Answer Projection & Extraction

NLP Systems and Applications
Ling573
May 15, 2014

Roadmap

- Integrating Redundancy-based Answer Extraction
  - Answer projection
  - Answer reweighting
- Answer extraction as Sequence Tagging
  - Answer candidate reranking
  - Answer span extraction

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
- 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

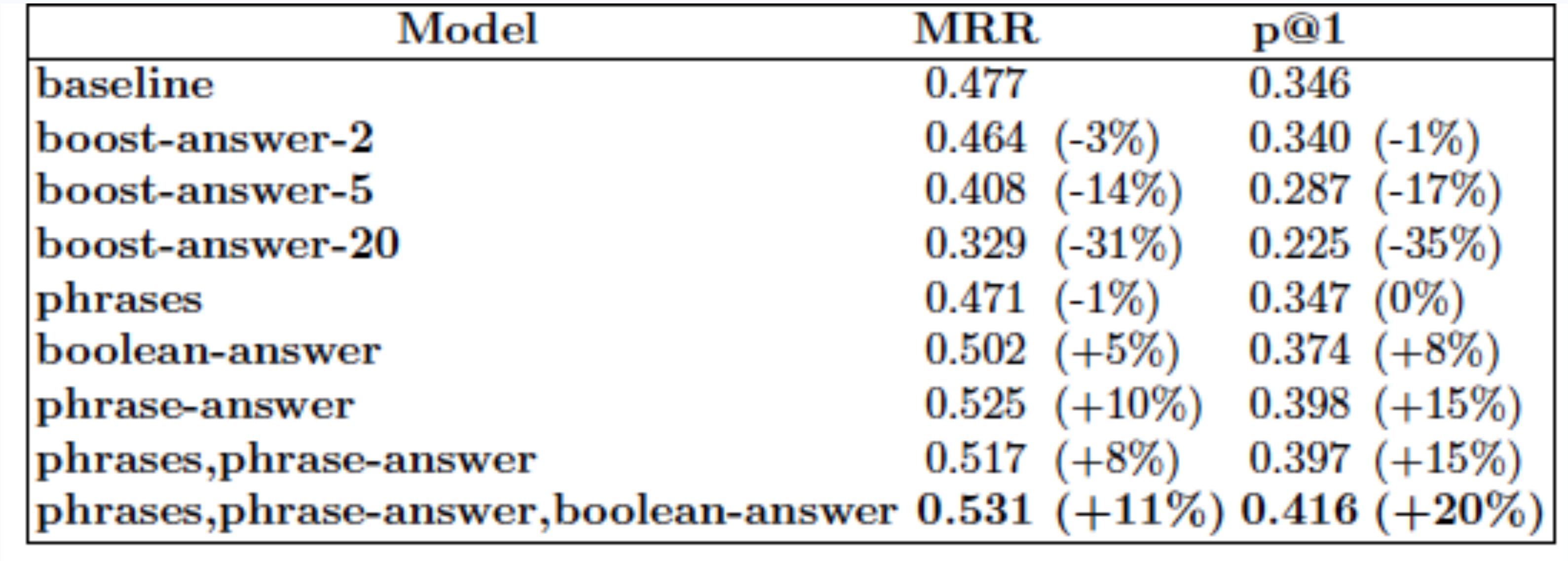
Answer Projection

- Modifications:
  - Not just retrieval status value
    - 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?
    - 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%)**

