that many times

Approach (Cont'd)

- Assign one element in cut to be 'Answer'
- Issue: Cut STILL may not be same length as Q
- Solution: (typical MT)
  - Assign each element a fertility
    - 0 – delete the word; > 1: repeat word that many times
- Replace A words with Q words based on alignment
- Permute result to match original Question
- Everything except cut computed with OTS MT code

Schematic

- Assume cut, answer guess all equally likely



Training Sample Generation

- Given question and answer sentences
- Parse answer sentence
- Create cut s.t.:
  - Words in both Q & A are preserved
  - Answer reduced to 'A_' syn/sem class label
  - Nodes with no surface children reduced to syn class
  - Keep surface form of all other nodes
- 20K TREC QA pairs; 6.5K web question pairs

Selecting Answers

- For any candidate answer sentence:
  - Do same cut process

Selecting Answers

- For any candidate answer sentence:
  - Do same cut process
  - Generate all candidate answer nodes:
    - Syntactic/Semantic nodes in tree

Selecting Answers

- For any candidate answer sentence:
  - Do same cut process
  - Generate all candidate answer nodes:
    - Syntactic/Semantic nodes in tree
  - What's a bad candidate answer?

Selecting Answers

- For any candidate answer sentence:
  - Do same cut process
  - Generate all candidate answer nodes:
    - Syntactic/Semantic nodes in tree
  - What's a bad candidate answer?
    - Stopwords
    - Question words!
  - Create cuts with each answer candidate annotated
  - Select one with highest probability by model

Example Answer Cuts

- Q: When did Elvis Presley die?
- S_{A1}: Presley died A_PP PP PP, and ...
- S_{A2}: Presley died PP A_PP PP, and ....
- S_{A3}: Presley died PP PP in A_DATE, and ...

- Results: MRR: 24.8%; 31.2% in top 5

Error Analysis

- Component specific errors:
  - Patterns:
    - Some question types work better with patterns
    - Typically specific NE categories (NAM, LOC, ORG..)
    - Bad if 'vague'
  - Stats based:
    - No restrictions on answer type – frequently 'it'
  - Patterns and stats:
    - 'Blatant' errors:
      - Select 'bad' strings (esp. pronouns) if fit position/pattern

Error Analysis

- Component specific errors:
  - Patterns:
    - Some question types work better with patterns
    - Typically specific NE categories (NAM, LOC, ORG..)
    - Bad if ‘vague’

Error Analysis

- Component specific errors:
  - Patterns:
    - Some question types work better with patterns
    - Typically specific NE categories (NAM, LOC, ORG..)
    - Bad if ‘vague’
  - Stats based:
    - No restrictions on answer type – frequently ‘it’

Error Analysis

- Component specific errors:
  - Patterns:
    - Some question types work better with patterns
    - Typically specific NE categories (NAM, LOC, ORG..)
    - Bad if 'vague'
  - Stats based:
    - No restrictions on answer type – frequently 'it'
  - Patterns and stats:
    - 'Blatant' errors:
      - Select 'bad' strings (esp. pronouns) if fit position/pattern

Combining Units

- Linear sum of weights?

Combining Units

- Linear sum of weights?
  - Problematic:
    - Misses different strengths/weaknesses

Combining Units

- Linear sum of weights?
  - Problematic:
    - Misses different strengths/weaknesses
- Learning! (of course)
  - Maxent re-ranking
    - Linear

Feature Functions

- 48 in total
- Component-specific:
  - Scores, ranks from different modules
    - Patterns. Stats, IR, even QA word overlap

Feature Functions

- 48 in total
- Component-specific:
  - Scores, ranks from different modules
    - Patterns. Stats, IR, even QA word overlap
- Redundancy-specific:
  - # times candidate answer appears (log, sqrt)

Feature Functions

- 48 in total
- Component-specific:
  - Scores, ranks from different modules
    - Patterns. Stats, IR, even QA word overlap
- Redundancy-specific:
  - # times candidate answer appears (log, sqrt)
- Qtype-specific:
  - Some components better for certain types: type+mod

