Answer Extraction & Projection

Ling573
NLP Systems and Applications
May 21, 2013

Deliverable #4

- Posted
- Notes:
  - Evaltest materials
    - Corpus: Aquaint**-2**
      - Should be installed soon: Pending David B.
      - DTD change: DOCID attributed vs element
    - Questions: TREC-2007
      - Available tonight
    - Answer patterns:
      - Available next Tuesday (avoid temptation!)

Roadmap

- Answer extraction
  - Learning Reranking I
  - Noisy channel extraction
  - Learning Reranking II
- Answer Projection
  - Strategies for document recovery

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

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

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

Feature Functions

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:

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:

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

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)

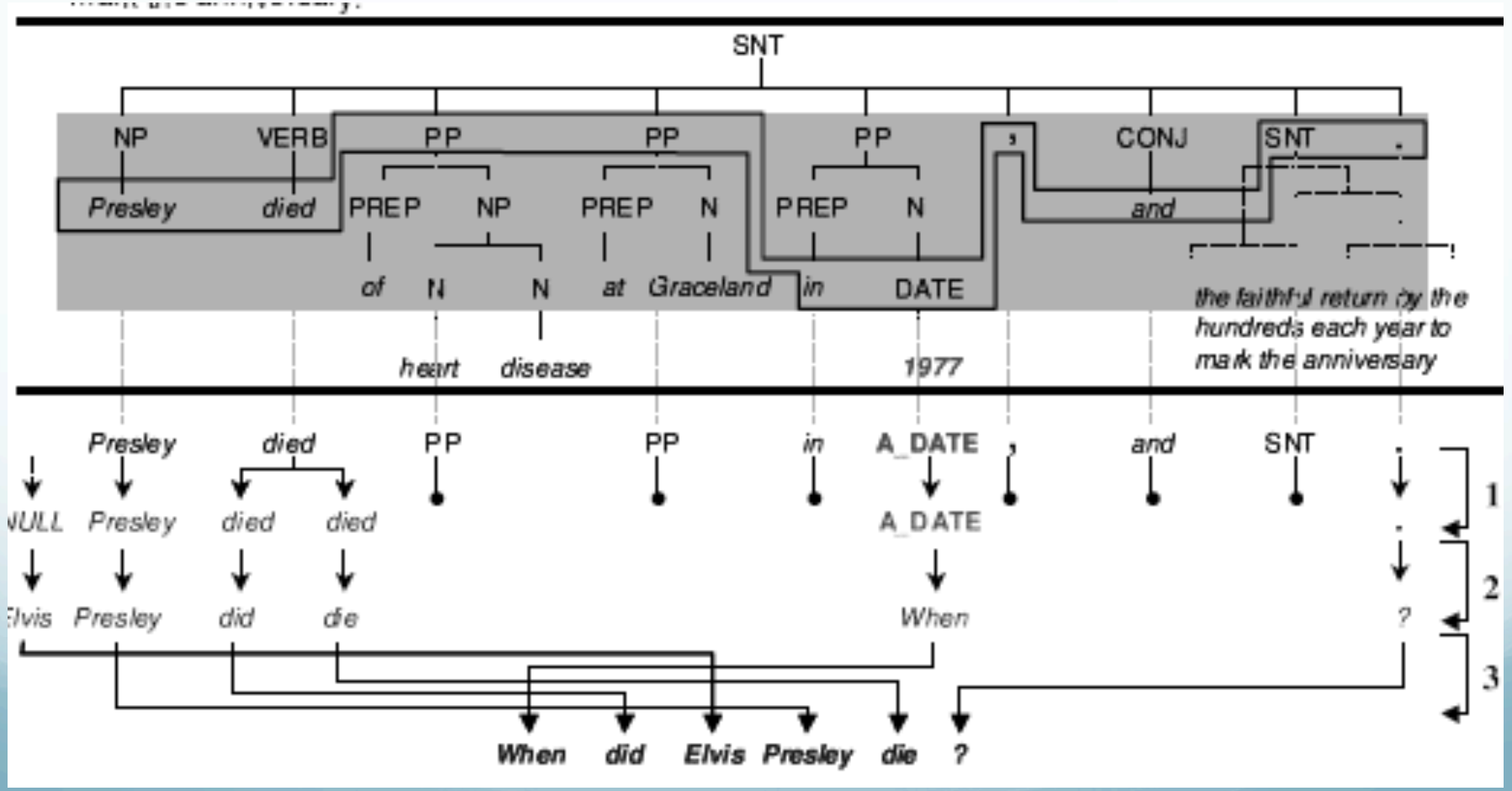
- Assign one element in cut to be 'Answer'
- Issue: Cut STILL may not be same length as Q
- Solution: (typical MT)
  - Assign each element a fertility
    - 0 – delete the word; > 1: repeat word that many times