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

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%

Answering by Sequence Tagging

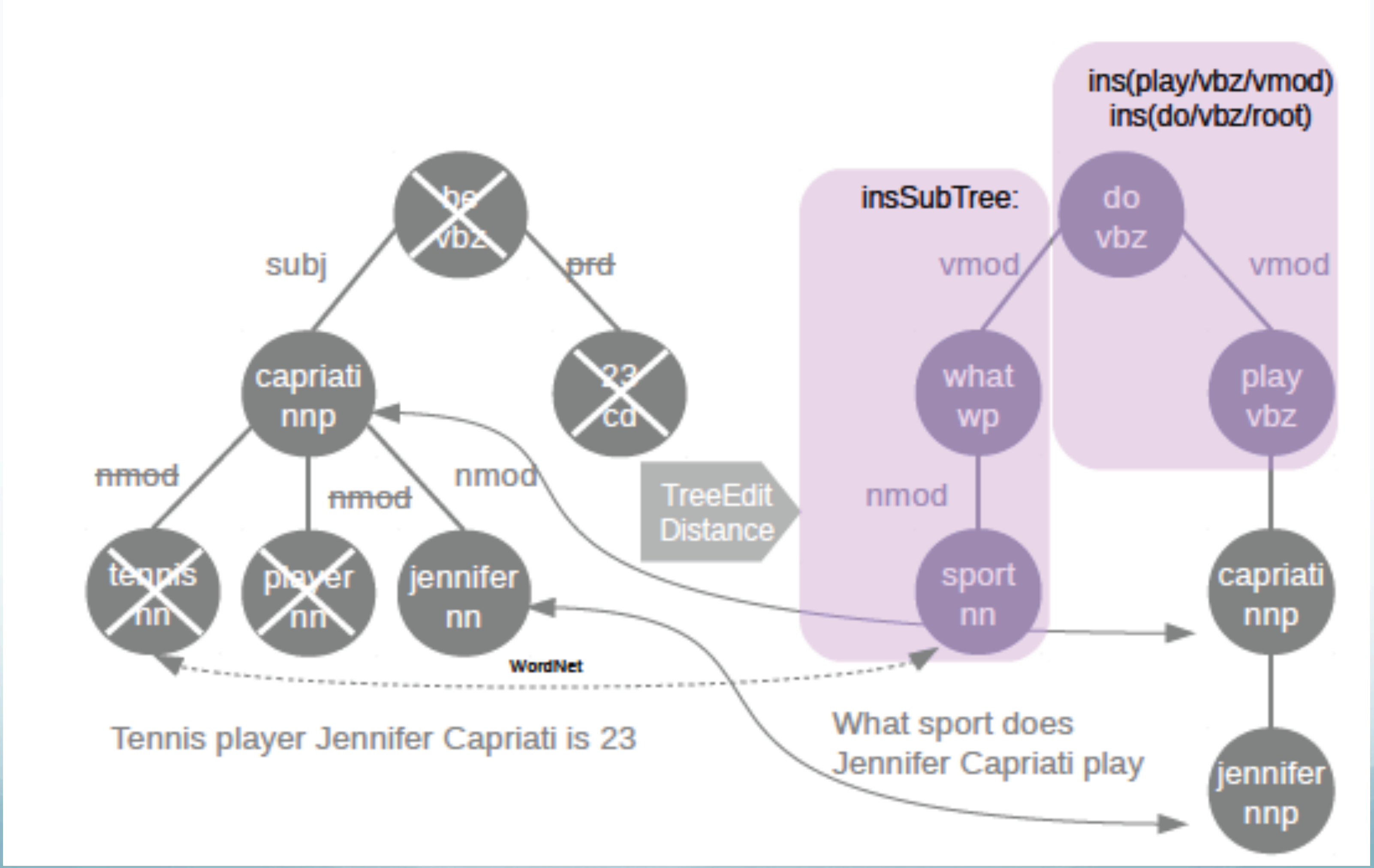
- Answer Extraction as Sequence Tagging with Tree Edit Distance
  - Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, Peter Clark

- Intuition:
  - Exploit dependency-level correspondence b/t Q & A
    - Modeled as Tree Edit Distance over dependency parses
  - Use to rank candidate answer sentences
  - Use as features in sequence tagging for answer extr.

Intuition

- Answer extraction assumes correspondence b/t Q&A
  - Many types of correspondence:
    - Pattern-based cued on answer type
    - Noisy-channel based surface word alignment
    - Syntactic parallelism of constituent tree paths
    - Semantic role parallelism of FrameNet frame elements
- Here, correspondence via dependency parse trees
  - Similarity between question and answer candidate
    - Tree Edit Distance:
      - Total cost of best transformation from Q tree to D tree
      - Transformation sequence: "edit script"