Feature Functions

- 48 in total
- Component-specific:
  - Scores, ranks from different modules
    - Patterns. Stats, IR, even QA word overlap
- Redundancy-specific:
  - # times candidate answer appears (log, sqrt)
- Qtype-specific:
  - Some components better for certain types: type+mod
- Blatant ‘errors’: no pronouns, when NOT DoW

Experiments

- Per-module reranking:
  - Use redundancy, qtype, blatant, and feature from mod

Experiments

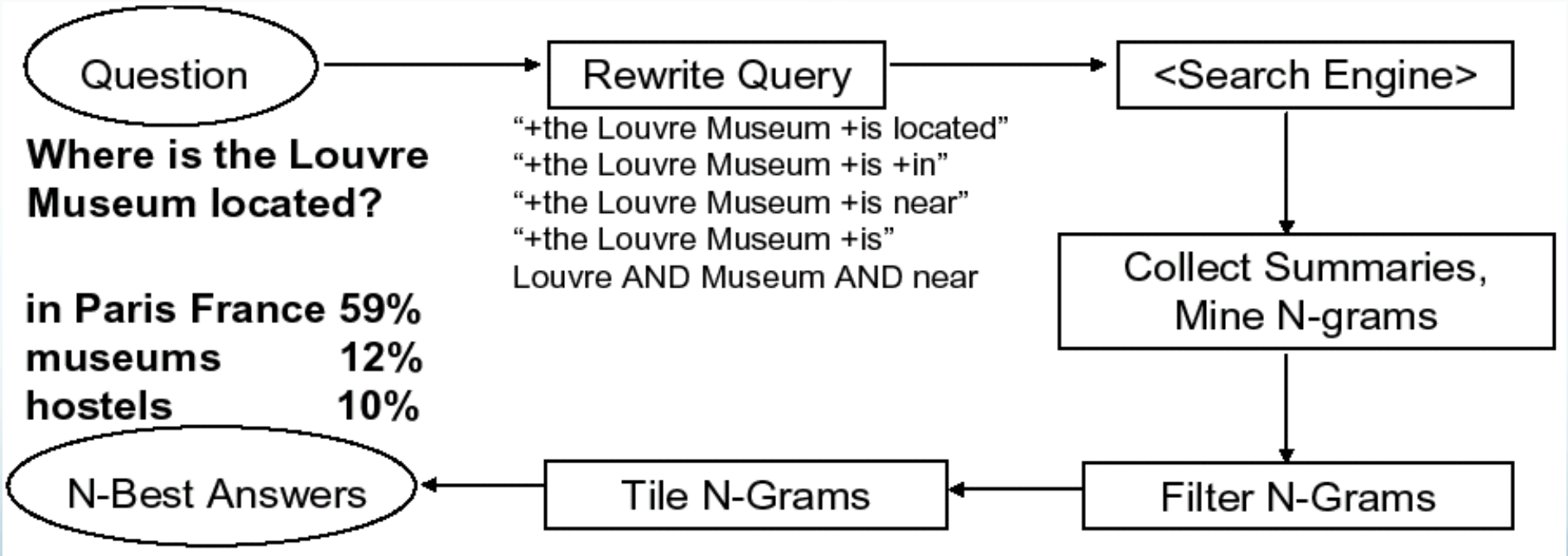
- Per-module reranking:
  - Use redundancy, qtype, blatant, and feature from mod
- Combined reranking:
  - All features (after feature selection to 31)

Experiments

- Per-module reranking:
  - Use redundancy, qtype, blatant, and feature from mod
- Combined reranking:
  - All features (after feature selection to 31)
- Patterns: Exact in top 5: 35.6% -> 43.1%
- Stats: Exact in top 5: 31.2% -> 41%
- Manual/knowledge based: 57%

Redundancy-based QA

- AskMSR (2001,2002); Aranea (Lin, 2007)



Redundancy-based QA

- Systems exploit statistical regularity to find “easy” answers to factoid questions on the Web

Redundancy-based QA

- Systems exploit statistical regularity to find "easy" answers to factoid questions on the Web
  - —When did Alaska become a state?
  - **(1) Alaska became a state on January 3, 1959.**
  - **(2) Alaska was admitted to the Union on January 3, 1959.**