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’

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%

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%
- Combined: 57%+

Roadmap

- Integrating Redundancy-based Answer Extraction
  - Answer projection
  - Answer reweighting

Redundancy-Based Approaches & TREC

- Redundancy-based approaches:
  - Exploit redundancy and large scale of web to
    - Identify ‘easy’ contexts for answer extraction
    - Identify statistical relations b/t answers and questions

Redundancy-Based Approaches & TREC

- Redundancy-based approaches:
  - Exploit redundancy and large scale of web to
    - Identify 'easy' contexts for answer extraction
    - Identify statistical relations b/t answers and questions
  - Frequently effective:
    - More effective using Web as collection than TREC
- Issue:
  - How integrate with TREC QA model?

Redundancy-Based Approaches & TREC

- Redundancy-based approaches:
  - Exploit redundancy and large scale of web to
    - Identify 'easy' contexts for answer extraction
    - Identify statistical relations b/t answers and questions
  - Frequently effective:
    - More effective using Web as collection than TREC
- Issue:
  - How integrate with TREC QA model?
    - Requires answer string **AND** supporting TREC document

Answer Projection

- Idea:
  - Project Web-based answer onto some TREC doc
    - Find best supporting document in AQUAINT

Answer Projection

- Idea:
  - Project Web-based answer onto some TREC doc
    - Find best supporting document in AQUAINT
- Baseline approach: (Concordia, 2007)
  - Run query on Lucene index of TREC docs

Answer Projection

- Idea:
  - Project Web-based answer onto some TREC doc
    - Find best supporting document in AQUAINT
- Baseline approach: (Concordia, 2007)
  - Run query on Lucene index of TREC docs
  - Identify documents where top-ranked answer appears

Answer Projection

- Idea:
  - Project Web-based answer onto some TREC doc
    - Find best supporting document in AQUAINT
- Baseline approach: (Concordia, 2007)
  - Run query on Lucene index of TREC docs
  - Identify documents where top-ranked answer appears
  - Select one with highest retrieval score

Answer Projection

- Modifications:
  - Not just retrieval status value

Answer Projection

- Modifications:
  - Not just retrieval status value
    - Tf-idf of **question** terms
    - No information from answer term
      - E.g. answer term frequency (baseline: binary)

Answer Projection

- Modifications:
  - Not just retrieval status value
    - Tf-idf of **question** terms
    - No information from answer term
      - E.g. answer term frequency (baseline: binary)
  - Approximate match of answer term
- New weighting:
  - Retrieval score x (frequency of answer + freq. of target)

Answer Projection

- Modifications:
  - Not just retrieval status value
    - Tf-idf of **question** terms
    - No information from answer term
      - E.g. answer term frequency (baseline: binary)
  - Approximate match of answer term
- New weighting:
  - Retrieval score x (frequency of answer + freq. of target)
- No major improvement:
  - Selects correct document for 60% of correct answers

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations
  - Baseline: All words from Q & A

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations
  - Baseline: All words from Q & A
  - Boost-Answer-N: All words, but weight Answer wds by N

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations
  - Baseline: All words from Q & A
  - Boost-Answer-N: All words, but weight Answer wds by N
  - Boolean-Answer: All words, but answer must appear

