Enabling Browsing in Interactive Systems

Joseph Henry Polifroni

Thesis submitted
in partial fulfillment of the requirements
for the degree of
Ph.D.

The Department of Computer Science
University of Sheffield
United Kingdom
December 2007
Contents

Title Page ................................................................. 1
List of Figures ............................................................... iv
List of Tables ............................................................... vii

Abstract xiii

Acknowledgments xiii

1 Introduction ................................................................. 2
  1.1 Motivation ............................................................. 2
  1.2 Hypotheses ............................................................ 9
  1.3 Approach and Methodology ........................................ 11
  1.4 Contributions ........................................................ 12
  1.5 Organization ........................................................ 13

2 Previous Work ............................................................. 15
  2.1 Browsing as an information-seeking strategy ................. 15
  2.2 Beyond data elicitation ............................................. 18
    2.2.1 Cooperative Answering ....................................... 18
    2.2.2 Knowledge Discovery from Databases .................... 18
    2.2.3 Cooperative Database Systems ............................. 19
    2.2.4 Data visualization ............................................ 21
  2.3 Summarization ...................................................... 22
  2.4 User modelling ..................................................... 28
  2.5 Dialogue management ............................................. 32
  2.6 Generating summary responses in dialogue systems .......... 34
  2.7 Conclusion .......................................................... 37

3 Testing the utility of summaries for structured data: A user
   study in the restaurant domain ....................................... 40
  3.1 Goals of experiment ................................................. 40
4 Testing a range of methods for automated response construction

4.1 Issues with the Refiner ........................................ 47
4.2 Applying machine learning to corpus .......................... 48
4.3 Advantages of using decision tree induction for dialogue systems 50
4.4 Database used .................................................. 52
4.5 Parameters in automatically constructing summaries ....... 54
   4.5.1 Number of attributes ...................................... 55
   4.5.2 Attribute Selection/Ranking ............................... 56
      4.5.2.1 Information gain .................................... 57
      4.5.2.2 Gain ratio .......................................... 61
      4.5.2.3 Forward stepwise selection .......................... 64
      4.5.2.4 Backward stepwise elimination ..................... 66
      4.5.2.5 Change in Error Rate in Forward Stepwise
               Selection and Backward Stepwise Elimination .... 67
      4.5.2.6 Issues with automated metrics ...................... 70
      4.5.2.7 “Refined” Refiner ................................... 71
      4.5.2.8 User model .......................................... 74
   4.5.3 Maximizing number of rules ............................... 78
   4.5.4 Constructing clusters ..................................... 80
   4.5.5 Cluster Scoring ........................................... 81
      4.5.5.1 Coverage ........................................... 82
      4.5.5.2 User model .......................................... 82
   4.6 Creating summaries .......................................... 89
4.7 Conclusion ..................................................... 93

5 Testing the effectiveness of summary types: A user study 

5.1 Experimental Method ............................................ 96
5.2 Experimental results ............................................ 99
   5.2.1 User judgments/input .................................... 99
   5.2.2 Experimental effects .................................... 101
   5.2.3 Comparison with generalized user model ............... 112
   5.2.4 User qualitative judgments ............................... 114
5.3 Conclusion ..................................................... 115
6 Learning to predict user summary preference: A classification experiment
6.1 Classification task ......................................................... 116
6.2 Determination of feature set .............................................. 117
6.3 Calculating utility .......................................................... 119
6.4 Results .............................................................................. 120
   6.4.1 Features associated with the query/response pair ............ 122
   6.4.2 Features associated with the user model ....................... 123
   6.4.3 Combining features ................................................... 125
6.5 Conclusion ......................................................................... 129

7 Technology for creating associative summaries from an automatically annotated corpus of news documents
7.1 Experiments with human-annotated data .............................. 131
7.2 Automated identification of keyphrases ............................... 132
7.3 Automatically annotated news documents ............................ 133
7.4 Algorithm .......................................................................... 134
   7.4.1 Attribute selection/ranking ........................................... 134
   7.4.2 Pre-processing data/Creating database ......................... 136
7.5 Natural language experiment .............................................. 139
7.6 Natural language experiment results .................................... 141
7.7 Alternative method of producing clusters in the news domain
   7.7.1 Algorithm for clustering text data ................................. 143
   7.7.2 Predictive Apriori ....................................................... 144
   7.7.3 Hierarchical agglomerative clustering ............................ 146
7.8 Combining/pruning clusters ................................................ 151
7.9 Conclusion ......................................................................... 154

8 Testing the effectiveness of associative summaries: An experiment with a news archive
8.1 Experimental setup .......................................................... 157
   8.1.1 Background ............................................................. 157
   8.1.2 Experimental population/protocol ................................. 159
   8.1.3 Experimental setup ................................................... 165
8.2 Data analysis ....................................................................... 166
   8.2.1 User judgments/input ................................................ 166
   8.2.2 Expert judgment ....................................................... 168
   8.2.3 Examining subjects by performance ............................. 171
List of Figures

3.1 Box plot showing the scores for the three response types evaluated by users. Displayed are the median score, quartiles, and extreme values for each response type. .................. 44

4.1 Tree representation of association rules computed for restaurants in Boston, using the attribute cuisine to predict neighborhood. ......................................................... 50

4.2 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by information gain. The y-axis shows coverage and the x-axis shows the number of rules with that coverage. ........................................... 61

4.3 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by gain ratio. The y-axis shows coverage and the x-axis shows the number of rules with that coverage. ........................................... 63

4.4 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by forward stepwise selection. The y-axis shows coverage and the x-axis shows the number of rules with that coverage. ........................................... 66

4.5 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by backward stepwise elimination. The y-axis shows coverage and the x-axis shows the number of rules with that coverage. ........................................... 68

4.6 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by the delta in forward stepwise selection. The y-axis shows coverage and the x-axis shows the number of rules with that coverage. ........................................... 69
4.7 Bar graph showing the average coverage of rules derived using the top-2 attributes scored by the delta in backward stepwise elimination. The $y$-axis shows coverage and the $x$-axis shows the number of rules with that coverage. .................................. 70

4.8 Bar graph showing the average coverage of rules derived using the top-2 attributes scored using the modified Refiner algorithm. The $y$-axis shows coverage and the $x$-axis shows the number of rules with that coverage. ......................... 74

4.9 Tree representation of rules derived from three runs of c4.5 using a different attribute as the dependent variable. ............. 79

4.10 Graph showing the coverage of rules describing all restaurants in the London database as function of their utility scored for user $AF$. ............................................................... 86

4.11 Graph showing the coverage of rules describing all restaurants in the London database as function of their utility scored for user $EJB$. ............................................................... 87

5.1 An example of a webpage used for the second restaurant domain experiment. ......................................................... 100

5.2 Graphical representation of the average scores for summary statements grouped by attribute selection algorithm and summary construction method. ................................. 102

5.3 Graphical representation of the average scores for summary statements grouped by attribute selection algorithm, summary construction method, and setsize. ......................... 104

5.4 Graphical representation of the average scores for summary statements grouped by attribute selection algorithm, summary construction method, and constraint introduced by query. 106

5.5 Graphical representation of the interaction between the use of user model for attribute selection and the presence or absence of poorly rated attributes in the response. ................. 109

5.6 Graphical representation of the interaction between the clustering method used to construct the response and the presence or absence of poorly rated attributes in the response. ....... 110

7.1 Dendrogram computed on dissimilarities shown in Table 7.10. 149

7.2 Tree representation of dendrogram computed using dissimilarities in Table 7.10 ................................................. 150
8.1 The interface to the *Full Associations* interface, showing clusters grouped by semantic type.

8.2 The interface to the *Combined Associations* interface, showing all clusters combined.

8.3 The interface subjects used in the *Baseline* system to query the Ontotext search facility with their own search terms.

8.4 The interface provided by Ontotext to their search facility. Subjects used the interface shown in Figure 8.3 to query this system, to preserve the look-and-feel of the other interfaces.

8.5 The interface to the actual set of news articles found in the Ontotext archive. All three experimental interfaces ultimately showed pages that resembled this.
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Example menu-driven dialogue in a used car domain. Taken from [Meng et al., 1996].</td>
<td>4</td>
</tr>
<tr>
<td>1.2</td>
<td>Example dialogue system in a restaurant domain for Boston.</td>
<td>4</td>
</tr>
<tr>
<td>1.3</td>
<td>Belkin’s characterization of information-seeking strategies, taken from [Belkin et al., 1995].</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>An example of two types of summary responses possible for dialogue systems in the restaurant domain. In SunType1, attributes are mentioned individually, with values listed. In SunType2, attributes are mentioned as associations of their values.</td>
<td>7</td>
</tr>
<tr>
<td>2.1</td>
<td>Example menu-driven dialogue in flight domain. Taken from [Seneff and Polifroni, 2000].</td>
<td>35</td>
</tr>
<tr>
<td>2.2</td>
<td>Example dialogue using Refiner method for summary construction, taken from [Polifroni et al., 2003].</td>
<td>36</td>
</tr>
<tr>
<td>2.3</td>
<td>Example dialogue system in an air travel domain, tailored for a business user and taken from [Demberg and Moore, 2006].</td>
<td>38</td>
</tr>
<tr>
<td>3.1</td>
<td>Examples or responses tailored to the three personas used in the initial experiment.</td>
<td>42</td>
</tr>
<tr>
<td>3.2</td>
<td>Scores for the three response types judged by subjects.</td>
<td>43</td>
</tr>
<tr>
<td>3.3</td>
<td>Results as a function of persona assumed.</td>
<td>45</td>
</tr>
<tr>
<td>4.1</td>
<td>Influence of number of attributes used to construct tree and average size of the resulting leaf clusters. The numbers are averaged using leaf clusters from trees built using the entire set of 596 restaurants in our dataset.</td>
<td>56</td>
</tr>
<tr>
<td>4.2</td>
<td>Values for cuisine, price, and neighborhood for the example database.</td>
<td>58</td>
</tr>
</tbody>
</table>
4.3 Values for instance entropy, information gain, class entropy, and gain ratio for the example database used to illustrate these concepts. ................................. 59
4.4 Values for information gain computed using all exemplars in the London restaurant domain. ............................. 60
4.5 Values for gain ratio for all tuples in London restaurant domain. ................................. 62
4.6 Values for forward stepwise selection for all tuples in London restaurant domain. ................................. 65
4.7 Values for backward stepwise elimination for all tuples in London restaurant domain. ................................. 67
4.8 Example of refiner ranking computation for price and food quality in the London restaurant domain. The number in parenthesis next to the highest scoring values is the ratio of the number of tuples with that value to the total number of tuples. ................................. 73
4.9 Values for Refiner for all tuples in London restaurant domain. ................................. 73
4.10 User models for subjects AF and EJB. FQ = food quality, SVC = service, DEC = decor, Nghd = neighborhood, CUI = cuisine. ................................. 78
4.11 Parameters associated with, and example of a response using the Refiner method of attribute selection and singleton clustering. ................................. 81
4.12 Parameters associated with, and example of a response using the user model method of attribute selection and singleton clustering. ................................. 82
4.13 Parameters associated with, and example of a response using the Refiner method of attribute selection and associative clustering. ................................. 83
4.14 Mappings from symbolic values in the database used to describe records and scalar values used to score association rules. ................................. 84
4.15 Weights assigned to attributes based on their position in the user model ranking. ................................. 85
4.16 Weights assigned to attributes based on their position in the user model ranking. ................................. 86
4.17 Rules scored for utility using two user models from subjects AF and EJB. Included in the table are the utility and number of restaurants represented by each rule. FQ = food quality. ................................. 88
4.18 Example summaries for the four different experimental conditions.

4.19 Example frame used by Genesis to generate the response *I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.*

5.1 Example summaries for the four different experimental conditions.

5.2 Table showing the four responses to the initial query configured for and evaluated by user *EJB*, whose ranked preferences were, in order, *food quality, cuisine* and *neighborhood*.

5.3 Table showing a breakdown of the eight query types evaluated by subjects for the second restaurant experiment. The number in parenthesis indicates the number of restaurants in the database matching the query constraint. The responses could mention a varying number of restaurants, depending on the subset tailored to subjects’ preferences.

5.4 Average scores for summary statements grouped by attribute selection algorithm and summary construction method.

5.5 Average scores for summary statements grouped by attribute selection algorithm, summary construction method, and size of dataset described in response.

5.6 Table showing the average utility of responses to particular query types, divided by method used to select the attributes to speak about.

5.7 Average scores from users for responses grouped by query type, dataset, and the presence or absence of poorly rated attributes.

5.8 Table showing attributes ranked by average user model preference and the percentage of responses containing those attributes as part of their descriptive statements.
6.1 Table showing the initial query seen by user EJB, whose ranked preferences were, in order, food quality, cuisine and neighborhood.

6.2 Scoring of UM-Assoc response clauses shown in Table 6.1 for user EJB. avgUtility for this response was .544.

6.3 Scoring of Ref-Sing response clauses shown in Table 6.1 for user EJB.

6.4 Classification results showing effects of different combinations of features from the query in context set.

6.5 Classification results showing effects of different combinations of feature from the utility set. * indicates significant improvement over baseline (P < .005).

6.6 Cosine vector similarities between each of the 16 subjects in the restaurant experiment.

6.7 Classification results showing effects of different feature sets on predicting summary type. * indicates significant improvement over baseline (P < .05).

6.8 Highest scoring clauses showing contrast between UM-Assoc and Ref-Assoc responses for user NW.

6.9 Example classification rules from System 3 in Table 6.7.

7.1 Example frame-based representation of news story from the MUC-3 corpus. Attributes not represented in this particular template are not shown.

7.2 Example rules derived from the C4.5 tree-induction algorithm applied to the MUC templates. The number in parentheses indicates the number of templates accounted for by the rule.

7.3 Table showing search terms used to determine a profile for semantic types in the Ontotext news domain.

7.4 Profile of the semantic types for key terms found for the three story topics used.

7.5 Top-10 terms given to search engines at yahoo.com and news.google.com.

7.6 Example frame-based representation of news story found using the search term “Sarkozy.”

7.7 Example Predictive Apriori rule found for documents found searching on ’Nicolas Sarkozy’.

7.8 Binary representation of Predictive Apriori on documents found searching on ’Nicolas Sarkozy’. 
7.9 Example *Predictive Apriori* rules found searching on 'Nicolas Sarkozy' showing associations among key terms. . . . . . . . 147
7.10 Manhattan Distance computed on rules shown in Table 7.9. 148
7.11 Rule groupings derived from the tree shown in Figure 7.2, with the corresponding key terms found in the rules in Table 7.9. 149
7.12 Statistical profile of clusters found on the three topic types used in the experiment, showing the number in the full set of discovered clusters, and the number and percentage reduction after initial pruning of duplicate clusters and those that did not represent a document in the archive. . . . . . . . . . . . . . 152
7.13 Clusters of length 3 discovered using key terms found in stories on Nicolas Sarkozy. . . . . . . . . . . . . . . . . . . . 152
7.14 Profile of number of clusters displayed in the *Full Associations* and *Combined Associations* interfaces for the three topic types. The total number of unique clusters represented in documents in the archive is shown first. Then, for each interface, the total after pruning is shown, along with the number from the total that was obtained by either c4.5 decision tree induction or *Predictive Apriori* + hierarchical agglomerative clustering. The numbers in parentheses are the percent reduction in number of clusters as a result of pruning. . . . . . 155

8.1 The scores users assigned to each interface, for overall rank, confidence, and reuse. . . . . . . . . . . . . . . . . . . . . . . . . 166
8.2 The scores subjects assigned each story for overall rank, confidence, reuse, and familiarity. The order in the table reflects the order in which the subjects saw each story topic. . . . . . 167
8.3 Rankings from the expert judge for each interface, on usefulness, richness, and originality. . . . . . . . . . . . . . . . . . . 169
8.4 Rankings from the expert judge for each topic, on usefulness, richness, and originality. . . . . . . . . . . . . . . . . . . 170
8.5 Rankings from users, shown by groups of reflecting those whose angles were ranked highly by the expert judge and those that were ranked poorly. . . . . . . . . . . . . . . . . . . 172
8.6 Example comments from subjects for each of the three interfaces. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174
8.7 Subjects use of clusters, divided by technology used to discover them. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 175
Abstract

In this thesis, we examine algorithms for content selection and information presentation that enable *browsing* as an information-seeking strategy. Our hypothesis is that users who wish to explore the contents of a particular domain might not be satisfied with traditional spoken dialogue systems, in which focal information is not described until a small number of the records comprising the domain are identified through a constraint-elicitation process. Specifically, we hypothesized that users would prefer responses in the form of incremental summaries of the focal information.

We developed a set of algorithms that automatically determine underlying parameters for attribute ranking/selection, maximizing for generalizability and/or user preference. To highlight relationships between attribute/value combinations, we employed decision tree induction, as well as a simpler algorithm that lists values for attributes by frequency.

We test responses whose content and presentation mode were automatically selected and/or configured in experiments eliciting user judgements in overhearer mode. When attributes are selected for their ability to generalize, we found that users prefer to hear them described individually. When attributes are chosen based on individual preferences, users ranked most highly responses that describe associations among attributes. We built a classifier, based on features from the query in context and the query’s utility, to predict response type and were able to achieve accuracy significantly above the baseline.

To test portability, we applied the algorithms to a corpus of news articles that had been automatically annotated for keyword and named entity. Evaluation was task-based with journalists as subjects, using the technology to search a news archive for background information to support a real breaking news story. The interfaces to the news archive included two that displayed discovered associations of key terms and one, Google-like baseline. Experimental results showed the effectiveness of the associative summaries.
Acknowledgments

I’d like to thank my advisor, Lyn Walker, whose comments and insights have greatly contributed to this thesis. It was her idea to this many years ago and it would not have happened without the impetus she gave me.

In Sheffield, I’m indebted to Hamish Cunningham, who was instrumental in setting up the collaboration with Ontotext that enabled a great deal of the work described here. I hope I’ve held up my end of that bargain. I’d also like to thank Hamish for his friendship and support in many other ways. I’m still not a big fan of beer, but I will greatly miss all that goes with going out for a pint.

I’d like to thank my panel members, Roger Moore and Rob Gaizauskas, for their helpful and encouraging input. They were always available, interested, and full of good advice. My conversations with both helped my understand the issues better and, I hope, contextualize them.

The experiment with journalists would not have happened without Emma Barker, to whom I’m deeply indebted and whose help I greatly appreciate. This work has benefited greatly from the conversations we had about evaluation and the insights she provided. Emma’s contacts with journalists were invaluable and her enthusiasm for the work kept me going. I will miss our collaboration.

At Ontotext, I’d like to thank Borislav Popov, Ilian Kitchukov, and Atanas Kiryakov. I can’t say enough good about Ontotext and the people there. Their technology is great, and I am deeply appreciative of the level of responsiveness and support I got from them. I’m happy to say that my findings show the effectiveness of their interface.

I’d like to thank Jonathan Foster, who served as the expert judge for our experiment with the news domain. Jonathan’s expert skills helped inform much of our understanding of how this technology should (ideally) be contextualized for journalists. I’ve learned a great deal from him.

I’d like to thank Mark Hepple, Ben Allison, Mark Sanderson, Yorick
Wilkes, Louise Guthrie, and Steve Whittaker for conversations that greatly helped me clarify issues. Thanks to Mark Greenwood for invaluable help in setting up the user experiments and webpages. Thanks to Cristi Ursu for the formatted Zagat data used in the restaurant experiments. Thanks to Sanaz Jabbari for always being interested; there were many times when that meant a lot. Thanks to Stu Wrigley for being a great officemate, and to Fleur Wrigley, for being a great baker. It helped make me feel at home here. Thanks to Louise Guthrie for being a good friend, for schlepping me around, and for showing me that some games can be fun. And thanks to Wim Peters, who always left me feeling happy that I had come to Sheffield just so I could get to know him.

In Boston, I want to thank Caroline Huang, Mike Phillips, Lee Hetherington, Sarah Hetherington, Judy Adler, and Jeff Marcus, for never letting me forget that I had deep roots there. Thanks to Caroline, whose phone calls always made me feel like someone at home was making sure I was okay. Thanks to Caroline, Mike, Lee, and Sarah, for never letting it get too long before pinging me, even when I was unresponsive, and for much practical help in living in two countries. I also want to thank Robert Phillips and Tommy Phillips, who always reminded me that I had something I wanted to go back to.

I’d like to thank Jim Glass, for giving me a place to work at MIT and for his advice along the way. Thanks to Chao Wang, who was always enthusiastic about what I was doing, something that often gave me an extra shot of energy when I needed it. Thanks to Mark Adler and Jamey Hicks, for being patient; I’ll make it worth it.

Finally, I’m deeply indebted to Victor Zue and Stephanie Seneff, for so many things. It was Victor and Stephanie, along with Mike Phillips, who introduced me to spoken dialogue systems many years ago. I couldn’t have predicted the path from there. Victor and Stephanie have been professional mentors, advisors, fonts of knowledge, and among the greatest friends anyone could imagine. They provided me with an ongoing career, and they threw in a home for me in Boston over the last three years. I will always be grateful to them.
Chapter 1

Introduction

1.1 Motivation

One of the most challenging problems in dialogue system design is the development of general algorithms for information presentation. This challenge arises from the fact that, in most dialogue states, the system can only provide a description of a subset of the information in focus. Furthermore, this subset must be selected and presented using algorithms that balance the user’s cognitive limitations on processing information, against the user’s need for enough high quality information to make progress on the task. In this work, we explore the hypothesis that summaries of the subset of the information that is in focus at any given point are a critical dialogue strategy for information presentation in dialogue systems. We will examine technology to enable browsing of information, a term we use in the sense of “semi-directed or semi-structured searching” [Ellis, 1989].

To formally define some terms used throughout this work, let $D$ be a domain comprised of a set of records $\{r_1, ... r_n\}$. These records make up what is often called the “backend data” for a dialogue system. Each record is described by a set of attributes $\{A_j, ..., A_n\}$. Attributes have associated values $v$ that are particular to a domain $D(A_i) = \{v_{i,1}, v_{i,2}, ..., v_{i,n}\}$.

In a dialogue system, a constraint is a value introduced by a user with an associated attribute either explicit or implied. A constraint $c$ is a function over records in $D$ such that $c_j(R)$ returns a record $r$ if $r \subseteq D$ and $r : A_i = c$. The set of records $R$ that satisfy all dialogue constraints $\{c_i, ..., c_n\}$ is the focal information at any point in the dialogue. If $Q_j$ is a query instantiating one of more constraints on attributes in $D$, the focal information $R$ is the extension of $Q_j$ in $D$.

A typical attribute in an information retrieval-type dialogue system would
be *cuisine*, the food type associated with a restaurant in a restaurant domain, or *airline*, the air carrier associated with a particular flight in an air travel domain. Values for these attributes include “French” or “Italian”, in the case of cuisine, or “British Airways” or “Ryanair”, in the case of airline. A user utterance instantiating a constraint on cuisine, e.g., “I’m interested in Chinese food”, would result in a set of records corresponding to restaurants that serve Chinese food, the extension of that query. One way to respond to such an utterance is to list each item in the extension individually. If the set is quite large, this is difficult. A response that describes the extension at a level of abstraction, e.g., highlighting relationships across attributes, would comprise an *intensional* response, e.g., “There are 35 Chinese restaurants in Leicester Square”. This work investigates automated means of discovering intensional answers.

Enabling these types of responses in dialogue systems would be a departure from the traditional paradigm followed in building such systems, as exemplified by the DARPA Communicator Project [Walker et al., 2002b] and incorporated into toolkits using VXML [Voxeo, 2003]. Traditionally, dialogue systems are designed within a menu-driven paradigm, where the system elicits a series of constraints from the user in a predefined order. Table 1.1 shows an example of one such system, developed at MIT and used as an interface to a database of used cars. Answers that contain information other than a brief numeric summary of the number of cars that match a set of constraints are delayed until a small subset of the data has been identified [Meng et al., 1996].

Many systems, especially those developed in industry, require users to provide the requested information before they will move on to another query. Some systems will allow users to deviate from the requested response, but usually only by providing another constraint or changing an existing one (as in the utterance marked *User4* in Table 1.1). What is common across all systems is that they typically do not provide any description of focal information until the user has sufficiently narrowed the choices to a number that can be easily described in some detail.

Aside from delaying the ability to learn about the domain, another pitfall associated with dialogue systems driven by constraint elicitation is exemplified in Table 1.2. This system shown here, also developed at MIT under the SpeechBuilder framework [Glass et al., 2004], provided an interface to a database of restaurants in Boston. The dialogue illustrates the problem users
Table 1.1: Example menu-driven dialogue in a used car domain. Taken from [Meng et al., 1996].

| System1: Welcome to the Electronic Automobile Classifieds. What car would you like? | User1: Can you show me some convertibles? |
| System2: There are 208 convertible cars. Can you provide a make or a make and model? | User2: Red ones. |
| System3: There are 39 red convertible cars. Which particular model are you interested in? | User3: How about black ones? |
| System4: There are 41 black convertible cars. Can you provide a make or a make and model? | User4: Any color |
| System5: There are 208 convertible cars. Can you provide a make or a make and model? | User5: A Corvette. |
| System6: There are 8 corvette convertibles. A price range would be helpful. | User6: Between twenty and thirty thousand. |
| System7: There are 4 Corvette convertibles costing between 20,000 dollars and 30,000 dollars. The first one is a.... |

Table 1.2: Example dialogue system in a restaurant domain for Boston.

| User1: Tell me about restaurants in Boston. |
| System1: What kind of cuisine are you interested in? |
| User2: Chinese. |
| System2: What neighborhood would you like? |
| User3: The North End. |
| System3: Do you have a preference for price range? |
| User4: I’d like something cheap. |
| System4: I’m sorry. I’ve found no cheap Chinese restaurants in the North End. |
face when generalizations about the data are not forthcoming from a system, and users are able to pursue paths that ultimately lead to presupposition failure. After the user requests Chinese food at utterance User2 in Table 1.2, a helpful system might have pointed out that 88% of Chinese restaurants that the system knew about in Boston were in the Chinatown neighborhood. Another helpful piece of information could have been provided after utterance User3, where the constraint on neighborhood is introduced. In this particular database, 89% of the restaurants in the North End serve Italian food. Instead, the system is configured simply to elicit constraints in a default order.

Dialogue systems designed in this way fail to address the needs of their users as consumers of information. Belkin [1995] characterizes information seeking strategies along four dimensions, based on his own research and that of others [Belkin et al., 1993, Ellis, 1989]. Each of the four dimensions is characterized schematically by a continuum of behavior or information, with the ends defined by contrasting types. Table 1.3 shows Belkin’s schema.

Menu-driven dialogue systems address only one part of the space of possible information-seeking strategies. Referring to the schema in Table 1.3, a typical menu-driven dialogue system would be most useful for a user who is searching (Mode of Interaction) for specific information (Goal of Interaction), and is happy recognize (Goal of Interaction) and specify (Mode of Retrieval) that information from a list of. Such a Power User would have little need for exploring the focal information, as she knows what she is looking for and can proceed straight to it. However, Belkin claims that, during an information-seeking session, users may also scan for interesting items in

<table>
<thead>
<tr>
<th>Method of Interaction</th>
<th>Goal of Interaction</th>
<th>Mode of Retrieval</th>
<th>Resource Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>scanning</td>
<td>learning</td>
<td>recognition</td>
<td>information</td>
</tr>
<tr>
<td></td>
<td>searching</td>
<td>specification</td>
<td>meta-information</td>
</tr>
</tbody>
</table>

Table 1.3: Belkin’s characterization of information-seeking strategies, taken from [Belkin et al., 1995].
the database, learn about the structure of the database, retrieve items by recognition rather than by specification, and use meta-level information to understand the structure and content of a database. This Browsing User would benefit from a spoken dialogue system that could provide summaries of information to enable the verbal equivalent of browsing in an interface where there is no visual component [Kamm et al., 1997]. Our hypothesis is that presenting information in stages throughout the dialogue in the form of summary statements is one way of addressing the needs of the Browsing User.

In a real-world setting, Solomon studied the behavior of users engaged in a variety of information-seeking tasks, including air travel planning [Solomon, 1999]. He defines a notion of “roundness”, which refers to the strategies individuals employ to seek out and contextualize information they use in their daily lives. According to Solomon, users construct “small information world[s]”, populating them by learning about the topic they are interested in and collecting data to make informed decisions. For example, the users he studied in the air-travel domain sought to understand what their options were before committing to a particular choice. A system that provides a mechanism for summarizing those options would fit into such a model of information seeking.

Instead of dialogue systems designed along the lines of those shown in Table 1.1 and Table 1.2, imagine that dialogue systems were capable of producing the sorts of summary responses as shown in Table 1.4. These types of responses give the user a glimpse into the contents of the domain data, as well as highlighting some possibly salient attributes and values. Such responses address the needs of Browsing Users.

The expansion of the definition of a dialogue interface to include the ability to summarize data at a meta-level makes sense intuitively. Research in both cooperative answering (see Section 2.2.1) and browsing as an information-seeking strategy (see Section 2.1) has supported this idea for many years. More recent work in Knowledge Discovery from Databases (see Section 2.2.2) suggests that detecting emergent patterns in data is an important aspect of systems that support search.

Ideas from as far afield as evolutionary ecology support the idea that human beings employ browsing-like behaviors when seeking information. The theory of information foraging [Pirolli and Card, 1999] is based on the assumption that humans employ the same mechanisms when searching for
User1: Tell me about restaurants in Boston.

SumType1: I know of 600 restaurants in Boston. Many of them are in the North End, Chinatown, and Back Bay. Some of cuisines served are Italian, Chinese, and American.

SumType2: I know of 600 restaurants in Boston. There are 40 medium-priced Italian restaurants in the North End and 30 inexpensive Chinese restaurants in Chinatown.

User2: Chinese.

SumType1: I know of 40 Chinese restaurants in Boston. Most of them are in Chinatown. Many of them are inexpensive.

SumType2: I know of 40 Chinese restaurants in Boston. There are 25 inexpensive ones whose food quality is not very good. There are also 15 medium-priced ones with highly rated food quality.

User3: The North End.

SumType1: I know of 85 restaurants in the North End. Many of them serve Italian food. All price ranges are represented.

SumType2: I know of 85 restaurants in the North End. There are 34 medium-priced Italian restaurants with good food quality and 20 expensive Italian restaurants with highly rated food quality.

Table 1.4: An example of two types of summary responses possible for dialogue systems in the restaurant domain. In SumType1, attributes are mentioned individually, with values listed. In SumType2, attributes are mentioned as associations of their values.
information that our animal ancestors used to find food. Humans detect the “scent” of information as they pursue paths, for example, while web surfing. The strength of the scent is used to evaluate the efficacy of the path and determine whether to continue or abandon one particular path in favor of another. As applied to web surfing, the strength of the information scent can be evaluated by scanning information on the page. When there is no visual interface, or when that visual interface does not contain enough cues to adequately assess the information scent (e.g., a map that simply shows dots indicating locations of restaurants), other mechanisms are necessary to select information that could potentially be useful to “informavores” [Dennett, 1991]. Our primary hypothesis can then be reformulated within the information foraging framework as the claim that summaries help informavores in the assessment of the information scent.

The growth in the number of information sources and the improvement in technology for extracting information is creating an enormous potential for dialogue systems. But traditional menu-driven approaches to dialogue management are severely constrained and possibly not sustainable. (For example, if a complete, or near-complete database of London restaurants were available, it might not be possible to narrow restaurant options to a small enough set to present to a user.) Much of the current research in this area has been aimed at developing schemas and toolkits to facilitate the development of spoken dialogue systems, or on learning dialogue strategies from a corpus of real or simulated interactions (see Section 2.5). Little work has been done on content selection for information presentation, an equally important aspect of dialogue management. This is an area where further research is needed, both in the underlying technology and in evaluating the effectiveness of the resulting dialogues.

This thesis frames the issues in dialogue management within the scope of general information-seeking strategies. In doing so, we explore algorithms for content selection and presentation that are lightweight, portable, and adaptable to user-imposed constraints, such as pre-existing preferences in certain contexts. We hope by doing so to help further the notion of human-machine dialogue as a collaborative activity.

Our work was initially motivated by work on dialogue systems providing interfaces to structured data, i.e., data that is represented in tabular, or attribute/value, format. The records in these databases are typically well-structured, and often correspond to generalizable semantic types, such
as location (i.e., city, neighborhood, etc.) or price_range (i.e., inexpensive, medium-priced, expensive). Our algorithms will first be tested on these types of data. In addition, we look at data that is derived from content that is unstructured, i.e., free-text found on the web. For this, we will make use of information extraction technology to derive a structured database, upon which we will apply our algorithms. Although these data eventually take on a structure, this structure is induced automatically, and semantic classes corresponding to entities within the structured database can be absent.

1.2 Hypotheses

Our primary hypothesis (H1, below) can be formulated in the claim that users prefer incremental summaries in dialogue systems. We test this hypothesis, and a series of others that relate to it, in the work described here. To begin with, we examine H1 and H1a, the latter designed as a refinement of the “one size fits all” nature of the primary hypothesis, to see if there are particular circumstances under which users prefer summary responses.

**H1** Users prefer incremental summaries in dialogue systems. (Chapters 3, 5 and 8)

**H1a** Users prefer summary type responses when they are unfamiliar with a domain. (Chapters 3 and 5)

Previous research in fields such as Cooperative Answering (see Section 2.2.1) have proposed algorithms that involve hand-coding of system parameters to generate specific types of content-rich responses. We feel it is important that content selection and information presentation be performed automatically using methods relating to data analysis. **H2** tests this claim.

**H2.** Algorithms for selecting content and presenting information in summary responses can be automated. (Chapters 4 and 7)

The following two hypotheses examine content selection in summary responses, testing two different ways of selecting attributes, and the claim that a user’s knowledge of the domain affects the choice of an optimal response.

**H3a.** When users are unfamiliar with a domain, the optimal summary type will describe attributes that generalize well across the focal information, irrespective of dialogue context. (Chapter 5)
H3b. When users have expressed preferences about attributes in a domain, the optimal summary type will describe attributes that conform to those preferences, irrespective of dialogue context. (Chapter 5)

Information presentation is addressed in hypotheses H4a and H4b, by which we test two automated ways of organizing summary information.

H4a. The optimal summary type will describe attributes singly. (Chapter 5)

H4b. The optimal summary type will describe attributes in association with other attributes. (Chapter 5)

We examine context as a factor in content selection in hypotheses H5a and H5b, testing intuitions relating to the amount of information being described and how that affects user preferences for the attributes in the response.

H5a. Large sets of focal information will be best described by attributes chosen because a small number of their values describe a large amount of the information, leading to summaries such as “Many of the restaurants are in Soho” or “Most of the articles mention lawsuits”. (Chapter 5)

H5b. Small sets of focal information will be best described by attributes that conform to user preferences. (Chapter 5)

If it is true that attributes are best chosen by their ability to generalize, the specific values of the attributes should be unimportant. With H6, we examine that claim, to see if the semantics of the attribute values themselves are also important.

H6. When choosing an attribute to use to describe a set of focal information, the particular values of that attribute are not as important as the ability of those values to cluster and generalize. (Chapter 8)

With hypothesis H7, we move away from algorithmic considerations of content selection and information presentation, to examine the claim that the parameters associated with response construction can be automatically selected in context.
H7. Given multiple ways of selecting attributes and creating summaries, a classifier can predict the optimal attribute selection/presentation algorithm. (Chapter 6)

Hypotheses H8 through H10 focus on claims relating to the portability of the technology developed in this work. H8 directly tests the portability of the technology by applying it data whose structure is induced by information retrieval techniques. To test H9, we look at two competing algorithms to create summaries in the new domain. Finally, with H10, we once again test a hypothesis related to content selection, to see if there are algorithmic ways to perform attribute selection in this new domain.

H8. Useful and informative summaries can be created when focal information is unstructured, i.e., a news archive. (Chapters 7 and 8)

H9. Real-time technology can be used to create summaries. (Chapter 7)

H10. The topic of the search will influence the semantic type of the search terms used in a news archive. (Chapter 8)

1.3 Approach and Methodology

In examining ways of selecting and automatically processing content for information presentation, we investigate the technical challenges involved in finding automated methods, the use of these methods in the construction of summary statements, and user evaluations of the resulting summaries. Our chosen method of presentation involves summarization based on attributes and values within the focal information, resulting in an intensional-type summarization (see Section 2.2.1). We choose this type of summarization strategy because it fits in with our desire to enable browsing. Furthermore, constructing a more detailed, informative-type summary typically requires a level of natural language processing that would be difficult to achieve in real-time or near real-time in a dialogue system.

We decompose the technical challenges into the determination of which attributes to summarize and how to perform the summarization. We examine algorithms for attribute selection, and methods for discovering relationships among attribute values. We test known methods from the literature, as well as original ones developed for this work. We investigate means of selecting attributes based on data analysis and also based on user preferences. We
also employ known scoring methods to novel data representations to determine content selection. These studies are done using a domain of restaurant data for London.

Once a set of algorithms for content selection and information presentation have been determined, we test these by eliciting user judgments on the information presented. User evaluations are elicited to determine whether (1) such summary statements are indeed preferred over other, more traditional response types; and (2) what makes for the most effective summary response. We then build a classifier to predict the type of summary response to use, based on data from user experiments.

We next port the technology to a domain of unstructured news articles, using automatically discovered keywords and named entities to characterize and build associative summaries. In addition, we implement a different set of algorithms, used in related work, to derive equivalent summaries in this domain and compare the use of each type of summary in user studies. These same user evaluations are also used to determine the effectiveness of the resulting summaries.

1.4 Contributions

We make contributions in the areas of algorithms and technology, applications, and evaluation, as they relate to the creation and use of summary statements. In the area of algorithms and technology, we develop and/or reuse existing technology to enable browsing, in the following ways.

- We make use of a novel, iterative method of creating association rules to discover a rich set of relationships in data.

- We formalize an algorithm for choosing domain attributes based on their ability to generalize.

- We employ a user model in a novel way to determine the “interestingness” of automatically acquired association rules.

- We explore the use of classification to predict the type of summary response to use in context.

- We test the usefulness of two competing technologies for summary construction in the news domain.
In the area of applications, we implement the technology in two domains chosen to demonstrate the applicability of the technology with very different corpora. The two domains are:

- a structured corpus of information about restaurants in London, derived from formatted web pages;
- an unstructured corpus of news articles, automatically annotated for keywords and named entities.

In treating the automatic annotations as domain attributes, we create a novel source of data for dialogue systems.

We employ three user studies to test the technology and underlying assumptions about its usefulness in context by eliciting user preferences for:

- responses that summarize focal information in dialogue systems vs. responses that elicit further constraint preferences;
- the specific type of summary responses created using the algorithms we developed;
- responses that summarize focal information in a news domain.

In the evaluation in the news domain, we create a novel methodology based on the professional needs of journalists and journalism students, who form the subject pool of the study. We also make use of expert judgments from a member of the academic staff of the Department of Journalism Studies at the University of Sheffield to evaluate the information from the user studies.

### 1.5 Organization

In Chapter 2.7 we review some of the previous work in various fields as it relates to the central work presented here. Chapter 3 describes a proof-of-concept experiment designed to determine if our primary hypothesis holds, i.e., that users prefer summary responses in interactive systems over constraint-elicitation prompts. In Chapter 4, we describe the development of technology to construct these summaries in the London restaurant domain. The domain data were obtained from structured sources on the Web. Chapter 5 describes a further experiment in the London restaurant domain to determine what types of summaries are the most effective. Using data collected in Chapter 5, we build a classifier described in Chapter 6 to predict the
correct class of response in context. In Chapter 7, we describe technical issues involved in porting the technology described in Chapter 4 to a corpus of news documents. The annotations for these documents were automatically derived, and thus the structure of the domain data were induced. We discuss the creation of the database and the modifications needed to successfully port the technology to a corpus of less structured data. Chapter 8 describes a user experiment designed to test the efficacy of associative summaries in the news domain. Chapter 9 revisits the hypotheses, concludes, and gives ideas for future work.
Chapter 2

Previous Work

Dialogue systems themselves are hybrids, combining a range of input and output modalities, each with its own underlying technology. The work here combines insights from multiple fields, as well, in an effort to find a new way to produce responses for dialogue systems. In the following sections, we describe some of the work in these fields, beginning with research that helps frame our definition of the term \textit{browsing} (Section 2.1). In Section 2.2, we look at what it means to provide a response to a query, what constitutes a cooperative answer, and computational models for doing so. Section 2.3 examines work done on automatic summarization, as a mechanism for information presentation in a compressed format. User modelling, as a means of tailoring information to the needs and preferences of specific users, is described in Section 2.4. In Section 2.5, we look at the concerns and issues central to the field of dialogue management. Finally, in Section 2.6, we look at previous work specifically aimed at constructing informative responses in dialogue systems.

