Widening the NLP pipeline for Spoken Language Processing

Srinivas Bangalore, AT&T-Research
Outline

- Natural Language Processing Pipeline
  - Text input
  - Speech input
- Uniform decoding framework
- Case Studies
  - Call-type classification
  - Speech translation
  - Multimodal language processing
Text-based Natural Language Processing

- Text Document(s)
- Web Documents
- Email Messages
- Blogs

Input-specific Normalization -> Text -> NLP Pipeline

Richly Annotated Text -> Task-specific Transduction

Summarization

Machine Translation

Question Answering

Information Extraction
NLP Pipeline: Beads on a String

- Tokenization
- Sentence Segmentation
- Part-of-speech tagging
- Noun/Verb Chunking
- Named Entity Detection
- Syntactic Parsing
- Word Sense Disambiguation
- Co-reference resolution
- Semantic Role Labeling
Spoken Language Processing

- Telephone conversations
- News broadcasts
- Voicemail messages
- Movies

Speech Recognition → ASR output → NLP Pipeline → Richly Annotated Text → Task-specific Transduction

- Summarization
- Machine translation
- Question answering
- Information extraction
- Interactive dialog
Widening the NLP pipeline

- Passing one-best solution is sub-optimal.
- **Error in processing models**
  - Most modules in the pipeline are not perfect
  - Error propagation down the pipeline
- **Ambiguity in NLP**
  - "John saw a man with a telescope"
  - Postpone ambiguity resolution down the pipeline
  - Until information is available to resolve the ambiguity
- **N-best solutions**
  - List of solutions ranked by some goodness criteria
- **Weighted packed representations**
  - Lattices for linear outputs
  - Forests for hierarchical outputs

- **N-best versus Lattices/Forests**
  - N needs to be very large for substantially different solution
  - Repeated computation is factored out
    - Significant parts are shared across n-best solutions
A word about decoders

- Specialized decoder for each task
  - Use weighted lattices as input
  - Produce weighted lattices as output
- Uniform decoding framework
  - Most NL processing steps can be encoded as token tagging tasks.
  - … word/?? …
- Approximation for other steps
  - Attachment in parsing
- Weighted finite-state transducers
Weighted Finite-State Transducers (WFST)

- Provide efficient ways of representing weighted ambiguous hypotheses.
- Closed under composition
  - straightforward integration of finite-state constraints.
  - allows for modular development without loss of optimality of the solution.
- Decoding: linear in the input size.
- Multi-tape finite-state automata used to represent constraints from different levels of language processing.
- Extensively used for speech and language processing.
Decoders as WFSTs

- Grammar based decoding models
  - Regular expressions (e.g. dates, telephone numbers, name lists)
  - Context-free grammars (syntactic parsers)
    - Approximation techniques (Nederhof 1997, Pereira and Wright 1997)
- HMM-based generative model
  \[ T^* = \arg\max \prod_{i=1}^{n} P(w_i \mid t_i) \cdot P(t_i \mid t_{i-1}) \]
  - (Schabes and Roche 1997)
- Discriminatively trained classification models
  \[ T^* = \arg\max \prod_{i=1}^{n} P(t_i \mid f(t_{i-1}^1, w_n^1)) \]
  - Decision Trees to FSTs (Sproat and Riley, 1996); Adaboost to FSTs (Bangalore, 2004)
  - Encode features and weights as context-dependent rewrite rules (CDR)
    \[ \phi \rightarrow \psi \mid \gamma \rightarrow \delta \]
  - Compile CDRs into FSTs (Johnson 1972, Kaplan and Kay 1994, Mohri and Sproat 1996)
Outline

- Natural Language Processing Pipeline
  - Text input
  - Speech input
- Uniform decoding framework
- Case Studies
  - Call-type classification
  - Speech translation
  - Multimodal language processing
Call-type classification

Calls are classified based on user’s response to an opening prompt.
- “How may I help you” (Gorin et.al. 1997); BBN call director (Natarajan et.al. 2002); (Chelba and Acero 2003); (Cox 2003)

Training data:

I would like to speak to an operator: Request(customer_care)
What is my account balance: Request(account_balance)
I’d like to have a copy of my March bill: Request(copy_bill)
How do I pay my bill: Ask(bill_payment)

Classification model:

\[ \text{topclass} = \arg \max_{\text{class}} P(\text{class} \mid \text{Ngrams}(\text{input})) \]

ASR output is classified
- one-best, n-best, word lattice
Call-type classification error rates  Results from (Haffner 2005)

Top class error rate after rejecting 30% low confidence examples:

- **3 inputs**: 1-best sentence, 10 best, full lattice
- **trigram** word features

