Learning Individual Adaptation in Dialogue Systems

François Mairesse
Marilyn Walker

Department of Computer Science
University of Sheffield, United Kingdom

Symposium on Dialogue Modelling and Generation
Amsterdam, July 7th 2005
Outline

- Motivation
- Training a sentence generator
- Results of individual models
  - Model performances
  - Qualitative analysis of preferences
- Adaptation in a full dialogue system
Motivation

- People adapt to conversational partner
  - Based on individual differences
    - Age, social group, intelligence, hierarchy
  - Affects many aspects of language
    - Acoustic parameters (Coulston et al. 02)
    - Lexicon (Brennan 96)
    - Syntax (Niederhoffer & Pennebaker 02)
Motivation

- User specific content selection (Rich 79)
- User prefer systems with same personality (Reeves and Nass 96)
  - Similarity-attraction effect
- Individual adaptation is useful

- Hypothesis: individual linguistic preferences can be modeled and analyzed
Trainable Sentence Generator

- **SPoT** (Walker et al. 02)
  - Stochastic sentence generation
  - Trainable sentence ranker
- **Input is a content plan**
Stochastic Sentence Generation

- Randomly generate sentence plan trees
  - Map rhetorical relations to clause combining operations
    E.g. justification $\rightarrow$ since, because
    inference $\rightarrow$ conjunction, period, merge
  - Nodes are ordered
Stochastic Sentence Generation

- Last step: realization of each alternative
  - Based on sentence plan tree and deep syntactic trees of each assertion
  - Add function words

"Chanpen Thai has the best overall quality among the selected restaurants since it is a Thai restaurant, with good service, its price is 24 dollars, and it has good food quality."

"Chanpen Thai is a Thai restaurant, with good food quality. It has good service. Its price is 24 dollars. It has the best overall quality among the selected restaurants."
Trainable Sentence Ranking

- Learn user preferences
- Associate features to alternatives
  - Node counts of sentence plan tree

![Diagram]

- Traversal (WITH-NS-infer, assert-reco-cuisine, assert-reco-service) = 1
- Leaves-under (WITH-NS-infer) = 2
- Leaves (assert-reco-cuisine, assert-reco-service) = 1
Trainable Sentence Ranking

- **Training the ranker**
  - Training data: user ratings of sentences
  - Learning algorithm: RankBoost (Freund et al. 98)
  - Generalizes user ratings for any new sentence
    - Compute ranking score

<table>
<thead>
<tr>
<th>Sentence</th>
<th>User A</th>
<th>User B</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sentence 1</td>
<td>4/10</td>
<td>6/10</td>
</tr>
<tr>
<td>Sentence 2</td>
<td>8/10</td>
<td>9/10</td>
</tr>
<tr>
<td>Sentence 3</td>
<td>7/10</td>
<td>5/10</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
User models

- RankBoost produces a set of rules

  If feature ≥ threshold then
  modify ranking score by α

- Compute the ranking score

  Given an utterance u1 with feature vector F = (2, 1, ...)
  and rules:
  - if F(1) ≥ 1 increase score by 0.8
  - if F(2) ≥ 0.5 decrease score by 0.3
  \[ h(u1) = 0.8 - 0.3 = 0.5 \]

  If for another utterance h(u2) = 0.3, u1 is preferred over u2
Quantitative Results

- Testing the models
  - Compare correct ranking of 300 sentences with models
  - 2 fold cross validation

\[
\text{Ranking Loss} = \frac{\text{Incorrectly ranked utterance pairs}}{\text{Total number of utterance pairs}}
\]

<table>
<thead>
<tr>
<th>Ranking Loss</th>
<th>A’s test data</th>
<th>B’s test data</th>
</tr>
</thead>
<tbody>
<tr>
<td>A’s model</td>
<td>0.16</td>
<td>0.51</td>
</tr>
<tr>
<td>B’s model</td>
<td>0.51</td>
<td>0.15</td>
</tr>
<tr>
<td>AVG model</td>
<td>0.27</td>
<td>0.26</td>
</tr>
</tbody>
</table>

→ Individual models perform better than averaged ones
Qualitative Analysis

Compare individual preferences

“Chanpen Thai has the best overall quality among the selected restaurants since it is a Thai restaurant, with good service, its price is 24 dollars, and it has good food quality.”

User A: 2/10 – Model: 1.6/10
User B: 8/10 – Model: 6.5/10

<table>
<thead>
<tr>
<th>Conditions of some rules of User B’s model</th>
<th>( \alpha )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Average Leaves Under (CW-CONJUNCTION-infer) ( \geq 2.8 )</td>
<td>0.52</td>
</tr>
<tr>
<td>2. Leaves ( assert-reco-best, assert-reco-cuisine ) ( \geq 1 )</td>
<td>0.50</td>
</tr>
<tr>
<td>3. Ancestor ( assert-reco, PERIOD-infer, PERIOD-infer ) ( \geq 1.5 )</td>
<td>-0.49</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Features associated with high \( \alpha \) values tend to be preferred by the user
Qualitative Analysis: Example

Rule 1: Average Leaves Under (CW-CONJUNCTION-infer) ≥ 2.8
→ increase alpha by 0.52

Rule 2: Leaves (assert-reco-best, assert-reco-cuisine) ≥ 1
→ increase alpha by 0.50

Leaf-assert-reco-best * assert-reco-cuisine = 1
→ User B’s Rule 2 increases alpha by 0.50
→ Models preference for the claim to be expressed first and before the cuisine

CW-CONJUNCTION-infer-avg-leaves-under = 3
→ User B’s Rule 1 increases alpha by 0.52
→ Models preference for combining assertions using conjunctions
Qualitative Analysis: More Examples

Why doesn’t User B like it?
- Doesn’t respect ordering
- No conjunctions
- Many periods

Rule 3: Ancestor ( assert-reco,
PERIOD-infer,
PERIOD-infer ) \geq 1.5
\rightarrow decrease alpha by 0.49

Other realization of the same content:

User A: 8/10 – Model: 8.1/10
User B: 4/10 – Model: 2.9/10

“Chanpen Thai is a Thai restaurant, with good food quality. It has good service. Its price is 24 dollars. It has the best overall quality among the selected restaurants.”
Future Work

- Problems
  - Time consuming to acquire feedback
    - 120 sentence ratings to get 0.2 ranking loss

- Integration in a full dialogue system
  - User utterances as direct feedback
  - Domain-independent features
  - Long term use
Future Work: Full System

- **New modules**
  - Feature detection
  - Communicate feature information to learning algorithm
  - Associate positive feedback on the fly
Conclusion

- Language generators can be trained for individual users
- Individual models perform better than averaged models
- User preferences can be analyzed
- Learn preferences in real-time?
References


Thank you! Questions?