2.1 Browsing as an information-seeking strategy

Our primary hypothesis, that users prefer incremental summaries in dialogue systems, finds support in research carried out to discover how users browse for information. This research has been conducted to determine not only what strategies are used when performing the task [Ellis, 1989, Belkin et al., 1995] but also what resources aid the task itself [Cockburn and Jones, 1996, Catledge and Pitkow, 1995, Tauscher and Greenberg, 1997, Makri et al., 2007].

Before the advent of the Web, research into browsing was most often characterized by attempts to describe its underlying motivation and intent.
As we have seen, Belkin [1995] described the activity along four dimensions, representing the methods, goals, modes of retrieval, and type of resource used when browsing (see Table 1.3). For Belkin, browsing is inherently interactive and dynamic. Users' requirements, and even the strategies they employ, will change over time and as they learn more about the topic of their search. The development of systems that enable users to learn about topics motivated our work here.

Belkin sees the interaction between the user and the system as a dialogue even when the interaction is not carried out via natural language. He claims there is a need for systems to provide guidance to users in helping them pursue their information-seeking goals. A useful function the system can play, according to Belkin, is the identification of appropriate ways of continuing the dialogue. Our hypothesis is that one way of doing this is by pointing out the sub-set of attributes that characterize the data in each dialogue state.

Bates [1989] uses the metaphor of “berry-picking” to describe browsing, a way of describing the way users move through the information space, acquiring pieces of information as they progress. Bates claims that the process itself necessitates a level of imprecision on the part of the user about the ultimate outcome of the search. This would argue for a system that has some ability to describe itself to users, as opposed to one in which user goals are assumed to be set a priori. For Bates, the “quintessential form of browsing” is something she calls “area scanning” in which a user may ignore the formal organization of information, but seek to impose an order of his/her own by finding the relatedness of information.

In analyzing how users interact with data retrieval systems, the importance of relationships discovered within data has been noted. Dworman cites the need for systems that can discover emergent patterns in data [Dworman, 1995]. His work focuses on data in document collections, for which he advocates the use of machine learning algorithms, along with automatic annotations, to learn about and present thematic groupings to the user. His work does not advocate any particular technological approach to the problem, whereas ours is grounded within a specific set of machine learning algorithms.

Other research has taken a more behavioral approach to the question of how users browse, by actually studying users in the process and querying them afterwards about their experience. Rice [1988] points out the importance users ascribe to “serendipity”, i.e., the discovery of unexpected infor-
ation when performing a search. Users may have predetermined goals, but are happy to allow their search to follow alternate paths when they find, or are presented with something unforeseen. Marchionini et al. [1993] found that users turn to electronic forms of data when they are in a hurry and, furthermore, when they have an unclear formation of their plans. He further claims that a system that provides overall insights into the contents of a dataset will help users achieve their goals faster.

In a separate study, Marchionini [1997] again analysed the way in which users interact with electronic media and concluded that the ways users browse in such environments has specific characteristics that require new types of interfaces. He claims that electronic environments blur the distinction between individual sources of data (e.g., a single document) and collections of data (e.g., a data archive), and that interfaces are needed that enable “seamless browsing” within and across individual data sources. Choo et al. [2000] make a stronger claim, that browsing can only take place once the relationships have been imposed on a dataset, allowing a user to see groupings.

Based on an investigation into the actual behavior of users in information seeking tasks, Makri et al. [2007] came up with a series of recommendation for information retrieval systems, among them being giving users the ability to see and browse meta-level information. Jones [2006] points out the importance of facilities for annotating documents with, among other things, keyterms, to make it easier to retrieve specific items in later searches.

The research on information-seeking strategies lends support to the idea that interactive systems that can provide meta-level information, point out relationships, and construct content-filled responses serve the needs of a user population that may find other, constraint-elicitation approaches to interaction hard to navigate. The research also gives us a definition of browsing that we can apply to our work, i.e., the process of searching for information, in a semi-structured or unstructured way, that relies on an ability to inform oneself about the nature of the topic one is interested in and, possibly, learn something about how aspects of the topic are related. These insights helped inform and contextualize the work that follows.
2.2 Beyond data elicitation

2.2.1 Cooperative Answering

Our work is motivated by the claim that content-rich responses will help users make sense of focal information in dialogue systems. This claim has been explored in the field of Cooperative Answering, which takes as its goal the creation of systems that can provide answers in as human-like a way as possible. In that regard, it is informed by the spirit of the Gricean conversational maxims [Grice, 1972] governing the cooperative principles that can be brought to bear in a dialogue. Early work in this field focussed on the process of creating intensional answers, by which responses are generated that not only answer a specific question, but also have the capability to describe the schema of the underlying database and the relationships that hold among its components. Logical implicature and presupposition failure are two phenomena that are often treated in cooperative answering.

The initial research laid the groundwork for the claim that it is acceptable, and even desirable, for an answer to be comprised of more than simply a natural language representation of database tuples (or, even less helpful, a simple “yes” or “no”) [Joshi et al., 1977, Webber and Joshi, 1982, Siklossy, 1978, Cohen et al., 1982, Wahlster et al., 1983, Cheikes and Webber, 1989]. Early research focussed primarily on justifying the concepts themselves and outlining the information and relationships that would have to be encoded in order to provide intensional answers [Lehnert, 1978, Hirschberg, 1991]. The information used in this previous work was hand-coded and the algorithms for processing it was domain-specific. For the most part, the work was not framed within an overall system that could enable data collection, for example, to test the claims made about the generated responses. Furthermore, claims about the usefulness of intensional answers were not tested under experimental conditions, contrasting them with responses that made no attempt to analyze focal information. In our work, we seek to develop automated ways of constructing intensional responses and test them in experiments involving user judgements.

2.2.2 Knowledge Discovery from Databases

While the work in Cooperative Answers provides a justification for creating intensional answers, the field itself does not provide a great deal of insight into how that might be done in a portable or machine learnable way. For
that we turn to the fields of Knowledge Discovery from Databases and Co-operative Database Systems. It is here that methods for operationalizing the idea of browsing can be found.

Knowledge Discovery from Databases presupposes that knowledge is inherent in the structure of databases and it is important, furthermore, to discover this knowledge in a user-centered way. Bachman and Anand see database access systems as “knowledge discovery support environments” and claim that natural language generation is an integral part of any such system [Brachman and Anand, 1996]. Frawley et al. [1992] see a critical need for systems that can induce patterns from database content, especially as access to data grows. They include rule induction as one such way to induce these patterns. Fayyad et al. [1996a] speaks of the importance of making knowledge discovery understandable to humans, including the use of natural language generation to formulate output. In our work, we test rule induction as a way of constructing cooperative answers, using both natural language and tabular format, to see if users find the induced relationships helpful.

2.2.3 Cooperative Database Systems

The relational model of database management has been a success based in large part on the increase in computational power and memory. Database access languages such as SQL provide simple interfaces for querying and updating very large datasets quickly and efficiently. With the growth in the number of relational databases, there has been an increased interest in developing models and languages for describing these databases in more abstract terms. The ultimate goal of the work described in the following chapters is to produce such a description of focal information.

Two paradigms for organizing and accessing databases have been proposed: deductive and object-oriented. The former is based on first-order logic, and includes rules for inferring new facts from existing ones explicitly encoded in the database [Barja et al., 1995]. The latter paradigm is characterized by its ability to represent rich hierarchies in the database itself, describing relations among the classes stored in the database [Abiteboul, 1989, S-C. Yoon and Park, 1994]. In both types, information is encoded by hand and inferences depend critically on this hand-crafted information.

Yoon et al. [1997] looked at ways of using information encoded in an object-oriented database to derive intensional answers in database query applications. Their work consisted of a series of steps, each dependent on
a concept hierarchy encoded by hand and updated manually when any new information is added to the database. The data-mining techniques used were specific to the implementation and were dependent on traversing a hand-encoded hierarchy, searching for increasingly higher levels of abstraction at which to describe the data. The algorithm is of interest in that it is applied iteratively over the course of a series of user queries, somewhat akin to a dialogue system. It is applied at each “turn”, in an attempt to describe the focal information at any given point. The methodology, however, is hand-crafted and would not generalize easily to another domain. In fact, it seems as though the process of porting to a new domain would require a completely new hierarchy, derived manually, as well as new rules for traversing the hierarchy. In contrast, as we claim in hypothesis H2 in Section 1.2 and describe in more detail in Chapters 4 and 7, it is possible to fully automate the methods for deriving such information.

In his survey of cooperative database systems, Motro looks at two types of queries: those that result in a database response that can be supplemented with additional knowledge and those that result in an empty response (i.e., no entries in the database satisfy the user constraints) [Motro, 1996]. For both, a cooperative answer is one that enables a user to browse the data for additional information. In addition, Motro specifically mentions the type of user who may wish to explore the contents of a database without a clear goal in mind.

Gaasterland et al. [1992a] expands the idea by defining an intensional answer as one that describes either the complete set of possible answers, or a subset of the possible answers. For this, they propose the identification and selection of high-level rules that describe the data at an abstract level of generalization. This same idea is presented within the context of a logic-based database access schema [Gaasterland et al., 1992b, Godfrey et al., 1994].

Han et al. [June, 1996] propose a model for intensional answering that is based on what is termed “knowledge-rich” databases. These databases make use of various algorithms to process existing data for the acquisition of knowledge at an abstract level. This processing allows for intelligent generalization and summarization of data. The techniques described in [Han et al., June, 1996] involve careful annotation of database entries, as well as complex ontologies describing the relationships among database attributes. The same holds true of other methods proposed for deriving intensional
answers [Acar and Motro, 2005, Motro, 1989, Benamara, 2004, Necib and Freytag, 2003]. Even the more automated methods [Shrum and Muntz, 1988] still depend on data that is annotated by hand. Our goal is consistent with that of the research done in cooperative database systems, and the focal information used to create responses is derived from a set of database records. Our work goes further by developing automated metrics for content selection and information presentation based on measures of generalizability and utility.

2.2.4 Data visualization

Our claims about the ability to perform content selection in a principled way have been explored to some extent in work done in Data Visualization. Early work in this field was concerned with selecting subsets of data for display to users. The purpose was to focus users on particular aspects of a problem, rather than providing complicated and dense representations of all the data in question [A. Buja and Stuetzle, 1991, Buja et al., 1996]. Later work treats the problem of data visualization as one of data summarization [Lesh and Mitzenmacher, 2004]. Logical subsets within the data are discovered, as in earlier work, but the emphasis in [Lesh and Mitzenmacher, 2004] is in the presentation to the user, including an iterative interaction via a GUI that seems very like a dialogue.

Selfridge, et al. [1996] describe a prototype environment, called IDEA, which stands for Interactive Data Exploration and Analysis. This system is designed to analyze business data and provide a user interface that promotes exploration of that data. The system requirements include functionality common in most dialogue systems, including the ability to understand user queries, the ability to filter based on those queries, and the ability to remember discourse context across multiple utterances. The analysis of the data is tailored towards business applications and the capabilities are ones that are possible via SQL queries, e.g., counting and averaging. There is no natural language component to this system. The authors claim that data analysis and data exploration should be coupled with an interactive component that provides insight for the user into focal information. The concern with the analysis of focal information for the purpose of aiding end users is one that we share.
2.3 Summarization

If the \textit{Incremental Summary Hypothesis} described in Section 1.2 is correct and a viable dialogue strategy consists in presenting topical descriptions of focal information, one area to investigate for ways of creating such descriptions is the field of automatic summarization. Automatic summarization is typically applied to document corpora, something we will investigate in Chapter 7. In describing how automatic summarization is performed, the distinction is made between \textit{informative} and \textit{indicative} summarization.

Informative summaries typically process the set of documents in question more deeply, with the aim of finding representative sentences from the text to use in the summary (also called \textit{extractive} summarization [Chen and Chen, 2008]), or of constructing a summary statement using a natural language generation component (also called \textit{abstractive} summarization). Cues for finding sentences that are relevant to the topic include word frequency, the position of the sentence within a document/paragraph, and certain cue phrases [Hovy and Lin, 1997]. Barzilay [1997] used information from WordNet, along with a shallow parse of the input documents, to construct \textit{lexical chains} that define semantic relationships among words. Sentences identified via heuristics applied to the lexical chains were used for summarization. In other work, target text is parsed and mapped to a predicate-argument structure. The predicate-argument structure is converted using hand-crafted rules to a functional representation that can be used to generate text [Barzilay et al., 1999].

It is important to note that the goal of informative summarization is often to replace the need for a reader to consult the original text [Mani, 2001] or to provide insights into information contained in a set of documents that might not be present in any one source text [Radev and McKeown, 1998]. For this to be possible, a rich semantic representation of the set of documents is necessary. Radev and McKeown [1998] use MUC-style systems to interpret their text, augmented with hand-coded templates for specifically targetted ancillary documents. Other informative summarization algorithms also make use of hand-crafted rules and full Penn Treebank-style parses [White et al., 2001].

Indicative summaries, on the other hand, are typically generated using information from structured data about a set of documents [Kan et al., 2001b, Hatzivassiloglou et al., 2001], such as tables of contents, headings, and tabular information within the text [Kan et al., 2001b]. Metadata as-
associated with, but not necessarily part of the documents, can also be used in constructing indicative summaries and include information such as title, author, and date of publication.

Where annotated bibliographies corresponding to documents are available, they have been used to derive metadata [Kan et al., 2002]. Kan et al. additionally construct topic trees from headings and subheadings within a document [Kan et al., 2001a]. All these techniques rely on a particular structure in the source documents (e.g., headings and subheadings) or on ancillary data that may not be available (e.g., author in the case of a newswire article). This limits the algorithms for summary construction to domains such as scholarly articles, where a certain structure is present and can be exploited.

Sparck Jones [1998] speaks of the importance of “shallow processing” techniques for summarization, such as the use of keywords for indicative summarization. She argues that emulating human summarization is not possible currently or “even in the medium to long term” and, therefore, advocates the use of technology that is currently available. This is not simply a short-term solution, either. Sparck Jones also speaks of the importance of taking the needs of the user into account when creating a summary. When those needs are for information at a coarse-grained level, summaries that make use of information such as keywords may be completely adequate for the purpose. We examine this claim in some detail in this work, testing it in a task-based evaluation scenario.

A more lightweight approach to indicative summarization can be found in the use of keywords to enable browsing a set of documents. Keywords were used by ScatterGather [Cutting et al., 1992] as a way of clustering documents (as opposed to summarizing their contents), and authors make the claim that the keywords themselves are a form of summarization. More recently, Witten and colleagues have developed the KEA system for automatic keyphrase extraction [Witten et al., 1999]. This, along with named entity recognition, enables the identification of a number of possibly important keyword and keyphrases within documents. (See [Cunningham, 2005] for a survey article on information extraction, including named entity recognition.)

In our work, we develop and test a lightweight alternative solution to the task of indicative summarization of documents. We use off-the-shelf technology to identify keywords and keyphrases (described in Chapter 7), and our main interest lies in how effectively these entities can be used to characterize
information. For this, we examine how indicative summarization technology has been evaluated. Evaluation in this field falls into three main categories: evaluation of the relevance of the keywords themselves as descriptors of the documents from which they were extracted, evaluation of the keyword-based technology for the purposes of browsing, and the evaluation of the technical aspects of the algorithms themselves (e.g., speed).

Jones and Paynter [2001] used human subjects to evaluate KEA keyphrases, comparing automatically generated keyphrases for six documents against author-generated keyphrases, and a control set of randomly selected keyphrases. Users were asked to read documents and then rank the sets of keyphrases by their relevance to the documents. It was found that subjects ranked the keyphrases generated by the documents' authors the highest, with the ranking of KEA keyphrases falling somewhat below. (Results were reported in the form of a graph and the precise values were difficult to determine.) However, when the results were analysed in terms of the proportion of judgments that were above 5 on a scale of 1-10, 80% of the automatically generated keyphrases were rated as acceptable.

Wacholder et al. [2000] performed a similar experiment, gathering user judgments on three sets of keywords derived using different algorithms, finding a relatively low ranking overall among all three. Other studies have looked at the same issue. Tuney [2000] cites an 80% rate of acceptability for automatically derived keyphrases, although subjects were only given three choices, “Good”, “Bad” and “No opinion” for each keyphrase and “Good” and “No opinion” votes were pooled to form the 80% acceptability rating. There was no comparison with a baseline using keyphrases generated by human annotators. Barker and Cornacchia [2000], on the other hand, found automatically generated keyphrases rated between 0.47 and 0.56 on a scale of 1-3, which they say indicates an overall negative reaction to the keyphrases themselves. In comparing automatically identified keywords with those assigned by human experts, Frank [1999] found that the automatic identification of keywords is more reliable within a topically coherent set of documents. None of these studies situate the user evaluation within a specific task, i.e., to determine how well keywords describing a document help a user understand the contents of the documents.

Pirolli et al. [1996] report on a study in which users were asked to use the Scatter/Gather interface to both locate specific documents within a collection and determine the topical structure of a set of documents. They found
that Scatter/Gather was not as effective as a simple word-based search engine as a stand-alone information retrieval interface. However, they did find evidence to support the claim that Scatter/Gather was able to communicate the topic structure of a text collection. They conclude that Scatter/Gather may be useful in aiding users in exploratory phases of their search, and should be used in conjunction with other information retrieval technologies.

Sanderson and Croft [1999] asked users to judge the interestingness of the hierarchies they derived by first identifying keywords and then inducing relationships among them. They found preferences for their induced hierarchies significantly higher than for randomly generated relationships. Users were asked to judge these relationships in absolute terms, i.e., whether the hierarchies themselves seemed interesting.

With the popularity and ubiquity of search interfaces such as Google, it has been difficult to find clear preferences for systems designed to enhance search when human evaluators compare those systems to a Google-like baseline. Gutwin et al. conducted an experiment in which users were asked to assess the coverage of a particular collection of documents returned by a search engine on a particular topic. Experimental subjects interacted with both a traditional search engine-type interface, displaying the titles of the documents and a short piece of text associated with each, and an interface in which keyphrases associated with the documents were displayed. Users did not show a significant preference for the interface that displayed the keyphrases automatically discovered by KEA [Gutwin et al., 1998]. When users were asked for their opinions on the two systems, most reported that they preferred the keyphrase-based system for certain types of searches. Specifically, users preferred seeing the keywords when they felt the need to familiarize themselves with a topic area or when they were just beginning their exploration.

In a related study, Joho et al. [2004] used human evaluators to judge the effectiveness of two algorithms to support query expansion in tasks involving on-line search. Subjects were asked to judge the usability of the two query expansion systems against a baseline that offered no support. Users showed a significant preference for the baseline system in this case. In a study of multiple information retrieval techniques, Hertzum and Frokjaer found that the searches that users performed were iterative and that a combination of multiple interfaces was preferred [Hertzum and Frokjaer, 1996].

In some cases, subjects are asked their opinions of systems that process
data but these systems are not compared with any baseline [Hearst and Pedersen, 1996, Pansanato and Fortes, 2007, Buyukkokten et al., 2000a]. In a user experiment with hand-held devices [Buyukkokten et al., 2000b], a combination of summaries derived from keywords, and keywords plus targetted sentences, was preferred, although there was no comparison with a simple baseline. In other cases, keywords are proposed as interfaces to document collections, but no evaluation is performed on their effectiveness. Keywords were used to describe clusters of documents in a legal domain, to enable a dialogue-like interface to aid users with web searches [2000], but the interface itself was not tested on users. Squirrel [Duke et al., 2007], a system designed to enable browsing in semantically annotated documents, makes use of automatically derived keywords, but was not tested with users.

Clustering of documents based on the display of automatically detected keywords was proposed as a way of enabling both summarization and browsing of search text [Kummamuru et al., 2004, Krishna and Krishnapuram, 2001]. The claim in this work was that the extracted keywords were predictive of the content of the document. The terms were presented to the user, who could then refine the search based on the set of terms. The actual user interface was not described in detail in the paper and there was no evaluation metric, other than time taken to perform the search and clustering.

Other evaluation of summarization and clustering technology focuses on the underlying technology. Zamir and Etzioni [1998] examined ways of grouping documents retrieved from web searches using a variety of clustering algorithms, including K-Means, Agglomerative Hierarchical Clustering, and Buckshot [Cutting et al., 1992]. They used words contained in the short document snippets returned by search engines and their evaluation concentrated on technical issues such as speed and precision, not on user preferences or effectiveness of the technique for a particular task. In other cases, descriptive statistics (e.g., number of identified keywords as a function of the length of the document) are used as the evaluation metric [Wacholder et al., 2001, Manana et al., 1999]).

Various systems have been developed to provide interfaces to online news corpora [Steinberger et al., 2005, McKeown et al., 2002, Doran et al., 2004, Diaz and Gervas, 2005, Reffell et al., 2003]. These systems typically summarize particular events [McKeown et al., 2002, Doran et al., 2004], chart changes in news content [Reffell et al., 2003], or provide pointers to articles of interest to individual users [Steinberger et al., 2005]. Often, no direct user
evaluation of these systems is performed [Steinberger et al., 2005, Doran et al., 2004, Reffell et al., 2003]. Diaz and Gervas [2005] describe a system which employs a user model, allowing users to provide positive or negative feedback on retrieved articles. The keywords contained in the positively rated articles are then assumed to be relevant to that particular user and more articles were retrieved using those keywords, although no user input was elicited to test this latter aspect of the technology.

Pansanato and Fortes [2007] developed a prototype system that combined a variety of UI devices to allow users to search through Wiki pages that had been annotated for both meta-data (e.g., author) and keywords. Among the tools available to users were ones that enable highlighting, sorting, and viewing the data from various perspectives associated with keywords/meta-data. Users were asked to find particular pieces of information within the Wiki and then rate the overall interface in terms of the relevance of the information found and the usefulness of the interface. There was no comparison with a baseline. Fifteen subjects rated the overall system quite highly (avg. 6.07 out of 7), as well as the relevance of the documents found (avg. 5.92 out of 7). Perhaps the most relevant finding for our research came from the authors’ examination of logfiles of their experimental interactions, to determine which of the tools subjects used. The tool used most often was what the authors called facet browse, which allowed users to filter Wiki pages by selecting particular elements within each and showing pages with combinations of those elements.

The relevance of generated summaries has been evaluated, determined by comparing summaries to the document from which they were generated [McLellan et al., 2001], or to a hand-crafted summary [Hovy and Lin, 1997]. The relevance of particular documents has also been evaluated, by constructing idealized user models for particular populations (e.g., educated, young, female) and determining if sets of documents predicted to be of interest to those subsets are judged to be so by actual exemplars from the targeted population [Ardissono et al., 2001].

Kruschwitz and al-Bakour [2005] found that users preferred a system that offered possible query refinements to aid their search, when they had quite specific tasks to solve (e.g., “find a list of political figures in Iraq before the last Iraq war started”). In a unique, task-based evaluation, McKeown et al. [2005] recruited subjects to write reports on four different topics, after a brief description of the topic and the posing of specific questions to be
answered in the report (e.g., “Who participated in the negotiations that produced the Geneva Accords?”). They found that users felt they wrote more effective reports after using an interface that provided a multi-document summary of each pre-determined set of documents, rather than the interface that simply listed the documents.

The field of automatic summarization is vibrant and contains many examples of systems that are able to condense large amounts of information algorithmically. Furthermore, it shows that a relatively light level of annotation (e.g., at the level of an attribute or a keyword) can help users get a sense of what information is contained in an archive or database. As Sparck Jones notes, however, it is ultimately the needs of consumers of this information that must be accounted for in the algorithms [1998]. We take this claim seriously, in grounding our evaluation in user-centered experiments. More research is needed in applying summarization techniques to large corpora and testing the effectiveness of the summaries generated in aiding the information-seeking strategies of users. We address this need in the work that follows.

2.4 User modelling

Our hypotheses H3b and H5b are concerned with content selection as a function of user preferences. Hypothesis H10 claims that user behavior is predictable based on the topic a user is investigating. These claims are both informed by the belief that user behavior can be modelled and that overall satisfaction is determined, at least in part, by how well system output conforms to user preferences. The field of user modelling provides multiple examples of how these claims have been investigated in the past.

Like cooperative answering, the field of user modelling has its roots in the desire to make human-machine interaction more human-like. Early work focussed on how to use data about user’s actions and preferences to provide cooperative answer-like responses [Cohen et al., 1982]. Rich [1998], an early proponent of user modelling, described ways to generalize across users to obtain a relatively small set of models for a system to use.

In interactive systems, user models have been employed as a way of tracking progress and modifying system parameters to address the needs of the user [Elzer et al., 1994, Chu-Carroll, 2000, Komatani et al., 2003]. They have also been used to model beliefs and aid in plan recognition [Taylor et al., 1996, Ardissono and Sestero, 1996, Lesh et al., 1999]. As part of col-
collaborative problem-solving tasks, user models attempt to understand both system capabilities, expertise and previous actions, as well as user goals and the progress users make as they move towards achieving those goals [Moore and Paris, 1994, Blaylock, 2002, Ferguson and Allen, 2005]. User models have also been employed to optimize information throughput based on measures of cognitive load [Keates and Robinson, 1997, Chin and Crosby, 2003]. Most recently, modelling user behavior has been applied to e-commerce, as a way of developing, among other things, recommender systems (see, e.g., [Cho et al., 2002, Haruechaisak et al., 2004, Markellou et al., 2005, Yang et al., 2007] for specific systems and [Kohavi, 2001, Srivastava et al., 2000, Montaner et al., 2003] for useful overviews of the types of information that it is possible to gather from web interactions and a survey of data mining applications applied to those data).

One of the earliest treatments of user modelling as it applies to natural language systems can be found in [Kass and Finin, 1998], which provides a useful overview of the issues involved in acquiring and representing knowledge that should go into a user model. This overview claims that among the capabilities that a user model should enable is the understanding of the user’s information-seeking goals. Fischer [March, 2001] extends the applications of user modelling to include dealing with information overload. His claim is that a user model is a way of determining and presenting to the user just the information that is relevant at the time of interaction. As he puts it, a user model is meant to determine “the right thing to say at the right time in the right way”. Billsus and Pazzani expand on this notion of information management, arguing that there is an even greater need now for user models to address the issue of information overload, with the growth of information available electronically [Billsus and Pazzani, 2000].

Kobsa [1990] claims that a cooperative system will highlight information that is most likely be of interest to a particular user. Tailoring that information to the interests of the user in the form of recommendation systems is one way of doing so, and guidelines for this can be found in theoretical work on evaluative argumentation. The structure and components of evaluative argumentation is described in [Mayberry, 2001, Stiff and Mongeau, 2003]. Our summaries are not intended to be recommendations, but we make use of the algorithms that have been applied to recommendation systems to derive some of the operational details of our summary construction method.

Specifically, we incorporate information about user preferences into our
content selection algorithms using Multi-Attribute Utility Theory (MAUT) [Keeney and Raiffa, 1976]. (Implementation details can be found in Section 4.5.5.) The central insight from (MAUT), which we will revisit in Section 4.5.5, is that decisions are based on the interaction of multiple individual factors. By understanding the importance users assign to those individual factors, intelligent decisions can be made about the selection of content for display to particular users.

In the field of dialogue systems, the use of MAUT was first used for text generation in the real estate domain in the work of Carenini and Moore [2001, 2006]. This work applied MAUT to score and select and organize content for presentation to users in generated recommendations. This work showed that tailoring an evaluative argument to a user’s preferences does increase its effectiveness. Carenini and Moore’s work was extended by Walker et al. and applied to content selection for recommendations and comparisons in a spoken dialogue system in a restaurant domain [Walker et al., 2004]. Moore et al. used MAUT for content selection in a flight domain, also in a spoken dialogue system [Moore et al., 2004].

This same model was used by Stent et al. in a restaurant domain to construct summaries [Stent et al., 2002]. In this work, it was found that summary statements based on MAUT were significantly less preferred than comparison or recommendation statements constructed using the same methodology. The authors attribute this to an artifact of the way the summaries were constructed and not a general principle about summaries based on user model information. The example summary statements they show are applied to a small set of restaurants (three, in the examples in the paper), whereas the summary statements in this work encompass focal information at all steps in the dialogue, including large sets of restaurants.

While we use MAUT, it is not the only way to select content in a principled way in an interactive system. Thompson and Göker [2000] use their own similarity measure along with an entropy-based measure called information gain (see Section 4.5.2.1 for an explanation of information gain) to retrieve items from a database they claim will be suitable to their users. Lesh and Mitzenmacher [2004] use discrepancy functions that are under the control of users to filter database content for visual display. Interactive user input is also used in the Candidate/Critique Agent method of content selection, where options are presented to users whose ratings of those options are then used by the system to filter future items [Linden et al., 1997, McCalla et al.,
User models based on Bayesian Networks are constructed iteratively as user preferences are inferred during the course of an interaction and operationalize the choice of content selection [Zukerman et al., 2000, Eugene Santos et al., 1999]. Moore and Demberg [2006] construct a tree of options tailored to user preferences and use that to propose options they judge will be of interest, making trade-offs explicit among attribute values.

The type of user preference information we will use is dispositional as opposed to situational, so an interactive way of inferring user preferences such as Candidate/Critique or Bayesian Networks is not necessary. Dispositional information can be elicited prior to the interaction (see Section 4.5.2.8 for details on how this was done). In our work, the user model does not control the selection of individual options or recommendations, but rather it is used to filter sets of association rules. As applied to the output of a machine learning algorithm, our work represents an extension of work that used heuristics to prune association rules to find subsets that may be of interest to users [Srikant et al., 1997, Liu et al., 1999, Klemettinen et al., 1994, Nazeri and Bloedorn, 2004, Kotsiantis and Kanellopoulos, 2006].

Various evaluation metrics have been proposed to measure the benefit of user models, many of which look at second-order effects such as whether or not the user adopts the suggestions made by the system [Carenini and Moore, 2000] or how quickly a user can get through a dialogue modified according to a user model [Komatani et al., 2003]. Explicit user judgments were elicited by Walker et al. to compare a system tailored to a specific user’s preferences vs. a system tailored to another, randomly selected user [Walker et al., 2004]. In that study, it was shown that users rated more highly responses generated using their preferences in determining content selection and presentation.

We make use of multiple insights from the field of user modelling in the work here. The use of summary-style responses helps address the issue of information overload, by the principled choice of attributes to describe to users. User modelling within the context of Multi-Attribute Utility Theory also informs the process of content selection by determining weights for attributes and an overall utility score for representations of database entities. We evaluate our use of user model information through user judgments, in an experiment designed to contrast responses tailored to user preferences against those constructed with an alternative attribute selection criterion.
The work described in this thesis is grounded in the idea that human-machine interaction is a dialogue in which both parties contribute to the achievement of an overall goal. Dialogue management can be seen as mediating the process of data acquisition (from the user) and content selection (from a domain, for presentation to the user). The process of data acquisition is typically achieved over multiple turns in the dialogue and the underlying structure necessary to retain relevant information and coordinate action among human language technology (HLT) components is a subject of ongoing research [Martin et al., 1999, Seneff et al., 1999, J.F. Allen et al., 2000].

Within the field of dialogue management, work has focused on two main areas: protocols and formalisms for making the management of the dialogue server easier to develop and manage, and ways of automatically acquiring dialogue information from either simulated or real corpora of interactions. DIPPER [Bos et al., 2003] is a formalism for mediating interaction among a suite of human language technology (HLT) servers in the dialogue management component of a dialogue system. DIPPER uses an information state update language to control, script, and maintain the interaction among the servers. DIPPER uses the Open Agent Architecture (OAA) and its design benefits lie in the ease and transparency of encoding its rules. DIPPER is an extension of the TrindiKit [Goteborg University Dialogue Systems Lab, 2004], another formalism for dialogue management that uses a scripting language to encode rules that are mediated by an OAA-like architecture. Efforts at Carnegie Mellon University [Denecke, 2002, Xu and Rudnicky, 2000, Bohus and A. Rudnicky, 2003] have focussed on configuring an underlying architecture that can process simple information elicited from the developer. This information may include simple forms, as well as more complicated representations for dialogue goals and ontologies. In all cases, the specific knowledge used by the dialogue manager is derived from hand-annotated data.

The SpeechBuilder framework at MIT [Glass et al., 2004] is an attempt to generalize more dialogue parameters with less input from the system developer, but there is still a significant amount of hand-editing required to configure a dialogue system, even within this framework. SpeechBuilder makes use of data arranged in CSV (i.e., Comma Separated Value, the format used for SQL tables) format to create an initial spoken dialogue system.
completely automatically. SpeechBuilder requires just one level of annotation to create the initial system: each general attribute in the database must be marked as either an attribute, i.e., an entity that a user might ask for (e.g., a telephone number in a personnel database) or a key, i.e., an entity that has attributes associated with it (e.g., the person who is associated with a particular telephone number). The resulting dialogue system makes use of a simple dialogue management server [Polifroni and Chung, 2002] but is able to answer basic questions about database tables, with discourse inheritance and simple disambiguation and clarification subdialogues, with no other intervention from a developer.

Other research takes a more direct approach at automating dialogue construction, by framing dialogue as a Markov Decision Process. Levin et al. [Levin et al., 1998] applied reinforcement learning to a corpus of simulated dialogues to automatically acquire a dialogue strategy in an air travel domain. Here, their criterion for success was determined by the ability of the system to narrow flight options to a number small enough to speak. Walker and colleagues at AT&T applied reinforcement learning to other aspects of dialogue strategy, e.g., summarization strategies in an email domain [Walker, 2000] and ways of presenting choices in a tourist domain [Singh et al., 2002]. For these experiments, measures of satisfaction were directly elicited from users of the system. These experiments used a relatively small number of parameters to represent the dialogue state at each turn.

Roy et al. [2000], explored methods for approximating the “belief state”, i.e., the reliability of the hypothesis from the automatic speech recognizer, at each turn in a spoken dialogue. This led to formalizing the problem of the automatic acquisition of dialogue strategy from observed or simulated data within the framework of the Partially Observable Markov Decision Process. This approach has been extended and applied to other corpora in work at Cambridge University [Williams and Young, 2007] and the University of Edinburgh [Lemon et al., 2006].

Inasmuch as this work can be applied to the automatic acquisition of dialogue strategy, it does so by using data to determine what to say at various points in the dialogue. It will do so by using the data contained in the focal information. The records in the domain are typically well-structured, or a structure can be induced (see, e.g., [Uren et al., 2006] for an overview of efforts at automatically acquiring structured data from the Web). Furthermore, data describing the domain is prone to change over
time. Techniques that make use of, and are robust to changes within those data, are less likely to need to be reconfigured if, for example, a change in the schema takes place.

### 2.6 Generating summary responses in dialogue systems

The idea of generating a summary-type response for a dialogue system has its roots in the research on intensional answering, which attempted to codify the ways in which domain data can be described and presented at an abstract level of generalization (see, for example, [Gaasterland and Minkler, 1991, Motro, 1996, Cheikes and Webber, 1989] for a discussion of some of these principles). An example of an intensional answer to a user utterance such as “I’m looking for a Chinese restaurant in London” would be “Many of the Chinese restaurants in London are located in Leicester Square and are medium priced.” This response contains no extensional information at all, i.e., no specific instances of database tuples are contained within it. However, the answer does summarize the items in the database matching the user constraints into a statement capturing generalization about neighborhood and price. This response provides the user with possibly more useful information than a simple, but possibly quite lengthy, enumeration of the restaurants that satisfy the constraints “Chinese” “London”.

Attempts have been made previously to develop dialogue systems that provide information incrementally to users. In the air-travel domain, some systems within DARPA Communicator Program [Walker et al., 2002b] provided additional information to users while still pursuing an essentially menu-driven approach. A partial example of one such system is shown in Table 2.1, adapted from [Seneff and Polifroni, 2000]. This system allows for much more flexibility of input than the one in Table 1.1, and there is even an attempt to summarize focal information. (cf., utterances System 3 and System 7). However, the summarization techniques were based on just one possible dialogue constraint (i.e., time) and were only applied when a sufficiently small set of tuples had been returned by the database. A completely menu-driven phase still preceded any description of the focal information. Furthermore, the dialogue management scheme exemplified in Table 2.1 was not meant to be portable or automatic.

Polifroni et al. [2003] developed a “Refiner” algorithm, in an attempt to
automate the process of content selection for information presentation, and
to enable summary responses in a tourist information domain (i.e., restaurants and hotels). Summary responses are constructed in terms of a stepwise refinement of user goals. The “Refiner” identifies attributes whose top four values account for 80% of the focal information and chooses from that a set to present to the user. Table 2.2 illustrates examples of responses constructed using the Refiner algorithm. Polifroni et al. claimed that summary responses, such as those in Table 2.2, help the user mine through large datasets to a few well-informed choices. However, this claim was not evaluated through user experiments.

The Refiner algorithm was deployed at MIT for a database of Boston restaurants, containing information about approximately 1000 restaurants, covering 106 cities in the Boston metropolitan area (e.g., Newton, Cambridge) and 45 neighborhoods (e.g., Back Bay, South End). Included in the database was information about cuisines, price ranges, and addresses.

The dialogue manager retrieved the entire set of database tuples matching a user constraint at each turn (or the full set, for initial queries). The Refiner sorted the values for each attribute by their frequency. For example,
Restaurants in Quincy: I have found 14 restaurants. Some of the options are American, Brazilian, Indian and Italian Pizza. None of them are expensive. Many of them are on Hancock Street, Adams Street, Billings Road and Franklin Street.

Restaurants with a medium price range: I have found 172 restaurants. Most of them are located in Boston and Cambridge. There are 18 choices for cuisine. They are predominantly in the North End, Back Bay, the Financial District and the South End.

Restaurants near the Prudential Center: I have found 21 restaurants. There are 11 choices for cuisine. Many of them are on Boylston Street, Newbury Street and Massachusetts Avenue.

Table 2.2: Example dialogue using Refiner method for summary construction, taken from [Polifroni et al., 2003].

cuisine values for restaurants in Back Bay included, in order, American, French, Italian, and Thai, indicating that American was the most common cuisine among those restaurants, followed by French, Italian, and Thai.

Content selection with the Refiner was made within the Genesis natural language generation component [Baptist and Seneff, 2000, Baptist, 2000]. Genesis is template-based, and the templates can be enhanced with information relating to lexical and grammatical choices. Genesis has a pre-processing phase in which information from the dialogue manager is analysed and parameters that will ultimately be used to generate the response are set within the frame. For the Refiner, this pre-processing phase involved consulting a set of domain attributes that had been pre-specified by system developers for use in a particular context. These attrsToConsider depended upon the constraint introduced by the user in the utterance. In the case of multiple constraints being introduced in a given utterance, attrsToConsider were chosen to associate with the first constraint in a pre-specified order. For example, if a user had instantiated a preference for the constraint city, e.g., “Brookline”, a Genesis template licensed the system to look, in order, at values for the attributes cuisine, price range, and neighborhood to determine what to say. This was one of three hand-coded and invariant parameters contained within the Refiner algorithm.

The actual selection of attribute values was also controlled by the generation template, for each attribute from the attrsToConsider list. Each attribute type in a Genesis template had two parameters associated with it, numVals, the number of values to consider for the attribute, and percent, the percentage of values for that attribute that must be accounted for by numVals. These two parameters, along with the specific attributes to use in context, were also hand-coded, determined a priori, and could not be
modified at run-time. For example, a particular attribute would be chosen for inclusion in a summary response if its top four most frequent values accounted for 80% of its possible values, both four and 80% set a priori. At most, three separate attributes were chosen for inclusion in a response. Once these attributes were chosen, the system returned the generation frame and a natural language response was constructed. (Details of the Refiner method of response construction were never published. However, for more details on how the Refiner is used in our experiments, see Section 4.5.4.)

Demberg and Moore [2006] explored the issue of both stepwise refinement and a user model for information presentation in an air travel domain. They constructed trees of options based on clusters of data representing flights in their domain. The tree structure determined the dialogue flow and, because it was based on values for attributes in the domain (e.g., airline), choices at branch points could be made sensitive to user preferences. Table 2.3 shows an example of a dialogue taken from the Demberg and Moore system, with summary responses tailored to a business user. A user evaluation of this system showed a marked preference for summary statements constructed using both stepwise refinement and a user model.

These efforts demonstrate the growing realization that dialogue systems can do more than elicit constraints and describe small sets of database records. To be truly interactive, a dialogue system must be able to reason about its domain and the preferences of its users. We will investigate how focal information can be exploited to achieve these goals.