ASR word accuracy about 70% for the three applications

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Input</th>
<th>App1 (82 classes)</th>
<th>App2 (97 classes)</th>
<th>App3 (64 classes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poly2 SVM</td>
<td>1-best</td>
<td>12.9</td>
<td>8.44</td>
<td>4.66</td>
</tr>
<tr>
<td>Poly2 SVM</td>
<td>10-best</td>
<td>11.3</td>
<td>7.45</td>
<td>4.37</td>
</tr>
<tr>
<td>Poly2 SVM</td>
<td>Lattice</td>
<td>10.2</td>
<td>6.68</td>
<td>3.37</td>
</tr>
</tbody>
</table>

Classification of ASR word lattices consistently outperforms classification of one-best ASR output.
Spoken Language Translation

- Two-step process (narrow pipe)

- Two-step process (broader pipe)

- Tight-coupling (integrated ASR+MT)
FST-based Spoken Language Translation

- Finite-state transducer based spoken language translation
  - **Lexical choice** and **reordering** are modeled using finite-state transducers
  - \( T \) estimated from bilingual phrases/tuples, source \( F \), decoded target \( E^* \)

\[
E_{\text{lex}} = \pi_1(\text{best}(F \circ T)) \\
E^* = \text{best}(\text{permute}(E_{\text{lex}}) \circ \text{LM}_E)
\]

- FST-based Eutrans II Italian-English task (Matusov, Kanthak, Ney ICASSP 2006)
  - 23.7% ASR word-error rate

<table>
<thead>
<tr>
<th>Method</th>
<th>WER(%)</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>One-best ASR output</td>
<td>37.4</td>
<td>51.3</td>
</tr>
<tr>
<td>Word-lattices ASR output</td>
<td>36.6</td>
<td>52.4</td>
</tr>
<tr>
<td>ASR+MT integrated</td>
<td>36.3</td>
<td>52.6</td>
</tr>
</tbody>
</table>
Multimodal Language Processing

Multimodal interfaces: allow for multiple modes of input
- Pen/hand gestures, handwriting and speech

Interpretation of input
- derived by fusing information distributed in multiple input modalities

Challenges:

- Interpretation ambiguity
  - Each combination of strokes as a candidate for handwriting and gesture recognition
  - Even simple inputs can have highly ambiguous interpretations

- Speech and gesture recognition errors

- Modality Synchronization
  - Alignment between input lattices
Representation of input and output streams

Speech:

Gesture:

Meaning:
Multimodal grammars

Multimodal context-free grammar

- Terminals are multimodal tokens consisting of three components:

  - Gesture stream (gesture symbols)
  - Speech stream (words)
  - Combined meaning (meaning symbols)

- Grammar rules encode
  - Gesture and speech alignment
  - Gesture-speech combined meaning

Speech Input

Gesture Input

SPEECH RECOGNITION

GESTURE RECOGNITION

MULTIMODAL INTEGRATION AND UNDERSTANDING

MULTIMODAL GRAMMAR

Meaning lattice
review: <info>review</info>
address: <info>address</info>
phone: <info>phone</info>
for: <obj>
show: <show/>
zoom: <zoom/>
in: 
here: <Garea><doc>
these: <Gsel>
those: <Gsel>
two: <two>
three: <three>
ten: <ten>
and: <and>

e:SEM:SEM
e:SEM:SEM

e:cmd

e:obj

e:obj

e:cmd

phone
for
these
two

restaurants

Gsel
rest
SEM

Gsel
rest
SEM

Garea
SEM

Garea
SEM

Ghw
O

<info>phone</info>
<obj>
<rest>
SEM
</rest>
</obj>
</cmd>
Finite-state edit machines: Robust interpretation

- Transform ASR output so that it can be assigned a meaning by the FST-based Multimodal Understanding model.

phone restaurant these to restaurants please

phone for ε these two restaurants ε

- Decoding: $s^* = \arg \min_s \lambda_s \circ \lambda_{Edit} \circ \lambda_{Grammar}$

- MATCH domain concept sentence accuracy
  (Bangalore and Johnston 2006)

<table>
<thead>
<tr>
<th></th>
<th>Concept Sentence Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Edit</td>
<td>38.9</td>
</tr>
<tr>
<td>One-best Edit</td>
<td>60.2</td>
</tr>
<tr>
<td>Lattice Edit</td>
<td>63.2</td>
</tr>
</tbody>
</table>

Edit machine (insert, substitute, delete, identity arcs).
Summary

• Widening the NLP pipeline for spoken language processing
  - Imperfect output from speech recognition and other processing components
  - Inherent ambiguity in language
  - N-best or lattice representations

• Extending decoders to cope with lattice input
  - FST as a uniform decoding framework
  - Grammar-based, HMM-based, Classification-based decoders

• Case Studies:
  - Call-type classification
  - Spoken Language Translation
  - Multimodal Language Processing

• Issues:
  - Combining weights across multiple disambiguation models
  - Search and prune during FST composition (Lazy evaluation)