Redundancy-based QA

- Systems exploit statistical regularity to find "easy" answers to factoid questions on the Web
  - —When did Alaska become a state?
  - **(1) Alaska became a state on January 3, 1959.**
  - **(2) Alaska was admitted to the Union on January 3, 1959.**

  - —Who killed Abraham Lincoln?
  - **(1) John Wilkes Booth killed Abraham Lincoln.**
  - **(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.**

Redundancy-based QA

- Systems exploit statistical regularity to find "easy" answers to factoid questions on the Web
  - —When did Alaska become a state?
  - **(1) Alaska became a state on January 3, 1959.**
  - **(2) Alaska was admitted to the Union on January 3, 1959.**
  - —Who killed Abraham Lincoln?
  - **(1) John Wilkes Booth killed Abraham Lincoln.**
  - **(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.**
- Text collection

Redundancy-based QA

- Systems exploit statistical regularity to find "easy" answers to factoid questions on the Web
  - —When did Alaska become a state?
  - **(1) Alaska became a state on January 3, 1959.**
  - **(2) Alaska was admitted to the Union on January 3, 1959.**
  - —Who killed Abraham Lincoln?
  - **(1) John Wilkes Booth killed Abraham Lincoln.**
  - **(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.**
- Text collection may only have (2), but web? anything

Redundancy-based QA

- Systems exploit statistical regularity to find "easy" answers to factoid questions on the Web
  - —When did Alaska become a state?
  - **(1) Alaska became a state on January 3, 1959.**
  - **(2) Alaska was admitted to the Union on January 3, 1959.**

  - —Who killed Abraham Lincoln?
  - **(1) John Wilkes Booth killed Abraham Lincoln.**
  - **(2) John Wilkes Booth altered history with a bullet. He will forever be known as the man who ended Abraham Lincoln's life.**
- Text collection may only have (2), but web?

Redundancy & Answers

- How does redundancy help find answers?

Redundancy & Answers

- How does redundancy help find answers?
- Typical approach:
  - Answer type matching
    - E.g. NER, but
    - Relies on large knowledge-based
- Redundancy approach:

Redundancy & Answers

- How does redundancy help find answers?
- Typical approach:
  - Answer type matching
    - E.g. NER, but
    - Relies on large knowledge-based
- Redundancy approach:
  - Answer should have high correlation w/query terms
    - Present in many passages
      - Uses n-gram generation and processing

Redundancy & Answers

- How does redundancy help find answers?
- Typical approach:
  - Answer type matching
    - E.g. NER, but
    - Relies on large knowledge-based
- Redundancy approach:
  - Answer should have high correlation w/query terms
    - Present in many passages
      - Uses n-gram generation and processing
  - In 'easy' passages, simple string match effective

Redundancy Approaches

- AskMSR (2001):
  - Lenient: 0.43; Rank: 6/36; Strict: 0.35; Rank: 9/36

Redundancy Approaches

- AskMSR (2001):
  - Lenient: 0.43; Rank: 6/36; Strict: 0.35; Rank: 9/36
- Aranea (2002, 2003):
  - Lenient: 45%; Rank: 5; Strict: 30%; Rank:6-8

Redundancy Approaches

- AskMSR (2001):
  - Lenient: 0.43; Rank: 6/36; Strict: 0.35; Rank: 9/36
- Aranea (2002, 2003):
  - Lenient: 45%; Rank: 5; Strict: 30%; Rank:6-8
- Concordia (2007): Strict: 25%; Rank 5

Redundancy Approaches

- AskMSR (2001):
  - Lenient: 0.43; Rank: 6/36; Strict: 0.35; Rank: 9/36
- Aranea (2002, 2003):
  - Lenient: 45%; Rank: 5; Strict: 30%; Rank:6-8
- Concordia (2007): Strict: 25%; Rank 5
- Many systems incorporate some redundancy
  - Answer validation
  - Answer reranking
    - LCC: huge knowledge-based system, redundancy improved

Redundancy-based Answer Extraction

- Prior processing:
  - Question formulation (class 6)
  - Web search
  - Retrieve snippets – top 100