2.7 Conclusion

Research into when and how users search for information point out the importance of browsing as a common information-seeking strategy. This research suggests that methods of describing corpora at a high-level could be useful to people whose familiarity with the domain is limited or who have undefined goals for the information they seek. Summarization is one way of enabling browsing and lightweight methods of producing summaries, such as those that use keywords to extract sentences, have been shown to be both technically feasible and methodologically interesting.

Research in cooperative answering suggests that answers containing generalizations of focal information are both useful and add a more human-like dimension to human-machine dialogue. The methods employed to derive such answers, however, are often hand-crafted and domain-specific. Ma-
User1: Hello. I’d like to book a flight from Edinburgh to Brussels for tomorrow morning arriving at round 3pm.

System1: There are four flights with availability in business class. None are on KLM. The only direct flight arrives at 5:30 pm, which is later than you requested. To arrive earlier, you’ll have to make a connection.

If you’re willing to travel economy, there is a direct flight on BMI, arriving at 1:10 pm. There’s also a KLM flight arriving at 1:50 pm, but it requires a connection in Amsterdam. All other economy flights take longer or arrive after 4 pm.

User2: Can you tell me more about the business class flights that arrive earlier?

System2: The two flights with shortest travel time are both on British Airways and cost 308 pounds. The first takes 4 hours and 20 minutes and arrives at 1:20 pm. The other takes 4 hours and 30 minutes and arrives at 3:30 pm.

There’s a cheaper flight on Lufthansa. It arrives at 3:10 pm and costs just 270 pounds, but it takes 5 hours and 35 minutes. All other business class flights arrive very early or too late.

Table 2.3: Example dialogue system in an air travel domain, tailored for a business user and taken from [Demberg and Moore, 2006].
machine learning and Knowledge Discovery from Databases has produced algorithms for the automatic acquisition of high-level information about large databases, and point the way to useful techniques that can be applied to dialogue systems.

The importance of taking the user into account in constructing a response is a topic addressed along multiple dimensions in the field of User Modelling. Models of information overload, level of expertise, and overall goals have been influenced the selection of content and the way it is presented to users. The application of user modelling we will employ is a decision-theoretic approach to content selection, which will be applied to the construction of summary statements.

The field of dialogue management has addressed the issue of automating the process of acquiring a dialogue strategy, but so far these dialogue strategies have, for the most part, been limited to the menu-driven, constraint-elicitation paradigm. Work has been done on determining optimal strategies for information presentation once a dataset has been sufficiently narrowed, but little work has been done on techniques for automatic content selection for browsing at any point in the dialogue. Data-driven approaches are popular within dialogue management but they focus on data from human-machine interactions or simulations of the same. Very little effort has been applied to exploiting the data contained in an application domain.

Recent work in spoken dialogue systems has begun to examine the use of domain data as it relates to content selection. Stepwise refinement is one means of constructing summaries that makes use of focal information. However, other than [Demberg and Moore, 2006], little research has been done in finding a domain-independent solution to content selection and information presentation. We address this issue in the remainder of this work.
Chapter 3

Testing the utility of summaries for structured data: A user study in the restaurant domain

3.1 Goals of experiment

We describe here an experiment designed to test hypothesis H1, the Incremental Summary Hypothesis. To confirm this hypothesis, we must first show that users want to hear summary responses when interacting with a system. Given a confirmation of the hypothesis, the next challenge is to find the best way to construct those responses.

This preliminary experiment is also designed to test hypothesis H1a, that users prefer summary type responses when they are unfamiliar with a domain. To gather data to answer these questions, we carry out an experiment to compare summary responses to a baseline, i.e., traditional menu-driven responses (e.g., What neighborhood are you interested in?) in a London restaurant domain.

3.2 Experimental design

There are multiple issues involved in determining judgments of particular response types and the design of an experiment to test this must control for these factors. Users’ familiarity with the domain, the quality of the interaction (e.g., how well the user is understood), the extent and relevance of the domain data D (e.g., are the data real), and the quality and style of the
responses themselves are all issues that affect how a response is perceived.

To control for the quality of the interaction, dialogues in this experiment were presented to users in “overhearer” mode, where only the text of the simulated user utterance and the set of responses to be judged were made available. This method of evaluation differs from a real interaction with the system in that the subject performing the evaluation examines query/response pairs that have already been created by the system. The subject does not interact directly with the system itself, and, in the case of this experiment, the subject reads the response, as opposed to hearing it spoken. To control for the ecological validity of the domain data, responses were constructed from data consisting of all restaurants represented in the online Zagat guide to London [ZagatSurvey, 2005] (Section 4.4 will provide a complete description of the database).

To control for user preferences, we ask users to assume a “persona” for each part of the experiment. This method of controlling for subject preferences has been used within the context of a dialogue system evaluation, when responses are tailored to a particular class of user [Hu et al., 2007, Demberg and Moore, 2006]. In this experiment, we ask users to assume three separate personas, chosen to typify a range of user types for a dialogue system in a restaurant domain. The complete description for each personas, as well as the dialogues presented to users, can be found in Appendix A.1.)

We describe each briefly below.

The first persona we refer to as the Londoner, representing someone who knows London and its restaurants quite well. The Londoner persona conforms to the information needs of the Power User defined in Belkin’s terms in Section 1.1. The Londoner is likely to know the specific information s/he is looking for and has no problems expressing preferences.

The second persona we call Generic tourist. The Generic Tourist doesn’t know London well and does not have strong preferences when it comes to selecting a restaurant. The Generic Tourist has information needs similar to the Browsing User described in Section 1.1, who may wish to learn about the structure of the domain and retrieve information by recognition rather than specification.

The third persona we call the UM tourist. This persona represents another instantiation of the Browsing User from Section 1.1. The UM Tourist is also unfamiliar with London but has had a chance to interact with the system and express preferences about restaurants. The UM tourist is con-
Table 3.1: Examples or responses tailored to the three personas used in the initial experiment.

Concerned with price and food quality (in that order), and has a preference for restaurants in Central London. After location, the UM tourist is most concerned with cuisine type. The cuisines the UM tourist prefers and the neighborhoods the system considered to be in Central London can be seen, along with the complete user model information for this persona, in Appendix A.1. Experimental subjects were asked to read these details, and the information was made available for reference, via a link, throughout each subjects’ interaction.

Eighteen subjects were recruited from among post-graduate students and full-time research staff in the Department of Computer Science at the University of Sheffield. Each subject was asked to assume all three personas, in randomized order, and remain within that persona throughout each pass through the dialogues. As each persona, subjects rated a series of four dialogues, each containing between three and four utterance/response pairs. Ratings were on a Likert scale from 1-7. Because we were interested in the relative rank of each option we did not allow tie votes among the three choices or responses. Once a particular number was chosen on the Likert scale, that number was disabled for the other two options. Users could clear all choices, but had to rank each of the three response types before moving on to the next utterance.

The three response types for each utterance were constructed by hand. The first is a standard, data-elicitation type response, e.g., What price range are you interested in? ¹ Hypothesis H1a would predict that this response

¹Menu-driven queries were always asked in the same order across all dialogues, but if a simulated user response contained information that had not yet been asked for, the query to elicit the same information was not used in subsequent turns in that dialogue. This conforms to
Table 3.2: Scores for the three response types judged by subjects.

<table>
<thead>
<tr>
<th>Response type</th>
<th>Average score</th>
<th>Standard dev.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner</td>
<td>4.41</td>
<td>1.95</td>
</tr>
<tr>
<td>Generic Tourist</td>
<td>4.46</td>
<td>1.65</td>
</tr>
<tr>
<td>UM Tourist</td>
<td>4.39</td>
<td>1.94</td>
</tr>
</tbody>
</table>

type will be most preferred by the native Londoner who is familiar with the domain. This hypothesis states that users prefer summary types when they are *unfamiliar* with a domain.

The second type of response is based on the Refiner method of attribute selection, explained in more detail in Section 4.5.2.7. Restaurant attributes in this type of response are chosen by analysis of the focal information at that point in the dialogue. Associations among attributes are discovered using a decision tree algorithm, explained in more detail in Section 4.2. Hypotheses H1a predicts that the Generic Tourist persona would highly prefer this response since the Generic Tourist is unfamiliar with the domain. Hypothesis H3a predicts that the Generic Tourist, who hasn’t indicated preferences, would prefer to hear about attributes chosen because they generalize well across the domain.

The third type of response is based on information contained in a user model, combined with decision trees to discover associations among preferred attributes. This type of response is explained in more detail in Section 4.5.2.8. These responses are tailored to the user model that subjects were asked to assume for the UM Tourist persona, and we hypothesize that they will appeal most strongly to that persona, again because of Hypothesis H1a, and also H3b, which predicts that users who have expressed preferences will value responses that highlight those preferences.

### 3.3 Experimental results

Overall results, showing the average score for each type of response across all personas, are shown in Table 3.2, graphically in Figure 3.1. The scores are close, and pair-wise T-tests between each pair showed no significant difference. This would be expected if the experimental hypotheses were true, i.e., that the three different response types were preferred in turn by

---

standard VXML-style functionality for menu-driven systems. [Voxeo, 2003]
the three different personas subjects were asked to assume. We analysed the scores in more detail, however, to determine how the individual scores for personas matched the hypothesized response types for each.

Table 3.3 shows scores for each response type as a function of the persona the subject was assuming when assigning them. Overall, subjects preferred the responses tailored to the personas they were in. When assuming the Londoner persona, the Londoner responses were preferred significantly more than the UM Tourist scores (using a paired $t$-test; $p < .05$, df = 95), but not more so than the Generic Tourist scores (also with paired $t$-test). This confirms our primary hypothesis, $H1$, that users prefer incremental summaries in dialogue systems. Further, it disconfirms Hypothesis $H1a$, that users prefer summaries only when they are unfamiliar with the domain. The fact that no difference was found between Londoner responses and those for the Generic Tourist indicates that the latter contain information that is perceived as useful even to a Power User.

When assuming the Generic Tourist persona, users also preferred the Generic Tourist responses, significantly more than the Londoner responses (paired $t$-test; $p < .05$, df = 95), but not significantly more than the UM Tourist responses (also paired $t$-test). This disconfirms Hypothesis $H3a$, 
<table>
<thead>
<tr>
<th>Persona</th>
<th>Londoner</th>
<th>Generic Tourist</th>
<th>UM Tourist</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner</td>
<td>5.02</td>
<td>4.55</td>
<td>4.32</td>
</tr>
<tr>
<td>Generic tourist</td>
<td>4.14</td>
<td>4.67</td>
<td>4.39</td>
</tr>
<tr>
<td>UM tourist</td>
<td>3.68</td>
<td>4.86</td>
<td>5.23</td>
</tr>
</tbody>
</table>

Table 3.3: Results as a function of persona assumed.

that the optimal summary type for users completely new to a domain will describe attributes that generalize well across the focal information. These findings indicate that user model information is helpful in summary responses for any *Browsing User*.

Finally, when asked to assume the UM Tourist persona, responses tailored to that persona were overwhelmingly preferred over those for Londoner persona (paired $t$-test; $p < .0001$, $df = 95$). However, UM Tourist responses were not significantly preferred in a paired $t$-test over responses tailored for the Generic Tourist. This confirms Hypothesis \textbf{H1a}, that users prefer summary responses when they are unfamiliar with the domain, but disconfirms Hypothesis \textbf{H3b}, that users will prefer summaries that describe attributes and values they are most concerned about, when they have had the opportunity to express those preferences. The results for both the Generic Tourist and the UM Tourist show that both types of browsing summaries contain useful information.

Hypothesis \textbf{H1} states that users prefer summary responses in dialogue systems, without reference to background knowledge of the domain. To test this hypothesis concerning the efficacy of summary type responses in general in dialogue systems, we make a further binary comparison; we compare data-elicitation responses (i.e., Londoner responses) against the higher scoring of the two summary responses in the same context, across the entire experimental dataset. When scores are examined in this binary way, data-elicitation responses scored 4.64, against an average score of 5.29 for summary-type responses. This difference was significant (paired $t$-test; $p < .0001$, $df=263$) and it confirms Hypothesis \textbf{H1}.  

45
3.4 Conclusion

We conducted an initial study to compare summary responses against traditional, menu-driven responses in the context of a simulated dialogue system interface to a London restaurant domain. Subjects were asked to assume three “personas”, each of which represented a possible type of user of such a system. Subjects read and judged responses as either a native Londoner, a tourist who was interacting with the system for the first time, or a tourist whose preferences were known to the system. The personas were designed to reflect differing ends of the spectra defined by Belkin to characterize information-seeking strategies. The Londoner is an example of a Power User who is very familiar with a domain, while the two tourist personas represent Browsing Users.

Subjects found summary responses useful across all personas, but especially when the personas were meant to be unfamiliar with the domain that the dialogue system targets. There was, furthermore, a significant preference for summary responses in general, across all personas. This indicates that menu-driven approaches to dialogue might be failing to address the information-seeking strategies employed by users.
Chapter 4

Testing a range of methods for automated response construction

The Incremental Summary Hypothesis was confirmed by the experiment described in the preceding chapter. We next turn our attention to the technological issues involved in creating such responses. To do so, we first discuss some of the limitations in the Refiner approach, a previous attempt to create summary responses automatically. We then define a set of parameters needed to create summary responses and examine each in turn. We conclude with an algorithm encompassing multiple parameters that can be varied to create a set of responses summarizing focal information.

4.1 Issues with the Refiner

The underlying motivation behind the Refiner was interesting and worth pursuing; the heuristic nature of the content selection process was a concern, however. The three most important parameters for response construction within the Refiner, attrsToConsider, numAttrs, and percent, were determined a priori by a system developer and hard-wired into the system itself at run time. For given subsets of the data, especially those that were retrieved at later stages in the dialogue when choices were more limited, it was possible that none of the pre-determined attributes met the thresholds for inclusion and so no summary response was constructed.

Another shortcoming in the Refiner method of response construction was that no attempt was made to analyse focal information as more than simple lists. As we noted in our example from the Boston restaurant domain
(Table 1.2), many of the ethnic restaurants in Boston were clustered in particular neighborhoods (e.g., Italian restaurants in the North End). This sort of knowledge was beyond the capabilities of the Refiner. All attributes chosen by the Refiner were described in isolation, as lists of values associated with one particular attribute (e.g., “Some of the neighborhoods are the North End and Back Bay. Some of the cuisines are American, French, and Italian”). The final shortcoming of the Refiner approach was that the claim that summary responses such as those in Table 2.2 help the user mine through large datasets to obtain a few well-informed choices was not evaluated experimentally.

We hypothesized that it would be possible to come up with an automated way to perform content selection in a way that guarantees a summary response for each turn in a dialogue. We also hypothesized that simple machine learning techniques could help us discover information about backend data that users would find useful. Specifically, we felt that the use of machine learning to discover associations among attributes would help users see patterns and make choices in a more informed way. In this chapter, we will address the issues in the Refiner method of attribute selection and also examine new methods of response construction. In the following chapter, we will test our hypotheses in the context of user experiments.

4.2 Applying machine learning to corpus

Much of the previous work in the field of cooperative and intensional answering has involved the extensive use of hand annotated data. These annotations are typically in the form of rules [Motro, 1989] or complex, domain-specific ontologies [Benamara, 2004]. The literature in Data Mining and Knowledge Discovery from Databases suggested that user-centered interfaces to data could be grounded in automated techniques for analyzing data specifically for the purpose of describing it at an abstract level (see [Fayyad et al., 1996b] for an overview of the field of Knowledge Discovery from Databases).

This literature led us to consider decision tree induction as a means of performing content selection for summary responses. Our initial interest in decision trees was motivated by the fact that they could discover associations among attributes in a dataset, precisely the sort of information we had noted was missing from the Boston restaurant domain. (See Section 4.3 for a more comprehensive discussion of the advantages of decision tree induction.
(as a machine learning algorithm for content selection.) As a first step in determining the efficacy of this method, therefore, we built a set of decision trees using the data from the Boston restaurant domain and examined the results.

The software used to construct the decision trees is c4.5 [Quinlan, 1992]. The algorithms incorporated into c4.5 (described in more detail below) have become a standard within the field and have been incorporated into the WEKA toolkit, a widely used software environment that provides an experimental testbed for machine learning algorithms [University of Waikato, 2004]. We used a standalone version of c4.5 that is freely available on the Web [Quinlan, 1993b] and written in C. We downloaded and compiled the source code for c4.5 and embedded it within a larger suite of servers within the Galaxy Communicator architecture [Seneff et al., 1999, Galaxy Communicator Architecture, 2003]. In both standalone and server mode, c4.5 takes as input a set of database tuples, along with a file describing the attributes to use to construct the decision tree and how to use them (i.e., which attribute to use as the dependent variable).

When building a decision tree, one attribute is designated as the dependent variable, with other attributes used to predict that variable. Each branch in the tree represents a cluster of the data that is described by the attribute/value pairs that predict the leaf node. Using the data from the Boston restaurant domain at MIT [Polifroni et al., 2003], we experimented with various configurations of decision trees.

Figure 4.1 shows one such decision tree, with \textit{cuisine} values used to predict \textit{neighborhood} in the Boston data. As can be seen, the relationship between the North End neighborhood and Italian cuisine is immediately apparent, as is that between Chinatown and Chinese cuisine. American cuisine is not quite as segregated as Chinese and Italian cuisine in Boston; American restaurants are predominantly in the Back Bay neighborhood, but there are also significant numbers in the Financial District and Allston.

The results from the Boston restaurant domain suggested that informative and possibly useful clusters might be discovered through the use of decision tree induction to identify relationships between two (or more) attributes. Encouraged by these results, we looked more closely at the issues involved in using decision tree induction as a way of constructing responses in dialogue systems.
4.3 Advantages of using decision tree induction for dialogue systems

Decision tree induction has long been used in data mining and has been shown to be an efficient way to learn associations among attributes [Gaines, 1996a, Kamber et al., 1997, Liu et al., 2000, Eardley et al., 2000]. In addition, decision tree induction has several advantages that are specific to dialogue management. In the first place, the algorithms are relatively fast. Decision tree induction algorithms in general, and c4.5 in particular, take $O(n \log n)$ time, and several parallel implementation algorithms have been proposed to speed them up even further [Ruggieri, 2002, Joshi et al., 1998, Shafer et al., 1996]. With a database of approximately 600 restaurants in London, there is no noticeable degradation in response time even when computing association rules on-the-fly for each utterance.

In the second place, the particular implementation we use in this work, c4.5, has been shown to be robust in the face of data containing attributes with many unknown values [Zheng and Low, 1999]. These types of attributes are known to occur in the London restaurant domain data, where 19% of the values for food quality and 20% of the values for service are unknown (see Section 4.4 for a description of database attributes). Though attributes with unknown values may be of limited value in summarizing our data to users, an underlying classifier known to have well-developed and tested ways of dealing with attributes of this type in training data [Quinlan, 1989] would be valuable for our dataset. This is also an important consideration if we
wish to devise a method of content selection that is applicable across multiple
domains, where unknown values may be even more common.

Another advantage of decision tree learners, albeit one that we have not
exploited in our current work, is that they can be applied to both categorical
and numeric data. Learned relationships can, therefore, include attributes
such as scalar values for average prices. The rules themselves can either
be in the form of equivalences (e.g., “when \textit{cuisine} equals ‘italian’ THEN
\textit{neighborhood} equals ‘north end’”), or arithmetic relationships (e.g., “if \textit{average price} is less than \$20.00, THEN \textit{service} is ‘poor’”). As with its ability
to deal with unknown values, we felt this was a useful characteristic for a
portable algorithm.

Finally, as opposed to many other types of classifiers where the decision
space may remain opaque, decision trees produce human-readable output.
The path from node to leaf can be translated relatively straightforwardly into
visual displays that characterize the data. This has been done in data mining
applications [Kosseim et al., 2001, Gaines, 1996b, Basak and Krishnapuram,
2005, Apte and Weiss, 1997], but, as far as we know, this is the first attempt
to use such rules in the context of dialogue systems. Association rules have
well-defined scoring metrics (see Section 4.5.5), as well, so that an “optimal”
set of rules can be discovered and generated into natural language.

It is important to note we are not using tree induction to derive an op-
timal classifier for the data. Rather, we are attempting to find a set of
association rules that, when converted to natural language, will provide the
best, or most helpful, insight into the data we are processing for our users.
In that respect, we are only concerned with the decision tree classifier’s per-
formance on training data. Specifically, we are interested in two evaluation
criteria: the size of the clusters at the leaf node and the information in the
paths leading to those nodes. Our interest in the size of the leaf nodes lies
in our desire to have a method of constructing generalized summaries of the
focal information. The size of the leaf nodes corresponds to the number of
database tuples that can be described using that branch of the tree, and the
larger that number, the higher the degree of generalization that can be ob-
tained. The information in the paths themselves is more difficult to assess,
but important to our goal of constructing summaries that will help users
understand the backend data better. We will examine these criteria in more
detail in the sections that follow.
4.4 Database used

The database used in the experiments described in this chapter, and in Chapters 3 and Chapter 5, were obtained from the Zagat website (www.zagat.com). The records in the domain data are comprised of information about 594 restaurants in the London. The restaurants are represented in a single-table database, with each restaurant having up to 24 attributes associated with it. The full set of attributes is:

- building
- city
- cuisine
- decor
- decor_range
- entrance
- food
- food_range
- food_type
- locale
- name
- neighborhood
- phone
- price_range
- prices
- avg_price
- rest_id
- restriction
- resttype
- service
- service_range
- street
- streetnum
- tube_station
All values for attributes are either strings or integers. Database values were preprocessed to expand abbreviations (e.g., “St. James” ⇒ “Saint James”) and to regularize names (e.g., “Belvedere, The” ⇒ “The Belvedere”). Street addresses were parsed into :streetnum and :street to allow us to filter on street names, if necessary.

Values for decor, food_quality, and service were represented as integers in the original Zagat data. Values for decor and service ranged between 9 and 27 in the data, and values for food_quality ranged between 10 and 28. Prices were given as ranges (e.g., “31 pounds - 50 pounds”) and post-processed to determine a single mid-range integer value for each.

We computed means and standard deviations for each of the four integer attributes, i.e., :decor, :food_quality, :price, and :service, and then used a hierarchical agglomerative clustering algorithm similar to that used in [Polifroni et al., 2003, Demberg and Moore, 2006] to create three bins for each of these values. These bins were intended to correspond impressionistically to poor, medium, and good (or inexpensive, medium-priced, and expensive in the case of prices). Any values that were two or more standard deviations away from the mean were placed in separate categories at each end of the scale, corresponding to very poor/very cheap or very good/very expensive. We then re-annotated the database, adding new fields for decor, cuisine, food_quality, and price corresponding to the symbolic values derived from the clustering. The symbolic values from the clustering were those used for natural language generation of the summary responses.

Rather than use all 24 attributes to evaluate selection criteria, we algorithmically chose a subset of attributes to use in the experiments described below. Among the 24 attributes are ones that are unique to a particular restaurant and not typically associated with user preferences (e.g., phone number, name). In addition, some attributes are quite specific to a certain type of restaurant (e.g., “traditional” or “modern” as restriction, and “bistro” or “brasserie” as restaurant type). These are are represented in fewer than half the restaurants (e.g., restaurant type is found in only 29% of restaurants in our database).

When unique attributes and those that apply to fewer than half of our restaurants are removed, we are left with the following nine attributes:

- Cuisine

- Decor (quantized into three bins: poor, good, and very good)
• Food quality
• Price (quantized into three bins: inexpensive, medium-priced, and expensive)
• Service (quantized into three bins: poor, good, and very good)
• Location/neighborhood
• Tube station
• Street name

In the experiments that follow on the full set of restaurants, this is the set of attributes used. ¹

### 4.5 Parameters in automatically constructing summaries

The basic problems we face in constructing summary responses is the fact that, in a dialogue system, the set of backend data that corresponds to the focal information is typically too large to be described in full. The first parameter in summary construction becomes, therefore, how many of the top-ranked attributes to use in a summary. Given the fact that there are many more attributes per database entity than can possibly be used in a summary, the next issue becomes the selection of which attributes to use, characterized here in terms of algorithms for attribute ranking. With a set of attributes selected for presentation to the user, we must next define a clustering algorithm defining how attributes are presented for speaking/displaying. The final parameter in response generation is the method used for scoring the groupings of data derived by the clustering algorithm.

Our algorithm for constructing summary responses thus involves the following steps:

- Take as input the subset of database entities comprising the focal information, as determined by the user constraints in context;

- Select number of attributes to speak about (Section 4.5.1);

¹In constructing summaries, all non-unique attributes are available for consideration by the attribute selection algorithms described below. In cases where the number of restaurants in the focal information is small, attributes such as restaurant_type can be algorithmically determined to be of use for summarization.
• Rank attributes relevant to current state (Section 4.5.2);

• Construct clusters using selected attributes (Section 4.5.4);

• Rank and select top-\(N\) clusters (Section 4.5.5);

• Construct frames for generation, perform aggregation on equivalent predicates and generate responses (Section 4.6).

Each of these steps will be described in more detail below. The steps themselves apply sequentially to produce a summary, but the analysis of the issues involved in any one step will, in some cases, necessarily involve reference to another. Some of the algorithms for attribute ranking, for example, take into account the specific clustering methods that will be used. Likewise, some of the cluster scoring algorithms presuppose a particular attribute ranking method.

4.5.1 Number of attributes

To derive groupings of data large enough to make generalizations possible, we first examined the size of clusters we could derive using tree induction. The purpose of this experiment was to determine an “optimal” number of attributes to use when inducing a tree. To do this, we constructed decision trees using all possible combinations of nine attributes in the London restaurant domain (see Section 4.5.2 for a discussion of how the nine attributes were determined). We then computed a value for coverage at each leaf node. Coverage, a traditional metric used in decision tree analysis, measures the number of instances at a given leaf node that are correctly classified [Witten et al., 1999]. In the remainder of this work, when we speak of the number of restaurants associated with a given rule (i.e., path from root to leaf), we are referring to coverage.

Table 4.1 shows the influence of number of attributes on resulting cluster sizes in the London restaurant domain. When using just two attributes, the average coverage of a cluster at the leaf node is 60.4. This number drops precipitously, even when just one more attribute is added. It, therefore, became important that we have a good way of ranking our attributes, so that we could choose the top 2-3 for inducing association rules. This number not only gave us a reasonable level of generalization across our dataset, but also is small enough to make comprehensible summaries that human subjects could hear and absorb.
<table>
<thead>
<tr>
<th>No. of attributes used</th>
<th>Avg. bin size</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>60.4</td>
</tr>
<tr>
<td>3</td>
<td>4.22</td>
</tr>
<tr>
<td>4</td>
<td>2.23</td>
</tr>
<tr>
<td>5</td>
<td>2.16</td>
</tr>
</tbody>
</table>

Table 4.1: Influence of number of attributes used to construct tree and average size of the resulting leaf clusters. The numbers are averaged using leaf clusters from trees built using the entire set of 596 restaurants in our dataset.

4.5.2 Attribute Selection/Ranking

Attribute selection has traditionally been studied as a way to optimize classifier performance. Attribute selection algorithms are typically divided into two categories: *filter* methods, which attempt to discover the best set of attributes based on an examination of the data alone [Kira and L.A.Rendell, 1992, Almuallim and Dietterich, 1994], and *wrapper* methods, which use the classification algorithm itself to discover the best set of attributes to use [Kohavi and John, 1998]. Given that the classification algorithm was known (i.e., tree induction, specifically c4.5), and domain data was available, we were able to examine both methods of attribute selection.

The algorithms for attribute selection tested were:

- *Information gain* [Witten and Frank, 2000], a filter method;
- *Gain ratio* [Witten and Frank, 2000], a filter method;
- *Forward stepwise selection* [Caruana and Freitag, 1994], a wrapper method;
- *Backward stepwise elimination* [Caruana and Freitag, 1994], a wrapper method;
- *Refiner* [Polifroni et al., 2003], a filter method;
- *User model* [Carenini and Moore, 2001], a method based on user input;

For each of the algorithms, we look at two features: the coverage at the leaf nodes for the top two attributes and the information in the paths leading
to those nodes. For purposes of summarization, we would like coverage to be relatively large. In addition, the information contained in the paths to the leaf nodes should be as useful as possible, to make the summaries both generalized and informative. Usefulness or interestingness is, of course, a subjective metric and can only really be measured by the end user. However, for each attribute selection criterion examined, we will construct summary statements using c4.5 rules and use these for insight into the usefulness of the selection criteria.

4.5.2.1 Information gain

Before examining either of the two filter models, information gain and gain ratio, we must first look at how entropy is calculated, since both make use of that metric. Entropy, in information theoretic terms, is a measure of the information required by a system to make a decision. In general, consider a set of objects $X$ that can have one of $n$ values, \{v_{i,1}, v_{i,2}, ..., v_{i,n}\}. Entropy expresses the smallest average number of bits that are needed to transmit a series of values drawn from $X$’s distribution:

$$
Entropy_{cell} = -\sum_{i=0}^{N} p_i \log_2(p_i)
$$

where $p_i$ is the probability of a given symbol. Expressed in log base 2, entropy measures bits of information.

To illustrate entropy, we created a small database of 14 restaurants, eight in Chelsea and six in Soho. Five of these restaurants serve Indian food, four French, three Chinese, and two Moroccan (comprising the cuisine attribute). Six of the restaurants are expensive, four are medium-priced, and four are inexpensive (comprising the price attribute). Finally, each restaurant has a unique phone number. Table 4.2 shows this database in tabular format. In the discussion of information gain that follows, we will be considering the problem of predicting neighborhood from the other attributes.

Of the five Indian restaurants, three are in Chelsea and two are in Soho. The calculation of the entropy for the node “indian” in the tree, would, therefore, be realized as:

$$
-\frac{2}{5} \log_2\left(\frac{2}{5}\right) - \frac{3}{5} \log_2\left(\frac{3}{5}\right)
$$

or .971.

Entropy considers the purity of a node in a tree, and it can be measured in two ways when building a tree. The first way, which we refer to as the
### Table 4.2: Values for cuisine, price, and neighborhood for the example database.

<table>
<thead>
<tr>
<th>Cuisine</th>
<th>Price</th>
<th>Neighborhood</th>
<th>Phone #</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chinese</td>
<td>Inexpensive</td>
<td>Chelsea</td>
<td>555-1001</td>
</tr>
<tr>
<td>Chinese</td>
<td>Medium</td>
<td>Chelsea</td>
<td>555-1002</td>
</tr>
<tr>
<td>Chinese</td>
<td>Medium</td>
<td>Soho</td>
<td>555-1003</td>
</tr>
<tr>
<td>Indian</td>
<td>Inexpensive</td>
<td>Soho</td>
<td>555-1004</td>
</tr>
<tr>
<td>Indian</td>
<td>Medium</td>
<td>Chelsea</td>
<td>555-1005</td>
</tr>
<tr>
<td>Indian</td>
<td>Inexpensive</td>
<td>Chelsea</td>
<td>555-1006</td>
</tr>
<tr>
<td>Indian</td>
<td>Expensive</td>
<td>Soho</td>
<td>555-1007</td>
</tr>
<tr>
<td>Indian</td>
<td>Medium</td>
<td>Chelsea</td>
<td>555-1008</td>
</tr>
<tr>
<td>Moroccan</td>
<td>Inexpensive</td>
<td>Chelsea</td>
<td>555-1009</td>
</tr>
<tr>
<td>Moroccan</td>
<td>Expensive</td>
<td>Soho</td>
<td>555-1010</td>
</tr>
<tr>
<td>French</td>
<td>Expensive</td>
<td>Chelsea</td>
<td>555-1011</td>
</tr>
<tr>
<td>French</td>
<td>Expensive</td>
<td>Chelsea</td>
<td>555-1012</td>
</tr>
<tr>
<td>French</td>
<td>Expensive</td>
<td>Soho</td>
<td>555-1013</td>
</tr>
<tr>
<td>French</td>
<td>Expensive</td>
<td>Soho</td>
<td>555-1014</td>
</tr>
</tbody>
</table>

Instance entropy, considers how decisions are made within each individual leaf node itself. In the second way, which we refer to as class entropy, decisions are considered at the level of the split in the tree, i.e., over the entire attribute, e.g., cuisine. Each is explained in more detail below.

Evaluation criteria such as information gain and gain ratio are applied to each attribute available for classification at each level of tree construction (i.e., any attribute that is not the dependent variable or has not been chosen higher up in the tree). Consider first the attribute cuisine. The first column in Table 4.2 shows the values for cuisine in the toy database. Instance entropy for cuisine, as with all attributes, is computed with reference to the dependent variable, in this case, neighborhood. To determine the instance entropy for cuisine, we begin by computing the entropy at each individual leaf node (i.e., “indian”, “chinese”, etc.).

Entropy values for each of the individual nodes at the level of cuisine (i.e., Indian, French, Moroccan, and Chinese) are computed separately, as above. Each is then multiplied by its probability; in the example with the Indian restaurants, the entropy value for the node containing Indian restaurants would be multiplied by $\frac{5}{14}$, the probability that one of the fourteen restau-
Table 4.3: Values for instance entropy, information gain, class entropy, and gain ratio for the example database used to illustrate these concepts.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Instance Entropy</th>
<th>Information Gain</th>
<th>Class Entropy</th>
<th>GainRatio</th>
</tr>
</thead>
<tbody>
<tr>
<td>cuisine</td>
<td>.972</td>
<td>.013</td>
<td>1.92</td>
<td>.007</td>
</tr>
<tr>
<td>price</td>
<td>.857</td>
<td>.128</td>
<td>1.56</td>
<td>.082</td>
</tr>
<tr>
<td>phone number</td>
<td>0</td>
<td>.985</td>
<td>3.81</td>
<td>.259</td>
</tr>
</tbody>
</table>

rants in the entire set is Indian. To compute the entropy of all the cuisine instances, the remaining entropy values for the other nodes are computed, multiplied by their probabilities and then all four are summed. Instance entropy at the level of cuisine is .972 in our example. Instance entropy values for all attributes in the toy database are shown in Table 4.3.

Entropy can also be calculated at the level of the split itself, i.e., at the level of the attribute/class being predicted (e.g., cuisine or price). At this level, we treat the nodes as classes (i.e., all Indian restaurants). The probabilities are associated with the values of the class, not with how each individual instance predicts neighborhood. In other words, we are looking at the information necessary to classify four cuisines, not 14 individual instances of cuisines. To calculate the class entropy in the example, i.e., the entropy involved in making the splits into the four different cuisines, we calculate the entropy for each of the four possible choices. Given the equation for entropy above, this works out to be

\[
-\frac{5}{14} \log_2 \frac{5}{14} - \frac{3}{14} \log_2 \frac{3}{14} - \frac{4}{14} \log_2 \frac{4}{14} - \frac{2}{14} \log_2 \frac{2}{14}
\]

for the five Indian restaurants, three Chinese, four French, and two Moroccan. The value of the class entropy for cuisine in our toy database is 1.92. Class entropy values for the variables in our example database are shown in Table 4.3.

The two types of entropy illustrated above are used to calculate information gain. Subtracting the instance entropy of the independent variable from the class entropy of the dependent variable neighborhood gives the information gain, i.e., the gain made at the level of cuisine in classifying the dependent variable (.985, in our example). Information gain values for all the independent variables in our example database are shown in Table 4.3.

Information gain favors attributes that have a large number of possible values and prefers splits producing the purest daughter nodes, i.e., nodes
<table>
<thead>
<tr>
<th>Attribute</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>street</td>
<td>1.514</td>
</tr>
<tr>
<td>service quality</td>
<td>0.836</td>
</tr>
<tr>
<td>tube station</td>
<td>0.808</td>
</tr>
<tr>
<td>food quality</td>
<td>0.704</td>
</tr>
<tr>
<td>cuisine</td>
<td>0.498</td>
</tr>
<tr>
<td>price</td>
<td>0.415</td>
</tr>
<tr>
<td>neighborhood</td>
<td>0.351</td>
</tr>
</tbody>
</table>

Table 4.4: Values for information gain computed using all exemplars in the London restaurant domain.

that have a high degree of homogeneity. As we can see, information gain, by favoring such splits, will score highly attributes whose values are unique, or account uniquely for a small set of database tuples. Information gain scores for the set of all data in the London restaurant corpus are shown in table 4.4.

Not surprisingly, when attributes ranked by information gain are used to generate rules, the resulting leaf node clusters are extremely small. Figure 4.2 shows the average coverage of rules derived using the top two attributes scored by information gain. Coverage values clustered towards the left on the x-axis indicate relatively poor generalization, i.e., there are many rules that account for only one restaurant each. For purposes of generalization, higher bars along on the right-hand side of the x-axis would indicate a larger number of restaurants covered by the rules, and, therefore, a larger number of rules to choose from when considering what to say about the dataset. As can be seen, the largest number of rules account for a very small number of restaurants (i.e., one).

The two highest scoring attributes from information gain result in summary statements that are not particularly general (considering that there are approximately 600 restaurants in our dataset). Using the two highest scoring attributes, street and service, and choosing the rules that account for the largest number of restaurants, we obtain summary statements such as the following using information gain: “There are seven restaurants with poor service quality on Fulham Road and six with poor food quality on Upper Street. There are also five with medium food quality on Old Brompton Road”.

---

60
It is doubtful that summary statements such as the ones above capture
generalizations that are of interest to users who are unfamiliar with London,
and, even for a user familiar with London, it is unlikely that the streets that
are chosen are of interest. The use of street as an attribute is very specific and
may miss generalizations that would also incorporate location information,
but that would be more meaningful for both subjects who are new to London
and those that know London well (e.g., neighborhood). Furthermore, the
overall number of restaurants described, 18 out of 594, is small, and might
only be useful to subjects who are very interested in quality of service.

4.5.2.2 Gain ratio

Because values that are scored highly by the information gain metric, such
as street, do not generalize to unseen data very well, they are not useful for
the purpose of classification. As an alternative to information gain, gain
ratio was proposed, as a way of taking into account the number and size of
the daughter nodes in a split. Gain ratio can be seen as a correction, for the
purposes of classification, to deal with the fact that choosing attributes that
score highly on information gain (and entropy) may result in the highest
purity in the resulting daughter nodes, but a great deal of information is
Table 4.5: Values for gain ratio for all tuples in London restaurant domain.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>service</td>
<td>0.419</td>
</tr>
<tr>
<td>food_quality</td>
<td>0.3445</td>
</tr>
<tr>
<td>price</td>
<td>0.2366</td>
</tr>
<tr>
<td>street</td>
<td>0.1835</td>
</tr>
<tr>
<td>tube station</td>
<td>0.1227</td>
</tr>
<tr>
<td>cuisine</td>
<td>0.1124</td>
</tr>
<tr>
<td>neighborhood</td>
<td>0.07</td>
</tr>
</tbody>
</table>

The reasoning behind this measure can be seen by looking at an extreme example of both entropy and information gain, *phone number*, in our toy database. The class entropy value for phone number is 3.81, the highest by far of the attributes for that measure, but the instance entropy is 0. This is because, in using *phone number* to predict *neighborhood*, no practical information (for purposes of classification) is gained, i.e., each value is unique and the decision is unambiguous. Each of the terms in the summation for instance entropy for *phone number*, \( \frac{1}{T} \log_2 \frac{1}{T} \), is zero. Information gain for phone number is, therefore, quite large, .985.

As an alternative to information gain, gain ratio is computed by dividing the attribute entropy at a particular node by the information gain at that node. This has the effect of taking the number and size of daughter nodes into account when choosing an attribute. As we have seen, information gain is quite large for attributes with many unique values, so its use in the equation for gain ratio will tend to minimize gain ratio values for those attributes.

Table 4.3 shows the values for gain ratio computed on the toy database. As these values show, *phone number* remains the highest scoring attribute, even for gain ratio. Computations of gain ratio in extreme cases like *phone number* and other unique identifiers can be (and are) corrected by ad hoc methods in the construction of decision trees. For our purposes, such attributes were already eliminated by the heuristic that eliminated all attributes whose values were unique or unaccounted for in over 50% of our data.

*Gain ratio* looked more promising for the purposes of generalization,
Using gain ratio to determine two highest scoring attributes for rule induction

Figure 4.3: Bar graph showing the average coverage of rules derived using the top-2 attributes scored by gain ratio. The y-axis shows coverage and the x-axis shows the number of rules with that coverage.

and, as can be seen in Table 4.5, attributes scored using this metric have a very different profile from those using information gain. Figure 4.3 shows average coverage of rules derived using the top two attributes scored by gain ratio, food\_quality and service. There are many more rules that cover a large number of restaurants (e.g., over 100). One interesting thing to note, however, is that, with just five values to choose among both the top two attributes (i.e., “very poor”, “poor”, “medium”, and “good”, and “very good”), and with those values closely correlated, a very limited number of rules account for all combinations of these attributes.