Answer to Question Edit



Tree Edit Distance

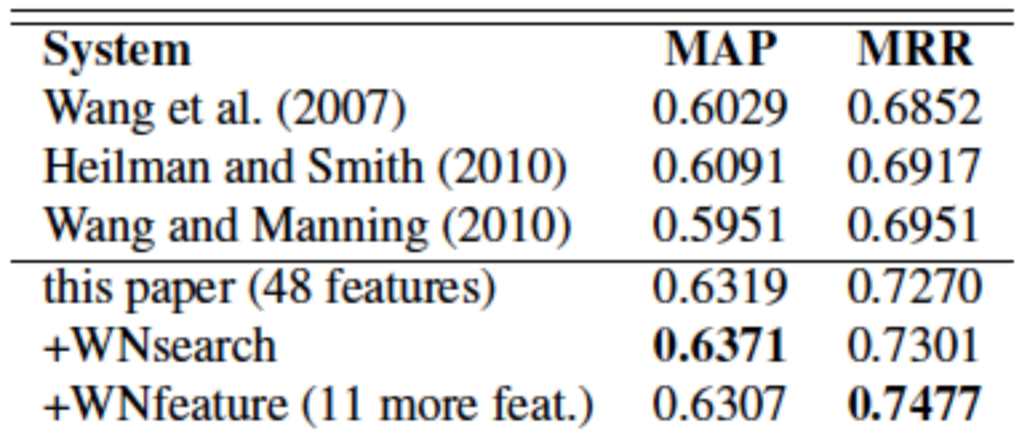
- Representation:
  - Node: lemma, POS, dependency relation to parent (DEP)
    - E.g., Mary ➔ Mary/nnp/sub
- Basic edits:
  - Insert or delete:
    - Leaf node, whole subtree, other node
  - Rename:
    - node POS, DEP, or both
- Costs assigned to each operation
- Standard dynamic programming solution: least cost, opt.

Answer Candidate Ranking

- Goal:
  - Given a question and set of candidate answer sents
  - Return ranked answer list
- Approach: learn logistic regression model
- Features:
  - Tree edit features from sentence to question
    - 48 edit types: broken down by POS, DEP (similar to prior)
  - WNSearch: TED, but allows alignment/renaming of lemmas that share WordNet relations: e.g. REN_..(sport, tennis)
  - WNFeatures:
    - # of words in each WN relation b/t question & answer

Answer Sentence Ranking

- Data: TREC QA
  - Sentences w/non-stopword overlap
  - Positive instances = pattern match
- Results:
  - Competitive w/earlier systems: WN promising



System	**MAP**	**MRR**
Wang et al. (2007)	0.6029	0.6852
Heilman and Smith (2010)	0.6091	0.6917
Wang and Manning (2010)	0.5951	0.6951
this paper (48 features)	0.6319	0.7270
+WNsearch	**0.6371**	0.7301
+WNfeature (11 more feat.)	0.6307	**0.7477**

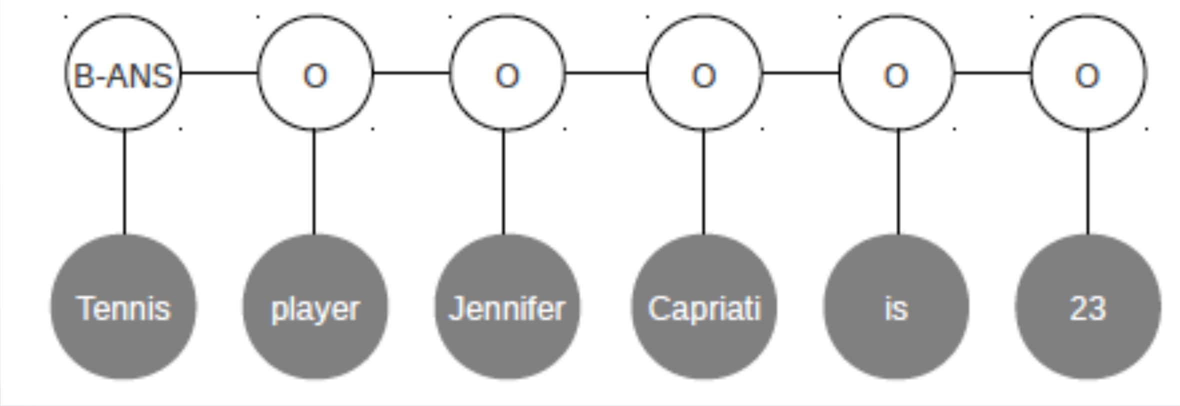
Answer Extraction

- Option 1:
  - Use tree alignment directly (like last class)
    - Answer is content word (subtree) aligned to Q-word
- Issue: Limited, not tuned for this:
  - F1: 31.4%
- Alternative:
  - Build CRF sequence tagger
  - Incorporate many features, including TED features

Answer Sequence Model

- Linear chain CRF model:
  - BIO model
  - Features over whole data

  - Example sequence tagging:



Features

- “Chunking” features:
  - Intuition: some chunks are more likely to be answers
    - E.g. “in 90 days” vs “of silly” (in “kind of silly”)
  - POS, NER, DEP features of current token
    - Unigram, bigram, trigram contexts
- Fine, but obvious gap.... No relation to question!
- Question-type features:
  - Combine q-type with above features (std. types)
  - Perform question classification for what/which