Redundancy-based Answer Extraction

- Prior processing:
  - Question formulation (class 6)
  - Web search
  - Retrieve snippets – top 100
- N-grams:
  - Generation
  - Voting
  - Filtering
  - Combining
  - Scoring
  - Reranking

N-gram Generation & Voting

- N-gram generation from unique snippets:
  - Approximate chunking – without syntax
  - All uni-, bi-, tri-, tetra- grams
    - Concordia added 5-grams (prior errors)

N-gram Generation & Voting

- N-gram generation from unique snippets:
  - Approximate chunking – without syntax
  - All uni-, bi-, tri-, tetra- grams
    - Concordia added 5-grams (prior errors)
  - Score: based on source query: exact 5x, others 1x
- N-gram voting:
  - Collates n-grams
  - N-gram gets sum of scores of occurrences
  - What would be highest ranked ?

N-gram Generation & Voting

- N-gram generation from unique snippets:
  - Approximate chunking – without syntax
  - All uni-, bi-, tri-, tetra- grams
    - Concordia added 5-grams (prior errors)
  - Score: based on source query: exact 5x, others 1x
- N-gram voting:
  - Collates n-grams
  - N-gram gets sum of scores of occurrences
  - What would be highest ranked ?
    - Specific, frequent: Question terms, stopwords

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error
- Question-type-neutral filters:

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error
- Question-type-neutral filters:
  - Exclude if begin/end with stopword
  - Exclude if contain words from question, except
    - ‘Focus words’ : e.g. units
- Question-type-specific filters:

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error
- Question-type-neutral filters:
  - Exclude if begin/end with stopword
  - Exclude if contain words from question, except
    - ‘Focus words’ : e.g. units
- Question-type-specific filters:
  - ‘how far’, ‘how fast’:

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error
- Question-type-neutral filters:
  - Exclude if begin/end with stopword
  - Exclude if contain words from question, except
    - ‘Focus words’ : e.g. units
- Question-type-specific filters:
  - ‘how far’, ‘how fast’: exclude if no numeric
  - ‘who’,’where’: exclude if not NE (first & last caps)

N-gram Filtering

- Throws out ‘blatant’ errors
  - Conservative or aggressive?
    - Conservative: can’t recover error
- Question-type-neutral filters:
  - Exclude if begin/end with stopword
  - Exclude if contain words from question, except
    - ‘Focus words’ : e.g. units
- Question-type-specific filters:
  - ‘how far’, ‘how fast’: exclude if no numeric
  - ‘who’,’where’:

N-gram Filtering

- Closed-class filters:
  - Exclude if not members of an enumerable list

N-gram Filtering

- Closed-class filters:
  - Exclude if not members of an enumerable list
  - E.g. ‘what year ‘ -> must be acceptable date year

N-gram Filtering

- Closed-class filters:
  - Exclude if not members of an enumerable list
  - E.g. ‘what year ‘ -> must be acceptable date year

- Example after filtering:
  - Who was the first person to run a sub-four-minute mile?

Candidate	Score
Bannister	137
Roger	114
Roger Bannister	103
English	26
...	...

N-gram Filtering

- Impact of different filters:
  - Highly significant differences when run w/subsets

N-gram Filtering

- Impact of different filters:
  - Highly significant differences when run w/subsets
  - No filters: drops 70%

N-gram Filtering

- Impact of different filters:
  - Highly significant differences when run w/subsets
  - No filters: drops 70%
  - Type-neutral only: drops 15%

N-gram Filtering

- Impact of different filters:
  - Highly significant differences when run w/subsets
  - No filters: drops 70%
  - Type-neutral only: drops 15%
  - Type-neutral & Type-specific: drops 5%

N-gram Combining

- Current scoring favors longer or shorter spans?

N-gram Combining

- Current scoring favors longer or shorter spans?
  - E.g. Roger or Bannister or Roger Bannister or Mr.....