The small number of rules is not necessarily a problem, if those rules are particularly informative. Using just the two highest scoring attributes from gain ratio, and choosing the rules that account for the largest number of restaurants from the entire set of London restaurants, the following summary statements can be constructed: “There are 149 restaurants with medium food quality and medium service and 125 restaurants with poor food quality and poor service. There are also 114 restaurants with unknown food quality and unknown service.”

There are three things about the summary statement above that are of concern. First, if a particular user was not concerned with food\_quality or service, the summary might be uninteresting and, furthermore, fail to cap-
ture some other generalization that might be valuable for a first-time visitor to London, for example. Secondly, the third sentence in the summary, while capturing a generalization about a large number of restaurants in London, seems to maximize for *uninformative* attributes, violating the second evaluation criterion, i.e., that the generalizations should be informative to users. Finally, the use of these two attributes captures *correlations*, but possibly not informative *generalizations*. An examination of the rules themselves (three of which are indicated by the summary statement above), show that equivalent values are the most common pattern, i.e., not surprisingly, the same level of *food_quality* and *service* are often found in a single restaurant.

### 4.5.2.3 Forward stepwise selection

*Forward stepwise selection*, a wrapper method of attribute selection is examined next. Recall that a wrapper method makes use of the algorithm itself to decide which set of attributes will maximize performance. We are able to examine wrapper methods of attribute selection because we know the algorithm we are going to use to classify our data, i.e., decision tree induction. For the experiments described below, c4.5 was used, and we score attributes based on their ability to maximize the performance of that algorithm.

In all cases, we are attempting to learn something about *combinations* of attributes with these metrics. To apply these metrics to classification, we would choose the set that produced the lowest error rate overall. As our experiments will show, that set is comprised of five attributes for *forward stepwise selection* and six attributes for *backward stepwise elimination*. Because we are interested in combinations of just two, or at most three attributes, we are using these algorithms to narrow the choice to a subset of 2 attributes to see what summary statements they provide.

Forward stepwise selection identifies the best set of attributes, beginning with a single attribute and working up to the full set of attributes. The number of attributes $N$ is incremented by one at each iteration. At each iteration, all possible combinations of $N$ attributes are examined.

To begin with, we compute $N$ error rates from running the classifier $N$ times with a single attribute, once for each of the $N$ attributes. We then select the single attribute that, on its own, produced the lowest error rate using data from the London restaurant domain and c4.5. Once that has been chosen, we iterate through the remaining groupings of attributes to find the ones that, in combination with the one(s) previously chosen, produce the
lowest error rate. At every iteration, one attribute is chosen to add to the assembled set, based on its effect on error rate.

Table 4.6 shows the error rate results when applying forward stepwise selection to the full set of restaurants in the London restaurant domain, using c4.5. For the purpose of summary construction, we chose the first two attributes introduced using forward stepwise selection, `price` and `food_quality`. Pair-wise, these are the two attributes with the lowest error rate according to this algorithm.

Figure 4.4 shows average coverage of rules derived using the attributes `price` and `food_quality`. Coverage is relatively good, above one hundred restaurants for some rules, but the number of rules that account for these large numbers is quite small. The profile is very similar to that for gain ratio (see Figure 4.3). As with the two top-scoring attributes for gain ratio, `food_quality` and `service`, the two here have a limited number of highly correlated values. A small number of rules account for all possible combinations, with equivalently ranked values co-occurring.

Using the top two attributes identified by forward stepwise selection to generate rules, and choosing from the resulting set the rules that account for the largest number of restaurants, we produced the following summary statements: “There are 140 medium-priced restaurants with medium service quality and 109 inexpensive restaurants with poor service quality. There are also 54 expensive restaurants with unknown service quality.”

Although `price` and `service` combined might provide a more informative
Using forward stepwise selection to determine two highest scoring attributes for rule induction

Figure 4.4: Bar graph showing the average coverage of rules derived using the top-2 attributes scored by forward stepwise selection. The $y$-axis shows coverage and the $x$-axis shows the number of rules with that coverage.

general summary to a user with no expressed preferences, we were still not satisfied with this as an attribute selection criteria, especially given that we were still producing summary sentences like the third one above, containing information about attributes with unknown values.

4.5.2.4 **Backward stepwise elimination**

The second wrapper method of attribute selection examined was *backward stepwise elimination*. In this method, all attributes are used to compute a baseline error rate and then attributes are removed iteratively to find the set of $N$ attributes that yields the lowest error rate at a steadily decreasing value of $N$. We consider the final two attributes identified as functionally equivalent to the first two attributes discovered using forward stepwise selection, since both algorithms attempt to rank attributes by their effect on classifier performance. Even though error rates are, for the most part, high when using smaller numbers of attributes, the attributes thus identified are optimized by their effect on classification error rate.

Table 4.7 shows error rate values for backward stepwise elimination in tabular format. Figures 4.5 displays a histogram of average coverage of rules derived using the top-2 attributes scored by backward stepwise elimination. Figure 4.5 reveals the same effect noticed with the two top-scoring values from forward stepwise selection and gain ratio. With backward stepwise
Table 4.7: Values for backward stepwise elimination for all tuples in London restaurant domain.

Elimination, \textit{food\_quality} and \textit{decor} are the highest; they are also both attributes for which there are only five possible values. Consequently, a single rule accounts for all of the pair-wise combinations of these two attributes.

A positive aspect of backward stepwise elimination is that there are rules that account for a large number of restaurants in the database. A summary statement derived from the top-2 attributes from backward stepwise elimination applied to our set of London restaurants is “There are 121 restaurants with medium food quality and medium decor and 114 restaurants with unknown food quality and unknown decor. There are also 80 restaurants with poor food quality and poor decor.” However, in addition to the dubious value of \textit{decor} as an attribute in a summary statement \footnote{When we elicit user rankings for these attributes for experimental subjects, \textit{decor} ranked last, on average, across all user preferences (Section 5.2.3.)} we also have the problem seen with other cover and wrapper models, that of summary statements about attributes that have unknown values.

\subsection*{4.5.2.5 Change in Error Rate in Forward Stepwise Selection and Backward Stepwise Elimination}

In investigating these attribute selection algorithms, our goal is to identify attributes that will be useful in summary statements about backend data. Wrapper algorithms, with their stepwise nature, give insight into attributes whose addition (in the case of Forward Stepwise Selection) or removal (in the
Using backward stepwise elimination to determine two highest scoring attributes for rule induction

![Bar graph showing the average coverage of rules derived using the top-2 attributes scored by backward stepwise elimination. The y-axis shows coverage and the x-axis shows the number of rules with that coverage.]

Figure 4.5: Bar graph showing the average coverage of rules derived using the top-2 attributes scored by backward stepwise elimination. The y-axis shows coverage and the x-axis shows the number of rules with that coverage.

case of Backward Stepwise Elimination) have the greatest effect on error rate. We hypothesized that, if the addition or removal of a particular attribute significantly changes error rate, perhaps that attribute captures something interesting about the underlying data, and might, therefore, be useful for summarization statements.

The two attributes from Forward Stepwise Selection that made the greatest change in error rate were food quality and tube station. When food quality was added to the set of attributes at Iteration #3 in Table 4.6, error rate dropped by 18.5% absolute. When tube station was added at Iteration #7, error rate dropped by 27.1% absolute. We investigated the generalizations possible using these attributes, selected by their effect on the change in error rate.

Table 4.6 shows a histogram of the coverage in number of restaurants for these two attributes as a function of the number of rules. As with information gain, a large number of unique rules account for a small number of restaurants (i.e., many restaurants fit unique patterns for these attributes). A summary statement constructed using these two attributes from the London database, maximizing for coverage, is: “There are 11 restaurants with poor food quality near the South Kensington tube station and 10 with medium food quality near the Green Park tube station. There are also 9 with medium food quality near the Sloane Square tube station.”

Although summaries constructed using tube station seem more general
Using largest deltas in forward stepwise selection to determine 2 attributes for rule induction

Figure 4.6: Bar graph showing the average coverage of rules derived using the top-2 attributes scored by the delta in forward stepwise selection. The y-axis shows coverage and the x-axis shows the number of rules with that coverage.

than those constructed using street (cf., information gain), in combination with food_quality, the summaries account for only 20 restaurants in total from a set of close to 600. Furthermore, food_quality itself does not seem like a useful attribute to mention in a broad summary of all restaurants in a database, unless a user is known to have a strong preference for good food.

The two attributes that made the greatest change in error rate using Backward Stepwise Elimination were decor (removed at Iteration #7, with a drop of 20.5% in error rate) and cuisine (removed at Iteration #6, with a drop of 15.6% in error rate). Table 4.7 shows the coverage in number of restaurants for these two attributes, again as a function of the number of rules that account for that coverage.

The pattern here is again very similar to that found with information gain, i.e., an attribute with a relatively large number of attributes (i.e., cuisine) is paired with one with a small number (i.e., decor). A summary statement constructed using these two attributes from our London database is: “There are 48 restaurants serving Italian cuisine with medium decor and 41 serving British cuisine with medium decor. There are also 39 restaurants serving European cuisine with medium decor.”

The degree of generalization using these two attributes is quite good, i.e., 128 restaurants mentioned among the three clusters. However, as we noted with the summaries constructed using information gain, the use of decor
Using largest deltas in backward stepwise elimination to determine 2 attributes for rule induction.

Figure 4.7: Bar graph showing the average coverage of rules derived using the top-2 attributes scored by the delta in backward stepwise elimination. The y-axis shows coverage and the x-axis shows the number of rules with that coverage.

may not be something that most users will find interesting. The preference for decor as an attribute here could be due to the fact that the values for this attribute were spread more evenly across the restaurants in our database than the values for price or food quality (which were binned in similar ways).

4.5.2.6 Issues with automated metrics

Up to this point, we have examined a set of attribute selection metrics that are derived from either how these attributes score within the dataset itself or on how combinations of these attributes affect error rate using a particular classification algorithm. Filter methods such as information gain and gain ratio either produce combinations of attributes that do not generalize particularly well (e.g., information gain, where the largest grouping found among 600 restaurants was 7), or they tend to pinpoint attributes that are highly correlated but possibly not of general interest to users (e.g., decor). More problematic than what the attribute is, however, are the values that a particular attribute may have. Whether or not decor or service are attributes users want to hear about in a summary, it is very unlikely they will want to hear about “restaurants with unknown values for decor” or “restaurants whose service quality is not in the database.”

Intuitively, it seemed that attributes selected via wrapper methods, which identify those that have a significant effect on error rate, might be more use-
ful in constructing summaries. With the wrapper methods, attributes such as *cuisine* and *tube station* (which roughly corresponds to *neighborhood*), appear among top scorers. However, both wrapper methods of attribute selection have a serious drawback: they are extremely costly to compute. Each requires multiple iterations through the algorithm, and for backward step-wise elimination and the algorithms that considered the deltas in error rate, they cannot be considered complete until all iterations have been performed. For use in a real-time system, where attributes are chosen that best reflect the focal information at each turn, this was a costly and serious drawback. Some, but not all pre-computation of values would be possible, but additions of new data would require that values be re-computed. For the latter stages in a dialogue, where multiple constraints may have been introduced, pre-computing all combinations could be impractical.

For the purpose of constructing generalizable, informative responses, the automated algorithms for attribute selection found in the literature and applied to decision tree induction were not satisfying. We needed to find attributes that would both generalize across the data and inform users about attributes whose values might be of interest to them. We, therefore, returned to the original Refiner method, to adapt it to perform attribute ranking, and we also examined the use of a user model to select attributes in a way tailored to user preferences.

### 4.5.2.7 “Refined” Refiner

The Refiner algorithm (see Section 2.6) was designed to find attributes that produce summaries accounting for relatively large numbers of database records. It did so in a heuristic way, and it could not be guaranteed to find a set of attributes for all subsets of data. In order to be generalizable across all datasets, the Refiner had to be adapted to incorporate a scoring mechanism. As a general principle, the scoring function should always return a ranked list of attributes for each dataset it evaluates. Furthermore, the scoring mechanism should find attributes that satisfy two criteria: they should generalize well over the data (i.e., produce large clusters of restaurants) and they should contain values of interest to users.

To address the first criteria, the scoring mechanism we developed began with a basic idea from the Refiner, i.e., for attributes to generalize well, they must self-cluster. The scoring mechanism, therefore, evaluates the most frequently occurring values for a particular attribute as a percentage of the
To address the second criteria, a penalty was incorporated into the scoring mechanism and applied to attributes with a large number of unknown values.

The ranking function itself is quite simple. As with the original Refiner, the ratio of the top four most frequently occurring values to the total number of values is calculated for each set of focal information. Unlike the Refiner, however, there is no \textit{a priori} threshold set on this ratio. In addition, the ratio of database instances accounted for by the single most popular value for each attribute to the total is calculated. Table 4.8 shows examples of these values for two of the attributes from the London restaurant database, \textit{price} and \textit{food quality}.

The total accounted for by the top-four most frequently occurring values for \textit{price} is 593 (out of 594), .998 expressed as a ratio. This number is used as the score for an attribute that has no unknown values among the top-four most frequently occurring values. The total accounted for by the top-four most frequently occurring values for \textit{food quality} is 573, also a substantial number of the total, .96 expressed as a ratio. However, the third highest ranking value for \textit{food quality} is “unknown”, and, in cases such as these, where the top-ranking values include the value “unknown”, we attempt to calculate the effect of the unknown values on our ability to generalize, using the following computation:

\[
\frac{\text{numTopOne}}{\text{totalNum}} - \frac{\text{numUnks}}{\text{totalNum}}
\]

where \text{numTopOne} is the number of values in the highest scoring attribute, \text{numUnks} is the number of values that are unknown in the top four, and \text{totalNum} is the total number of tuples in the database.

The first ratio is the total accounted for by the single most frequent value. For \textit{food quality} in our example, this number is 229 out of 594, or .386. The second ratio represents the percentage of unknown values in the top four, 117 out of 594, or .196, again using \textit{food quality}. When an unknown value is in the top four values, the difference between these two ratios is used as the score for the attribute. This effectively penalizes attributes with unknown values in the top four, by an amount that reflects how extensively they are represented in that set. For \textit{food quality}, this score is .386 - .196, or .19.

Table 4.9 shows values ranked by the revised Refiner method for all restaurants in London. Figure 4.8 shows the coverage of these attributes as a function of the rules generated using them. Although we still have quite a few...
<table>
<thead>
<tr>
<th>Rank</th>
<th>Price</th>
<th>Food quality</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Value</td>
<td>Number</td>
</tr>
<tr>
<td>1</td>
<td>medium</td>
<td>243 (.409)</td>
</tr>
<tr>
<td>2</td>
<td>inexpensive</td>
<td>187</td>
</tr>
<tr>
<td>3</td>
<td>expensive</td>
<td>138</td>
</tr>
<tr>
<td>4</td>
<td>very expensive</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 4.8: Example of refiner ranking computation for *price* and *food quality* in the London restaurant domain. The number in parenthesis next to the highest scoring values is the ratio of the number of tuples with that value to the total number of tuples.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>price</td>
<td>0.998</td>
</tr>
<tr>
<td>cuisine</td>
<td>0.522</td>
</tr>
<tr>
<td>neighborhood</td>
<td>0.259</td>
</tr>
<tr>
<td>food quality</td>
<td>0.19</td>
</tr>
<tr>
<td>decor range</td>
<td>0.182</td>
</tr>
<tr>
<td>tube station</td>
<td>0.175</td>
</tr>
<tr>
<td>service</td>
<td>0.172</td>
</tr>
</tbody>
</table>

Table 4.9: Values for Refiner for all tuples in London restaurant domain.
Using refiner to determine 2 highest scoring attributes for rule induction

Figure 4.8: Bar graph showing the average coverage of rules derived using the top-2 attributes scored using the modified Refiner algorithm. The $y$-axis shows coverage and the $x$-axis shows the number of rules with that coverage.

rules that cover only one restaurant each, there is relatively broad coverage for individual rules, up to 49 restaurants. An example summary constructing using the Refiner method for all restaurants in London is: “There are 49 medium-priced Italian restaurants in London and 40 medium-priced European restaurants. There are also 36 medium-priced British restaurants.”

The original Refiner was designed to identify attributes whose values produce large clusters. The addition of a scoring function makes the bias towards attributes that have many values among the most frequently occurring ones algorithmic. Furthermore, by incorporating a penalty applied to attributes whose top-scoring values include many unknown values, the Refiner also now targets informative attributes. However, as a generic function, the Refiner may miss attributes that are of interest to particular users, an important consideration in dialogue systems (see Section 2.4). To obtain a set of attributes tailored for particular users, we turned to user modelling, and, in particular, Multi-Attribute Utility Theory.

4.5.2.8 User model

Information about user preferences has long been used in interactive systems and shown to measurably enhance the user experience [Carenini and Moore, 2006, Walker et al., 2002a, Komatani et al., 2003, Elzer et al., 1994, Thompson and Goker, 2000]. It has also been shown to improve the discovery of useful association rules [Chen and Wu, 2006]. We hypothesized that user
model information could help us identify content that would be of interest in the context of our summary construction methods.

Preferences are typically associated with individual values of the attributes the system knows about, e.g., cheap (price) or Mayfair (neighborhood). Knowing users’ preferences in regard to these attributes enables a system to evaluate the usefulness of individual database tuples, or groups of tuples for presentation to the user [Keeney and Raiffa, 1976, Carenini and Moore, 2001] (See Section 4.5.5 for the way in which user model information is used to score content selected from our database.)

There are multiple ways to gather information about user preferences in interactive systems, among them: infer it from ongoing interactions [Thompson and Goker, 2000]; acquire it during the interaction by explicitly eliciting user feedback about choices presented to them [Linden et al., 1997]; or elicit the information in a separate process and store it for use when enrolled users engage with the system [Walker et al., 2002a, Carenini and Moore, 2006].

Given the fact that we did not have a live system with which to gather data, we used the third method, i.e., elicit information about users in an explicit pre-processing phase. Before doing so, we first narrowed the selection of attributes further, to make the scope of the task easier to understand for our subjects. Recall that, after removing attributes whose values were unique within the corpus (e.g., phone number) and attributes that applied to fewer than 50% of our restaurants, the following set of attributes remained and were used for the experiments described above:

- **Cuisine**
- **Decor**
- **Food quality**
- **Price** (quantized into three bins: inexpensive, medium-priced, and expensive
- **Service**
- **Location/neighborhood**
- **Tube station**
- **Street name**
From these attributes, we chose six to use in summaries containing user model information: *cuisine, decor, food quality, price, service* and *neighborhood*. We conflated three attributes, *tube station, street, and direction*, to one parameter associated with restaurant location, reasoning that *neighborhood* would be an attribute that users would be more likely to have knowledge about, and a preference for, than either *street* or *tube station*. This seemed especially true of tourists who might not be familiar with either the street or subway map of London. *Neighborhood* captures location information in a way that native Londoners understand, as well.

Although *neighborhood* has been used elsewhere as the sole location attribute in a user model [Walker et al., 2002a], we were aware that its selection from among the attributes we had available in our restaurant domain could be construed as introducing domain-specific knowledge into a set of algorithms we wished to keep as domain-independent as possible. However, we were able to use c4.5 to induce an ontology within our data that enabled us to automatically decide on *neighborhood* as the attribute to use among the four in consideration (i.e., *street, tube station, and neighborhood, and direction*). We ran c4.5 on every pair of attributes in our database that had unequal numbers of unique values, in each case using the attribute with a larger number of values to predict the one with a smaller number of values. In this way, we hoped to learn which attributes were “redundant,” with each other, reasoning that, if an attribute with many values predicted very well an attribute with a smaller number of values, the latter value was more general. In these pair-wise comparisons, the use of *street* to predict *tube station* and the use of *tube station* to predict *neighborhood* were the only two pairs to achieve error rates below 20% (as compared to an average error rate of 56.3% for other predictor pairs such as *cuisine* used to predict *price range* or even *neighborhood* used to predict *price range*). The lowest error rate, 0.0%, was achieved when using *neighborhood* to predict direction, and we did use both attributes in acquiring our user model, allowing users to substitute a *direction* preference for a specific *neighborhood* one (see below).

Users were asked to rank the six attributes using a standard procedure called SMARTER [Edwards and Barron, 1994] that has been used previously in the restaurant domain [Walker et al., 2002a] and shown to be an effective way of eliciting preferences to build a model using Multi-Attribute Utility Theory [Carenini and Moore, 2006]. Recruited subjects worked through a series of webpages, the first one of which contained the following scenario.
Imagine that you've had the bad luck to have to eat at the worst possible restaurant in the city. The price is 50 pounds per head and you don't like the type of food or the neighborhood. The food itself is dreadful and the decor is appalling. Furthermore, the service is terrible. Now imagine that a good fairy comes along who will grant you one wish. You can use that wish to improve the restaurant to the highest level, but only on one of the following dimensions. Which dimension would you choose?

- Food quality
- Service
- Price
- Location
- Decor
- Cuisine type

When a subject selected an attribute on the first page, that attribute was recorded as his/her top preference among restaurant attributes. A new page was displayed to the user with the same scenario but with the previously chosen attribute no longer among the possibilities for improvement. By working through all attributes until there was just one left, subjects provided a ranking for the six attributes.

After providing rankings, subjects were also asked to look at the 56 cuisine choices available in the London restaurant domain and mark those that they particularly liked and particularly disliked. On average, users chose 10 cuisines they preferred and 7 cuisines they disliked. We then asked users to rank the London neighborhood choices in the database in a similar way, by those they preferred and those they disliked. For neighborhood preferences, we allowed subjects to choose direction (i.e., north, south, east, west, or central) as their preference if they were unfamiliar with London neighborhoods, or if their preferences could naturally be expressed in those terms. In over 80% of the cases, subjects chose this method of specifying neighborhood preferences. There was an unambiguous many-to-one mapping in the database between neighborhood and direction, so we were able to recover specific neighborhoods from such preferences.
<table>
<thead>
<tr>
<th>User</th>
<th>FQ</th>
<th>SVC</th>
<th>DEC</th>
<th>Cost</th>
<th>Nghd</th>
<th>CUI</th>
<th>CUI</th>
<th>Nghd Likes</th>
<th>Nghd Dislikes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJB</td>
<td>.41</td>
<td>.10</td>
<td>.06</td>
<td>.03</td>
<td>.24</td>
<td>.16</td>
<td>Central London</td>
<td>Chinese, French, Japanese, Italian</td>
<td></td>
</tr>
<tr>
<td>AF</td>
<td>.24</td>
<td>.16</td>
<td>.03</td>
<td>.41</td>
<td>.06</td>
<td>.10</td>
<td>Central London, South London</td>
<td>Greek, Indian, Mexican, Vegetarian</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.10: User models for subjects AF and EJB. FQ = food quality, SVC = service, DEC = decor, Nghd = neighborhood, CUI = cuisine.

Table 4.10 shows an example of user models gathered from two experimental subjects. Included in the table are the weights assigned to each attribute and preferences and dislikes for neighborhood and cuisine. Across 16 subjects recruited for the experiment described in Chapter 5, no two user models were identical, when the categorical preferences for neighborhood and cuisine are taken into account. For constructing responses, the two attributes available in the context and scored the highest by preference are used. The profile of coverage by number of rules will, of course, vary from user to user as the two top-scoring attributes differ across users. In section 4.5.5, we describe how user preferences define a scoring metric based on utility, and discuss the relationship between coverage and the resulting utility scores.

### 4.5.3 Maximizing number of rules

Attribute selection alone is not the only method employed in the literature to improve classifier performance. Another method, involving randomizing the selection of attributes at branch (i.e., split) points has also been shown to be effective. By introducing randomness at branch points in tree construction, i.e., either by choosing the split attribute at random or choosing different selection criteria at different splits, classification performance has been improved, especially when the technique was used in combination with boosting [Dietterich, 2000a,b, Pappa et al., 2002]. Although the main interest here is not the quality of the classifier, these techniques provided insight into the way c4.5 worked, and led to experimentation in increasing the richness of a ruleset, i.e., the number of rules describing a dataset that is available for scoring and selection.

To introduce randomness in the implementation of a decision tree classifier, we chose not to modify the algorithm itself. Rather, we ran c4.5 in
Figure 4.9: Tree representation of rules derived from three runs of c4.5 using a different attribute as the dependent variable.

Figure 4.9 shows an example of three rules derived running c4.5 iteratively. Each path represents a different iteration, with a different attribute (from among three) used each time as the dependent variable (price range, location, and cuisine, respectively). The dataset was drawn from the subset of restaurants labeled with the value central for the attribute direction in the London restaurant domain. Each of the rules in Figure 4.9 is unique to the iteration in which it was found. Even though the same data are available with each iteration, because of the different splitting options introduced by varying the dependent variable, different groupings of attributes are discovered.

In experiments with the full set of restaurants and several subsets, we found an average of 42.6% more unique rules when running the system in iterative mode. In addition, on the full dataset of close to 600 restaurants, using three attributes, each iteration of c4.5 took 1 millisecond of real time on a Dell Pentium processor running Linux. Given the speed with which these
iterations could be calculated, combined with the enhanced content they made available, iterative c4.5 was used for creating rules in the experiments described below.

4.5.4 Constructing clusters

The original Refiner method of response construction (see Section 2.6) included both a heuristic means of scoring attributes and a way of describing those attributes within a summary statement, i.e., by listing the values for each attribute singly. We modified the scoring function, but maintained the description method for attributes and their values. We refer to this method as the singleton method of response construction. It represents the simplest algorithm for producing clusters.

In the singleton method of response construction, the top three attributes are used, scored by either the user model or the Refiner ranking method. When using the Refiner method of ranking, attribute values are ordered by frequency and the specific values used in the response are determined by calculating the percentage of the total accounted for by each, starting with the most frequent. First, the algorithm looks to see if the top-four values account for all exemplars. If so, it creates an attribute-dependent summary statement indicating complete coverage of all values for that attribute (e.g., “All price ranges are represented”). Failing that, the algorithm examines values, in order of frequency, to find a set that account for at least 80% coverage, in order to create a statement using the quantifier “most” (e.g., “Most are inexpensive or medium-priced”). If the top four values do not comprise 80% of the total, the algorithm then tries to find a set that comprises more than 50% of the total, for a statement using “many” (e.g., “Many serve Italian or European cuisine”). If a group of that size is not found, the algorithm simply lists the top three values for that attribute using the quantifier “some” (e.g., “Some of the neighborhoods represented are Chelsea, Knightsbridge, and Soho”). Table 4.11 shows parameters associated with, and an example of, a response constructed using the Refiner method of attribute selection and the singleton method of clustering.

When using a user model, the single top-scoring value for each of the three attributes is mentioned in an utterance or clause of its own, e.g., “There are 6 restaurants with good food quality. There are 21 inexpensive restaurants and 12 restaurants with good service.” Table 4.12 shows parameters associated with, and an example of a response constructed using
Table 4.11: Parameters associated with, and example of a response using the Refiner method of attribute selection and singleton clustering.

The alternate method for clustering attributes is the use of decision tree induction to find associations among the chosen attributes. We refer to this as the associative method of response construction. The way association rules are created was explained in Section 4.5.3 and the way the rules are scored is explained in Section 4.5.5.

### 4.5.5 Cluster Scoring

The final parameter in our response construction algorithm is the method used to score the association rules discovered in our data. As mentioned, we are interested in maximizing two rather impressionistic parameters, the degree of generalization in the response, and the degree of “interestingness” of the information. For these, we examine two metrics, one directly measuring the number of database tuples accounted for in a particular response, and the other measuring the degree to which a particular response conforms to a user’s preferences. When using the associative method of response construction, these measures are applied to the rules discovered using c4.5. When using the singleton method of response construction, these measures are applied to the attribute values themselves. Each method is explained in more detail below.
I know of 594 restaurants in London. There are 70 restaurants with good to very good food quality. There are 105 Italian restaurants and 188 inexpensive to very inexpensive restaurants.

Table 4.12: Parameters associated with, and example of a response using the user model method of attribute selection and singleton clustering.

**4.5.5.1 Coverage**

For responses using association rules, the degree of generalization is measured using a metric known as *coverage* (also called *support*) [Witten and Frank, 2000]. *Coverage*, first mentioned in Section 4.5.1, measures the number of instances (i.e., leaf nodes) that are correctly classified by the rule. For responses constructed using the associative method, we use *coverage* as the baseline metric for rule scoring. In other words, given no user model, we order rules by the number of instances that they correctly classified. This is consistent with the desire to maximize a notion of generalization in presenting summary information to users. Table 4.13 shows parameters associated with, and an example of a response constructed using the Refiner method of attribute selection and the associative method of clustering.

**4.5.5.2 User model**

Given knowledge about ranked user preferences (see Section 4.5.2.8), Multi-Attribute Utility Theory (MAUT) provides a straightforward way to operationalize the use of those preferences to tailor information to a specific user [Keeney and Raiffa, 1976]. The key insight behind MAUT is that users overall preferences are determined by the interplay of multiple individual factors. By knowing how users feel about each of those factors, weights can
Refiner ranking, associative clustering

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Attributes</th>
<th>Score</th>
<th>Num. values used per statement</th>
<th>Value selection method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>price</td>
<td>0.998</td>
<td>1</td>
<td>Coverage</td>
</tr>
<tr>
<td></td>
<td>cuisine</td>
<td>0.522</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table 4.13: Parameters associated with, and example of a response using the Refiner method of attribute selection and associative clustering.

be assigned that lead to an overall score for a particular entity.

The use of Multi-Attribute Utility Theory to derive a utility score for particular database instances has been explored elsewhere [Walker et al., 2002a, Carenini and Moore, 2006, Thompson and Goker, 2000, Jameson et al., 1995]. This resulting score is used as a means of choosing content in recommender systems, where the goal is to find a small group of individual items that most closely conform to user preferences. In the field of data mining, systems have been developed that employ information elicited from users to filter association rules and discover ones that reflect the interest of the data consumer [Liu et al., 1999, Srikant et al., 1997, Chen and Wu, 2006]. These systems query users interactively and filter rule results based on input that targets particular values for attributes. In our work, we are interested in scoring groupings of attribute/value pairs (i.e., association rules) that describe generalized sets of database instances, rather than specific records from the database. To do so, we use MAUT in the same way as it has been used in recommender systems, but apply it to a different set of entities. As far as we know, this is the first attempt made to employ MAUT to operationalize the selection of association rules to maximize for “interestingness”.

Converting symbolic values to scalar

In order to weigh the values for each attribute in the user model (i.e., cuisine, decor, food quality, location price and service) they must be assigned
Table 4.14: Mappings from symbolic values in the database used to describe records and scalar values used to score association rules.

<table>
<thead>
<tr>
<th>Symbolic value</th>
<th>Scalar value</th>
</tr>
</thead>
<tbody>
<tr>
<td>“very poor” /”very expensive”</td>
<td>.2</td>
</tr>
<tr>
<td>“poor” /”expensive”</td>
<td>.4</td>
</tr>
<tr>
<td>“medium”</td>
<td>.6</td>
</tr>
<tr>
<td>“good” /”inexpensive”</td>
<td>.8</td>
</tr>
<tr>
<td>“very good” /”very inexpensive”</td>
<td>1</td>
</tr>
</tbody>
</table>

scalar values. Recall that values for decor, food quality, price and service were scalar in our original database. However, for generating responses to users, we quantized those values into three bins, corresponding to small, medium, and large, with outliers at either end forming two further bins (see Section 4.4).

The values assigned to these bins were maintained for scoring our clusters, rather than reverting back to the original scalar values. This decision was made based on two criteria. In the first place, these quantized values were used for generating descriptions to users, i.e., restaurants were described as “inexpensive” rather than “having average prices between 10 and 15 pounds”. It seemed correct to use the same terms to score the attributes themselves as were used to describe them. Secondly, our interest was in describing large clusters of our data, and the quantized values were more generalizable than the individual scalar values/ranges used in the original database.

Each of the five symbolic values for decor, food quality, price and service was converted into a scalar value spaced evenly between .2 and 1.0. Table 4.14 shows this mapping. Values for cuisine and neighborhood were represented as .1 if they were on the list of values the subject disliked, .9 if they were on their preferred list, and .5 if the subject did not rate the attributes at all. This is similar to how values were quantized in [Walker et al., 2002a].

**Weighing attributes**

As explained in Section 4.5.2.8, we asked subjects to provide ranked preferences for six attributes from the restaurant domain. These rankings were converted to weights ($w$) for each attribute, using the formula below, which
Table 4.15: Weights assigned to attributes based on their position in the user model ranking.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.41</td>
</tr>
<tr>
<td>2</td>
<td>0.24</td>
</tr>
<tr>
<td>3</td>
<td>0.16</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
</tr>
<tr>
<td>5</td>
<td>0.06</td>
</tr>
<tr>
<td>6</td>
<td>0.03</td>
</tr>
</tbody>
</table>

guarantees that the total sum of the weighted attributes sum to 1:

$$w_k = \frac{1}{K} \sum_{i=k}^{K} \frac{1}{i}$$

where $K$ equals the number of attributes in the ranking. Table 4.15 shows the weights assigned to each attribute by its rank in the user model.

The weights from the user model and the scalar values for the attributes in the rules are combined using a simple additive function, to yield an overall utility $U$ for an entity indexed by $h$:

$$U_h = \sum_{k=1}^{K} w_k u_k(x_{hk})$$

In our case, $U$ is computed on associations rules derived from a decision tree (see 4.5.4). The overall utility $U$ is computed for each instance of an association rule. Individual weights ($w$, as computed above) are applied to the values for each attribute in question ($k$). The function $u$ maps particular attribute values to utilities, for each value $x_{hk}$ in the association rule.

Table 4.16 shows an example of an association rule scored for a user whose first two preferences in the restaurant domain are price and food quality, respectively (user AF from Table 4.10). For the first rule in Table 4.16, the price attribute is very cheap, which is scored as 1, the highest value it can have. Food quality is relatively low in this rule, with a symbolic value of medium which translates into a scalar value of .6. To score the rule, the two weighted values are summed: (.41 * 1) + (.24 * .6) = .554. The second rule in Table 4.16 shows the trade-off when the second-highest ranked attribute has a very good score. In the second rule, the food quality attribute has its highest possible value, very good, which translates to a scalar value of 1.
Figure 4.10: Graph showing the coverage of rules describing all restaurants in the London database as function of their utility scored for user $AF$.

The value for the price attribute, expensive, scores relatively low, .4, since this user is concerned about price. Summed together, $(.41 \times 1) + (.24 \times .4)$ gives a utility score for this rule of .404.

**Coverage as a function of utility**

When evaluating the other attribute selection criteria above, we looked at the number of rules generated at various levels of coverage, i.e., at the number of restaurants accounted for by the rules. For user model scoring, we also look at coverage, but this time as a function of the user model score, i.e., we look at the number of restaurants that are covered at the varying utility levels defined by the user model score. Just as the number of rules at varying levels of coverage measured generalization given the attributes used, the number of rules at varying levels of utility measures generalization given user model utility.

Figure 4.10 shows restaurant coverage for all restaurants in the London Preferences: (1) food quality (2) price

<table>
<thead>
<tr>
<th>Rule1:</th>
<th>price = v. inexpensive + food quality = medium</th>
<th>Scoring:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule2:</td>
<td>price = expensive + food quality = very good</td>
<td>Scoring:</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Rule1:</th>
<th>price = v. inexpensive + food quality = medium</th>
<th>Scoring:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rule2:</td>
<td>price = expensive + food quality = very good</td>
<td>Scoring:</td>
</tr>
</tbody>
</table>

Table 4.16: Weights assigned to attributes based on their position in the user model ranking.

Table 4.16: Weights assigned to attributes based on their position in the user model ranking.
Figure 4.11: Graph showing the coverage of rules describing all restaurants in the London database as function of their utility scored for user EJB.

database as a function of user model utility score for user AF. Figure 4.11 shows the same for user EJB whose top ranked preferences were food_quality and cuisine (see Table 4.10). The top two preferences in the user models for these two subjects represent three of the four attributes ranked most highly across all subjects. Furthermore, user AF has highly ranked attributes whose values are drawn from a small set of possibilities (i.e., at most five for both price and food_quality), whereas user EJB highly ranked one attribute cuisine that has 56 possibilities.

As both figures show, there are a large number of restaurants at the upper end of the coverage scale with relatively high user model scores. Although the highest user model score for AF accounts for just a single restaurant, the second highest score for that user accounts for a group of 78 restaurants. There are 85 restaurants in total in the three groups at the highest utility level for user AF. The total number of restaurants at the three highest utility values for user EJB is higher, 137, partly because subject EJB’s cuisine preferences were well-represented in the database. Across all subjects, the average number of restaurants accounted for by the three highest utility values was 41.

Table 4.17 shows examples of rules with attributes chosen, and utility scored by these subjects’ user models. For subject AF, the three rules with the highest utility are unique for the values contained within each. For subject EJB, the third rule, referring to “medium” food_quality and “Italian” cuisine scored equivalently with eight other rules at that same utility. In cases such as these, we use coverage to choose one rule, if necessary. For
Subject: **AF**; Preferences: price, FQ

<table>
<thead>
<tr>
<th>Num. rests.</th>
<th>Rule</th>
<th>Weight Value</th>
<th>Weight Value</th>
<th>Weight Value</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>price = v. inexpensive + FQ = medium</td>
<td>.41 * 1</td>
<td>.24 * .6</td>
<td></td>
<td>0.554</td>
</tr>
<tr>
<td>78</td>
<td>price = inexpensive + FQ = poor</td>
<td>.41 * .8</td>
<td>.24 * .2</td>
<td></td>
<td>0.396</td>
</tr>
<tr>
<td>6</td>
<td>price = expensive + FQ = v. good</td>
<td>.41 * .2</td>
<td>.24 * 1</td>
<td></td>
<td>0.322</td>
</tr>
</tbody>
</table>

I know of 594 restaurants in London. There is one very inexpensive restaurant with medium food quality and 78 restaurants that are inexpensive but with poor food quality. There are also 6 restaurants that are expensive with very good food quality.

Subject: **EJB**; Preferences: FQ, cuisine

<table>
<thead>
<tr>
<th>Num. rests.</th>
<th>Rule</th>
<th>Weight Value</th>
<th>Weight Value</th>
<th>Weight Value</th>
<th>Utility</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>FQ = v. good + cuisine = French</td>
<td>.41 * 1</td>
<td>.24 * .9</td>
<td></td>
<td>0.626</td>
</tr>
<tr>
<td>9</td>
<td>FQ = good + cuisine = Japanese</td>
<td>.41 * .8</td>
<td>.24 * .9</td>
<td></td>
<td>0.544</td>
</tr>
<tr>
<td>53</td>
<td>FQ = medium + cuisine = Italian</td>
<td>.41 * .6</td>
<td>.24 * .9</td>
<td></td>
<td>0.462</td>
</tr>
</tbody>
</table>

I know of 594 restaurants in London. There are 6 French restaurants with very good food quality and 9 Japanese restaurants with good food quality. There are also 53 Italian restaurants with medium food quality.

Table 4.17: Rules scored for utility using two user models from subjects AF and EJB. Included in the table are the utility and number of restaurants represented by each rule. FQ = food_quality.
Based on the problems discovered in the various filter and wrapper methods examined, and the results obtained using the Refiner ranking function and user model utility scoring, we settled on the latter two attribute ranking methods as the most promising. These two ranking functions maximize two parameters we hypothesize are important in content selection and summary construction, i.e., the degree of generalization and the degree of interest in a response. We have two different parameters associated with the way attributes are grouped, as well: the singleton method wherein each attribute is described individually, and the associative method, associations among attributes found via tree induction are described. These two parameters test another of our hypotheses, i.e., that users will find associations among attributes in a dataset interesting.

Combining these parameters leads to four different possible summary types:

- Refiner ranking + singleton clustering
- Refiner ranking + associative clustering
- User model ranking + singleton clustering
- User model ranking + associative clustering

Table 4.18 shows an example of each of the four types of summaries, along with the parameters used to create them, for the set of focal information corresponding to restaurants that serve Indian cuisine in London.

In the summaries we create, we mention at most three clauses/utterances in any statement for succinctness. When using the singleton method of response construction, this meant using the three highest scoring attributes. For the associative method, this meant using the two highest scoring attributes, and speaking of the three highest scoring rules.

Summary Ref-Sing in Table 4.18 is an example of a Refiner-type summary, using the Refiner method of attribute selection and singleton attribute
Table 4.18: Example summaries for the four different experimental conditions.