Features II

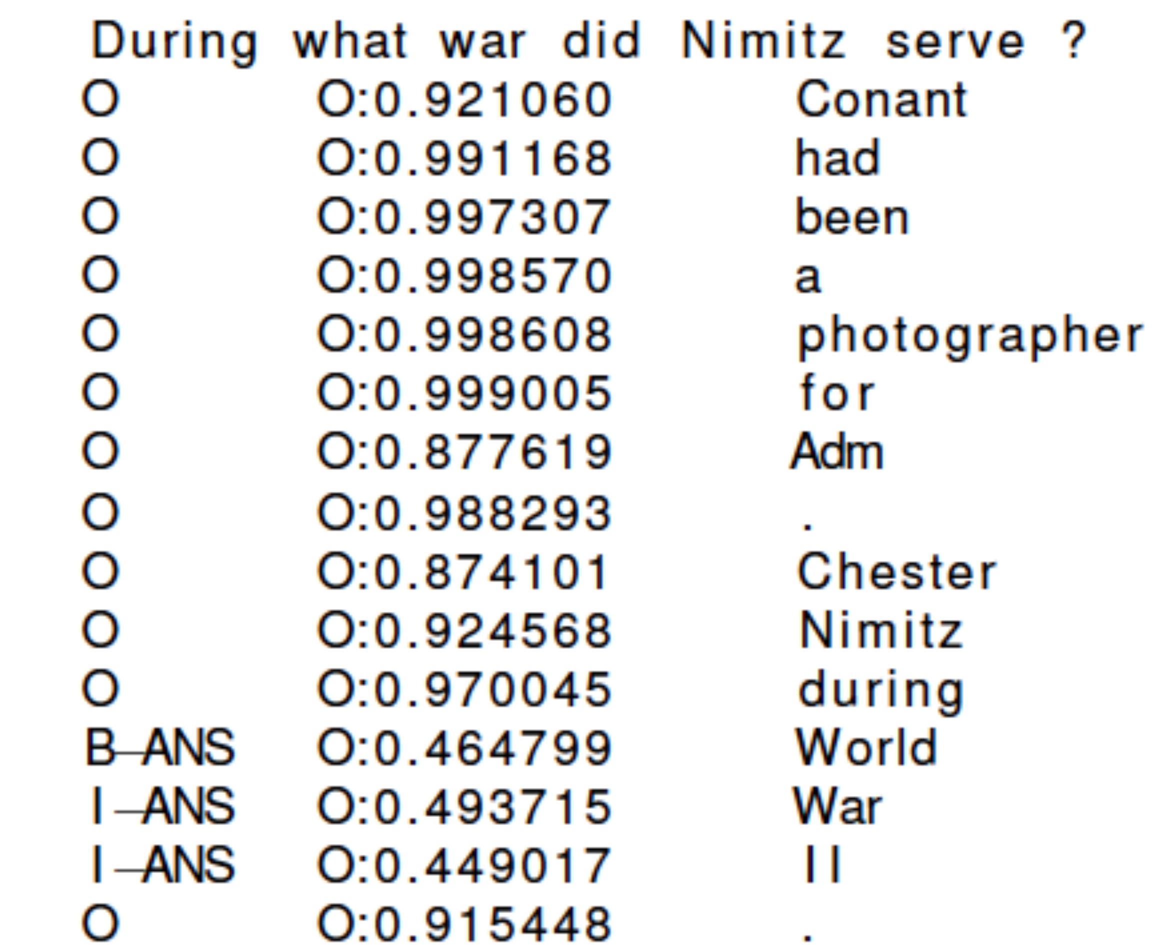
- Tree Edit Features:
  - Each token associated with edit operation from trace
    - Deleted, renamed, or aligned
      - E.g. Deleted term likely to be … answer
    - Variety of features also tied to POS/NER/DEP
- Alignment features:
  - Intuition: Answers often near aligned tokens
  - Distance to nearest aligned word (integer)
  - Also POS/NER/DEP feature of nearest aligned word

Answer Selection

- Run CRF tagging on high ranked answer sentences
  - Assume all produce answers
  - What do we do with multiple answers?
  - Weighted voting: (cf. redundancy-based approach)
    - Add partial overlap = #overlap/#words
  - What if sentence produces NO answer?
    - Insufficient prob mass for answer BI
    - "Force" candidate: outlier span
      - Threshold by multiple of Median Absolute Deviation
        - MAD = median(|x – median(x)|), sequence x
    - Weight score by 0.1

