Parsing – Beyond WSJ

• Why? Not all text looks like WSJ!
  – Speech vs. text issues, formality, …

• Challenges:
  – Domain wording/tokenization differences & sparse training
  – Words may be incorrect (speech, OCR, MT) or purposely misspelled/abbreviated (chat, blogs)
  – Punctuation and case may be unreliable (unknown sentence boundaries)
  – Disfluencies and incomplete sentences
Speech and Text are not SO Different

• Uncertainties in text
  – OCR systems & handwriting recognizers make errors (not to mention typos)
  – Punctuation conventions vary with sources
  – Non-alpha characters and font help communicate intent, informal text drops case

• Domain variations (newswire vs. blogs vs. chat)
  – Different register ➔ different wording choices, non-standard sentences
Blog:
Interestingly, four Republicans, including the Senate Majority Leader, joined all the Democrats on the losing end of a 17-12 vote. Welcome to the coven of secularists and atheists. Not that this has anything to do with religion, as the upright Senator ___ swears, … I guess no one told this neanderthal that lying is considered a sin by most religions.

Chat: (online collaboration)
Brian: we should get a RC person if we can
Sidney: ah, nice connection
Sidney: Mary ___
Brian: she's coming to our mini-conference too
Sidney: great. we can cite her :)
Brian: copiously
Alan: ha
...
Sidney: Brian, check out ch 2 because it relates directly to some of the task modeling issues we're discussing. the lit review can be leveraged
Solutions

• Statistical parsing provides robustness
• Domain differences addressed by semi-supervised learning
• Issues discussed here
  – Evaluation & word uncertainty
  – Parsing & word uncertainty
  – Sentence segmentation
  – Disfluencies
EVALUATION
Evaluating Parses

Types of errors in Parseval
- P/R: inserted & missing brackets
- Count of crossing brackets

[ S [NP I ][VP was [ADJP personally acquainted [PP with [NP the people]]]]]

[ S [NP I ][VP was [ADJP personally acquainted]] [PP with [NP the people]]]

[ S [NP I ][VP was personally] [VP acquainted [PP with [NP the people]]]]
What if the words don’t match?

I was personally acquainted with the people

I * personally acquaint Edwin the people yeah

Inserted bracket or constituent error?

Missing brackets

Crossing brackets

Missing word, so no PP

Crossing bracket?
SPARSEVAL (Roark et al.)

• SParseval is a tool for scoring a parse when the words don’t match the reference.

• Multiple scoring functions implemented.

• My favorite:
  – Translate parse to dependency representation.
  – Compute P/R from errors on word-dependency triples (dependent word, relation, head word).
  – A bit like a weighted word error rate, with higher weight on head words.
Mapping Trees to Dependency Graphs
Simple Example
Sparseval Dependency Score

• REF:
  I really think so.

• HYP 1:
  I really think yeah.

• HYP 2:
  I really sink so.

• REF triples:
  (I, S/NP, think)
  (really, VP/AdvP, think)
  (think, <s>/S, <s>)
  (so, VP/AdvP, think)

• HYP 1: WER=1/4, F=3/4
  (I, S/NP, think)
  (really, VP/AdvP, think)
  (think, <s>/S, <s>)
  (yeah, S/DM, think) *

• HYP 2: WER=1/4, F=0
  (I, S/NP, sink) *
  (really, VP/AdvP, sink) *
  (sink, <s>/S, <s>) *
  (so, VP/AdvP, sink) *

Word errors impact parse score; headwords get more weight!
Word Uncertainty
Parsinig Speech: Preliminaries

- Early work on parsing and ASR either
  - Used reference transcripts, so didn’t deal with word or sentence error problems
  - Focused on parsers as a language model, so didn’t deal with parse scoring

- Speech parsing may have two objectives
  - WER (parsing as a language model)
  - SParseval (parsing for parsing’s sake)
  but Sparseval is impacted by word errors
Handling Word Uncertainty

- Direct-in-decoding parser (left-to-right)
  - Almost-parsing (super-tag) language model
  - Word+parse history conditioning
- Lattice or confusion network rescoring (left-to-right)
- N-best rescoring (any parser)
  - Allows discriminative reranking w/ non-local features
Word vs. Parse Hypothesis Richness

• Consider
  – N-best ASR sentences
  – M-best parses/sentence

• Oracle
  SParseval F
  – Better to increase N than M
  – Large M only helps with N>30
Discriminative Parse Reranking

From Johnson-Charniak ’05 (following Collins ’00)

• Generate M best parses, each with a probability

• For each parse, compute vectors of features, e.g.
  – 50,000 headed-constituent counts: $c(l,n,h,d,v)$
    count of constituent type $l$, with $n$ words, height $h$,
    right side depth $d$ and lexical label $v$

• Can use maximum entropy model, average perceptron, SVN, etc. in reranking
ASR Hypothesis Reranking

- Same as parse reranking, except:
  - Objective is $\text{min WER or Parseval F}$
  - Parse features are expected values (analogy to parse LM)
    
    \[
    P(W) = \sum_T P(W,T) \\
    E(C_i(W)) = \sum_T C_i(W,T)P(W,T)
    \]
    
    where $C_i$ is the i-th feature in the 50k vector

- Optionally, can incorporate prosodic features
WER vs. SParseval Reranking

- Parsing helps more with bigger N
- Improving WER improves SParseval
- Non-local features:
  - helps parsing only for N=1
  - always helps ASR
Example Corrections