When using either the Refiner or a user model method, attributes are ranked using focal information. When using the Refiner method of attribute ranking, the top scoring attributes are returned with values ranked by frequency, and the Refiner score is assigned to each attribute. When using the user model, specific attribute values are ranked by user preference and the top two chosen. If a particular attribute had already been specified by a user query, the next highest ranking attribute is used.
When using the singleton method of response construction, the three highest scoring attributes are spoken of in separate clauses/utterances. When using the Refiner method of response construction multiple values are spoken of with each attribute, if applicable, using the algorithm described in Section 4.5.4 (e.g., “Many of the restaurants are Italian or French”). When using the user model, a single value is selected for each attribute, based on user model utility (e.g., “There are 12 restaurants with good food quality.”).

When using the associative method of response construction, sets of decision tree rules are created iteratively and all rules are concatenated together into a single data structure. Duplicates are removed algorithmically. C4.5 typically outputs a large number of rules, even for a small dataset; on the 594 restaurants in the London restaurant domain, using between two and three attributes to create a decision tree classifier, we obtain, on average, 90 unique rules per run. In order to use these rules efficiently, each set is processed into a hash table structure that allows for sorting on any attribute contained within the rule. Scores are assigned each set of rules, using either coverage or user model utility as the metric, depending upon the method of attribute selection. Rules are then resorted using one of the scoring metrics. The data structure also allows for the sorting of rules by the values of specific attributes (e.g., rules whose cuisine values are “french”).

Genesis templates were constructed that were able to generate each of the response types shown in Table 4.18. Basic word order is determined by the templates (e.g., “preds” which are modifiers that precede the core noun phrase). Genesis is also able to detect certain flags set within the clause and perform simple if-then-else operations on these flags. Genesis templates are domain-dependent, but the underlying mechanisms for generating natural language have been applied to a variety of domains and languages [Baptist and Seneff, 2000].

With responses that use the Refiner method of attribute selection and the singleton method of response construction (the “traditional” Refiner method, closest to that used in the original system at MIT), thresholds are set within Genesis templates, but used only to determine the specific quantifier (e.g., some, many) that appears in the response. The thresholds are based on the total percentage accounted for by the top three values by frequency.

Table 4.19 shows an example frame used by Genesis to create the response I know of 35 restaurants in London serving Indian food. There are 4
I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.
medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality. The response clause shown in Table 4.20 was created from the output from the c4.5 tree induction system, and is an example of a response using associative clustering. Rules are scored and the top three are processed before being sent to Genesis. Table 4.20 shows an example of some of the rules used to create the same response.

The top-level rule, corresponding to the name of the first clause, is the first rule that is triggered. Genesis then works through the rest of the frame, creating a final string to be spoken to the user. Aggregation of predicates is done as a pre-process, with predicates at the same level in the hierarchy marked as silent if they have the same value as a predicate that has already been seen. Predicates whose values correspond to those at the lower end of the value hierarchy in the user model weighting are marked to be spoken in concession statements (e.g., concession: “but” in the clause marked third_best in Table 4.19). After the response string is created, Genesis is able to apply a set of rewrite rules to make final changes in the string presented to the user. Combined with the templates, these constituted a domain-dependent generation component to the system that was independent of the dialogue manager.

In the experiment described in the next chapter, the responses were constructed by hand. This decision was made because the experiment was conducted in “overhearer” mode and a full system was not required to generate the summaries.

4.7 Conclusion

Automating the process of constructing summaries for dialogue systems depends on a selection of attributes to speak about, clustering methods over those attributes to discover ways of speaking it, and scoring methods to determine what to say. In this chapter, we examined technological challenges associated with the automatic construction of summary responses from structured data, beginning with a critical assessment of the Refiner method of response construction which attempted to summarize data at every dialogue turn in list format.

We decomposed the problem into parameters and algorithms for ranking and selecting from the attributes available, and for combining attributes and values in a summary response. We looked at how to determine the number
| speak_three | >speak_count :agg_values |
| speak_count | "I have found" :num_found :restaurant_up |
| restaurant_up | "I know of" :num_found >restaurant_prepreds :restaurant >rest_postpreds "altogether") |
| restaurant_prepreds | :price |
| restaurant_postpreds | :price_range :food_quality |
| best | "There are" :int >cff_prepreds :core >cff_postpreds |
| cff_prepreds | cff_pricerange cff_cuisine |
| cff_postpreds | cff_foodrange |
| cff_prepreds_template | :name :topic >preds |
| cff_pricerange | $if(:clausify >pricerange_clause) >pricerange |
| price_range | :name |
| pricerange_clause | "that" :be :name |
| cff_foodrange | $if(:concession >concession_foodrange) >foodrange |
| foodrange | "with" :name :core |
| concession_foodrange | "but" :have :name :core |

Table 4.20: Example Genesis rules applied to frame in Table 4.19 and used to generate the response *I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.*
of attributes to include in a response, and multiple methods for ranking those attributes. Our ultimate goal was the selection of a subset to use in the actual responses. Of the attribute selection algorithms examined, the two that seemed to produce the most useful set for summary construction were one based on an expansion of the Refiner method and one based on the use of user model information. The particular user model employed could be applied easily to tailoring responses in a multi-attribute domain.

We also looked at scoring methods for the resulting clusters. In the absence of information about user preferences, we use coverage as a way of determining the set of data that generalizes the best over focal information. When user model information is available, we employ MAUT in a novel way, to score association rules for overall utility. In cases where there are tied utility scores, we use coverage to choose from the competing rules.

Using the two methods of attribute selection and the two methods of clustering gave us four possible summary types, whose construction was operationalized. In the next chapter, we discuss experiments conducted to determine users’ preferences of summary responses constructed in the ways investigated in this chapter. We believe the methods described here are domain-independent and plan to test that hypothesis in Chapters 7 and 8.
Chapter 5

Testing the effectiveness of summary types: A user study

In Chapter 3, we report on an experiment that tested the hypothesis that users prefer responses that provide summary information about system data and found it to be true. In Chapter 4, we investigated the technology needed to create summaries automatically. In this chapter, we report on experiments designed to determine the sorts of summary responses users prefer.

5.1 Experimental Method

The four parameters associated with attribute selection and clustering methods used to construct the summaries in Table 5.1 (repeated here from Chapter 4) allow us to test the following hypotheses (related parameter shown in parenthesis):

- **H4a** The optimal summary type will describe attributes singly (Singleton clustering);

- **H4b** The optimal summary type will describe attributes in association with other attributes (Associative clustering);

- **H5a** Large sets of focal information will be best described by attributes chosen because a small number of their values describe a large amount of the information, leading to summaries such as “Many of the restaurants are in Soho” or “Most of the articles mention lawsuits” (Refiner ranking);

- **H5b** Small sets of focal information will be best described by attributes that conform to user preferences (User model ranking).
<table>
<thead>
<tr>
<th>ID</th>
<th>Ranking method</th>
<th># of attributes</th>
<th>Clustering method</th>
<th>Cluster scoring</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ref-Sing</td>
<td>Refiner</td>
<td>3</td>
<td>Single value</td>
<td>Cluster size</td>
<td>I know of 35 restaurants in London serving Indian food. All price ranges are represented. Some of the neighborhoods represented are Mayfair, Soho, and Chelsea. Some of the nearby tube stations are Green Park, South Kensington and Piccadilly Circus.</td>
</tr>
<tr>
<td>Ref-Assoc</td>
<td>Refiner</td>
<td>2</td>
<td>Associative</td>
<td>Cluster size</td>
<td>I know of 35 restaurants in London serving Indian food. There are 3 medium-priced restaurants in Mayfair and 3 inexpensive ones in Soho. There are also 2 expensive ones in Chelsea.</td>
</tr>
<tr>
<td>UM-Sing</td>
<td>User model</td>
<td>3</td>
<td>Single value</td>
<td>User model</td>
<td>I know of 35 restaurants in London serving Indian food. There are 6 with good food quality. There are also 12 inexpensive restaurants and 4 with good service quality.</td>
</tr>
<tr>
<td>UM-Assoc</td>
<td>User model</td>
<td>2</td>
<td>Associative</td>
<td>User model</td>
<td>I know of 35 restaurants in London serving Indian food. There are 4 medium-priced restaurants with good food quality and 10 with medium food quality. There are also 4 that are inexpensive but have poor food quality.</td>
</tr>
</tbody>
</table>

Table 5.1: Example summaries for the four different experimental conditions.

To test these hypotheses, we construct four different summary-type responses along the lines of those shown in Table 5.1 for eight individual query/response sets in the London restaurant domain. Each of the eight sets, or “tasks”, has a single user query. The eight tasks were selected to utilize a range of attributes in the database that included cuisine, location, and price range. Four of the tasks (large set tasks) led to in-focus subsets of the database larger than 100 entities and four of the tasks (small set tasks) resulted in in-focus subsets smaller than 100 entities. Table 5.3 shows the specific attributes and values used in each of the eight tasks, along with the dataset size for each. The order in which tasks were presented in the experiment is reflected in the table.

Table 5.2 shows an example of the initial query and response set evaluated by user EJB, whose user model ranked food quality, cuisine, and neighborhood as the most important attributes, respectively. The complete set of eight queries and the corresponding response sets for user EJB, along
Table 5.2: Table showing the four responses to the initial query configured for and evaluated by user EJB, whose ranked preferences were, in order, food quality, cuisine and neighborhood.

with this subject’s entire set of user model preferences, can be found in Appendix A.3. Because two responses in each task were tailored to a particular subject, all query/response pairs for all users are not included.

Sixteen experimental subjects were recruited from among students and staff members from the University of Sheffield not involved in the experiment and unaware of the experimental hypotheses. We collected user model data from the 16 subjects using the procedure described in Section 4.5.2.8 and used, for example, in [Walker et al., 2002a].

Each task was presented to the subject on its own web page with the four potential system responses representing the four experimental conditions displayed, in randomized order, as text on the web page. Subjects were asked to read and rate each of the four summary responses on a Likert scale of 1...5 in response to the statement, This response contains information I would find useful when choosing a restaurant. The subjects were
<table>
<thead>
<tr>
<th>Query type</th>
<th>Value</th>
<th>Setsize</th>
<th>Query number</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>All</td>
<td>large (594)</td>
<td>1</td>
</tr>
<tr>
<td>neighborhood</td>
<td>Soho</td>
<td>small (36)</td>
<td>2</td>
</tr>
<tr>
<td>cuisine</td>
<td>Italian</td>
<td>large (105)</td>
<td>3</td>
</tr>
<tr>
<td>price</td>
<td>cheap</td>
<td>large (187)</td>
<td>4</td>
</tr>
<tr>
<td>neighborhood</td>
<td>Chelsea</td>
<td>small (47)</td>
<td>5</td>
</tr>
<tr>
<td>cuisine</td>
<td>Indian</td>
<td>small (35)</td>
<td>6</td>
</tr>
<tr>
<td>price</td>
<td>expensive</td>
<td>large (163)</td>
<td>7</td>
</tr>
<tr>
<td>neighborhood</td>
<td>Knightsbridge</td>
<td>small (23)</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 5.3: Table showing a breakdown of the eight query types evaluated by subjects for the second restaurant experiment. The number in parenthesis indicates the number of restaurants in the database matching the query constraint. The responses could mention a varying number of restaurants, depending on the subset tailored to subjects’ preferences.

Also asked to indicate which response they considered the most useful and which response the least, and were given the opportunity to provide free-text comments about each response. Each response established its own context within the set of restaurants in London, i.e., context history was not maintained across queries.

5.2 Experimental results

5.2.1 User judgments/input

Table 5.4 shows average user scores for each of the four possible summary types. The scores clustered as pairs, the two highest scoring summary types being (1) Refiner ranking and singleton attributes (Ref-Sing; average score: 3.4) and (2) user model ranking and association rules (UM-Assoc; average score: 3.25). A paired t-test showed no statistical difference between these two scores. This disconfirms hypotheses H3a and H3b, that an optimal summary strategy will either involve attributes chosen from data analysis or from user preference. It also disconfirms hypotheses H4a and H4b, that an optimal summary strategy will involve either listing attributes singly or describing them in association with other attributes. Instead, there is no one single optimal summary strategy.

In terms of the technology used, the two top-scoring summary types rep-
Figure 5.1: An example of a webpage used for the second restaurant domain experiment.
resent opposites. Summaries using singleton attributes and Refiner ranking are the simplest of the types, requiring no user model information or decision tree induction. These are the most “generic” of the response types. The other highly preferred type summarizes information using associative clustering and user model ranking. These summaries require user input and the added step of decision tree induction.

The two lowest scoring summary types were (3) user model ranking and singleton attributes (UM-Sing; average score: 2.95); and (4) Refiner ranking and association rules (Ref-Assoc; average score: 2.90). A paired $t$-test on these two scores showed no significant difference, although there is a significant difference between each of the two top-scoring methods and each of the two lower-scoring methods ($p < .005, df = 125$, in all cases).

### 5.2.2 Experimental effects

We performed an ANOVA with score as the dependent variable and the two parameters used for response construction, attribute ranking method and clustering method, as the independent variables. There was a significant interaction between attribute ranking method and clustering method ($df = 1, F = 21.384, p < .00001$). This interaction is illustrated in Figure 5.2. At either end of both crossing lines are the response types representing different values for each of the four parameters.

The experimental data show that subjects preferred responses highlighting associations among attributes when those attributes ranked highly among their stated preferences. When the attributes were determined by an analysis of the data, i.e., via the Refiner method, subjects preferred to have the attributes in the responses described using the singleton method. However, neither type was preferred overall, suggesting that there were other factors affecting user judgments. We hypothesized that features associated with the query/response pairs could help us understand these user judgments.

<table>
<thead>
<tr>
<th></th>
<th>+ User model</th>
<th>- User model</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Associations</td>
<td>3.25</td>
<td>2.90</td>
</tr>
<tr>
<td>- Associations</td>
<td>2.95</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Table 5.4: Average scores for summary statements grouped by attribute selection algorithm and summary construction method.
Figure 5.2: Graphical representation of the average scores for summary statements grouped by attribute selection algorithm and summary construction method.
There were three features associated with the responses that characterized the number and type of restaurants described in each. The first of these, as mentioned above, divided the data quite crudely into two groups defined as large and small setsize. “Large” setsize refers to responses that described 100 or more restaurants and “small” refers to responses that described fewer than 100 restaurants. In addition to setsize, there was a second size-related parameter, acctFor, a scalar value representing the actual number of restaurants described in each response. This parameter varied across user models, as different sets of restaurants matched different user preferences. A third parameter, qtype (for “query type”) describes the constraint that is mentioned in the user query (e.g., cuisine or price). It was always a single value and used in a context-independent way in the simulated dialogue.

We performed an ANOVA with subject score as the dependent variable and attribute ranking method (user model vs. Refiner), clustering method (association rules vs. singleton attributes), and setsize (large vs. small) as the independent variables. There was a main effect for set size ($df = 1, F = 5.86, P < .05$), with summaries describing small datasets getting higher scores. There was also a significant interaction between attribute selection method and dataset size. Figure 5.3 illustrates this interaction. Once again, the crossing lines show how the scores vary as the values for user model and setsize change. Small dataset responses score higher when attributes are selected using the Refiner method, and large dataset responses score higher when attributes are selected using the user model.

We did not find a significant interaction between clustering method (association rules vs. singleton attributes) and setsize in any of our tests. However, the use of user model information to determine which attributes to speak about clearly affects user judgments, in combination with setsize. Although a bias indicating preference for user model information based on dataset size was not surprising, the specific direction of that bias seemed counter-intuitive. We had hypothesized (H5a and H5b) the opposite of what we observe in the data. Because the Refiner method is designed to capture generalizations across datasets, we hypothesized that those data-driven generalizations would be most helpful with large numbers of database tuples. For smaller sets, we hypothesized that user model information would

---

1The entire set of features extracted from the experimental data is described in Section 6, where we use them for classification.
allow subjects to hone in on specific choices more easily. Instead, we found that subjects showed an overall preference for hearing about attributes conforming to user model preferences with large datasets, and for hearing about attributes determined by an analysis of the data with smaller datasets.

To investigate this effect further, we looked at the breakdown in scores for all summaries describing small and large datasets, regardless of their type. The average score for summaries of small datasets was 3.22 against an average score of 3.03 for summaries of large datasets (see Table 5.5 for a complete breakdown of scores along this dimension). This was a significant difference ($p < .05$, $df = 251$). We re-examined the results of the initial restaurant experiment in the light of these findings to see if summaries in general are not preferred with smaller datasets. We did not not find, however, that users preferred the short, menu-driven responses in cases where the dataset was below 100 restaurants in that experiment. In fact, for two out of three query/response pairs where this was the case, users preferred the summary responses overall. The two sets of experimental data are not
Table 5.5: Average scores for summary statements grouped by attribute selection algorithm, summary construction method, and size of dataset described in response.

directly comparable, but the results are consistent within the first dataset.

To examine the effect of setsize with a more refined metric, we next did an ANOVA with score as the dependent variable and attribute selection method and acctFor as independent variables. The feature acctFor reflects the explicitly stated number of restaurants described in each response. There was a main effect for acctFor (df = 103, $F = 1.403$, $P < .05$) but, unlike setsize, there was no significant interaction between it and attribute selection method. The fact that there was a significant interaction with setsize but not with acctFor led us to suspect that the interaction might be due to the type of information being described, rather than the amount.

To examine this, we looked at the interaction between query type (i.e., cuisine, neighborhood, etc.) and the response construction parameters. There was a significant interaction, illustrated in Figure 5.4, between query type and attribute selection method. As shown in Figure 5.4, user model information was preferred overall in responses to queries with no constraints (i.e., all) and those with a cuisine and a price constraint. For queries containing a neighborhood constraint, there was a preference for responses constructed using the Refiner method of attribute selection. The differences for price and neighborhood queries were significant ($p < .001$, df = 67; $p < .05$, df = 101, respectively), indicating that the attribute selection method was especially important in judgments of these responses.

Given that there was a strong correlation between the average utility of the response and the score in the data (Pearson correlation, $p < .0001$), one hypothesis is that Refiner responses to the queries that included a neighborhood constraint somehow reflected user model information more closely than the Refiner responses to queries that included a price constraint. Conversely, the Refiner responses to price queries might have particularly low user model utility and user model responses might have had particularly high utility for price queries. These effects together could account for the
Figure 5.4: Graphical representation of the average scores for summary statements grouped by attribute selection algorithm, summary construction method, and constraint introduced by query.
To test this, we first examined the average utility for all responses by query type, for responses that incorporated user model information and those that did not. (See Section 6.2 for how this is calculated.) Table 5.6 shows the average utility for all user model- and Refiner-based responses, divided by \textit{qtype}. As can be seen, the highest overall average utility by query type is found in the first utterance, where no constraint is mentioned in the user query.

The difference between the average utility of responses using the two attribute selection methods is significant for each query type \((p < .0001, \text{in all cases})\), showing that, overall, the two types of responses are capturing different information about the in-focus dataset. The average utility of the neighborhood responses using Refiner-based attribute selection was the highest among the query types for Refiner-based responses, significantly higher than the next highest average utility for Refiner-type responses \((i.e., \text{price}; p < .005, df = 33)\). This supports the hypothesis that Refiner-based responses are somehow better at capturing user preferences for responses involving neighborhoods. However, it is also the case that the Refiner-based neighborhood responses only match user model preferences relatively well, when compared to Refiner-based responses to other query types. Their average utility is still significantly worse \((p < .0001, df < 67)\) than the responses using user model information for neighborhood queries.

Price responses using user model information do not have significantly higher average utility than the other responses. Price responses using Refiner-based attribute selection have the lowest overall average utility, but it is not significantly lower than either of the other two constraint-based query types for that attribute selection method. Average utility could certainly explain some of the differences observed for attribute selection method and query types, but there are no clear significant differences in the utility scores for these responses. The difference, however, could lie in the actual wording used in the responses.

In several post-experiment interviews, individual subjects expressed a general dislike for responses containing attributes whose values were on the lower range of the goodness scale, such as those that referred to restaurants.

\footnote{To avoid the confounding factor of a simulated user query introducing a constraint that was disliked by a particular subject, constraints introduced in the neighborhood and cuisine queries ("Chelsea", "Soho", "Knightsbridge"; and "Italian" and "Indian", respectively) were chosen to fall within the top-ten preferred choices across all users for those attributes.}
Table 5.6: Table showing the average utility of responses to particular query types, divided by method used to select the attributes to speak about.

<table>
<thead>
<tr>
<th>Query type</th>
<th>Avg. utility of responses using user model attribute selection</th>
<th>Avg. utility of responses using Refiner attribute selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>.3516</td>
<td>.2247</td>
</tr>
<tr>
<td>cuisine</td>
<td>.3282</td>
<td>.2093</td>
</tr>
<tr>
<td>neighborhood</td>
<td>.3201</td>
<td>.2294</td>
</tr>
<tr>
<td>price</td>
<td>.3215</td>
<td>.1653</td>
</tr>
</tbody>
</table>

“with poor food quality” or “poorly rated service”. Such responses are possible with either attribute selection method. The Refiner chooses values for attributes that account for a large amount of the data. Especially for cases of inexpensive restaurants, for example, or for certain cuisines, a large percentage of the restaurants scored poorly on a variety of attributes. Because of the nature of Multi-Attribute Utility Theory, such responses are possible for user model-based responses, as well, for example, when a given set of restaurants scores very highly on something the user values highly (e.g., cost), but low on an attribute that is not as important to the user. A typical response of this type can be seen in the fourth response in Table 4.18, “There are also 4 Indian restaurants that are inexpensive but with poor food quality.” The user for whom this response was tailored ranked price (i.e., inexpensive price) as the most important attribute when choosing a restaurant and food quality as the second-most important. Among the set of Indian restaurants in the database, those that were inexpensive but had poor food quality had a higher overall utility than those that were medium-priced but with medium-food quality. (There were no restaurants that were medium-priced with highly rated food quality.)

The presence or absence of a poorly rated attribute correlates significantly with score, on both Pearson Correlation and Kendall’s tau ($p < .0001$ for both). We also performed an ANOVA with score as the dependent variable and response construction parameters (user model and associations) as one independent variable, along with a new parameter, poor, indicating the presence or absence of an attribute with a poorly rated value in the response. There was a significant interaction between attribute selection method (user model vs. Refiner) and poor ($p < .05$, $F = 5.6$, $df = 1$), illus-
The presence of a poorly rated value brings all scores down, regardless of the attribute selection method; the drop is steeper in responses constructing using the Refiner method. This finding disconfirms hypothesis H6, that the particular value of an attribute is not important when choosing an attribute that generalizes well.

There was a stronger interaction between response construction method and poor \((p < .005, F = 9.477, df = 1)\), as well. Figure 5.6 shows this interaction. Here, also, the presence of a poorly rated attribute brought scores down generally, but used in conjunction with associations, the scores were much lower. Associative responses were always constructed with the poorly rated attributes mentioned last, in a concession statement (i.e., “but with poorly rated service”). This type of response might have served to highlight the poorly rated attribute, drawing even more attention to something that users already found unacceptable. With responses using the singleton method, the specific value “poor” would appear inside a list alongside other, less negative values.
Figure 5.6: Graphical representation of the interaction between the clustering method used to construct the response and the presence or absence of poorly rated attributes in the response.
Finally, the presence or absence of a poorly rated attribute may explain some of the effects seen with query type and dataset size. Table 5.7 contains a breakdown of percentage of responses mentioning a poorly rated attribute by query type and task. For each, the overall percentage of responses with a poorly rated attribute is given, along with the percentages of such responses using either the user model or Refiner method of attribute selection. Neighborhood, with relatively low-scoring user model-based responses, had a relatively large percentage of responses containing poorly rated attributes constructed using the user model method. Price, on the other hand, had quite a small number of such responses for its user model-based responses and those responses scored higher. Responses to price queries using the Refiner method of attribute selection, on the other hand, had a very high percentage describing poorly rated attributes, and subjects preferred the user model responses over them.

The responses themselves reveal the algorithmic ways the scoring function underlying these two attribute selection criteria create responses highlighting poorly rated attributes. One set of responses users were asked to judge followed a request for “cheap” restaurants. For responses using the user model, it was relatively easy to find groups of restaurants that had higher values for other attributes. The “inexpensive” constraint was a given; MAUT scoring on the set of association rules describing the focal information assigned higher utility to the (admittedly few) that had relatively higher values on the attributes users preferred, such as food quality or service. The Refiner, however, had no way of penalizing individual values for attributes that described the focal information, so the large clusters discovered among inexpensive restaurants that also had poor service quality, for example, were mentioned algorithmically.

Small datasets, also characterized by low-scoring user model-based responses, had a significantly larger number of responses with poorly rated attributes than did large datasets ($p < .05$, $df = 268$). The breakdown by attribute selection criterion shows that there were no responses mentioning poorly rated attributes in small dataset responses using the Refiner, the responses preferred by subjects. The percentage of responses mentioning poorly rated attributes was smaller among large dataset responses using user model attribute selection and those were the responses preferred for that set. It is clear that that user judgments are being influenced by the presence of these poorly rated attributes.
<table>
<thead>
<tr>
<th>Query type</th>
<th>% responses w/ poorly rated attrs.</th>
<th>% responses w/ poorly rated attrs. using UM selection</th>
<th>% responses w/ poorly rated attrs. using Refiner selection</th>
</tr>
</thead>
<tbody>
<tr>
<td>all</td>
<td>2.9</td>
<td>5.9</td>
<td>0</td>
</tr>
<tr>
<td>cuisine</td>
<td>16.4</td>
<td>32.8</td>
<td>0</td>
</tr>
<tr>
<td>neighborhood</td>
<td>7.8</td>
<td>15.7</td>
<td>0</td>
</tr>
<tr>
<td>price</td>
<td>25.5</td>
<td>1.4</td>
<td>50</td>
</tr>
<tr>
<td>Task</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>large</td>
<td>17.4</td>
<td>9.6</td>
<td>25</td>
</tr>
<tr>
<td>small</td>
<td>10.4</td>
<td>20.9</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 5.7: Average scores from users for responses grouped by query type, dataset, and the presence or absence of poorly rated attributes.

Poorly rated attributes in responses is not the only factor affecting user judgment. Cuisine responses had the highest overall percentage of poorly rated attributes mentioned in user model responses (32.8%), yet user model-based cuisine responses were preferred overall by users. With the responses describing the entire dataset, the percentage of user model-based responses containing poorly rated attributes was small overall, but greater than that for Refiner-based responses. However, user model-based responses were preferred overall in this set. Among the highest rated responses overall in the user study, 11% contained phrases mentioning poorly rated attributes. The amount and type of information, as well as the specific constraint being addressed by the response also factor into user judgments. In Section 5.2.4, we examine subjects’ qualitative judgments to help understand this interplay of factors.

### 5.2.3 Comparison with generalized user model

Summaries containing attributes and values that users care about, as measured by utility, are correlated with highly ranked scores (Section 5.2.2). However, examining the attributes contained in summaries constructed without input from user models yet highly ranked by users reveal how dispositional preferences can be overridden in specific responses. Here, we compare the information in the most highly ranked response type (i.e., **Ref-Sing**) to
Table 5.8: Table showing attributes ranked by average user model preference and the percentage of responses containing those attributes as part of their descriptive statements.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>% of Ref-Sing responses with this attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>food_quality</td>
<td>0</td>
</tr>
<tr>
<td>price</td>
<td>75</td>
</tr>
<tr>
<td>service</td>
<td>12.5</td>
</tr>
<tr>
<td>cuisine</td>
<td>75</td>
</tr>
<tr>
<td>neighborhood</td>
<td>62.5</td>
</tr>
<tr>
<td>decor</td>
<td>0</td>
</tr>
</tbody>
</table>

Taken together and averaged, the overall rankings across all subjects’ user models for the six restaurant attributes were, from most to least preferred:

1. *food_quality*
2. *price*
3. *service*
4. *cuisine*
5. *neighborhood*
6. *decor*

Table 5.8 shows how these attributes are represented in *Ref-Sing* responses. Given the low ranking of *neighborhood*, it is interesting to note that neighborhood information appeared in 62.5% of the *Ref-Sing* responses. *Cuisine*, ranked just above *neighborhood* (4th out of 6), appeared in 75% of the same type of responses. Information about *food_quality*, on the other hand, appeared in none of the *Ref-Sing* responses.

When information about these attributes was present in the favored response, it appears to be unrelated to the characteristics users value in specific restaurants. The fact that subjects chose either responses tailored to their user models or responses that contained information that was ranked
lower in general, seems to indicate that the task of browsing, at least in this domain, involves an interaction between dispositional and situational preferences.

5.2.4 User qualitative judgments

As a final step in analyzing the results of the user study, we examined free-text comments users optionally provided for each task, paying particular attention to comments about the two top-ranking summary types. For both summary types, users reported favorably on the “overview” nature of the information presented. When users preferred UM-Assoc responses, they mentioned positively the way “trade offs” were highlighted among attributes. Users also commented on the importance of seeing a synthesis of information among the attributes that mattered to them in the responses. This supports the hypothesis that pointing out associations among attributes is valuable to users, but only when those attributes are tailored to their preferences.

When users disliked UM-Assoc responses, they referred to the “disparate” nature of the information. They also expressed a dislike for information at “too high a level of detail”. Individual subjects mentioned a preference for hearing about locations and cuisine instead, even when these were not highly ranked in their user models. One subject mentioned getting “bogged down” in quality/service combinations, the two highest attributes in his user model. Another subject, who had rated Japanese cuisine highly, wondered why a Japanese restaurant with very good food quality was mentioned in a response. This indicates that dispositional preferences were important to users, but not in all contexts.

In comments about the Ref-Sing responses, the characteristics users mentioned favorably included the fact that information was presented “in a general way”. Many subjects specifically noted location or cuisine information as helpful. Contrastingly, when they disliked these responses, that same information was termed “too vague” or “too open”. One subject’s comment on the Ref-Sing response for Indian restaurants was that he could have guessed at that information contained in the summary (i.e., that all price ranges were represented, that some Indian restaurants were in Mayfair, Soho, and Chelsea, and that some had good service). It seems that general information, based solely on an analysis of focal information, can be useful in a summary response, but there are other factors involved in user judgments.
5.3 Conclusion

We describe an experiment eliciting human judgments for four types of summary responses. We tested four hypotheses related to user preferences for summaries containing attributes selected by user model utility or by their ability to generalize, as well as for summaries highlighting associations among attributes vs. those that mentioned the attributes separately.

Our data show that users had equal preferences for responses that used the Refiner method of attribute selection, when those attributes were described in singleton clusters, and responses that used the user model method of attribute selection, when those attributes were described as associative clusters. An analysis of factors associated with the responses indicated that the attribute selection criteria and the clustering method strongly affected one another. Other features associated with the responses influenced user judgments, including the size of the dataset being described. Users prefer responses based on user model utility when those responses describe larger datasets. Another strong effect on user judgment was the presence of poorly rated values for particular attributes in the response. Across all responses, users ranked responses that mentioned poorly rated attributes lower.

The statistical analysis of subject rankings suggests that there is an interplay of factors in determining the choice of preferred response type. No one type of response construction was preferred by a significant margin, suggesting that those factors vary across dialogue contexts. In the next chapter, we look at this interplay from another perspective; we use the data from the user experiment to build a classifier to predict which response type to produce in a given dialogue context.
Chapter 6

Learning to predict user summary preference: A classification experiment

The previous chapter reports the results of a statistical analysis of our second restaurant experiment, testing which of the four summary types was the most effective. The results indicate that there is not a single, clear “winner” among the summary types that fits all dialogue contexts. Various features, e.g., dataset size, showed significant interaction effects with response construction parameters; in this chapter, we parameterize features underlying each task from the experiment and use them to test Hypothesis H7, that a classifier can predict an “optimal” summary type.

The data suggest that characteristics associated with the set of restaurants being described are important, as well as utility information derived from application of a user model. The performance of a classifier in predicting summary type will indicate if trends we discovered among user judgements carry over to an automated means of selecting which response type to use in a given context.

6.1 Classification task

One way of looking at the problem of learning to predict user summary preferences is to frame it as a classification problem, where the class to be predicted is a summary type and the features are derived from parameters within the dialogue context. For each task performed by the user in the second restaurant experiment, we identify the top-choice summary response, and use this response as the correct class to be predicted in classification
We have four classes of responses to predict, Ref-Sing, a response constructed using the Refiner method of attribute selection, with attributes referred to singly, Ref- Assoc, a response constructed using the Refiner, with associations among attributes described, UM-Sing, a response constructed using a user model for attribute selection, with attributes referred to singly, and UM-Assoc, a response constructed using a user model, with associations among attributes described. Table 6.1 show an example of the four response types for the first task (i.e., all restaurants in London) for user EJB, as well as features computed for each response. The features are explained in more detail below.

6.2 Determination of feature set

We divide the information associated with each task into two types: (1) information described by the query in context, i.e., the dialogue context and data used to construct the response; and (2) information described in the utility of the response and derived using information in the user model. For the former, we derive four categories of features that yield a total of seven query in context features altogether. For the latter, we derive 5 categories of features, for a total of 30 utility features altogether.

The query in context features are:

- **numRests**: four features representing the number of restaurants mentioned in each response type;

- **task**: one feature representing the type of constraint used to generate the in-focus subset of the database (e.g., cuisine, price range);

- **dataset**: one feature representing the size of the in-focus subset (i.e., big, if the number of restaurants matching the task constraints was over 100, small otherwise);

- **poorP**: one feature representing the presence or absence of a poorly rated attribute within the response;.

---

1In cases of ties, we consulted the optional free-form text input responses, where users were asked to indicate the response they liked best and least. In all but one case, this resolved the tie. We eliminated the one response that we were not able to resolve in this way from the dataset used for the classification experiment.
Ref-Sing

I know of 594 restaurants in London. All price ranges are represented. Many serve Italian and British cuisine. Some of the neighborhoods represented are Chelsea, Mayfair, and Soho.

<table>
<thead>
<tr>
<th>numRests</th>
<th>task</th>
<th>dataset</th>
<th>poorP</th>
<th>avgUtility</th>
<th>hiUtility</th>
<th>loUtility</th>
</tr>
</thead>
<tbody>
<tr>
<td>594</td>
<td>all</td>
<td>large</td>
<td>0</td>
<td>0.194</td>
<td>0.432</td>
<td>0.054</td>
</tr>
</tbody>
</table>

Ref-Assoc

I know of 594 restaurants in London. There are 49 medium-priced Italian restaurants in London and 40 medium-priced European restaurants. There are also 36 medium-priced British restaurants.

<table>
<thead>
<tr>
<th>numRests</th>
<th>task</th>
<th>dataset</th>
<th>poorP</th>
<th>avgUtility</th>
<th>hiUtility</th>
<th>loUtility</th>
</tr>
</thead>
<tbody>
<tr>
<td>689</td>
<td>all</td>
<td>large</td>
<td>0</td>
<td>0.19</td>
<td>0.234</td>
<td>0.102</td>
</tr>
</tbody>
</table>

UM-Sing

I know of 594 restaurants in London. There are 70 restaurants with good to very good food quality. There are 105 Italian restaurants and 188 inexpensive to very inexpensive restaurants.

<table>
<thead>
<tr>
<th>numRests</th>
<th>task</th>
<th>dataset</th>
<th>poorP</th>
<th>avgUtility</th>
<th>hiUtility</th>
<th>loUtility</th>
</tr>
</thead>
<tbody>
<tr>
<td>957</td>
<td>all</td>
<td>large</td>
<td>0</td>
<td>0.229</td>
<td>0.328</td>
<td>0.144</td>
</tr>
</tbody>
</table>

UM-Assoc

I know of 594 restaurants in London. There are 6 French restaurants with very good food quality and 9 Japanese restaurants with good food quality. There are also 53 Italian restaurants with medium food quality.

<table>
<thead>
<tr>
<th>numRests</th>
<th>task</th>
<th>dataset</th>
<th>poorP</th>
<th>avgUtility</th>
<th>hiUtility</th>
<th>loUtility</th>
</tr>
</thead>
<tbody>
<tr>
<td>662</td>
<td>all</td>
<td>large</td>
<td>0</td>
<td>0.544</td>
<td>0.626</td>
<td>0.462</td>
</tr>
</tbody>
</table>

Table 6.1: Table showing the initial query seen by user EJB, whose ranked preferences were, in order, food quality, cuisine and neighborhood.
Table 6.1 contains examples of these four features for responses to the first experimental task for one subject.

The utility features derived from information associated with the user model are:

- **umInfo**: Six features representing the rankings for each attribute from the user model for the particular user, indicating that user’s overall preferences for attributes in the domain. For example, a summary type whose user had rated *food quality* most highly would receive a ’5’ for the feature *food quality*;

- **avgUtility**: Four features representing an average utility score for each summary type, based on the information contained in the summary (computation described in Section 6.3 below). Each clause or statement in each response, regardless of the method used to construct it, was scored against the user model to determine how well it matched a user’s preferences. We averaged this score across all clauses/statements within each response. Four values are associated with **avgUtility**, as well as for all other utility-based features. These include the value associated with the highest rated response, as well as those for each of the three competing summary responses;

- **hiUtility**: Four features representing the highest utility score among the three clauses mentioned in the response, for all response types;

- **loUtility**: Four features representing the lowest utility score among the three clauses mentioned in the response, for all response types;

- **allUtility**: 12 features, representing the composite of the high, low, and average utility scores from the previous three feature sets.

### 6.3 Calculating utility

When using the associative method of response construction, utility is computed on the associated attributes and values contained in each statement, as described in Section 4.5.5. Utility is not computed on the grounding statement i.e., *I know of 594 restaurants in London*, which is the same across all responses. Table 6.2 shows how utility was calculated for the three non-grounding statements that make up the **UM-Assoc** response shown in
UM-Assoc
Statement 1  (hiUtility)
There are 6 French restaurants with very good food quality...

| food_quality | = | v. good | + | cuisine | = | French |
|------------|----|--------|+|---------|==|--------|

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]
\[
.41 \times 1 + .24 \times .9 = 0.626
\]

UM-Assoc
Statement 2
...and 9 Japanese restaurants with good food quality.

| food_quality | = | good | + | cuisine | = | Japanese |
|------------|----|------|+|---------|==|---------|

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]
\[
.41 \times .8 + .24 \times .9 = 0.544
\]

UM-Assoc
Statement 3  (loUtility)
There are also 53 Italian restaurants with medium food quality.

| food_quality | = | medium | + | cuisine | = | Italian |
|------------|----|--------|+|---------|==|--------|

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]
\[
.41 \times .6 + .24 \times .9 = 0.462
\]

Table 6.2: Scoring of UM-Assoc response clauses shown in Table 6.1 for user EJB. \(\text{avgUtility}\) for this response was .544.

Table 6.1. The statements that generated the \(\text{hiUtility}\) and \(\text{loUtility}\) features are marked.

For responses using the singleton method of response construction, each statement is considered as a set of attribute/value pairs, as in an association rule. Thus, the statement, \textit{Many of the restaurants serve Italian, British, or French cuisine}, would be treated as a series of three association rules for the attribute \textit{cuisine} for the purposes of computing utility. Table 6.3 shows how statements were described as rules, and utility calculated for the Ref-Sing response shown in Table 6.1. As with the UM-Assoc example, the statements that generated the \(\text{hiUtility}\) and \(\text{loUtility}\) values are marked.

6.4 Results

We experimented with a variety of classifiers contained in the WEKA toolkit [University of Waikato, 2004]. For the baseline, we use a Naive Bayes clas-
Ref-Sing

Statement 1  (*loUtility*)
All price ranges are represented.

<table>
<thead>
<tr>
<th>price</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>inexpensive</td>
<td>.03 * .8 +</td>
</tr>
<tr>
<td>medium</td>
<td>.03 * .6 +</td>
</tr>
<tr>
<td>expensive</td>
<td>.03 * .4 = 0.054</td>
</tr>
</tbody>
</table>

Ref-Sing

Statement 2
Many serve Italian and British cuisine.

<table>
<thead>
<tr>
<th>cuisine</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>italian</td>
<td>.24 * .9 +</td>
</tr>
<tr>
<td>british</td>
<td>.24 * .1 = 0.24</td>
</tr>
</tbody>
</table>

Ref-Sing

Statement 3  (*hiUtility*)
Some of the neighborhoods represented are Chelsea, Mayfair, and Soho.