Forced Vote Example

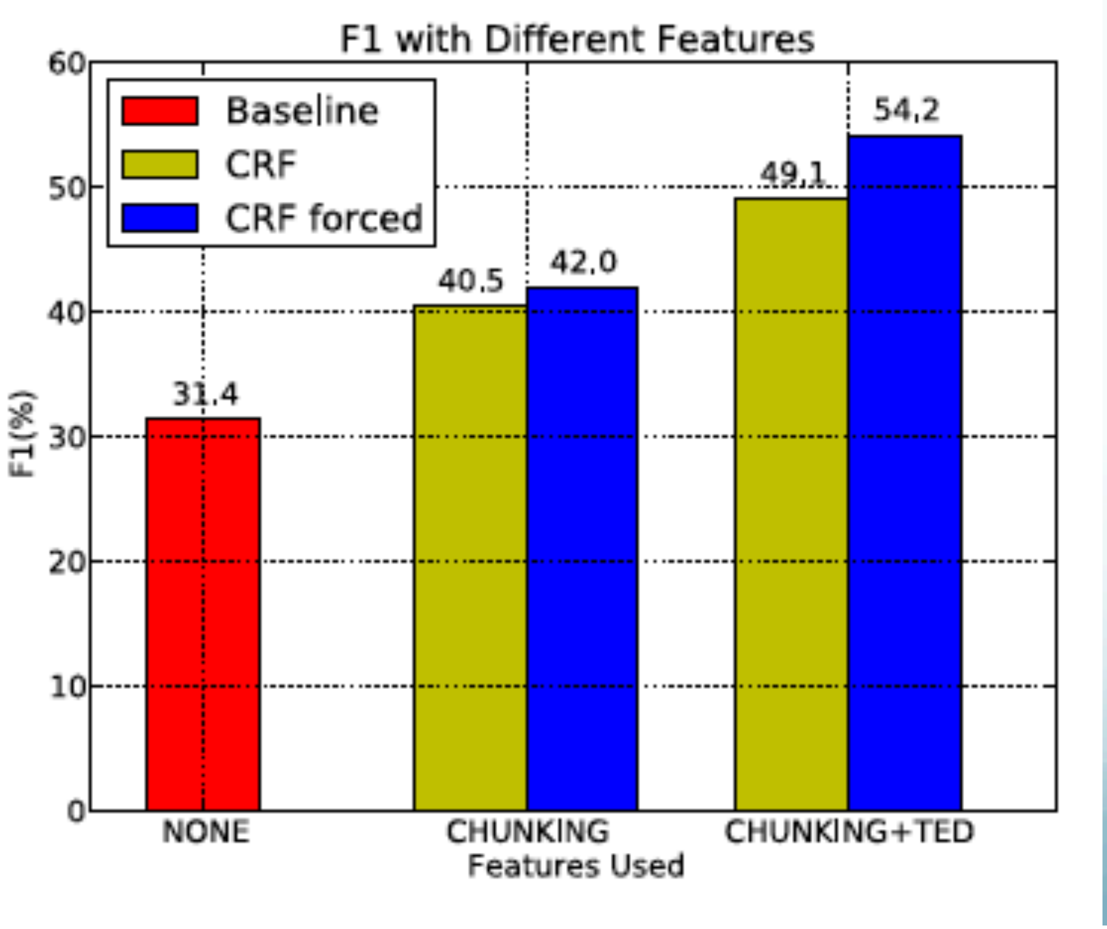
- Sequence



```
During  what  war  did  Nimitz  serve  ?
O          O:0.921060        Conant
O          O:0.991168        had
O          O:0.997307        been
O          O:0.998570        a
O          O:0.998608        photographer
O          O:0.999005        for
O          O:0.877619        Adm
O          O:0.988293        .
O          O:0.874101        Chester
O          O:0.924568        Nimitz
O          O:0.970045        during
B-ANS      O:0.464799        World
I-ANS      O:0.493715        War
I-ANS      O:0.449017        II
O          O:0.915448        .
```

Results

- All improve over baseline alignment approach
  - Chunk/Q features ~10%; TED features + ~10%


F1 with Different Features
Baseline
CRF
CRF forced
F1(%)
NONE
CHUNKING
CHUNKING+TED
Features Used