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations
  - Baseline: All words from Q & A
  - Boost-Answer-N: All words, but weight Answer wds by N
  - Boolean-Answer: All words, but answer must appear
  - Phrases: All words, but group 'phrases' by shallow proc

Answer Projection as Search

- Insight: (Mishne & De Rijk, 2005)
  - Redundancy-based approach provides answer
  - Why not search TREC collection **after** Web retrieval?
    - Use web-based answer to improve query
- Alternative query formulations: Combinations
  - Baseline: All words from Q & A
  - Boost-Answer-N: All words, but weight Answer wds by N
  - Boolean-Answer: All words, but answer must appear
  - Phrases: All words, but group 'phrases' by shallow proc
  - Phrase-Answer: All words, Answer words as phrase

Results

Model	MRR	p@1
baseline	0.477	0.346
boost-answer-2	0.464 (-3%)	0.340 (-1%)
boost-answer-5	0.408 (-14%)	0.287 (-17%)
boost-answer-20	0.329 (-31%)	0.225 (-35%)
phrases	0.471 (-1%)	0.347 (0%)
boolean-answer	0.502 (+5%)	0.374 (+8%)
phrase-answer	0.525 (+10%)	0.398 (+15%)
phrases,phrase-answer	0.517 (+8%)	0.397 (+15%)
phrases,phrase-answer,boolean-answer	**0.531 (+11%)**	**0.416 (+20%)**

Results

Model	MRR	p@1
baseline	0.477	0.346
boost-answer-2	0.464 (-3%)	0.340 (-1%)
boost-answer-5	0.408 (-14%)	0.287 (-17%)
boost-answer-20	0.329 (-31%)	0.225 (-35%)
phrases	0.471 (-1%)	0.347 (0%)
boolean-answer	0.502 (+5%)	0.374 (+8%)
phrase-answer	0.525 (+10%)	0.398 (+15%)
phrases,phrase-answer	0.517 (+8%)	0.397 (+15%)
phrases,phrase-answer,boolean-answer	**0.531 (+11%)**	**0.416 (+20%)**

- Boost-Answer-N hurts!

Results

Model	MRR	p@1
baseline	0.477	0.346
boost-answer-2	0.464 (-3%)	0.340 (-1%)
boost-answer-5	0.408 (-14%)	0.287 (-17%)
boost-answer-20	0.329 (-31%)	0.225 (-35%)
phrases	0.471 (-1%)	0.347 (0%)
boolean-answer	0.502 (+5%)	0.374 (+8%)
phrase-answer	0.525 (+10%)	0.398 (+15%)
phrases,phrase-answer	0.517 (+8%)	0.397 (+15%)
phrases,phrase-answer,boolean-answer	**0.531 (+11%)**	**0.416 (+20%)**

- Boost-Answer-N hurts!
  - Topic drift to answer away from question
- Require answer as phrase, without weighting improves

Web-Based Boosting

- Harabagiu et al 2005
- Create search engine queries from question
- Extract most redundant answers from search
  - Augment Deep NLP approach

Web-Based Boosting

- Harabagiu et al 2005
- Create search engine queries from question
- Extract most redundant answers from search
  - Augment Deep NLP approach
- Increase weight on TREC candidates that match

Web-Based Boosting

- Harabagiu et al 2005
- Create search engine queries from question
- Extract most redundant answers from search
  - Augment Deep NLP approach
- Increase weight on TREC candidates that match
  - Higher weight if higher frequency
- Intuition:
  - QA answer search too focused on query terms
  - Deep QA bias to matching NE type, syntactic class

Web-Based Boosting

- Create search engine queries from question
- Extract most redundant answers from search
  - Augment Deep NLP approach
- Increase weight on TREC candidates that match
  - Higher weight if higher frequency
- Intuition:
  - QA answer search too focused on query terms
  - Deep QA bias to matching NE type, syntactic class
  - Reweighting improves
- Web-boosting improves significantly: 20%