<table>
<thead>
<tr>
<th>neighborhood</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chelsea</td>
<td>.16 * .9 +</td>
</tr>
<tr>
<td>Mayfair</td>
<td>.16 * .9 +</td>
</tr>
<tr>
<td>Soho</td>
<td>.16 * .9 = 0.432</td>
</tr>
</tbody>
</table>

Table 6.3: Scoring of Ref-Sing response clauses shown in Table 6.1 for user EJB.
<table>
<thead>
<tr>
<th>System</th>
<th>Features Used</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Top-choice class</td>
<td>52.9</td>
</tr>
<tr>
<td>QR1</td>
<td>task</td>
<td>47.1</td>
</tr>
<tr>
<td>QR2</td>
<td>dataset</td>
<td>47.1</td>
</tr>
<tr>
<td>QR3</td>
<td>numRests</td>
<td>50.0</td>
</tr>
<tr>
<td>QR4</td>
<td>task + dataset</td>
<td>47.1</td>
</tr>
<tr>
<td>QR5</td>
<td>task + dataset + poor</td>
<td>50</td>
</tr>
<tr>
<td>QR6</td>
<td>task + numRests</td>
<td>51.5</td>
</tr>
<tr>
<td>QR7</td>
<td>numRests + dataset</td>
<td>47.1</td>
</tr>
<tr>
<td>QR8</td>
<td>task + dataset + numRests</td>
<td>51.5</td>
</tr>
</tbody>
</table>

Table 6.4: Classification results showing effects of different combinations of features from the *query in context* set.

The classifier, whose performance matches the performance achieved when, at any given choice point, the highest scoring classification scheme (i.e., *Ref-Sing*) is used. The baseline performance has an accuracy of 52.9%. Decision tree induction classification achieves the highest accuracy on the dataset from among the applicable classifiers in WEKA. ² The experiments described below compare the baseline Naive Bayes classifier to the decision tree classifier.

### 6.4.1 Features associated with the query/response pair

We look at each set of features separately, as well as the effects of combining them. The first set is the *query in context* features associated with the query/response pairs. Table 6.4 shows the performance of the classifier using combinations of these features.

These features, alone and in combination, did not improve classifier performance significantly over the baseline. (Significance for all classification experiments was measured using ten-fold cross-validation and a two-tailed t-test, with $P < .05$). In fact, in all cases, performance degrades (though not significantly). Since one of the top-scoring responses is based on information about user preferences, it is not surprising to find that information from the user model cannot be excluded from a classifier.

We were somewhat surprised to find that the information is not useful at

²For the experiments described, the decision tree induction algorithm found in the WEKA toolkit was used rather than c4.5, on which it is based.
### System Features Used | Accuracy (%)
---|---
Baseline | Top-choice class | 52.9
UM1 | allUtility | 47.1
UM2 | hiUtility | 42.6
UM3 | loUtility | 46.3
UM4 | avgUtility | 47.1
UM5 | umInfo | 66.9*
UM6 | allUtility + umInfoAll | 62.5*
UM7 | hiUtility + umInfoAll | 64.0*
UM9 | loUtility + umInfoAll | 61.0*
UM9 | avgUtility + umInfoAll | 64.0*

| System | Features Used | Accuracy (%)
---|---|---
Baseline | Top-choice class | 52.9
UM1 | allUtility | 47.1
UM2 | hiUtility | 42.6
UM3 | loUtility | 46.3
UM4 | avgUtility | 47.1
UM5 | umInfo | 66.9*
UM6 | allUtility + umInfoAll | 62.5*
UM7 | hiUtility + umInfoAll | 64.0*
UM9 | loUtility + umInfoAll | 61.0*
UM9 | avgUtility + umInfoAll | 64.0*

### Table 6.5: Classification results showing effects of different combinations of feature from the utility set. * indicates significant improvement over baseline \((P < .005)\).

all, however. We performed a further test to see if the information associated with just the query/response sets is useful in predicting the subset of summary types where no user model information was used. We partitioned the data into just the examples where responses using the Refiner method of attribute selection scored highest. Here classification performance is 82.93%, high but not significantly higher than the baseline performance, which is 81.5%. The difference between the users’ preferences within the two non-user model related responses was so great that little could be learned by the classifier, given the amount of data present.

#### 6.4.2 Features associated with the user model

Classification performance is significantly improved using information based on the user model, as can be seen in Table 6.5. The single set of features that had the greatest effect on classifier accuracy, was that describing the relative rankings of the attributes from the user model (system UM5) in Table 6.5. In other words, how users feel about food quality, service, etc., in a relative way, is a reasonable predictor of how they feel about the responses.

Although the utility of the response was shown to correlate highly with user score on statistical tests (see Section 5.2.2), avgUtility by itself (system
Table 6.5) was not a good predictor of user preferences for classification purposes. The degree to which a users’ preferences are met by a particular response will be reflected in the rank the user assigns to that response, but utility score is just one factor in determining which response is chosen as the top choice. Combining the various utility-based features with information about user model ranking, however, does produce significant improvements in classification performance (systems UM6 through UM9 in Table 6.5).

As with the set of query in context features, we tested the set of utility features on the subset of data for which the highest scoring response used user model information. Overall classifier accuracy is, once again, quite high (74.7%), although not significantly higher than the baseline performance (71.0%). As noted with the Refiner-only responses, the difference between users’ preferences for the these two types of response (i.e., UM-Assoc and UM-Sing) is quite high to begin with, giving the classifier little opportunity to learn.

It seemed somewhat surprising that UMInfo, a feature reflecting a relative preference, would have such a strong effect on classification performance on its own (cf UM5 in Table 6.5). We examined the user models more carefully to see if there were similarities that could explain the importance of this feature. We computed the pair-wise cosine vector similarity between vectors representing user model weights for all subjects. The cosine similarity between two vectors $\mathbf{A}$ and $\mathbf{B}$ is defined as the dot product of $\mathbf{A}$ and $\mathbf{B}$ normalized by the Euclidean distance between $\mathbf{A}$ and $\mathbf{B}$:

$$\frac{\mathbf{A} \cdot \mathbf{B}}{||\mathbf{A}|| ||\mathbf{B}||}$$

This results in a number between 0 and 1, with scores closer to 1 indicating a greater similarity between the two matrices.

Table 6.6 shows the cosine similarities between each experimental subject. The average pair-wise vector cosine similarity between all user models is .846, with a standard deviation of .126. $^3$ Price or food quality were among the top-three attributes in all of the user models, and the two attributes were ranked one or two (in varying order) in 44% of the user models. Across all user model, food quality was the most highly valued attribute in 77.8% of the time.

$^3$There were only 6 cases of identical user models among the 240 possible pair-wise vector similarity calculations between models not associated with the same person.
Given this general level of general agreement in relative preference across subjects, there was still enough variation across users to make **UMInfo** a useful predictor of response type. We experimented with creating training and test sets of users with similar and divergent user models, but were unable to replicate the average level of classification accuracy using the entire training set with ten-fold cross-validation. The datasets used for this experiment were quite small, however, averaging just four subjects' data per set, so it is difficult to draw conclusions from this. It is intriguing, however, that relative attribute ranking helps classification in this task.

### 6.4.3 Combining features

Table 6.7 shows the result of combining the two sets of features. Features from both the *query in context* and *utility* sets significantly increases accuracy in all cases. The highest accuracy is achieved using a combination of five features (system **All11** in Table 6.7), two *utility*-based and three *query in context*-based. Performance fluctuated among the various feature combinations, but several features emerged as good predictors. The use of the feature **hiUtility** performed better than **allUtility**, **avgUtility**, **loUtility**, for example (cf., System **All9** against Systems **All6**, **All7**, and **All8**). Although it is difficult to draw inferences from classifier performance, this seems to indicate that the *highest* valued statements were what users are focussing on in arriving at their judgements. Table 6.8 shows an example of the highest ranked statements in two response types for one particular user.

An examination of the classification rules themselves suggests what might be underlying users’ decisions. Table 6.9 shows three rules for System **All9** in

<table>
<thead>
<tr>
<th>Table 6.6: Cosine vector similarities between each of the 16 subjects in the restaurant experiment.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>al</td>
</tr>
<tr>
<td>as</td>
</tr>
<tr>
<td>ba</td>
</tr>
<tr>
<td>cab</td>
</tr>
<tr>
<td>dgm</td>
</tr>
<tr>
<td>ejb</td>
</tr>
<tr>
<td>hs</td>
</tr>
<tr>
<td>nw</td>
</tr>
<tr>
<td>os</td>
</tr>
<tr>
<td>pw</td>
</tr>
<tr>
<td>rc</td>
</tr>
<tr>
<td>rg</td>
</tr>
<tr>
<td>sat</td>
</tr>
<tr>
<td>snw</td>
</tr>
<tr>
<td>stc</td>
</tr>
<tr>
<td>wp</td>
</tr>
<tr>
<td>System</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Baseline</td>
</tr>
<tr>
<td>All1</td>
</tr>
<tr>
<td>All2</td>
</tr>
<tr>
<td>All3</td>
</tr>
<tr>
<td>All4</td>
</tr>
<tr>
<td>All5</td>
</tr>
<tr>
<td>All6</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>All7</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>All8</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>All9</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>All10</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>All11</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Table 6.7: Classification results showing effects of different feature sets on predicting summary type. * indicates significant improvement over baseline ($P < .05$).
UM-Assoc Response, statement with highest utility

There are 4 medium priced [Indian] restaurants with medium food quality.

UM-Assoc Rule

\[
\text{FQ} = \text{good} + \text{price} = \text{medium}
\]

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]

\[
.41 \times .8 + .24 \times .6 = 0.472
\]

Ref-Assoc Response, statement with highest utility

There are 6 inexpensive [Indian] restaurants in Soho.

Ref-Assoc Rule

\[
\text{neighborhood} = \text{soho} + \text{price} = \text{cheap}
\]

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]

\[
.06 \times .9 + .24 \times .8 = 0.246
\]

Ref-Sing Response

Some of the neighborhoods represented are Soho and Chelsea.

Ref-Sing Rule

\[
\text{neighborhood} = \text{Soho} + \text{neighborhood} = \text{Chelsea}
\]

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]

\[
.06 \times .9 + .06 \times .9 = 0.108
\]

UM-Sing Statement

Many have good food quality.

UM-Sing Statement

\[
\text{FQ} = \text{good}
\]

\[
\text{Weight} \times \text{Value} + \text{Weight} \times \text{Value} = \text{Utility}
\]

\[
.41 \times .8 = 0.328
\]

Table 6.8: Highest scoring clauses showing contrast between UM-Assoc and Ref-Assoc responses for user NW.
Table 6.9: Example classification rules from System 3 in Table 6.7.

Table 6.7, using $hiUtility$. The first rule is very simple, simply stating that, if the highest utility value of the Ref-Sing response is lower than a particular threshold, then use the UM-Assoc response. In other words, if one of the two highest scoring response types has a low utility, use the other.

The second rule in Table 6.9 shows the effect that the number of restaurants in the response has on summary choice. In this rule, the Ref-Sing response is preferred when the highest utility value of that response is over a particular threshold. This feature is not the only consideration determining response choice, however. The other determining factor has to do with the number of restaurants mentioned in the corresponding UM-Assoc response, and its high utility value. What is interesting is that the UM-Assoc response is not preferred when the number of restaurants it mentions is below a particular threshold, even when the average utility of that response is quite high. This is consistent with the evidence from the statistical analysis of these data, showing, in general, that users preferred Refiner-based responses for small datasets.

The final rule in Table 6.9 predicts Ref-Assoc, the lowest overall scoring response type. When the number of restaurants accounted for by UM-Assoc, as well as the highest utility for that response, are both below a certain threshold, and the highest utility for the Ref-Assoc response is above a certain threshold, then use Ref-Assoc. The utility for any summary type using the Refiner method of attribute selection will most likely be lower than those using the user model, since overall utility is not taken into account in summary construction (see Section 5.2.2). However even a relatively low score for a Ref-Assoc response might indicate a summary that mentions at
least one of the attributes the user finds important. That, combined with the other information in the summary, may make that summary type preferable over one constructed to maximize user model utility.

To illustrate these trade-offs, Table 6.8 shows an excerpt from two summary responses created for the Indian restaurant task, for user NW. In the experiment, user NW ranked the Ref-Assoc response higher in this context than the UM-Assoc response. The ranked preferences for this user were, from most to least highly valued, food_quality, price, service, decor, neighborhood, and cuisine. In Table 6.8, the highest utility within the UM-Assoc response was for the statement, There are 4 medium priced [Indian] restaurants with medium food quality, with a utility of 0.472. The highest utility within the Ref-Assoc response was associated with the statement There are 6 inexpensive [Indian] restaurants in Soho. At 0.246, the highest utility for the Ref-Assoc response is lower than the one tailored specifically to the user’s preferences in the UM-Assoc, but it contained a highly preferred attribute, price, with a high-scoring value, as well as a preferred neighborhood for this user.

### 6.5 Conclusion

The results from the user study described in Chapter 5 suggested that there was not a single correct choice of summary type and that users’ preferences were influenced by factors such as dataset size and the attribute values contained in the response. In this chapter, we framed the question of user preferences in summary type in terms of classification. Using data from the user study as input, we derived features associated with each query/response pair, as well as features associated with utility scored using user model information. Experimenting with a combination of these features to predict summary type, the best classification performance was 16.2% above the baseline, a significant improvement in predicting the top-choice response. We thus confirmed H7, that a classifier can predict summary type response. The amount of data available for classification experiments was small, but the performance is another indicator that the process of content selection and presentation for dialogue systems can be made algorithmic confirming hypothesis H2, as well.
Chapter 7

Technology for creating associative summaries from an automatically annotated corpus of news documents

The results from both sets of experiments with the restaurant corpus show that users find summary responses useful. Furthermore, the experiments show that real-time and automatic procedures can create summary responses within a dialogue system, and that procedural parameters can be determined from either data-driven or user-elicited methods. H8 states that useful and informative summaries can also be created when focal information is unstructured.

To test H8, and the generalizability of the algorithms, we next look at browsing interactions in a very different domain, that of text-based news archive. The motivation for this came from the CubReporter project at the University of Sheffield [Barker et al., 2006]. CubReporter was designed as a tool to help reporters quickly and easily find information to use to place breaking news in the context of previous and/or related events. A human-human corpus collected for CubReporter indicated that journalists are often concerned with discovering “angles”, i.e., with finding patterns in news archives and making connections within existing data and between existing data and new events. We hypothesized that the algorithms for finding associations automatically in restaurant data would be well suited to this task.
7.1 Experiments with human-annotated data

The most straightforward way to apply the browsing techniques to news documents was to treat the articles themselves as database entries, with each article represented as a set of attribute-value pairs. To explore this idea, we first experimented with the MUC-3 corpus [MUC, 1991, Lehner et al., 1991]. The corpus includes hand-annotated templates for approximately 1700 newswire stories in the terrorist domain. The templates include information such as date and location, and also information on perpetrator, victim, and instruments used. The average story length was 12 sentences.

The first step was to convert the MUC data to a Galaxy database frame identical to the one used for the restaurant data. This required some data regularization on the MUC templates (i.e., to convert terms that had multiple representations in the corpus, “FMLN”, “Farabundo Marti National Liberation Front”, and “Farabundo Marti Liberation”, for example, to a single representation). Table 7.1 shows an example frame derived from a MUC story template. We then ran the resulting frames through the iterative c4.5 process, sorted the rules by the number of documents they referred to, and examined the results.

It was immediately apparent that attribute selection would be a significant issue for the MUC domain and for news archives in general. Even after removing attributes that were unique to each story (e.g., story id), and

Table 7.1: Example frame-based representation of news story from the MUC-3 corpus. Attributes not represented in this particular template are not shown.
Table 7.2: Example rules derived from the C4.5 tree-induction algorithm applied to the MUC templates. The number in parentheses indicates the number of templates accounted for by the rule.

Although these were reasonable summaries of the MUC templates, it was unclear that these sorts of summaries, of a set of documents already known to be about terrorist acts, would be of use to someone browsing the archive to get an idea of what it contained. Our main concern, however, was that it would not be possible, at least in the near term, to derive the level of annotation of the MUC templates automatically in a domain-independent way. Thus, we next investigated the possibility of applying clustering algorithms to a corpus of automatically annotated documents.

7.2 Automated identification of keyphrases

Indicative summarization traditionally is done using information contained in structured representations of text such as table of contents, headings, and ontologies [Kan et al., 2001a, Kan, 2003, White et al., 2001]. As the technology for identifying named entities and keywords has matured, these have been increasingly used for generating indicative summaries [Witten et al., 1999]. Named entity recognition is done using a combination of table look-
up (i.e., gazetteers) and shallow natural language processing (see [Maynard et al., 2001] for a discussion of the issues involved and one such system for extracting named entities). Keywords, which are the common nouns that characterize a particular text, are often identified in one of two ways: (1) using some form of trigger phrase, pioneered by [Hearst, 1992] and expanded to use word collocation information [Turney, 2000]; and (2) using statistical information and TF/IDF scores [Witten et al., 1999].

We are interested in how these technologies might be used to enable browsing of large text corpora. In this chapter, we describe experiments that combine IE technology for keyword/named entity extraction with the algorithms developed for describing restaurant data, to provide support for browsing news archives. Our goal here is not to cluster documents, although the algorithms used will have that effect. By finding associations among keywords and named entities within documents, we will be essentially constructing polythetic clusters, i.e., clusters that are defined by a set of words or phrases. (Polythetic clusters are usually contrasted with monothetic clusters, where groupings are based on a single feature). Others have used both polythetic and monothetic clustering to group text [Cutting et al., 1992, Sanderson and Croft, 1999, Kummamuru et al., 2004, Lawrie et al., 2001] mainly as a way of discovering the topic structure within large sets of documents. Topics thus discovered may aid a user in browsing a corpus of documents, but, as far as we know, the work described here is the first attempt to use decision tree induction to find clusters in a news corpus and test them in a task-oriented evaluation.

### 7.3 Automatically annotated news documents

Our goal is to test associative summaries on a domain of news articles that had been automatically annotated for keywords and named entities. For the experiments described below, we made use of the semantic annotation provided by Ontotext Corporation [Ontotext Corporation, 2006]. Ontotext Corporation provides an interface to roughly 500,000 news articles from sources such as Reuters, the PA, ABC News, the BBC, and CNN. Each document has been automatically labelled with a set of keyphrases. Each key term has an associated high-level semantic label from Ontotext, one of

---

1We will use the term key term here to refer to the sets of keywords and named entities we obtain from Ontotext. Key terms may be either single words (e.g., tsunami, germany) or combinations of words (e.g., search engine, angela merkel).
The semantic annotation technology from Ontotext makes use of the KIM Platform implemented within GATE [Cunningham et al., 2002, Bontcheva et al., 2004]. KIM, and the semantic annotation underlying it, are described in more detail in [Kiryakov et al., 2004, Popov et al., 2004]. For our experiments, Ontotext provided a Java applet that enabled us to query the archive by key term and receive a set of semantically annotated documents in XML format in return.

7.4 Algorithm

The algorithm for processing the set of automatically annotated news articles closely parallels the one used to construct summary responses in the restaurant domain, with several differences that we describe in more detail in the following sections. It consists of the following steps:

- Take as input the set of annotated news articles for the topic in question;
- Extract annotations for selected semantic entities found in each article, keywords, locations, and persons; (Section 7.4.1);
- Perform simple stemming, remove stop words, compute the most frequently occurring examples of each type (Section 7.4.2);
- Create a database representing each document by the presence or absence of entities (Section 7.4.2);
- Construct clusters of entities within documents, broken down by entity type (Section 7.7.1);
- Optionally combine clusters;
- Rank clusters by number of documents they comprise;

7.4.1 Attribute selection/ranking

As noted, Ontotext provides annotations for four semantic types, “KeyPhrase”, “Location”, “Person”, and “Time”. From among these, we had to choose
Table 7.3: Table showing search terms used to determine a profile for semantic types in the Ontotext news domain.

<table>
<thead>
<tr>
<th></th>
<th>People</th>
<th>Organizations</th>
<th>Countries</th>
</tr>
</thead>
<tbody>
<tr>
<td>KeyPhrases</td>
<td>15669</td>
<td>13435</td>
<td>8543</td>
</tr>
<tr>
<td>Location</td>
<td>11292</td>
<td>2569</td>
<td>8662</td>
</tr>
<tr>
<td>Person</td>
<td>10976</td>
<td>3671</td>
<td>2743</td>
</tr>
<tr>
<td>Time</td>
<td>6540</td>
<td>4407</td>
<td>2957</td>
</tr>
</tbody>
</table>

Table 7.4: Profile of the semantic types for key terms found for the three story topics used.

the set to use as input to the clustering algorithms. We first processed a set of 1000 articles obtained from Ontotext using a variety of search terms. The search terms were divided among three broad categories, people, organizations, and countries, and chosen to provide a range within each. The three categories, and the specific search terms used are shown in Table 7.3.

Table 7.4 shows the numbers found for each of the semantic types in the Ontotext annotation data, divided by the category of the search term used to find them. The semantic type with the largest representation among all stories was “KeyPhrase”. The type with the next largest representation in the annotations varies from story to story: “Location” for stories found by searching on people and countries, “Time” for stories found by searching on organizations.

“Time” posed problems for automatic processing of the data. The “time” semantic type includes phrases such as “Friday”, which would have to be canonicalized with reference to some absolute date. We did not have access to the original text of the news article (see Section 7.4.2), and, therefore, had limited resources to use to perform such a task. In addition, the semantic entity “Time” was also applied to phrases such as “last minute” and “first 100 days”, which would be of limited value for users trying to browse the archive to get a general sense of what it contained.

For insight into the semantics of search on a large-scale, we examined
the top-10 search terms used at yahoo.com during the years 2005 and 2006. All 20 terms (some replicated in the two years) were named entities. For 2005, these terms were all proper names of people, whereas for 2006 the list included an organization (WWE) and a television show (American Idol).

For a look at searches in a news domain, we also examined the top-10 search terms used on Google News for the same two years. For 2005, these terms were, for the most part, proper names, but also included what would be classified as a “KeyPhrase” by Ontotext, e.g., tsunami. The year 2006 contained several such key terms, including cancer and autism. Table 7.5 shows the top-10 search terms for these two years for Google and Yahoo.

Based on the information contained in the annotations themselves, and the evidence from real-world search engines, we selected the three semantic types “KeyPhrase”, “Location”, and “Person” to use in the experiments described below. We hypothesized that these types would be familiar to users accustomed to search engines such as Google and Yahoo, and would help users browse the contents of the news archive. We were then left with the choice of which combinations of these entities to use to cluster the documents. Rather than decide this a priori, especially for the purpose of finding clusters, we construct clusters based on all possible combinations of the three entities, resulting in the following groupings of semantic entities:

1. KeyPhrases only
2. Locations only
3. People only
4. KeyPhrases + Locations
5. KeyPhrases + People
6. Locations + People
7. KeyPhrases + Locations + People

7.4.2 Pre-processing data/Creating database

News articles corresponding to search terms were first harvested using the Java applet provided by Ontotext. For the experiment described in Section 7.5 and Appendix A.4, the search terms used were “earthquake”, “earthquake AND indonesia”, “earthquake AND pakistan”, “microsoft”, and “microsoft AND china”. These terms were chosen to reflect newsworthy items
### Top-10 Search Items Per Year

<table>
<thead>
<tr>
<th></th>
<th>Yahoo</th>
<th>Google News</th>
</tr>
</thead>
<tbody>
<tr>
<td>2005</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Britney Spears</td>
<td>Paris Hilton</td>
</tr>
<tr>
<td>2</td>
<td>50 Cent</td>
<td>Hurricane Katrina</td>
</tr>
<tr>
<td>3</td>
<td>Mariah Carey</td>
<td>tsunami</td>
</tr>
<tr>
<td>4</td>
<td>Jessica Simpson</td>
<td>xbox 360</td>
</tr>
<tr>
<td>5</td>
<td>Paris Hilton</td>
<td>Brad Pitt</td>
</tr>
<tr>
<td>6</td>
<td>Eminem</td>
<td>Michael Jackson</td>
</tr>
<tr>
<td>7</td>
<td>Ciara</td>
<td>American Idol</td>
</tr>
<tr>
<td>8</td>
<td>Lindsay Lohan</td>
<td>Britney Spears</td>
</tr>
<tr>
<td>9</td>
<td>Jennifer Lopez</td>
<td>Angelina Jolie</td>
</tr>
<tr>
<td>10</td>
<td>Kelly Clarkson</td>
<td>Harry Potter</td>
</tr>
<tr>
<td>2006</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Britney Spears</td>
<td>Paris Hilton</td>
</tr>
<tr>
<td>2</td>
<td>WWE</td>
<td>Orlando Bloom</td>
</tr>
<tr>
<td>3</td>
<td>Shakira</td>
<td>cancer</td>
</tr>
<tr>
<td>4</td>
<td>Jessica Simpson</td>
<td>podcasting</td>
</tr>
<tr>
<td>5</td>
<td>Paris Hilton</td>
<td>Hurricane Katrina</td>
</tr>
<tr>
<td>6</td>
<td>American Idol</td>
<td>bankruptcy</td>
</tr>
<tr>
<td>7</td>
<td>Beyonce Knowles</td>
<td>Martina Hingis</td>
</tr>
<tr>
<td>8</td>
<td>Chris Brown</td>
<td>autism</td>
</tr>
<tr>
<td>9</td>
<td>Pamela Anderson</td>
<td>2006 NFL draft</td>
</tr>
<tr>
<td>10</td>
<td>Lindsay Lohan</td>
<td>Celebrity Big Brother 2006</td>
</tr>
</tbody>
</table>

Table 7.5: Top-10 terms given to search engines at yahoo.com and news.google.com.
that might be of general interest to a broad range of subjects. For the experiment described in Chapter 8, the search terms used were “sarkozy”, “microsoft”, and “china AND pollution”. These terms were chosen to represent a range of semantic types and to conform to a set of breaking news stories, described in more detail in the next chapter. Searches based on these terms returned a set of topically coherent documents from within the news archive. The Java applet returns a set of document annotations, not the documents themselves. The annotations contain the identified key terms, the semantic type they are associated with (i.e., “KeyPhrase”, “Location”, etc.) and the position in which they occur in the document, in XML format. Overall document length can be derived from information within the annotations.

Before processing the key terms, we performed stemming and regularization on the strings. Specifically, we conflated singular and plural forms of the same noun, and converted variant forms of the same name within a particular class (e.g., “u.s.”, “u.s.a”, and “united states of america” to “united states”) to a single representation. We then created an initial frame-like representation of each document, in which each semantic type and corresponding word(s) is an attribute/value pair. We preserve information about the position (in words from the top) in which each attribute/value pair appear in the original document, as well as the length of each document in words.

Documents within each topic group were then processed to identify the top-10 most frequently occurring key terms for the semantic types “KeyPhrase”, “Location”, and “Person”. This results in 30 possible key terms per document within each topic group. For consideration among the top-ten, counts on key terms were done in the top 1/3 of the document. The rationale for this is based on previous work, where position in the document has been applied specifically to the task of summarization in news delivery systems [Diaz and Gervas, 2005, Hovy and Lin, 1997, Doran et al., 2004]. In addition, we base this decision on the traditional “inverted pyramid” style of journalistic writing, in which the most relevant information for a particular story is provided at the beginning, with background information typically presented later in the article.

Using 30 possible values for each topic group, we create a database in which documents are represented by the presence or absence of each of the keyphrases. In other words, the keyphrases themselves were the attributes
Table 7.6 shows an example of a frame-based representation of a news story found using the search term “sarkozy”. In this example, the entities KeyPhrase and Location are used to construct the frame that describes a news article in which Nicolas Sarkozy is mentioned along with the Key Phrases “violence” and “youth” and the Location “Marseille”. This representation is functionally identical to the representation of entities used in the restaurant domain. Restaurants and documents are both represented as attribute/value pairs, but with the former the attributes have a semantic class (e.g., cuisine) and the values are instantiations within that class, whereas with the latter the attributes are discovered keyphrases and the values are boolean primitives denoting the presence or absence of that keyphrase within the document.

7.5 Natural language experiment

After constructing the database of documents, we designed an experiment based on the second user study done in the restaurant domain (described in Chapter 5), to test the hypothesis that associative summaries constructed from documents would help users browse the information contained in a news archive. A concern noted with the preliminary investigation using MUC data is that a user model based on dispositional preferences did not seem to be applicable in a domain whose attributes were “KeyPhrase”, “Location”, and “Person”. It was unlikely that users would be able to say which of those they preferred, even if they knew the topic of the overall search. This fundamental difference between the two domains carried over into other aspects of the experimental design, as well. With restaurants, it was relatively easy for subjects to understand the task at hand, i.e., finding an interesting place to eat in the city of London by browsing through a database of restaurants. For the news archive, the goal was less clear. If the goal was to find an “interesting” news article, we were faced with the issue of controlling for the varying definition of “interesting” users from a general population might employ.