N-gram Combining

- Current scoring favors longer or shorter spans?
  - E.g. Roger or Bannister or Roger Bannister or Mr.....
    - Bannister pry highest – occurs everywhere R.B. +
- Generally, good answers longer (up to a point)

N-gram Combining

- Current scoring favors longer or shorter spans?
  - E.g. Roger or Bannister or Roger Bannister or Mr.....
    - Bannister pry highest – occurs everywhere R.B. +
- Generally, good answers longer (up to a point)
- Update score: S_c += ΣS_t, where t is unigram in c
- Possible issues:

N-gram Combining

- Current scoring favors longer or shorter spans?
  - E.g. Roger or Bannister or Roger Bannister or Mr.....
    - Bannister pry highest – occurs everywhere R.B. +
- Generally, good answers longer (up to a point)
- Update score: S_c += ΣS_t, where t is unigram in c
- Possible issues:
  - Bad units: Roger Bannister was

N-gram Combining

- Current scoring favors longer or shorter spans?
  - E.g. Roger or Bannister or Roger Bannister or Mr.....
    - Bannister pry highest – occurs everywhere R.B. +
- Generally, good answers longer (up to a point)
- Update score: S_c += ΣS_t, where t is unigram in c
- Possible issues:
  - Bad units: Roger Bannister was – blocked by filters
    - Also, increments score so long bad spans lower
- Improves significantly

N-gram Scoring

- Not all terms created equal

N-gram Scoring

- Not all terms created equal
  - Usually answers highly specific
  - Also disprefer non-units
- Solution

N-gram Scoring

- Not all terms created equal
  - Usually answers highly specific
  - Also disprefer non-units
- Solution: IDF-based scoring

  $S_c = S_c$ * average_unigram_idf

N-gram Scoring

- Not all terms created equal
  - Usually answers highly specific
  - Also disprefer non-units
- Solution: IDF-based scoring

  $S_c = S_c$ * average_unigram_idf

After combining

Candidate	Score
Roger Bannister	354
Sir Roger Gilbert Bannister	286
Sir Roger Bannister	280
Bannister Sir Roger	278
...	...

N-gram Scoring

- Not all terms created equal
  - Usually answers highly specific
  - Also disprefer non-units
- Solution: IDF-based scoring

  $S_c = S_c$ * average_unigram_idf

After combining

Candidate	Score
Roger Bannister	354
Sir Roger Gilbert Bannister	286
Sir Roger Bannister	280
Bannister Sir Roger	278
...	...

After scoring

Candidate	Score
Roger Bannister	2377
Englishman Roger Bannister	1853
Sir Roger Gilbert Bannister	1775
Sir Roger Bannister	1768
...	...

N-gram Reranking

- Promote best answer candidates:

N-gram Reranking

- Promote best answer candidates:
  - Filter any answers not in at least two snippets

N-gram Reranking

- Promote best answer candidates:
  - Filter any answers not in at least two snippets
  - Use answer type specific forms to raise matches
    - E.g. ‘where’ -> boosts ‘city, state’
- Small improvement depending on answer type

Summary

- Redundancy-based approaches
  - Leverage scale of web search
  - Take advantage of presence of ‘easy’ answers on web
  - Exploit statistical association of question/answer text

Summary

- Redundancy-based approaches
  - Leverage scale of web search
  - Take advantage of presence of 'easy' answers on web
  - Exploit statistical association of question/answer text
- Increasingly adopted:
  - Good performers independently for QA
  - Provide significant improvements in other systems
    - Esp. for answer filtering

Summary

- Redundancy-based approaches
  - Leverage scale of web search
  - Take advantage of presence of 'easy' answers on web
  - Exploit statistical association of question/answer text
- Increasingly adopted:
  - Good performers independently for QA
  - Provide significant improvements in other systems
    - Esp. for answer filtering
- Does require some form of 'answer projection'
  - Map web information to TREC document

Summary

- Redundancy-based approaches
  - Leverage scale of web search
  - Take advantage of presence of 'easy' answers on web
  - Exploit statistical association of question/answer text
- Increasingly adopted:
  - Good performers independently for QA
  - Provide significant improvements in other systems
    - Esp. for answer filtering
- Does require some form of 'answer projection'
  - Map web information to TREC document
- Aranea download:
  - http://www.umiacs.umd.edu/~jimmylin/resources.html