• Anecdotal examples:
  – well they *don’t* [---] always
  – *used* [nice] to live in Colorado
  – *we’re* [where] the old folks now
  – *they* (uh) [yeah yeah] really get into it

• Conclusions:
  – Using structure helps overcome bias towards frequent words
  – Using syntax helps recover main verbs & pronouns (important for entity tracking)
Sentence Boundary Uncertainty
Sentence Segmentation

- Pause-based segmentation is not equivalent to sentence segmentation
- Informal online text is challenging for sentence segmentation
- Sentence (SU) boundary detection:
  - At each word boundary, predict probability of +/- sentence boundary (sequence labeling problem)
  - Statistical sequence (tagging) model, combines lexical and orthographic or prosodic cues
  - Precision/recall operating point depends on application
Segmentation Strategies

- Pause-based segmentation (standard strategy)
- Automatic SU detection w/ HMM-like model
- Oracle detection (hand-labeled SUs)
- Auto SU detection performance:

<table>
<thead>
<tr>
<th></th>
<th>F-measure</th>
<th>SER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pause-based</td>
<td>.56</td>
<td>.69</td>
</tr>
<tr>
<td>Auto</td>
<td>.81</td>
<td>.35</td>
</tr>
</tbody>
</table>
Impact on Parsing

• Auto SUs reduces error (1-F) relative to pause segmentation by roughly 20%
• Half the loss relative to oracle is recovered

![Bar chart showing error (1-F) for Charniak and Bikel models with different segmentation methods.]

• no explicit edit detection
• standard Parseval scoring
• similar results for ASR transcripts
Impact on Parsing (cont.)

• Auto SUs reduces bracket crossings relative to pause segmentation by 36%
• Half the loss relative to oracle is recovered

![Bracket Crossing Chart]

- no explicit edit detection
- standard Parseval scoring
Optimizing Segmentation

• The best precision/recall operating point depends on the application

• Higher recall (shorter sentences) benefits:
  – Parsing \( T=0.35 \) (Harper et al., JHU ’05 workshop)
  – MT \( T=0.2 \) (Matusov et al. ’07)
  – Entity extraction (NYU-ICSI)

• Higher precision (longer sentences) benefits:
  – Name extraction \( T=0.5 \) (NYU-ICSI)
Sentence Detection w/ Multiple Word Hypotheses
Disfluencies
Edit Disfluencies

oh it’s you know + we’re about to do like the + the uh fiesta bowl there ||

oh we’re about to do like the uh fiesta bowl there.

Some observations:
• Repeat/rephrase edits are modeled well by a “rough copy” operation
  (Frequent, but good language cues: we were + I was....)
• Restarts have little relation to the next words (but less frequent)

Claims: (Charniak/Johnson)
• PCFGs are not well suited to the “rough copy” operation
• Explicit edit detection (and filler removal) should improve parsing
Alternative Edit Handling Strategies

• Include edits (and fillers) in the grammar like other constituents
  – With vs. without adding IPs in the word stream

• Use separate edit/filler detection model, parse cleaned-up text
  – With vs. without detected IPs
  – Serially or jointly

• Types of edit models
  – Word/POS copy (I did + you did // the + the)
  – Tree adjoining grammar (TAG)
How well do we detect edits?

<table>
<thead>
<tr>
<th></th>
<th>Edit F</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train on original</td>
<td></td>
</tr>
<tr>
<td>PCFGe</td>
<td>65.8</td>
</tr>
<tr>
<td>PCFG + TAG</td>
<td>77.1</td>
</tr>
<tr>
<td>PCFGe + oracle IP</td>
<td>84.0</td>
</tr>
</tbody>
</table>

- PCFGe alone is a poor edit detector (as Charniak claimed)
- Acoustic cues (IP) can potentially help a lot!
- NIST RT04 Eval results: (Lease, Charniak & Johnson)
  - PCFG+TAG+autoIP beats PCFG+TAG a little
  - PCFG+TAG+autoIP beats autoIP+copy model (Liu et al.)
Edit Strategies and Parsing

Relaxed-edit Parseval score

<table>
<thead>
<tr>
<th>Remove Edits?</th>
<th>Model</th>
<th>F-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>no</td>
<td>PCFGe</td>
<td>76.6</td>
</tr>
<tr>
<td>no</td>
<td>PCFGe w/ oracle IP</td>
<td>77.8</td>
</tr>
<tr>
<td>yes</td>
<td>PCFGe + PCFG</td>
<td>84.4</td>
</tr>
<tr>
<td>yes</td>
<td>TAG + PCFG</td>
<td>84.8</td>
</tr>
<tr>
<td>yes</td>
<td>oracle edit + PCFG</td>
<td>87.0</td>
</tr>
</tbody>
</table>

Removing edits before training (learning on clean text) is more important than the specific edit detector.
Edit Findings

- PCFGs alone (no acoustic cues) are not good edit detectors
- But, PCFGs can help in edit detection combined with acoustic & copy cues
- Much room for improvement in auto edit detection (F<80)
- Most important effect: training on cleaned up text
Summary of Biggest Effects

• For one recognition system w/ N-best alternatives, the issues order by impact are:
  – Sentence segmentation
  – Edit clean-up
  – Word alternatives
  – Parse alternatives

• BUT, improvements to the recognition system can have a bigger impact than any of these