Despite these concerns, we piloted an experiment designed to contrast discovered associations with singleton values. Rather than having four choices at any given point in the dialogue, as was the case with the restaurant experiment, users were asked to compare two types of summaries, associations and singleton values, across seven possible attribute combinations (see
Table 7.6: Example frame-based representation of news story found using the search term “Sarkozy”.

```json
{q story
    :index all_0000813
    :violence 1
    :youths 1
    :candidate 0
    :police 0
    :constitution 0
    :law 0
    :students 0
    :referendum 0
    :unions 0
    :protests 0
    :united_states 0
    :brussels 0
    :europe 0
    :iraq 0
    :germany 0
    :britain 0
    :marseille 1
    :italy 0
    :spain 0
    :seine 0
}
}
Section 7.4.1). Users were asked to imagine a dialogue system interface
to a news archive, where queries about news-related topics such as “earth-
quakes” or “Microsoft” were answered with information about the keywords
and named entities found in stories concerning those topics. The summary
responses were framed in natural language similar to that used in the the
restaurant experiment. Details of this pilot experiment are contained in
Appendix A.4.

7.6 Natural language experiment results

The results from this pilot experiment suggested that users had very different
notions of the goals of the simulated task and also that the summary types
created were not helping them browse the news archive. The overwhem-
ing majority of people preferred singleton value responses over associations
(67.8% to 32.1%). However, the more interesting finding from this prelimi-
nary experiment was the strength with which people disliked both types of
responses. On a scale of 1-7, the average ranking for the responses with
singleton values was 4.3 and the average for associations was 3.2 (significant
at \( p < .005 \)). We had encouraged users to use the full range of values on the
Likert scale when ranking the responses, but they rarely moved above 5 for
either type of response. 65.6% of the rankings were 4 or below; 87.5% were
5 or below.

Two issues emerged in the optional feedback provided via a free-form
text interface. Users spoke of the “confusing” and “verbose” nature of the
responses, as well as the “lack of apparent structure”. Users also disliked the
limited set of options they were given, complaining that nothing of interest
was shown among the choices they saw. For example, several users spoke
of wanting to see “scientific” information about earthquakes rather than
the geopolitical information that was implied by the responses they saw
(samples of these responses are in Appendix A.4). Others complained about
entities found for Microsoft, saying that their interest in Microsoft was in its
software, not in its relationship to “China” or the “internet”, both of which
were keywords discovered and highlighted by both response construction
algorithms.

This quantitative and qualitative feedback led us to re-examine the ex-
perimental design and hypotheses. We will return to these concerns and
describe the revised experimental design in Chapter 8. In addition, there
was a technical consideration informing the experimental design, specific to
the news domain. Recent work has looked at a method of discovering associations among keywords in a news archive that was quite different from, and more computationally expensive than decision tree induction [Alipio, 2004]. This algorithm (described in Section 7.7) is capable of producing the same sort of clustered key terms as decision tree induction, but has not been tested in user studies. By incorporating this algorithm into the process by which we compute associations, we can test it against the more lightweight decision tree induction method. We describe this alternative method in the next section.

### 7.7 Alternative method of producing clusters in the news domain

For constructing natural language summaries in the restaurant domain, we made use of clusters discovered by decision-tree induction [Quinlan, 1993a]. As we (and others [Kamber et al., 1997, Liu et al., 2000, Ruggieri, 2002]) have noted, decision trees are able to offer insights into the dependencies found among attributes in a dataset, and lend themselves to easy interpretation. For describing a set of entities in a tourist domain such as restaurants, the rules discovered using decision tree induction are sufficient; as we have noted, we are not interested in the ability of the decision tree to classify a new, unseen restaurant. Rather, we are simply interested in learning about the broad generalizations contained within the training data, and in the ability to easily convert those generalizations into a summary for users.

The motivation for using a machine learning algorithm remained the same in the news domain, i.e., we wished to discover associations that facilitate our subjects’ ability to browse. In that regard, c4.5 appeared to be sufficiently powerful. It is used in this domain in the way described in Section 7.4. However, a variant of c4.5, called Predictive Apriori, was developed to give added weight to associations judged to be more indicative of unseen data [Agrawal et al., 1993a,b], and an algorithm for clustering news article was designed using it.

The algorithm, described in [Alipio, 2004], makes use of Predictive Apriori combined with hierarchical agglomerative clustering. It was deployed on a set of “published executive news texts”, but it is not clear how large the dataset is. Furthermore, only keywords were associated with documents (i.e., no named entities) and those keywords were chosen by hand to de-
scribe each document at the time it was written. We felt there was a need to test the algorithm on a larger dataset that was automatically annotated. Furthermore, to our knowledge, no one had ever done a user evaluation of the usefulness of the resulting clusters.

The were several risks involved in using this algorithm. In the first place, the algorithm (described in Section 7.7.1) is computationally expensive. It includes an agglomerative clustering step that takes $O(n^2)$ time; for large bodies of news articles, this could result in a significant lag, especially in an interactive system. By contrast, decision tree induction algorithms in general, and c4.5, in particular, take $O(n \log n)$ time, and several parallel implementation algorithms speed them up even further [Ruggieri, 2002, Joshi et al., 1998, Shafer et al., 1996]. Secondly, the way in which the algorithm is constructed can result in clusters that show only potential relationships among keywords found in documents (see Section 7.7.3). Relationships are found among words, not among words found within documents. Therefore, a particular set of words may cluster across multiple documents but never appear together within a single document. As useful as those clusters might be for discovering unforeseen relationships, they could prove to be of limited use as a browsing aid, since the discovered clusters of words do not necessarily appear together in any one document. Therefore, using such a cluster for searching could potentially not lead to a corresponding document in the archive. (see Section 7.8 for how we dealt with this issue).

Because of our interest in a browsing interface that is deployable in a real-time interactive system, it is important to know the capabilities and the costs of the technology used. Furthermore, the unstructured nature of the text domain is significantly different from the restaurant domain, and it is possible that users’ strategies in browsing within the two domains are different and require different solutions. The database itself is different, given that documents are represented as binary vectors of semantic entities. We thus implemented this second algorithm for clustering keywords in news documents, described in the next section. In Chapter 8, we describe an experiment, one of whose goals is to test the two ways of discovering clusters in a way that was transparent to the user but measurable after the fact.

### 7.7.1 Algorithm for clustering text data

In [Alipio, 2004], the algorithm for combining Predictive Apriori with bottom-up agglomerative clustering algorithm to arrive at a set of clusters proceeds
as follows:

- Compute Predictive Apriori rules;
- Convert each Predictive Apriori rule to a binary vector, by presence/absence of a particular keyphrase (Section 7.7.2);
- Compute Manhattan distance on binary vectors, resulting in vector of dissimilarities (Section 7.7.3);
- Cluster dissimilarities, using metric designed to find similar clusters, producing tree of dissimilarities (Section 7.7.3);
- Cut tree into groups of related attributes (Section 7.7.3);
- Recover values for key terms for clustering (Section 7.8);

In addition, we added the following steps to enable us to test the algorithm in a browsing interface and to contrast it with straightforward decision tree induction:

- Remove clusters that do not represent groups of documents in archive (Section 7.8)
- Combine clusters with those discovered using c4.5, maintaining history of how cluster was derived
- Prune clusters (Section 7.8)

### 7.7.2 Predictive Apriori

*Predictive Apriori* adds another refinement to c4.5, in automating a measure of interestingness of rules. The measure of interestingness used by *Predictive Apriori* is based on the assumption that, in evaluating the output of a classifier, users are concerned with finding rules that will work most effectively on new, unseen data. In scoring association rules in the restaurant domain, *coverage/support* was used as the evaluation metric. Another traditional metric for evaluating association rules is *confidence* [Witten and Frank, 2000]. *Confidence*, also referred to as *accuracy*, measures the number of instances that are correctly classified expressed as a percentage of all instances that the rule applies to.
The basic idea behind Predictive Apriori is that confidence, computed on training data, is most likely overly optimistic as a measure of performance on new data and should thus be discounted. The degree to which it should be discounted is dependent on support, i.e., the greater the support for a rule, the more reliably the confidence models the predictive accuracy. (See Section 4.5.5 for a review of these terms.)

Scheffer [2001] defines predictive accuracy in the following way:

Let $D$ be a database whose individual records are generated by a static process $P$, let $X \Rightarrow Y$ be an association rule. The predictive accuracy $c(X \Rightarrow Y = \Pr(r \text{ satisfies } Y | r \text{ satisfies } X))$ is the conditional probability of $Y \subseteq r$ given that $X \subseteq r$ when the distribution of $r$ is governed by $P$ [Scheffer, 2001].

Equation 7.7.2 shows how Predictive Apriori is computed.

$$E(c([x \Rightarrow y])|\hat{c}([x \Rightarrow y]), s(x)) = \frac{\int cB[c, s(x)](\hat{c}(r))\pi(c)dc}{\int B[c, s(x)](\hat{c}(r))\pi(c)dc}$$

where $c([x \Rightarrow y])$ is the observed confidence of a rule, $\hat{c}([x \Rightarrow y])$ is the expected accuracy of the rule, and $s(x)$ is the support of the rule. $\pi(c)$ is the histogram of accuracies of all association rules over $x$ and $y$ in the given database. $B[c, s](\hat{c})$ is the binomial distribution, i.e., the chance of a correct prediction for a value $x$ in a database, given the observed accuracy $c$. The probability of the confidence, given the support and the predictive accuracy, is simply the likelihood that a given rule will classify an instance correctly or not. Because Equation 7.7.2 returns a lower, “corrected” value for predictive accuracy, based on a normalization constant computed by integrating over the space of all association rules with a given accuracy, it has also been called Bayesian frequency correction [Scheffer, 2001].

Equation 7.7.2 quantifies how strongly the confidence of a rule has to be corrected, given the support. The details of how the integrals are approximated using discretisation can be found in [Scheffer, 2001, Mutter et al., 2004]. Predictive Apriori have been shown to work well on a variety of datasets [Kotsiantis and Kanellopoulos, 2006].

The input and output specifications for Predictive Apriori are the same as for c4.5. Input is a set of documents, represented as binary vectors (see Section 7.4.2). The output is a set of rules of the form $x \Rightarrow y$, which can also be represented as a tree. Our implementation used the version of Predictive Apriori contained in the WEKA toolkit [University of Waikato, 2004].
students=1 && unions=1 && marseille=1  -->  violence=1
suburbs=  && seine=1  && youths=1  -->  violence=1

Table 7.7: Example *Predictive Apriori* rule found for documents found searching on ‘Nicolas Sarkozy’.

<table>
<thead>
<tr>
<th>students</th>
<th>unions</th>
<th>suburbs</th>
<th>youths</th>
<th>violence</th>
<th>marseille</th>
<th>seine</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.8: Binary representation of *Predictive Apriori* on documents found searching on ‘Nicolas Sarkozy’.

### 7.7.3 Hierarchical agglomerative clustering

We represent the individual rules that form the output of *Predictive Apriori* in the same way we represent individual documents in the news archive, i.e., each rule becomes a set of boolean values denoting the presence or absence of any key term found. Table 7.7 shows an example of a rule derived using *Predictive Apriori*. In Table 7.8 we show these same two rules represented as binary vectors.

Once we have converted the rules, we have, in a sense, another database representing the associations of rules governing key terms discovered using *Predictive Apriori*. This database is used, rather than the one representing the set of documents, for the clustering steps described below. At this point, our data representation no longer reflects individual documents in the archive. The individual records in the new database correspond to a rule that represents an association of key terms found within some number of documents in the archive.

For hierarchical agglomerative clustering, we use algorithms contained in the software package R [R Project, 2006]. The first step in this type of clustering is to create a matrix of dissimilarities among the rules, using a specified distance metric. We use Manhattan distance, which simply sums the number of differences between two vectors in terms of a straightforward match of individual components within the vector [Mitchell, 1997]. Given that the rules were binary vectors, the difference between any two of them can be represented by Manhattan distance.

We then cluster the dissimilarities, using Ward’s minimum variance met-
Table 7.9: Example *Predictive Apriori* rules found searching on ‘Nicolas Sarkozy’ showing associations among key terms.

To illustrate the hierarchical agglomerative clustering process, we use a toy database, based on a grouping of keywords and locations from the Sarkozy stories. Table 7.9 shows a subset of the rules from *Predictive Apriori* for this set of stories. For ease of reading, we have represented each set of stories by their attribute names when those attributes were present, rather than the binary value '1'. When that attribute was missing, the attribute is represented as '0'. The rules are given line identifiers on the left to make it easier to refer to them in the text.

A casual perusal of this table shows a thematic group comprised of lines
Table 7.10: Manhattan Distance computed on rules shown in Table 7.9.

1, 2, and 3, around the key terms “unions”, “violence”, and “marseille”; and another comprised of lines 4, 5, and 6, including the terms “suburbs”, “youths”, “violence” and “seine”. The distance matrix computed for these rules (Table 7.10) confirms this impression. The cell at the intersection of each row/column contains the Manhattan distance between the two rules defined by that intersection. The larger the number, the greater the difference between the rules.

A dendrogram created using this distance matrix is shown in Figure 7.1. The dendrogram shows how each line was grouped based on dissimilarity. The height of the lines corresponds to the distance between the groups. As can be seen, the thematic groups observed previously are present in the dendrogram, with lines 1 and 2 at the same level (being identical at all values), and line 3 separated by one level from 1 and 2 (corresponding to the value, '1', of the dissimilarity among these three rules (see Table 7.10). Likewise, lines 5 and 6 are clustered at the same level, separated from line 4 by 2 levels in the dendrogram.

Figure 7.2 shows a tree created by cutting the dendrogram at level 3, i.e., cutting the dendrogram into groups that are no further than 3 levels apart in the tree. In Figure 7.2, rules 5 and 6 and rules 7 and 8 form two units, as they are at the same level in the tree. Rules 1, 2, and 3 form one unit, as they are separated by only one level of height, and the tree was cut at level 3. Rule 4 forms its own group, given that its distance (i.e., height in the dendrogram) from the other groups is uniquely 4. Table 7.11 shows the groupings of key terms derived by combining key terms found in rules grouped by the tree in Figure 7.2.

Using these groups as search terms (along with “Sarkozy”) we queried the
Figure 7.1: Dendrogram computed on dissimilarities shown in Table 7.10.

<table>
<thead>
<tr>
<th>Rule groupings</th>
<th>Terms in rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3</td>
<td>“unions”, “suburbs”, “violence”, “marseille”</td>
</tr>
<tr>
<td>5, 6</td>
<td>“suburbs”, “youths”, “violence”, and “seine”</td>
</tr>
<tr>
<td>7, 8</td>
<td>“violence”, “marseille”</td>
</tr>
<tr>
<td>4</td>
<td>“students”, “unions”, “suburbs”, “violence” and “seine”</td>
</tr>
</tbody>
</table>

Table 7.11: Rule groupings derived from the tree shown in Figure 7.2, with the corresponding key terms found in the rules in Table 7.9.
Ontotext news archive. We found stories describing two separate incidents that were found using the first two groupings in Table 7.11. The first set of stories describe Nicolas Sarkozy’s reaction, as Interior Minister, to union-led protests in Marseille. The second describe his reaction to riots among youths in Parisian suburbs in the Seine-Saint-Denis area.

It is important to note that, at the stage of hierarchical agglomerative clustering, we have lost the direct connection between combinations of keywords discovered and the documents themselves. Both decision tree induction and Predictive Apriori classify data that represents document profiles, so the coverage of the rules they discover in the training data is a direct indication of the number of documents in which the groupings are found. The process of hierarchical agglomerative clustering, however, acts on the entities found within the rules and may discover single terms that co-occur across rules but not within individual documents.
7.8 Combining/pruning clusters

In deriving the final set of clusters for any topic, we first run c4.5 in the iterative fashion described in Section 4.5.4 and used in the restaurant domain. In a separate process, we run the Predictive Apriori-based method and combine the output of the two algorithms. In combining the two sets, we maintain the provenance of each cluster (i.e., the algorithm used to obtain it). This information enables us to assess cluster use by technology in Section 8.3.1.

The set of all clusters is then divided into two subsets. In one set, distinctions are maintained between associative clusters found using different combinations of semantic types (e.g., “Key Phrase” only, “Key Phrase” plus “Location”, etc.). There are seven groupings in this set, following the combinations described in Section 7.4.1. This set is used to construct the Full Associations interface shown to users, described in Section 8.1. The second set combines all clusters, irrespective of the semantic entities used. This set is used to construct the Combined Associations set of clusters described in Section 8.1.

Once clusters are combined, identical clusters found by both algorithms are removed. This results in an average reduction of 64.6% in clusters derived using the three search terms used for the experiment described in Chapter 8 (“sarkozy”, “microsoft”, and “china AND pollution”). Any cluster whose combination of key terms is not represented by a document in the news archive is then removed, with an average reduction of 39.7% in the number of clusters across topics. The latter step is done to insure that the set of clustered keywords enables browsing the archive. The result of these two initial pruning steps is shown in Table 7.12.

For the three topic types, there is an average of 241 clusters of key terms after this initial pruning. This is still far too many to display to users. Asking users to examine 241 clusters would defeat the purpose of an interface designed to condense information in the archive and provide a quick overview of the contents. Furthermore, with an average of only 27 unique key terms used in clusters for each topic, many of the clusters of the same length and with the same entity types are very similar. The top set of clusters in Table 7.13 represents the full set of clusters of length 3, comprised “Key Phrases” only, within documents about Nicolas Sarkozy. As can be seen, many of the key terms occur multiple times throughout this set, with minor variations in associations.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Clusters in full set</th>
<th>Clusters after duplicates removed</th>
<th>% reduction</th>
<th>Clusters after unrepresented removed</th>
<th>% reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarkozy</td>
<td>1850</td>
<td>561</td>
<td>69.7</td>
<td>387</td>
<td>31</td>
</tr>
<tr>
<td>Microsoft</td>
<td>499</td>
<td>298</td>
<td>60.3</td>
<td>161</td>
<td>46</td>
</tr>
<tr>
<td>China</td>
<td>532</td>
<td>299</td>
<td>63.7</td>
<td>175</td>
<td>41.5</td>
</tr>
</tbody>
</table>

Table 7.12: Statistical profile of clusters found on the three topic types used in the experiment, showing the number in the full set of discovered clusters, and the number and percentage reduction after initial pruning of duplicate clusters and those that did not represent a document in the archive.

Table 7.13: Clusters of length 3 discovered using key terms found in stories on Nicolas Sarkozy.
To reduce the number of clusters for display to users, we developed a pruning algorithm that applies to each set of clusters of equal length. Pruning is performed iteratively, using Equation 7.1.

\[ \text{score} = N_{\text{intersection}} \times \left( \frac{1}{\text{count}} \right) \]  

(7.1)

The overall score is computed by multiplying the number of key terms that are in the intersection between the two clusters by the reciprocal of the count of the number of key terms in the clusters. The resulting score is a number between 0 and 1. The closer to 1, the larger the overlap between the two lists.

Clusters are first sorted and then pair-wise comparisons are performed iteratively. Clusters whose scores are over .5 (meaning that more than half their key terms match) are eliminated. With the set of clusters used for the Full Associations interface, the pruning algorithm is applied to clusters that are of equal length and contain the same semantic types. For the Combined Association interface, all clusters of the same length are compared and pruned, regardless of the semantic types used to construct them.

Table 7.13 shows the results of pruning clusters derived from the “Key Phrase” semantic type only in the Nicolas Sarkozy stories. Thirteen clusters are reduced to 3, a reduction of 76.9% for that particular subset of clusters. No attempt is made to take semantics into account, and it is possible that a cluster containing a particularly salient content word will be removed if it happens to overlap with other clusters that contain the same key terms. However, we felt it was important to show a small, and algorithmically determined, set of clusters to users.

To ensure that information was not lost in the process of pruning clusters, a post hoc test is performed on the pruned clusters to see if any key term from the original set had been completely removed from the pruned set. We found only one case where this occurred for a key term represented in a cluster that described a set of documents in the archive. As a final step, we replaced the cluster containing that key term in the overall set. ²

Table 7.14 shows the result of the pruning algorithm on clusters found in each of the three topics used in the experiment. Included is the percentage reduction in number of clusters from the full set of unique clusters rep-

²Because any such post hoc test would add to the complexity and elapsed time of the underlying technology, we checked to see if the cluster found this way was used by subjects. Although information based on such a small set of data is unreliable, it is interesting to note that the cluster recovered in this was not used by an experimental subject.
senting documents in the archive. The reduction in clusters achieved by pruning is quite large, with the Combined Associations interface having the fewest clusters shown to users. As mentioned, this reduction is intentional; the interface is designed to condense information for presentation to users.

Table 7.14 also shows the number of clusters for each interface that were derived from the two clustering algorithms. The numbers indicate those clusters discovered by c4.5, and the unique clusters discovered using Predictive Apriori and hierarchical agglomerative clustering. If a cluster was discovered by both algorithms, it was included in the count for c4.5. The number of clusters that represent documents in the archive is much smaller for the Predictive Apriori-based algorithm. The clusters of associated key terms, described statistically in Table 7.14, are included in Appendix A.6. Both Full Associations and Combined Association sets for the three topics used in the experiment described in the next chapter are shown.

7.9 Conclusion

For technology to be effective as a way of enabling browsing structured data, it should be portable. In this chapter, we looked at the issues involved in porting technology used for browsing a database of structured data in the restaurant domain to a domain of news documents whose annotation scheme was automatically induced. We examined the technological issues that arise from using the algorithms described in Chapter 4 to a database created from attribute/value pairs, derived automatically using information retrieval technology, and describing the contents of articles in a news archive.

We first applied techniques described in Chapter 4 to a hand-annotated corpus of news articles. Attribute selection metrics used in the restaurant domain were not applicable here and the degree of specificity that could be obtained from hand annotations did not seem necessary for browsing. Furthermore, a completely automated browsing solution requires automatically annotated data, which we describe.

Accessing an online archive of news articles automatically annotated for named entities and keywords, we downloaded a set of topically coherent documents resulting from a keyword search. We then converted the annotation for each document into a binary vector of key/value pairs, with each pair indicating the presence or absence of a named entity or keyword in the particular document. The same method of decision tree induction used in the restaurant domain was employed here to discover associated entities in this
Table 7.14: Profile of number of clusters displayed in the *Full Associations* and *Combined Associations* interfaces for the three topic types. The total number of unique clusters represented in documents in the archive is shown first. Then, for each interface, the total after pruning is shown, along with the number from the total that was obtained by either c4.5 decision tree induction or *Predictive Apriori* + hierarchical agglomerative clustering. The numbers in parentheses are the percent reduction in number of clusters as a result of pruning.

<table>
<thead>
<tr>
<th>Clusters represented in archive</th>
<th>Sarkozy</th>
<th>Microsoft</th>
<th>China</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clusters displayed in</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Full Associations Interface</em></td>
<td>387</td>
<td>161</td>
<td>175</td>
</tr>
<tr>
<td><em>Full Associations Interface</em></td>
<td>68 (82.5)</td>
<td>38 (76.4)</td>
<td>59 (66.3)</td>
</tr>
<tr>
<td>from c4.5 clustering</td>
<td>60</td>
<td>24</td>
<td>47</td>
</tr>
<tr>
<td><em>Full Associations Interface</em></td>
<td>8</td>
<td>14</td>
<td>12</td>
</tr>
<tr>
<td>from Pred. Apriori clustering</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clusters displayed in</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>Combined Association Interface</em></td>
<td>34 (91.3)</td>
<td>23 (85.8)</td>
<td>31 (82.3)</td>
</tr>
<tr>
<td><em>Combined Association Interface</em> from c4.5 clustering</td>
<td>25</td>
<td>17</td>
<td>22</td>
</tr>
<tr>
<td><em>Combined Association Interface</em></td>
<td>9</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>from Pred. Apriori clustering</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The entities were optionally grouped by their semantic types. In addition, associative summaries were discovered within the same data using Predictive Apriori and hierarchical agglomerative clustering. Associations were then pruned to produce a reduced set representing relationships within particular sets of documents. In the next chapter, we will evaluate the usefulness of the discovered associative clusters in a task-based experiment with human subjects.
Chapter 8

Testing the effectiveness of associative summaries: An experiment with a news archive

8.1 Experimental setup

8.1.1 Background

Much of the research in summarization techniques has focussed on underlying technology (see Section 2.3). Where this work has been evaluated, it has often been in the context of laboratory-created corpora, with an emphasis on evaluation of the underlying technology [MUC, 1991, 1992]. Our work is concerned with the technology, but we also want to ensure that any solution we discover is effective in a real-world setting. The DARPA Communicator project pointed out the necessity of true, task-based evaluation [Walker et al., 2002b]. Although that evaluation paradigm was in the context of spoken dialogue systems, the principle holds for any interactive system.

The experience gained from the natural language experiment (Section 7.5) helped inform a new design for testing summaries in a news domain. One of the main lessons from that pilot experiment was that it was unclear if associative summarization using keywords was useful at all for a news archive. We needed to confirm Hypothesis H8, that useful and informative summaries could be created when focal information is unstructured. To do so, an experiment similar to the initial experiment in the restaurant domain was needed (see Chapter 3), i.e., one that compared responses highlighting associations against a baseline interface that was familiar to users and did not use summarization. Experiments have examined the use of keywords pre-
sented as singleton values, and compared that to a baseline [Cutting et al.,
1992, Jones and Paynter, 2001], but no work has, to our knowledge, tested
the use of associations discovered among keywords, either as a standalone
technology or as compared to a baseline.

The second lesson from the natural language experiment was that users
bring a range of preconceived interests to topics and, in that experiment,
they tended to judge the technology by how well it addressed those interests,
rather than by how well it provided browsing support for the contents of the
archive. This pointed out the necessity of testing our technology with users
who have a professional interest in the information contained in a news
archive.

Thirdly, subjects spoke of being confused by the interface used in the
natural language experiment. It appeared that natural language is not the
best way to frame responses in this domain. The domain is rich and the
associations are broad, and formulating responses in the form of small sets
of natural language statements limits the amount of information provided
to the user. For users searching a text archive, a screen display using a web
browser, with associations displayed as browsable lists of key terms, seemed
the logical interface.

Given the concerns raised by the natural language experiment, we re-
designed the experiment to (1) have a baseline that would be familiar to
subjects and could be used as a point of comparison with the summarization
technology; (2) be framed in terms of a scenario-based task with subjects
who would be familiar with that task; and (3) present associative sum-
maries in tabular format. The experimental hypothesis remained the same,
i.e., that summarizations based on attributes automatically discovered in an
unstructured corpus can aid the process of browsing.

To make the evaluation task-based, it is designed around a scenario in
which journalists are meant to use the technology to assist them in finding
background information for a news story. Background information typically
accompanies a breaking news story at some time delay, gathered and con-
structed after-the-fact by searching through information related to the story
and found in a variety of sources. The process by which journalists gather
background information has not been studied systematically that we know
of. [Attfield and Dowell, 2003] describes in general terms how journalists
use information without addressing the task of background seeking. Here,
we follow the paradigm described by [Barker et al., 2006], which described
the task as involving a search for “angles”, i.e., subtopics that relate to the breaking news story and help contextualize it for readers. Typically, a user involved in information seeking for this task is under time pressure associated with a news story deadline.

8.1.2 Experimental population/protocol

The experiment was designed to assess: (1) subjects’ perception of the utility of each interface as a mechanism for searching for background information; and (2) the quality of the information provided by each interface. Sixteen subjects were recruited from among M.Sc. students in the Department of Journalism Studies at the University of Sheffield, as well as two professional journalists working for the Sheffield Star, for a total of 18 subjects. Subjects were asked to read three “breaking” news stories (real news stories, chosen from AP newswire via Google, from within two weeks of the date of the start of the experiment) and then find background angles for each, with a time limit per story.

Of the three breaking news stories, one was about riots in France following the election of Nicolas Sarkozy as President, one was about a threatened lawsuit by the European Union against Microsoft, and one about new Chinese government measures to address pollution (the complete set of stories is shown in Appendix A.5). These topics were chosen from a number of possible news stories to comprise a set containing a range of event types/entities. The three included one that focussed primarily on a person (i.e., Nicolas Sarkozy), one that focussed on an organization and a political entity entity (i.e., Microsoft and the EU), and one that focussed on a country and a keyword (i.e., China and pollution). The categories that these three topics fell into conformed to the three used to analyse the underlying annotation technology, described in Section 7.4.1.

Subjects carried out their information seeking tasks using one of three possible interfaces to the Ontotext news archive, two that showed the associative summaries and one, Baseline, that allowed subjects to query the Ontotext archive with their own key terms. Associative summaries were provided to the subjects with two interfaces. One interface, Full Associations, shows clusters of key terms organized by the semantic entities used to find them (i.e., the seven groupings listed in Section 7.4.1). The other interface, Combined Associations, combines all clusters found for a particular document and algorithmically pruned similar ones. All clusters shown
to users for the two associative interfaces are shown in Appendix A.6.

Using the Full Associations interface, subjects interacted with a webpage that had a graphic division between the seven groupings of associative summaries (e.g., KeyPhrases, Locations Keyphrases + Locations). Subjects were able to navigate to any of these groupings of associative summaries by scrolling, or via a frame menu on the left-hand side of the page. Figure 8.1 displays a screenshot of the Full Associations interface.

Using the Combined Associations interface, subjects interacted with a webpage in which all associations found for a particular topic are combined in a single frame. Figure 8.2 shows a screenshot of the Combined Associations interface. Users are able navigate through the window by scrolling.

For the Baseline interface, users constructed search terms themselves for the breaking news story and typed those directly into an interface to Ontotext’s search facility. Figure 8.3 shows a screenshot of the interface subjects used to interact with the baseline Ontotext search engine. Rather than use the interface provided by Ontotext (shown in Figure 8.4), we designed a separate page that preserves the look-and-feel of the other two interfaces.
Figure 8.2: The interface to the Combined Associations interface, showing all clusters combined.
Figure 8.3: The interface subjects used in the Baseline system to query the Ontotext search facility with their own search terms.
Figure 8.4: The interface provided by Ontotext to their search facility. Subjects used the interface shown in Figure 8.3 to query this system, to preserve the look-and-feel of the other interfaces.

Each subject completed three tasks by interacting with each of three interfaces in turn, in a within-subject design. We varied the interface order across subjects in order to assess the effects of the interface on user behavior and experimental judgment. Across the 18 subjects, each interface was used six times as the first, second, or third interface, respectively. To mitigate the confounding effects of story type on subjects’ perception of the interface, we did not also vary story type. Each subject completed the Nicolas Sarkozy task first, the EU/Microsoft task second, and the China task third.

The webpage for each of the associative summary interfaces included a banner heading that showed the default search term(s) used to find the original set of news stories for which the associative summaries were constructed. These search terms were “sarkozy”, “microsoft”, or “china AND pollution”. (These default search terms were also explained in the instructions.) In addition, the Full Associations interface also displayed banner headings describing the semantic entities, or combinations thereof, used to
construct the associative summaries (i.e., “Keywords”, “Locations”, etc.)

The discovered key terms were displayed in green. Within each cluster, the word “AND” appeared between each key term, in a smaller font and different color. This was done to reinforce the fact that the terms in the clusters co-occurred in the documents. A “Search” link was displayed with each cluster. By clicking on this link, subjects were able to initiate a live search of the Ontotext archive for documents containing those search terms.

Figure 8.5 shows the Ontotext webpage interface to documents found via a key term search. Documents found in the Ontotext archive are displayed by headline.

1We used the term “Keyword” to describe what Ontotext termed “KeyPhrases”, on the assumption that subjects were more familiar with the former term.
8.1.3 Experimental setup

Before beginning the experiment, subjects were asked to read a set of instructions (see Appendix A.7) that explained the task scenario and how to complete it. In addition, subjects were given a sample breaking news story (also shown in Appendix A.7) as a “warm-up”. The three interfaces used in the experiment had been configured for the warm-up story, and subjects were given a much time as they wanted to work through the warm-up task while familiarizing themselves with the interfaces. Experimenters were present to answer questions at this point.

Subjects read the breaking news story and then used one (or all, at this point) of the three interfaces to search for and find angles to use in writing background for the story. When they located an angle, they wrote a brief description of the angle itself (e.g., “Previous chemical spill in river in China”) in a text box and then attached the story or stories they found as background to support the angle. Each experimental interface was set up with an identical text box to use for the angles and supporting documents, although subjects were also free to use any text editor they wanted to gather the information and then cut-and-paste it into the text box for submission.

After completing the warm-up, subjects returned to the main experimental page, where they were asked to indicate, in general, how familiar they were with each of the topics used in the experiment before beginning. They rated their familiarity on a Likert scale from 1 to 5, with 5 being “Very familiar”. Topics were presented to the users as simply “Nicolas Sarkozy”, “Microsoft and its corporate practices”, and “China and pollution”.

Subjects then began the experiment, under time constraint (fifteen minutes per story, with warnings at five, two, and one minute remaining). When the fifteen minutes were up, subjects were asked to answer two questions about each interface, using a 5-point Likert scale:

- How confident are you that you were able to fully explore the contents of the corpus? (with ‘1’ indicating Not confident and ‘5’ indicating Very confident)

- Would you use such a system again? (with ‘1’ indicating Not likely and ‘5’ indicating Very likely)

User input on the first of these is analyzed as the confidence metric in Section 8.2; the second is the reuse metric. After completing all three tasks and seeing all three interfaces, subjects ranked each interface by its usefulness,
Table 8.1: The scores users assigned to each interface, for overall rank, confidence, and reuse.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Rank (average)</th>
<th>Confidence (average)</th>
<th>Use again (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Associations</td>
<td>3.11</td>
<td>3.11</td>
<td>3.06</td>
</tr>
<tr>
<td>Combined Associations</td>
<td>2.94</td>
<td>3.17</td>
<td>3.0</td>
</tr>
<tr>
<td>Baseline</td>
<td>3.28</td>
<td>3.33</td>
<td>3.7</td>
</tr>
</tbody>
</table>

again on a Likert scale from 1 to 5 (with ‘1’ being *Not useful* and ‘5’ being *Very useful*). This is used as the *rank* metric in Section 8.2.

Finally, users were asked to tell us what they liked best and least about each interface, using a free-form text box. This last set of questions was optional, but all subjects except one provided feedback here. When answering this last series of questions, subjects could use a link on each question to look back at the particular interface, to see once again the look-and-feel of each interface.

### 8.2 Data analysis

#### 8.2.1 User judgments/input

Overall, the two cluster-based interfaces were ranked as top-choice by our subjects 56% of the time. This number does not include those times when users ranked one of the processed interfaces equivalently to the *Baseline*. When we factor in those rankings, i.e., the times that one of the cluster-based interfaces was either ranked as top-choice absolutely, or as top-choice in conjunction with the *Baseline*, 67% of the time subjects preferred one of the cluster interfaces.

The average rank users assigned to each of the interfaces is shown in the second column of Table 8.1. Overall, the highest ranking interface was the *Baseline* system. Preference for the *Baseline* was not significant, however, compared with the *Full Associations* interface, based on paired *t*-tests and a multivariate analysis of variance (MANOVA; Wilks' Lambda, $F(2,16) = .423, p = .662$). This lack of significant difference indicates that subjects had no strong preferences among the three interfaces.

Users’ confidence in the usefulness of each interface for exploring the archive was also not significantly different in paired *t*-tests and a MANOVA
### Table 8.2: The scores subjects assigned each story for overall rank, confidence, reuse, and familiarity. The order in the table reflects the order in which the subjects saw each story topic.

<table>
<thead>
<tr>
<th>Topic/order</th>
<th>Rank (average)</th>
<th>Confidence (average)</th>
<th>Reuse (average)</th>
<th>Familiarity (average)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarkozy</td>
<td>3.06</td>
<td>2.94</td>
<td>3.22</td>
<td>2.722</td>
</tr>
<tr>
<td>Microsoft</td>
<td>2.83</td>
<td>3.06</td>
<td>2.83</td>
<td>2.22</td>
</tr>
<tr>
<td>China</td>
<td>3.44</td>
<td>3.61</td>
<td>3.67</td>
<td>2.33</td>
</tr>
</tbody>
</table>

(Wilks’ Lambda, $F(2,16) = .242$, $p = .788$). The third column of Table 8.1 shows these scores. For the reuse metric, reflecting users’ response to the question about using each particular interface again, averages are shown in the fourth column of Table 8.1. As with rank and confidence, paired $t$-tests and a MANOVA showed no significant differences (Wilks’ Lambda, $F(2,16)=1.8$, $p = .198$).

We next performed a series of ANOVAs using each of the subjective measures elicited from users as the dependent variable, and type (i.e., Full Associations, Combined Associations, or Baseline) and story topic (i.e., Sarkozy, Microsoft, or China) as independent variables. Table 8.2 shows these subjective measures as they correspond to story topic. We did not find significant effects or interactions with the independent variable rank. However, we found a marginal effect of story topic on confidence ($p = .098$, $F = 2.24$, $df = 2$), although no interaction effects. We also found a slightly stronger, though still marginal effect of story topic on reuse ($p = .066$, $F = 3.4$, $df = 2$), again with no interaction effects.

Recall that, because stories were always presented to users in the same order, story topic is a proxy for order in our analysis. Although there was just a marginal effect of story topic on confidence, there was a significant correlation between users’ confidence in the systems and story topic (i.e., order). Users’ confidence increased monotonically over the course of the experiment, regardless of the order of the interface (see Table 8.2). The effect of order on user judgment has been seen elsewhere in search-based tasks [Borlund and Ingwersen, 1997], although with a much smaller subject population. For our subjects, confidence grew as they progressed through the experiment. This suggests that a longitudinal study using these interfaces might yield interesting results.
One possible explanation for the effects shown by story topic is the users' familiarity with the story itself. The final column in Table 8.2 shows familiarity scores by topic, which were not significantly different. Familiarity had a marginal effect on confidence (ANOVA, $p = .06$, $F = 2.9$, $df = 3$), but no effect on rank or reuse. The topic users expressed the greatest familiarity with *a priori*, Nicolas Sarkozy, was also the one that had the lowest confidence scores. The interface to stories about China, about which users had expressed a lower degree of familiarity, had the highest confidence scores.

Taken together, these findings confirm Hypothesis H8, that useful and informative summaries can be created when focal information is unstructured. The lack of a significant difference in any of our subjective measures matches what has been found elsewhere the literature [Gutwin et al., 1998, Joho et al., 2004, Kan, 2003] when comparing interfaces that process data to a Google-like baseline. The fact that two associative summary interfaces were preferred more than half the time was a positive indicator of the utility of associative summaries. A simple, user-induced keyword search is a well-known and powerful interface on its own, and it is not easy in an hour-long experiment to show the clear efficacy of a new interface compared with something that users are very familiar with and may use very often. Users' confidence grew as they progressed through the experiment, even when they were using the associative summaries in later stages. This indicates that all interfaces met subjects’ information-seeking needs to some degree. An optimal interface may combine the baseline interface with the ones that process news stories, to maximize flexibility and, possibly, user satisfaction.

### 8.2.2 Expert judgment

In addition to the ratings we elicited from subjects, we also asked a Senior Lecturer in the Department of Journalism Studies at the University of Sheffield to serve as an expert judge on subject output. This Senior Lecturer teaches two core third-year modules whose curriculum includes extensive treatment of the topic of angles in news stories and the identification of good angles. This expert judge was presented with 54 separate “packages” of documents, one for each of the stories (3) used by each of the subjects (18) to complete their tasks. Each package consisted of a set of angles, followed by the stories the subject found to support each angle. A minimal amount of formatting was performed on each packet, to remove any control characters and to highlight the angles themselves. We did not change the
<table>
<thead>
<tr>
<th>Interface</th>
<th>Usefulness</th>
<th>Richness</th>
<th>Originality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full Associations</td>
<td>2.67</td>
<td>2.22</td>
<td>2.06</td>
</tr>
<tr>
<td>Combined Associations</td>
<td>2.72</td>
<td>2.61</td>
<td>2.28</td>
</tr>
<tr>
<td>Baseline</td>
<td>3.22</td>
<td>3.28</td>
<td>2.83</td>
</tr>
</tbody>
</table>

Table 8.3: Rankings from the expert judge for each interface, on usefulness, richness, and originality.

angle itself, its wording, or order in the file, nor did we change the stories selected to back up each angle in any way.

The expert judge read the breaking news story for each topic/interface and answered three questions about each package of angles and background stories. Answers to the questions, listed below, were on a Likert scale of 1 to 5, with 1 indicating a negative opinion:

- How would you rank this package for its usefulness in building a background for the breaking news story?
- How would you rank this package for richness/comprehensiveness of background?
- How would you rank this package for originality/unexpectedness (i.e., does it contain something that is both novel and helps contextualize the event)?

Each package was examined blindly, i.e., the expert had no idea who created the package or what interface was used to find the stories.

The results of the expert judgments were used to associate measures of usefulness, richness, and originality (corresponding, respectively, to the questions above) with the other experimental variables. Table 8.3 shows the expert measures as a function of the interface used for the package. Table 8.4 shows the expert ratings for usefulness, richness, and originality, divided by topic.

Examined by interface, the baseline performs best along all dimensions. The differences are not significant, however, for usefulness or originality,

---

2A concern in this part of the experimental protocol was that it was not possible to elicit judgments from more than one expert, given the level and specificity of expertise needed to rate background material, and the time required to examine 52 sets of background angles and supporting documents. Because this is the first time, to our knowledge, that this technology has been both used and evaluated by experts in the same field, we felt that one set of judgments here would contribute to an understanding of the usefulness of the technology, while helping refine an evaluation protocol for use in follow-up experiments.
Table 8.4: Rankings from the expert judge for each topic, on usefulness, richness, and originality.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Usefulness</th>
<th>Richness</th>
<th>Originality</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sarkozy</td>
<td>2.83</td>
<td>2.22</td>
<td>2.11</td>
</tr>
<tr>
<td>Microsoft</td>
<td>2.78</td>
<td>2.78</td>
<td>2.5</td>
</tr>
<tr>
<td>China</td>
<td>3.0</td>
<td>3.11</td>
<td>2.56</td>
</tr>
</tbody>
</table>

although a MANOVA indicates significant differences in richness (Wilks’ Lambda, $F(2,16) = 4.6, p < .05$). Paired t-tests showed a significant difference between the richness scores for Full Associations vs. Baseline interfaces ($p = .006, df = 17$) and for Combined Associations vs. Baseline interfaces ($p = .048, df = 17$).

Scores associated with topic also show significant differences only for richness measured by a MANOVA (Wilks’ Lambda, $F(2,16) = 3.8, p < .05$). Paired t-tests showed significant differences in richness between Sarkozy angles and China angles ($p < .05, df = 17$).

The expert judge used in this experiment was able to provide insight into the interaction between topic type and richness, in an interview conducted after the judgments were elicited. When assessing the angles for richness, the expert judge looked for unexpected or unusual information, rather than examples of where very similar events had occurred in the past. He hypothesized that the topic with the highest scores along all dimensions, “China and pollution”, lent itself naturally to the type of background information that he would score highly for richness. He further hypothesized that angles found for the other two topics would be, by their nature, not as interesting from his perspective.

For the story about Nicolas Sarkozy, which described riots that greeted his election as President, it was common for all subjects to find examples of previous unrest surrounding Nicolas Sarkozy’s candidacy. There were many such stories in the archive. General stories of political unrest in France were also common. Therefore, the angles many subjects found for this story did not highlight unusual or unexpected circumstances in the information they contained.

The breaking news story about Microsoft, which dealt with threats of sanctions by the EU, suffered from the same sort of profile, i.e., a story whose background angles were easily found among past stories in a very similar vein. The corpus contained a variety of documents describing previ-
ous attempts by both the EU and the US government to curb what it saw as monopolistic practices on the part of Microsoft. This background information on Microsoft is interesting in terms of contextualizing the contents of the breaking news story, but wouldn’t serve to uncover any novel element to the relationship between Microsoft and the EU.

The China story, however, did turn up, with all interfaces, stories that contained information not only about the problems of pollution in China, but also about how those problems have spread across the globe, and how those problems were seen outside of China. It was this larger perspective that appealed to the expert judge.

After rating each of the packets, we asked the expert judge to go back through the angles found for each of the three topics and flag the angle+story combinations that he thought were most interesting. Not surprisingly, he found none that he felt were outstanding along this dimension for either the Sarkozy or the Microsoft story. However, he did find two for the China story, both from the same subject. These two stories were both found using the Full Associations interface, and, furthermore, were found by a subject that rated that interface the highest for usefulness. We were pleased to note that these stories were found using the Full Associations interface. The fact that the only angles felt to be truly outstanding in an expert opinion were found by the same subject, suggested that individual subject performance might be an interesting dimension to investigate.

8.2.3 Examining subjects by performance

Previous research has examined the effects of issues such as personality [Goren-Bar et al., 2006, Sultan and Chan, 2000], and experience [Lippert and Forman, 2005, Hoscher and Strube, 2000, Choo et al., 2000] on users’ acceptance of a variety of technologies related to information presentation. Here, we investigate correlations between measures from the expert judge, which evaluate subjects’ abilities to identify background for stories, and the preferences expressed by users.

Out of 18 subjects, four fell into a group that scored cumulatively highest on the measures of usefulness, richness, and originality. Those four subjects are classified as the high achieving set. Four subjects fell into a group that

---

3One of these angles had to do with China’s economy and the possible effects pollution controls would have on production. The other mentioned a long-term forecast predicting a drought for the coming year in China, adding another source of strain to China’s environment.
Table 8.5: Rankings from users, shown by groups of reflecting those whose angles were ranked highly by the expert judge and those that were ranked poorly.

<table>
<thead>
<tr>
<th>Interface</th>
<th>Avg. rank</th>
<th>Confidence</th>
<th>Reuse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Full associations</td>
<td>3.50</td>
<td>3.50</td>
<td>3.50</td>
</tr>
<tr>
<td>Combined associations</td>
<td>3.00</td>
<td>2.75</td>
<td>3.00</td>
</tr>
<tr>
<td>Baseline</td>
<td>2.00</td>
<td>3.00</td>
<td>2.50</td>
</tr>
</tbody>
</table>

High-achieving subjects ranked the two associative clustering interfaces the highest. The differences here are significant between the rank of the Full Associations and the Baseline interfaces \((p < .05, df = 3)\), with the baseline scoring significantly lower. Low-scoring subjects tended to prefer the baseline (although not significantly). Although the dataset is small, and we have only the opinion of one expert judge, we were encouraged by this result, which seems to indicate that our interface was able to be used effectively by high-achieving students.

### 8.2.4 User qualitative judgments

At the end of each session, subjects were asked for optional feedback on what they liked best and least about each test condition, using a free-form text interface. This qualitative evaluation material was quite interesting. Experimental subjects, being journalists or journalism students, were comfortable writing, and provided a rich set of comments confirming some of the original, informal hypotheses we had about the various interfaces. Once they reached this phase of the experiment, subjects were nearly done and, per-
haps because they were asked to write about each interface in order starting with Full Associations, then Combined Associations, and finally the Baseline, the greatest number of comments, and the longest, were for the Full Associations interface. Not surprisingly, the shortest comments were for the Baseline.

Table 8.6 shows a representative sample of these opinions. We were happy to note that, for the Full Associations interface, subjects mentioned the help they thought the interface gave them in their search. It was also satisfying to see a subject who thought that the interface might have suggested ideas she would not have come up with on her own. On the negative side, subjects mentioned that the discovered search terms might be too constraining in this interface, suggesting that an associative summary interface might be useful in combination with the baseline, rather than as a replacement.

For the Combined Associations interface, subjects commented favorably on its “simplicity”, presumably in contrast to the Full Associations interface. This could be a matter of individual user preference, and could be addressed by making either interface available to subjects. Among the things subjects disliked about the Combined Associations interface was that it was more time-consuming and “less aesthetically pleasing” than the Full Associations interface. Once again, personal preference must be taken into account, as some subjects found the same interface less confusing (also presumably as contrasted with the Full Associations interface).

Favorable comments on the Baseline interface centered around subjects’ ability to control the search and choose terms themselves. It could be that such subjects would find the clusters informative but would then prefer to initiate their own searches, using either discovered search terms or ones of their own choosing. What subjects disliked about the Baseline interface was something we hoped would be addressed by the other two interfaces, i.e., that even a shallow processing of a news archive might suggest lines of inquiry that subjects could easily miss with the Baseline.

8.3 Evaluating the technology

8.3.1 Comparing algorithms

This work has been informed by the claim that browsing is inherently an interactive process [Belkin et al., 1995]. Sparck Jones has noted that an important aspect of a summarization system is its ability to be used inter-
<table>
<thead>
<tr>
<th>Interface</th>
<th>Liked Most</th>
<th>Liked Least</th>
</tr>
</thead>
</table>
| **Full Associations**      | “Inclusion of names/places gave me entry points into the subject area I may not have known if no keywords were suggested.”  
 |                            | “It was good to see a list of articles that corresponded to keywords that I could search through instead of trying to think up key words without knowing if these would produce any results.”  
 |                            | “It brought up ideas I wouldn’t have thought about if i was using a less intelligent interface. The categories meant you could systematically think about background to do with people, and then different places etc.”  
 |                            | “Very accessible and easy to use. Gave me a very detailed search and provided me with what felt like a useful search. Logically set out.” | “May have precluded me from searching under other keywords, and still presented a lot of irrelevant results.”  
 |                            | “Perhaps the different aspects of a particular topic were too rigidly broken down with this search tool as despite the fact you could search a combination of topics, a broader search is sometimes required to provide the writer with the general background of what they are writing about before they focus on a particular angle. In this sense, this search tool could be used alongside the general Microsoft/Google type general search.”  
 |                            | “Too many categories, too much information to look through” |                                                                                                                                                                                                         |
| **Combined Associations**  | “there seemed a better spread of subjects and the layout was less confusing...easier for the eye to pan down quickly through all the key words.”  
 |                            | “Its simplicity, and its guidance, selecting terms which would provide a good return.”  
 |                            | “very easy to narrow down the topic and find the specific information I need efficiently”  
 | **Baseline**               | “I liked being able to develop my ideas logically by trying out keywords in the search engine. It allowed me to get the basic facts before I moved on to looking at more detailed articles through increasing the number of keywords I used.”  
 |                            | “I had the power to choose the words defining the search, so could find out instantly what would come up for what I felt were the most important keywords for the topic.”  
 | **Baseline**               | “Clearly more time consuming to go through results, some keywords that would have been suggested under other 2 interfaces I no doubt didn’t think of.”  
 |                            | “It is perhaps too broad and presents the researcher with a high volume of information they then have to painstakingly sift through until they find the angle they are looking for.”  
 |                            | “You are more restricted by the ideas you have yourself - the other interfaces are probably more likely to spark off a line of enquiry that you hadn’t thought of already.” |                                                                                                                                                                                                         |

Table 8.6: Example comments from subjects for each of the three interfaces.
Table 8.7: Subjects use of clusters, divided by technology used to discover them.

<table>
<thead>
<tr>
<th>Technology used</th>
<th>c4.5</th>
<th>Predictive Apriori +Clustering</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Full associations</td>
<td>156 (82%)</td>
<td>34 (18%)</td>
</tr>
<tr>
<td>Combined associations</td>
<td>102 (74%)</td>
<td>36 (26%)</td>
</tr>
</tbody>
</table>

actively by subjects [Sparck Jones, 1998]. It is, therefore critical that the technologies underlying any browsing/summarization system be relatively fast.

Hypothesis H9 claims that lightweight technology can be used to construct summaries in interactive systems. This hypothesis was confirmed in the restaurant domain where the use of decision tree induction to find associations was done in near real-time. However, computational complexity and speed are an issue with the associative summarization method using the Predictive Apriori-based algorithm employed in the news domain. If clusters derived using this method were preferred by subjects, the experience of interactive browsing would be significantly slowed in the news domain.

As we created each webpage for the three topics, we maintained a record of how each cluster was created, to gather data to test Hypothesis H9. We also kept a record of which clusters were used by subjects in the experiment. Table 8.7 shows the numbers and percentages for clusters used by subjects, broken down by technology used to derive them. 82% of the clusters used by subjects in the Full Associations interface were derived using c4.5. The corresponding number for the Combined Associations interface is 74%. Thus, the overwhelming number of clusters actually selected by subjects were derived using the lightweight method of decision tree induction. This confirms Hypothesis H9, that lightweight technology can be used to create summaries.

8.3.2 Comparing use of semantic types

In the restaurant domain, a user model was used for attribute selection for two types of responses, focussing on the set of attributes that were known to be of interest to a subject. In the news domain, the attributes are specific keywords and named entities, and it is unlikely that a user concerned with
exploring the contents of a news archive would have dispositional preferences associated with this type of attribute. However, these keywords and named entities have broader semantic types associated with them, and a user may have an interest in one of those categories. For example, in searching for an angle on a story centering around a person, a journalist may be interested in related persons.

In selecting the topic types to use in the experiment, we intentionally chose three that were different semantically, i.e., a person, an organization, and a country. Using these topics, we hoped to gather data to test Hypothesis H10, that the semantics of the topic of the search will influence the semantic type of the search terms used. If Hypothesis H10 were true, it would mean that an attribute selection process similar to that used for user modelling could be used in the news domain either automatically or through explicit user elicitation, to narrow the set of attributes used in response construction.

In addition to maintaining a record how each cluster was constructed, we also recorded the semantic entities associated with the clusters selected by subjects. This information was available for the Full Associations interface only, where the separation of clusters by semantic entity was maintained. Six users interacted with the Full Associations interface for each topic type. Table 8.8 shows the average number of clusters chosen by subjects by entity type across the three topics. Some of the trends were interesting, especially in light of the data we have from Google and Yahoo, showing the importance of named entities in general web searches.

Among clusters that contained only one semantic entity, those containing keywords only were the most popular across all three topics. Subjects used clusters containing locations only or persons only quite rarely, across all topics. Among the clusters that contained two entity types, the ones that combined keywords with one other entity were the most popular. For the most part, these differences were not significant across topics. Paired t-tests showed only one significant difference along this dimension, only partially confirming Hypothesis H10. The significant difference was found with the use of the keyword-only cluster in searching for angles for the Sarkozy story and for the China story. Keywords alone were used significantly more often with the story about Nicolas Sarkozy (p < .05, df = 5).

There were more significant differences among entity groupings within topics. The keyword-only cluster for Sarkozy stories was preferred signifi-
Table 8.8: Semantic entities used by subject, divided by story topic.

<table>
<thead>
<tr>
<th></th>
<th>Sarkozy</th>
<th>Microsoft</th>
<th>China</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keywords only</td>
<td>42</td>
<td>9</td>
<td>11</td>
</tr>
<tr>
<td>Locations only</td>
<td>1</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>Persons only</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Keywords + locations</td>
<td>5</td>
<td>25</td>
<td>4</td>
</tr>
<tr>
<td>Keywords + persons</td>
<td>27</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Locations + persons</td>
<td>8</td>
<td>3</td>
<td>8</td>
</tr>
<tr>
<td>Keywords + locations + persons</td>
<td>7</td>
<td>12</td>
<td>14</td>
</tr>
</tbody>
</table>

cantly over every other type of entity grouping \((p < .05, df = 5)\) in that topic. There were no significant differences among entity groupings for the Microsoft stories. For the China stories, the clusters comprised of keywords+locations+persons were the most popular, but this preference was only significant in relation to one other cluster type, that with persons alone \((p < .05, df = 5)\).

The dataset is quite limited for this sort of analysis, and it is not clear if the few correlations found between the topic of the breaking news story and the entity types used in searches would hold for different topics. It is also unclear how those preferences could be determined reliably \textit{a priori}. It is more likely that preferences for particular semantic entities would emerge as users became more familiar with the contents of the domain for a particular topic. Subjects searching for angles about Nicolas Sarkozy showed the most skewed preference for one entity grouping, and this could be due to the fact that the name the main actor/person in the breaking news story, Nicolas Sarkozy, was used as a search term to gather the resulting documents, and he was most closely associated with one particular location (i.e., France). The most important person and location could, therefore, have been considered implicit in the clusters using keywords. Locations, at least in conjunction with keywords, were considered relatively important by subjects when looking for angles on the China breaking news story, however, and here a location (i.e., “China”) was used as a search term for all resulting documents.

In a truly interactive system, subjects might be given the option of choosing entities and entity groupings they wish to see, and the interface could limit itself to computing associations on just those subsets. This sort of interface could also be less confusing to subjects who might be overwhelmed
8.4 Conclusion

In this chapter, we tested the hypotheses that subjects would find associative summaries of sets of new articles useful for the task of discovering background information for a breaking news story. We also assessed the quality of the information found by our subjects. Post-graduate journalism students and professional journalists were given a real breaking news story and used each of three different interfaces to explore a news archive. The purpose of this exploration was to find “angles” that could be used in writing a longer piece about the information contained in the breaking news story. Two of the interfaces enabled searching based on automatically derived clusters of keywords and named entities, also discovered automatically. A baseline interface enabled user-initiated searches to the archive.

Overall, users preferred some form of associative summarization in 67% of the sessions. The baseline interface (Baseline) and an interface that displayed associative clusters grouped by the semantic entities they contained (Full Associations) were preferred equally. Users were also equally confident, using all interfaces, that they were able to fully explore the news archive. The order in which the interfaces were presented to subjects varied, but the overall ranking subjects assigned to each interface rose across the three interfaces, regardless of order. This suggested that there was an effect of learning.

Information “packets”, comprised of a short statement of the angle discovered and supporting documents from the news archive, were examined by a professor of journalism and rated for usefulness and originality. The Baseline and the Full Associations interfaces scored equivalently for these measures. Rated for richness of content, stories discovered using the baseline were judged significantly higher than those discovered using both interfaces showing associative summaries.

Subjects whose work was most highly rated by the expert judge preferred the Full Associations interface by a significant amount over the baseline. Subjects scored lower by the expert judge preferred the baseline, although not significantly. This could indicate individual differences based on subject ability.

Finally, we examined the associative summaries in terms of the technology used to derive them. The more lightweight method of deriving associ-
ations was used in, on average, 78% of the associations chosen by subjects as search terms. Associative summaries using keywords were used most commonly by subjects, along with associative summaries that combined keywords with one other type of named entity (i.e., location or person). In general, however, few significant differences were discovered that related to a possible general preference for associative summaries of a particular semantic entity by topic.
Chapter 9

Conclusion and future work

9.1 Hypotheses revisited

In this thesis, we have examined algorithms for content selection and information presentation that enable browsing as an information-seeking strategy. It was our hypothesis that users who wish to explore the contents of a particular domain might not be satisfied with traditional spoken dialogue systems, in which focal information is typically not described until a small number of the records comprising the domain are identified through a constraint-elicitation process. Specifically, with \textbf{H1}, our primary hypothesis, we examined the claim that users prefer responses in the form of incremental summaries of the focal information.

\textbf{H1} was tested in various experiments that elicited user judgements on response types whose underlying parameters were designed to explore issues in content selection and information presentation. The first of these experiments compared incremental summaries with the type of data-elicitation type responses traditionally used in spoken dialogue systems. This experiment was carried out in the context of a London restaurant domain and confirmed \textbf{H1}. It disconfirmed \textbf{H1a}, that users prefer incremental summaries in dialogue systems when they are unfamiliar with the domain, however. The evidence from this experiment indicated that users prefer incremental summaries under all circumstances.

Hypotheses \textbf{H3a} and \textbf{H3b} concern the type of information users prefer, given their knowledge of the domain and the system’s knowledge of their preferences. \textbf{H3a} states that the optimal summary type for users unfamiliar with the domain will describe attributes that generalize across the focal information. \textbf{H3b} predicts that users who have had the chance to express preferences about attributes in a domain will prefer summary types that are
tailored to those preferences. Neither hypotheses was confirmed, although neither was disproved. Rather, we discovered that responses describing domain attributes that are chosen for their ability to generalize across the focal information are preferred equally to responses chosen to conform to a set of pre-specified preferences. This shows that learning about focal information in unanticipated ways is something users value, as well as hearing about attributes that they have expressed a particular interest in.

To test H2, that the process of content selection and information presentation can be automated, we developed a set of algorithms to determine underlying parameters associated with these aspects of response construction. The parameters determined attribute ranking/selection, as well as how particular values for attributes were chosen and presented. For attribute selection/ranking, we tested a range of algorithms from the literature, settling on two. One of these was developed for this work as an extension of a stepwise refinement algorithm used elsewhere in spoken dialogue systems. The other made use of Multi-Attribute Utility Theory in combination with an elicited model of user preferences. For choosing attribute values, we maximized for generalizability and/or user preference. To highlight relationships between attribute/value combinations, we employed decision tree induction, as well as a simpler algorithm that listed values for attributes by frequency. User experiments were conducted to test the hypotheses associated with the efficacy of these automated algorithms.

The two sets of hypotheses H4a through H5b were tested in an experiment in the London restaurant domain that specifically contrasted two parameters for information presentation and two for content selection. To test Hypotheses H4a and H4b, we examined ways of presenting information to users. H4a claims that the optimal summary type will describe attributes singly, while H4b claims that the optimal summary type will describe attributes in association with others. We found evidence confirming both hypotheses. When attributes are selected for their ability to generalize, users prefer to hear them described individually. When attributes are chosen based on individual preferences, users ranked most highly responses that describe associations among those attributes.

Hypotheses H5a and H5b were formulated to test assumptions about how the amount of focal information would affect user preferences. H5a claims that large sets of focal information will best be described by attributes chosen to generalize across the data. When the set of focal information is
small, \textbf{H5b} claims that the best set of attributes for describing it will conform to user preferences. Surprisingly, we discovered that subjects preferred to hear about highly preferred attributes when the focal information encompassed a large number of restaurants. When focal information was relatively small, users preferred to hear about attributes chosen to generalize well over that information. Both hypotheses were disconfirmed, and the evidence suggested that an explanation could be found in testing \textbf{H6}, that the specific values of the attributes were not as important as their ability to generalize. We found that responses that mentioned poorly rated attributes were disliked by users, even when those attribute/value combinations described a large part of the focal information.

These results indicated that the parameters associated with the optimal summary type vary in context. Using data derived from the experiments with human subjects, we tested \textbf{H7}, that a classifier could predict which summary type to use in a given dialogue context. Deriving features from both the query in context and the query’s utility, as measured within the Multi-Attribute Utility Theory model, we were able to increase classifier performance significantly above the baseline. The dataset was small, but the results were encouraging.

In porting the technology to a new domain, we tested the final three hypotheses. To do so, we applied the algorithms used in the restaurant domain to a corpus of news articles that had been automatically annotated for keyword and named entity. The representation of the records in the domain was necessarily different for this unstructured data. Each news article was represented as a binary vector whose attributes were the discovered keywords and named entities. In addition, an alternative algorithm for finding associations was implemented and tested. This alternative was more computationally costly than the one used for the restaurant domain and a comparison of the two enabled us to test \textbf{H9}, that real-time technology could be used to create summaries.

Experiments using associative summaries in this domain were task-based and used post-graduate journalism students and/or professional journalists. Subjects used the technology to search a news archive for background information to support a real breaking news story. The interfaces to the news archive included two that displayed discovered associative summaries of key terms that could be used to search for specific articles, as well as one baseline, in which subjects interacted with a Google-like interface to find articles.
using key terms of their own choosing. Experimental results confirmed H8, showing the effectiveness of associative summaries in a task-based scenario using the news corpus. H9 was also confirmed, in that the overwhelming majority of associations used by experimental subjects to complete the experimental task were derived using lightweight technology. The use of the news corpus also allowed us to test H10, which was disconfirmed. The topic of the search in the news domain did not influence the semantic type of the search terms used by subjects.

9.2 Conclusion

These results show that content selection and information presentation in interactive systems can be tailored to the information-seeking goals and strategies of browsing users. The algorithms described here are lightweight and portable, making them suitable to interactive systems. Grounding the experimentation in task-oriented scenarios indicate that the results are applicable to real world situations.

It is our belief that the need to browse information can only grow, as more information resources become available on the Web and the technology for annotating these resources matures. It is also our belief that dialogue systems can address this need through the careful processing of domain data. Traditionally, a “data-driven” dialogue system was one whose parameters were learned from annotated human-human or simulated human-machine interactions. These parameters were typically associated with constraint-elicitation order and a strategy for confirming user input. In expanding the notion of what it means for a dialogue system to be data-driven, we have examined ways of using domain data to automatically configure content-rich, summary-type responses to user queries and to present information in a structured way. We feel these are interesting new directions for dialogue systems and ones that make particular sense from the point of view of human-machine interaction. The analysis of large amounts of data and the discovery of associations and correlations within datasets are capabilities made possible by advances in information extraction and machine learning. Interactive systems that exploit these capabilities enable a principled and portable treatment of content selection and information presentation to meet the varying needs of users.
9.3 Future work

Our results point encouragingly toward more work that can profitably be done in this area. The first, and obvious area of future work is in developing and deploying on-line systems using the technology described here. Our research shows that dispositional preferences cannot always be relied on to determine attribute selection in all contexts. Furthermore, these dispositional preferences may not apply in all domains. With lightweight technology, it is possible to design systems that allow users to interactively choose particular attributes and values to examine in more detail through associative clustering. Users with strong dispositional preferences may wish to explore trade-offs among particular preferences in more detail using these techniques.

The results from the second experiment in the restaurant domain suggest that choosing attributes for their ability to generalize may ignore user attitudes towards the values of those attributes, regardless of their personal preferences. Specifically, our experiment indicated that users are not interested in hearing about poorly rated attributes even when those attributes describe a large proportion of the focal information. In future work, it would be interesting to see if this finding is an artifact of the way the information is presented, i.e., if hearing about poorly rated attributes is somehow more salient when natural language is used to describe the information. A graphical display may be convey the same information in a more neutral way, and may be more acceptable to users. Alternately, user model scoring might be used after Refiner attribute selection, to find particular attribute/value pairs that are generalizable across the set of records in focus and of interest to users.

In general, the use of multiple output modalities is an obvious extension of this work. We examined the use of natural language and simple tabular summaries, but the information presented to users could also be overlaid onto maps, graphs, or other types of tables. The analysis of data that led to some of the natural language summaries highlighted information that was assumed to be important to particular users. A prosodic analyzer could make use of this information to configure a more natural spoken version of these responses.

Decision tree induction has also been used as a way of performing query relaxation without recourse to hand-crafted ontologies [Muslea, 2004]. An implementation of this in the London restaurant domain was promising, but
ultimately outside the scope of the thesis. Pursuing this further would add another dimension to the automation of cooperative answers in interactive systems.

This work implemented models of “interestingness” for content selection that maximized metrics of coverage, generalizability, or user preference. For a particular class of user, however, rarity is a valuable characteristic. A contrasting measure of interestingness would target outliers, e.g., rules that apply infrequently, attributes that occur rarely, or combinations of attribute/value pairs that are vastly different in utility. The algorithms described here would apply to this type of content selection; the scoring mechanism would simply have to be modified.

Combining information extraction and opinion mining, domains could be populated with both subjective and objective data that could also be explored for the purposes of summarization. On-line shopping sites such as amazon.com already provide brief, Refiner-like summaries of various products (e.g., the number of digital cameras as a function of manufacturer). Combined with opinion mining and more sophisticated natural language generation, summarization could take the form of automated consumer reports, pointing out correlations and trade-offs in attributes with objective and subjective values.
Appendix

A.1 Personas used in initial restaurant experiment
Londoner

This time, you’re someone who knows London fairly well, but you can still be surprised at times. Your knowledge of London restaurants is not encyclopedic and you are open to trying new things. You usually know what you’re looking for in a restaurant, and have no problem expressing those preferences. However, you’ve not registered your preferences with the system.

Generic tourist

You’re a tourist in London. You don’t know the city well, but you like to explore. You don’t have any particularly strong preferences when it comes to restaurants.

Tourist with user model

This time through, you’re a tourist who has had a chance to register your preferences when looking for a restaurant (the most important at the top and the least important at the bottom).

- Price
- Food quality
- Location/neighborhood
- Cuisine type
- Service quality
- Decor

You don’t know London well, so you’d like to stay within central London. The restaurant database we’re using considers the following sections of London to be central:

- Chelsea
- Chinatown
- Covent Garden
- Hampstead
- Knightsbridge
- Mayfair
• Notting Hill Gate
• Saint James
• Soho

Of the cuisines the system knows, you’ve indicated that you like the following:

• British
• Chinese
• Eclectic
• European
• Italian
• Spanish
• Thai
A.2 Dialogues used in initial restaurant experiment
**First utterance used to introduce all dialogues**

<table>
<thead>
<tr>
<th><strong>User</strong></th>
<th>I want a restaurant in London.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Londoner:</strong></td>
<td>What neighborhood are you interested in?</td>
</tr>
<tr>
<td><strong>Generic Tourist:</strong></td>
<td>I know of 594 restaurants in London. There are 49 medium-priced Italian restaurants in London and 40 medium-priced European restaurants. There are also 36 medium-priced British restaurants.</td>
</tr>
<tr>
<td><strong>UM Tourist:</strong></td>
<td>I know of 594 restaurants in London. There are 6 expensive restaurants with very good food quality and 1 very inexpensive restaurant with medium food quality. There are also 30 expensive restaurants with good food quality.</td>
</tr>
<tr>
<td>Dialogue 1</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td></td>
</tr>
<tr>
<td><strong>User:</strong></td>
<td>How about Soho?</td>
</tr>
<tr>
<td><strong>Londoner:</strong></td>
<td>What price range are you interested in?</td>
</tr>
<tr>
<td><strong>Generic Tourist:</strong></td>
<td>I know of 36 restaurants in Soho. There are 5 inexpensive ones near the Leicester Square tube station and 3 near the Piccadilly Circus tube station. There are also 2 medium-priced ones near the Oxford Circus tube station.</td>
</tr>
<tr>
<td><strong>UM Tourist:</strong></td>
<td>I know of 36 restaurants in Soho. There are 8 inexpensive restaurants but with poor food quality and 1 expensive one with very good food quality. There are also 8 medium-priced restaurants with medium food quality.</td>
</tr>
<tr>
<td><strong>User:</strong></td>
<td>I'd like someplace cheap.</td>
</tr>
<tr>
<td><strong>Londoner:</strong></td>
<td>What cuisine are you interested in?</td>
</tr>
<tr>
<td><strong>Generic Tourist:</strong></td>
<td>I know of 21 inexpensive restaurants in Soho. There are 2 near the Leicester Square tube station with medium food quality. There are also 3 near the Leicester Square tube station but with poor food quality. There are also 2 near the Piccadilly Circus tube station with medium food quality.</td>
</tr>
<tr>
<td><strong>UM Tourist:</strong></td>
<td>I know of 21 inexpensive restaurants in Soho. There are 2 Thai restaurants with medium food quality and 1 Japanese restaurant with medium food quality. There is also 1 Chinese restaurant with medium food quality.</td>
</tr>
<tr>
<td><strong>User:</strong></td>
<td>Chinese.</td>
</tr>
<tr>
<td><strong>All three responses (not rated):</strong></td>
<td>There is only 1 inexpensive Chinese restaurant in Soho.</td>
</tr>
<tr>
<td><strong>(not rated):</strong></td>
<td>Joy King Lau is located at 3 Leicester Street in Soho, near the Leicester Square tube stop. It serves inexpensive Chinese cuisine and dim sum.</td>
</tr>
</tbody>
</table>
### Dialogue 2

<table>
<thead>
<tr>
<th>User:</th>
<th>I'd like something cheap.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner:</td>
<td>What neighborhood are you interested in?</td>
</tr>
<tr>
<td>Generic Tourist:</td>
<td>I know of 187 inexpensive restaurants in London. There are 8 inexpensive Chinese restaurants in Chinatown and 5 inexpensive Italian restaurants in Chelsea. There are also 3 inexpensive European restaurants in Notting Hill Gate.</td>
</tr>
<tr>
<td>UM Tourist:</td>
<td>I know of 187 inexpensive restaurants in London. There are 2 with very good food quality in Hampstead and 7 with medium food quality in Soho. There are also 4 with medium food quality in Hampstead.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User:</th>
<th>What neighborhood are you interested in?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner:</td>
<td>I'm interested in Chelsea.</td>
</tr>
<tr>
<td>Generic Tourist:</td>
<td>I know of 23 inexpensive restaurants in Chelsea. There are 2 inexpensive Italian restaurants with medium food quality and 2 with poor food quality. There is also 1 inexpensive restaurant serving eclectic cuisine but with poor food quality.</td>
</tr>
<tr>
<td>UM Tourist:</td>
<td>I know of 23 inexpensive restaurants in Chelsea. There are 2 inexpensive Italian restaurants with medium food quality and 1 inexpensive Spanish restaurant but with poor food quality. There is also 1 inexpensive British restaurant but with poor food quality.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User:</th>
<th>I'll take Italian.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner:</td>
<td>What food quality are you interested in?</td>
</tr>
<tr>
<td>Generic Tourist:</td>
<td>I know of 7 inexpensive Italian restaurants in Chelsea. There are 2 near the South Kensington tube station serving contemporary Italian cuisine and 2 near the Sloane Square tube station serving traditional Italian cuisine.</td>
</tr>
<tr>
<td>UM Tourist:</td>
<td>I know of 7 inexpensive Italian restaurants in Chelsea. There are 2 with medium food quality but poor service and 2 with poor food quality and poor service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User:</th>
<th>I guess contemporary Italian cuisine.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All three responses (not rated):</td>
<td>There are 5 restaurants in Chelsea that serve inexpensive contemporary Italian cuisine. Calzone at 335 Fulham Road, Carpaccio at 4 Sydney Street, Friends at 6 Hollywood Road, and Made in Italia at 249 King’s Road are all near the South Kensington tube station. Pucci Pizza at 205 King’s Road is near the Sloane Square tube station.</td>
</tr>
</tbody>
</table>

### Dialogue 3

<table>
<thead>
<tr>
<th>User:</th>
<th>I want an expensive restaurant with good food quality.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Londoner:</td>
<td>What neighborhood are you interested in?</td>
</tr>
<tr>
<td>Generic Tourist:</td>
<td>I know of 30 expensive restaurants with good food quality. There are 4 French restaurants with good service quality and 3 Italian restaurants with good service quality. There are also 2 Japanese restaurants with good service quality.</td>
</tr>
<tr>
<td>UM Tourist:</td>
<td>I know of 30 expensive restaurants with good food quality in London. There is a French restaurant in Knightsbridge and 2 Chinese restaurants in Mayfair. There is also a British restaurant in Soho.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User:</th>
<th>I’d like to be in Soho.</th>
</tr>
</thead>
<tbody>
<tr>
<td>All three responses (not rated):</td>
<td>There are 2 expensive restaurants in Soho with good food quality. L’Escargot at 48 Greek Street near the Leicester Square tube Station, and Sugar Club at 21 Warwick Street near the Piccadilly Circus tube station.</td>
</tr>
</tbody>
</table>
**Dialogue 4**

| User: | I want to be in central London. |
| Londoner: | What neighborhood are you interested in? |
| **Generic Tourist:** | I know of 240 restaurants in central London. There are 24 expensive restaurants in Mayfair and 18 inexpensive restaurants in Soho. There are also 18 medium-priced restaurants in Covent Garden. |
| **UM Tourist:** | I know of 240 restaurants in central London. There is 1 extremely inexpensive restaurant with medium food quality and 30 inexpensive restaurants but with poor food quality. There are also 50 medium-priced restaurants with medium food quality. |

| User: | How about British food? |
| Londoner: | What neighborhood are you interested in? |
| **Generic Tourist:** | I know of 31 British restaurants in central London. There are 4 serving modern British cuisine in Covent Garden and 4 serving traditional British cuisine in St. James. There are also 4 serving traditional British cuisine in Covent Garden. |
| **UM Tourist:** | I know of 31 British restaurants in central London. There are 7 inexpensive British restaurants but with poor food quality and 3 expensive British restaurants with good food quality. There are also 6 medium-priced restaurants with poor food quality. |

| User: | I’d like good food quality. |
| **All three responses:** | There are 3 restaurants in central London serving highly rated British food. The Grill Room at 53 Park Lane in Mayfair, Rib Room at 2 Cadogan Place in Belgravia, and Wilton’s at 55 Jermyn Street in Saint James’s. |

| **(not rated):** | |
A.3 Example dialogues used in the second restaurant experiment
User model information used to construct some of the responses below

Attributes, ranked by preference:

- Food quality
- Cuisine type (Specific choices: Asian, Barbecue, Brazilian, Greek, Italian, Japanese, Mediterranean, Mexican, Middle Eastern, Moroccan, North African, Persian, South American, Spanish, Sudanese, Thai, Turkish, Vietnamese)
- Price
- Decor
- Location/neighborhood (Specific choices: Camden Town/Chalk Farm/Kentish Town, Golders Green/Finchley, Hampstead/Kilburn, Highgate/Muswell Hill, Islington, Saint John’s Wood, Stoke Newington, Barnes, Battersea, Brixton/Clapham, Chelsea, Earl’s Court, Fulham, Kensington, Pimlico, Putney/Richmond, South Kensington, Wandsworth/Balham, Bayswater, Chiswick, Hammersmith, Notting Hill Gate, Olympia, Shepherd’s Bush, Belgravia, Bloomsbury, Chinatown, Covent Garden, Knightsbridge, Marylebone, Mayfair, Piccadilly, Saint James’s, Soho, Victoria, Westminster, Camden Town/Chalk Farm/Kentish Town, Canary Wharf/Docklands, Chelsea, Chinatown, Earl’s Court, Fulham, Hammersmith, Islington, Kensington, Knightsbridge, Marylebone, Mayfair, Notting Hill Gate, Putney/Richmond, Saint James’s, Soho, South Kensington, Tower Bridge, Westminster)
- service quality
### Task 1

**User:** What do you know about restaurants in London?

**Refiner response, singleton attributes:**

I know of 594 restaurants in London. All price ranges are represented. Many serve Italian, British, European, and French cuisine. Some of the neighborhoods represented are Chelsea, Mayfair, and Soho.

**Refiner response, associations:**

I know of 594 restaurants in London. There are 49 medium-priced Italian restaurants in London and 40 medium-priced European restaurants. There are also 36 medium-priced British restaurants.

**User model response, singleton attributes:**

I know of 594 restaurants in London. There are 70 restaurants with good to very good food quality. There are 105 Italian restaurants and 188 inexpensive to very inexpensive restaurants.

**User model response, associations:**

I know of 594 restaurants in London. There are 6 French restaurants with very good food quality and 9 Japanese restaurants with good food quality. There are also 53 Italian restaurants with medium food quality.
**Task 2**

**User:** What can you tell me about restaurants in Soho?

**Refiner response, singleton attributes:**

I know of 36 restaurants in Soho. Most are inexpensive or medium-priced. Many are near the Leicester Square and Picadilly Circus tube stations. Many serve Italian or European cuisine.

**Refiner response, associations:**

I know of 36 restaurants in Soho. There are 5 inexpensive ones near the Leicester Square tube station and 3 near the Piccadilly Circus tube station. There are also 2 medium-priced ones near the Oxford Circus tube station.

**User model response, singleton attributes:**

I know of 36 restaurants in Soho. There are 4 with good to very good food quality. There are 6 Italian restaurants and 18 inexpensive restaurants.

**User model response, associations:**

I know of 36 restaurants in Soho. There is one Vietnamese restaurant with medium food quality and one Japanese restaurant with medium quality. There are also 4 Italian restaurants with medium food quality.
Task 3

User: I’m interested in an Italian restaurant.

Refiner response, singleton attributes:
I know of 105 Italian restaurants in London. All price ranges are represented. Most serve contemporary or traditional Italian cuisine. Some of the neighborhoods represented are Chelsea, Knightsbridge, and Soho.

Refiner response, associations:
I know of 105 Italian restaurants in London. There are 49 medium-priced ones. 61 serve contemporary Italian cuisine.

User model response, singleton attributes:
I know of 105 Italian restaurants in London. There are 8 with good food quality and 21 inexpensive ones. There is 1 with good decor.

User model response, associations:
I know of 105 Italian restaurants in London. There are 5 expensive Italian restaurants with good food quality and 33 medium-priced ones with medium food quality. There are also 9 Italian restaurants that are inexpensive but with poor service quality.
<table>
<thead>
<tr>
<th>Task 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User:</strong> I’d like to find a cheap restaurant.</td>
</tr>
<tr>
<td><strong>Refiner response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 187 inexpensive restaurants in London. Some of the cuisines represented are British, Italian, and Chinese. Many have poor service quality. Some of the neighborhoods represented are Soho, Chelsea, and Notting Hill Gate.</td>
</tr>
<tr>
<td><strong>Refiner response, associations:</strong></td>
</tr>
<tr>
<td>I know of 187 inexpensive restaurants in London. There are 106 with poorly rated service quality. There are 16 restaurants serving British cuisine with poor service quality and 12 Italian restaurants with poor food quality.</td>
</tr>
<tr>
<td><strong>User model response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 187 inexpensive restaurants in London. There are 6 with good food quality. There are 21 inexpensive Italian restaurants. There are also 2 inexpensive restaurants with good decor.</td>
</tr>
<tr>
<td><strong>User model response, associations:</strong></td>
</tr>
<tr>
<td>I know of 187 inexpensive restaurants in London. There is one Persian restaurant with good food quality and one Italian restaurant with good food quality. There is also 1 Sudanese restaurant with medium food quality.</td>
</tr>
<tr>
<td>Task 5</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td><strong>User:</strong> What do you know about restaurants in Chelsea?</td>
</tr>
<tr>
<td><strong>Refiner response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 47 restaurants in Chelsea. All price ranges are represented. Many are near the Sloane Square and South Kensington tube stations. Many serve Italian, French, and European cuisines.</td>
</tr>
<tr>
<td><strong>Refiner response, associations:</strong></td>
</tr>
<tr>
<td>I know of 47 restaurants in Chelsea. There are 7 medium-priced ones near the Sloane Square tube station and 5 near the South Kensington tube station. There are also 5 inexpensive ones near the South Kensington tube station.</td>
</tr>
<tr>
<td><strong>User model response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 47 restaurants in Chelsea. There are 4 with good food quality. There are 15 Italian restaurants and 15 inexpensive restaurants.</td>
</tr>
<tr>
<td><strong>User model response, associations:</strong></td>
</tr>
<tr>
<td>I know of 47 restaurants in Chelsea. There is one French restaurant with very good food quality and ten Italian restaurants with medium food quality. There are also 2 Thai restaurants with medium food quality.</td>
</tr>
<tr>
<td>Task 6</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td><strong>User:</strong> I’m interested in Indian food.</td>
</tr>
<tr>
<td><strong>Refiner response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 35 Indian restaurants in London. All price ranges are represented. Some of the neighborhoods represented are Mayfair, Soho, and Chelsea. Some of the tube stations represented are Green Park, South Kensington, and Piccadilly Circus.</td>
</tr>
<tr>
<td><strong>Refiner response, associations:</strong></td>
</tr>
<tr>
<td>I know of 35 Indian restaurants in London. There are 3 medium-priced ones in Mayfair and 3 inexpensive ones in Soho. There are also 2 expensive ones in Chelsea.</td>
</tr>
<tr>
<td><strong>User model response, singleton attributes:</strong></td>
</tr>
<tr>
<td>I know of 35 Indian restaurants in London. There are 6 with good food quality and 12 inexpensive ones. There are also 2 with good to very good decor.</td>
</tr>
<tr>
<td><strong>User model response, associations:</strong></td>
</tr>
<tr>
<td>I know of 35 Indian restaurants in London. There are 4 medium-priced Indian restaurants with good food quality and 10 medium-priced ones with medium food quality. There are also 4 Indian restaurants that are inexpensive but with poor food quality.</td>
</tr>
<tr>
<td>Task 7</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td><strong>User:</strong> What about expensive restaurants?</td>
</tr>
</tbody>
</table>
| **Refiner response, singleton attributes:**  
I know of 163 expensive restaurants in London. Many serve Italian, French or European cuisine. Many are in Mayfair, Chelsea, and Knightsbridge. Some of the tube stations represented are Green Park, Knightsbridge, and South Kensington. |
| **Refiner response, associations:**  
I know of 163 expensive restaurants in London. There are 5 French restaurants in Mayfair and 5 Italian restaurants in Belgravia. There are also 4 British restaurants in Covent Garden. |
| **User model response, singleton attributes:**  
I know of 163 expensive restaurants in London. There are 45 with good to very good food quality. There are 35 Italian restaurants. There are also 37 with good to very good decor. |
| **User model response, associations:**  
I know of 163 expensive restaurants in London. There are 4 expensive Japanese restaurants with good food quality and 3 expensive French restaurants with very good food quality. There are also 14 expensive Italian restaurants with medium food quality. |
<table>
<thead>
<tr>
<th>Task 8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>User:</strong></td>
</tr>
<tr>
<td><strong>Refiner response, singleton attributes:</strong></td>
</tr>
<tr>
<td><strong>Refiner response, associations:</strong></td>
</tr>
<tr>
<td><strong>User model response, singleton attributes:</strong></td>
</tr>
<tr>
<td><strong>User model response, associations:</strong></td>
</tr>
</tbody>
</table>
A.4 Natural language experiment in the news domain
Summary responses were constructed for each of the following simulated user queries:

1. I’m interested in stories about earthquakes.
2. I’m interested in stories about earthquakes and Pakistan.
3. I’m interested in stories about earthquakes and Indonesia.
4. I’m interested in stories about Microsoft.
5. I’m interested in stories about Microsoft and China.

Queries that introduced more than one constraint (queries 2, 3, and 5 above) were designed as follow-ons to a preceding query. For queries 2 and 3, the added constraint (i.e., “Pakistan” and “Indonesia”, respectively) was one that had been mentioned in responses to query 1; for query 5, the added constraint (i.e., “China”) had been mentioned in responses to query 4.

Two types of summaries were created for each query, associations and singleton values, across seven possible attribute combinations (e.g., “Keyword”, “Keyword + Location”, etc; see Section 7.4.1 for the complete set). This resulted in a total of fourteen possible responses, framed in natural language similar to that used in the experiment described in Chapter 5. Table A.1 shows an example of responses constructed using each semantic type individually; Table A.2 shows an example of responses constructed using pairs and triples of the semantic types. Each table shows both the singleton and associative summaries.

Twelve subjects were recruited from among post-graduate students in Computer Science or computer professionals. Subjects saw each pair of associative/singleton summaries on separate web pages, and were asked to choose one in response to the query “Which response is better?” The order of each pair of responses, singleton values vs. associations, was varied randomly for each pair across a particular users’ set of pages.

After they had looked at each set of pairwise responses, subjects were asked to look at the entire set of 14 responses on one page. Here we asked them to read each response again and rank each by how helpful they would find it for locating an article quickly in a news archive. Rankings were elicited on a Likert scale from 1 to 7, with 1 being “Very unhelpful” and 7 being “Very helpful”.


I’m interested in stories about earthquakes.

<table>
<thead>
<tr>
<th>Responses using KeyPhrases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singleton: I’ve processed 1000 news articles about earthquakes. 61 of the stories spoke of tsunamis and 51 mentioned people. 41 of the stories mentioned volcanoes.</td>
</tr>
<tr>
<td>Associative: I’ve processed 1000 news articles about earthquakes. 3 stories mention aftershocks and magnitudes and villages and 3 mention aftershocks and buildings. 1 story mentions magnitude and tsunami and damages.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responses using Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singleton: I’ve processed 1000 news articles about earthquakes. 177 of the stories mentioned Pakistan and 155 mentioned Indonesia. 115 of the stories mentioned the United States.</td>
</tr>
<tr>
<td>Associative: I’ve processed 1000 news articles about earthquakes. 72 of the stories mentioned Islamabad and Pakistan and 50 of the stories mentioned Java and Indonesia. 32 of the stories mentioned Tehran and Iran.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responses using Persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singleton: I’ve processed 1000 news articles about earthquakes. 73 of the stories mentioned US President George Bush and 26 mentioned Pakistani President Pervez Musharraf. 17 stories mentioned Iranian regional head of emergency services Ali Barani.</td>
</tr>
<tr>
<td>Associative: I’ve processed 1000 news articles about earthquakes. 11 of the stories mention US President George Bush and Pakistani President Pervez Musharraf and 3 stories mention George Bush and al-Qaeda No. 2 Ayman Al-Zawahri. 3 stories mention Pervez Musharraf and Pakistani Prime Minister Shaukat Aziz.</td>
</tr>
</tbody>
</table>

Table A.1: Example natural language responses using single semantic types.
<table>
<thead>
<tr>
<th>Responses using KeyPhrases + Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 7: I've processed 1000 news articles about earthquakes. 61 of the stories spoke of tsunamis and 51 mentioned people. 41 of the stories mentioned volcanoes. 177 of the stories mentioned Pakistan and 155 mentioned Indonesia. 115 of the stories mentioned the United States.</td>
</tr>
<tr>
<td>S 8: I've processed 1000 news articles about earthquakes. 26 of the stories mentioned a tsunami and Indonesia and 11 stories mentioned volcanoes and Indonesia. 6 stories mentioning villages and Pakistan.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responses using KeyPhrases + Persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 9: I've processed 1000 news articles about earthquakes. 5 stories mention Iranian regional head of emergency services Ali Barani and tsunamis and 5 stories mention US President George Bush and tsunamis. 3 stories mention magnitude and villages and aftershocks.</td>
</tr>
<tr>
<td>S 10: I've processed 1000 news articles about earthquakes. 61 of the stories spoke of tsunamis and 51 mentioned people. 41 of the stories mentioned volcanoes. 73 of the stories mentioned US President George Bush and 26 mentioned Pakistani President Pervez Musharraf. 23 stories mentioned Iranian regional head of emergency services Ali Barani.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responses using Locations + Persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 12: I've processed 1000 news articles about earthquakes. 199 of the stories mentioned the US and 177 mentioned Pakistan. 155 stories mentioned Indonesia. 73 stories mentioned US President George Bush, 26 mentioned Pakistani President Pervez Musharraf, and 17 mentioned Iranian regional head of emergency services Ali Barani.</td>
</tr>
<tr>
<td>S 11: I've processed 1000 news articles about earthquakes. 74 of the stories mentioned the US and al-Qaeda No. 2 Ayman Al-Zawahri and 23 mentioned Indonesia and Oleg Kotosanov. 16 stories mention Pakistan and General Peter Pace.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Responses using KeyPhrases + Locations + Persons</th>
</tr>
</thead>
<tbody>
<tr>
<td>S 14: I've processed 1000 news articles about earthquakes. 61 of the stories mentioned tsunamis and 51 mentioned people. 41 of the stories mentioned volcanoes. 177 of the stories mentioned Pakistan and 155 mentioned Indonesia. 115 of the stories mentioned the United States. 73 of the stories mentioned US President George Bush and 26 mentioned Pakistani President Pervez Musharraf. 23 stories mentioned Iranian regional head of emergency services Ali Barani.</td>
</tr>
<tr>
<td>S 13: I've processed 1000 news articles about earthquakes. 3 of the stories mentioned Indonesia and Oleg Kotosanov and tsunamis. 7 of the stories mention Pakistan and Oleg Kotosanov and tents. 6 of the stories mention Indonesia and al-Qaeda No. 2 Ayman Al-Zawahri and volcanoes.</td>
</tr>
</tbody>
</table>

Table A.2: Example natural language responses using combinations of semantic types.
A.5 Breaking news stories used in news archive experiment
NICOLAS SARKOZY STORY

VIOLENCE GREET SARKOZY’S VICTORY

Riot police fired tear gas into a crowd gathered at the Place de la Bastille following conservative Nicolas Sarkozy’s presidential election victory tonight.

French television described the crowd as "militant anarchists" apparently upset by the victory of a man of the right. Police used tear gas to try to disperse the gathering.

The Paris square often is a starting point for strikes and serves as a hub for popular demonstrations.

Security was reinforced on Election Day with police visibly patrolling the streets of the French capital and maintaining a discreet presence in the suburbs where many housing project are located.

MICROSOFT STORY

NEW FINES THREAT ESCALATES TRANS-ATLANTIC TRADE FIGHT

The European Union today escalated its trans-Atlantic fight with Microsoft today, threatening new multi-million fines against the software maker over claims it fails to offer rivals a fair deal on licenses for helping servers work with Windows.

In response, Microsoft charged that the treatment it receives from the EU is unmatched around the world and harmed Europe’s efforts to become a thriving high-tech economy.

The EU said Microsoft could face fines as high as 2 million
a day, accusing the software company of trying to protect its interests by overcharging rivals for complete and accurate interface documentation - which would allow them to interoperate with Windows PCs and servers - on reasonable and non-discriminatory terms.

The EU complained that three years after a landmark antitrust ruling to open up the market, the U.S. software giant still refused to cooperate.

The company has four weeks to reply to the Commission’s preliminary finding, EU Commission spokesman Jonathan Todd said.

CHINA STORY

CHINA EXPECTS POLLUTION TO IMPROVE SOON

BEIJING (AP) -- China should see a "turning point" this year in its fight against pollution and will likely meet its clean air and water goals in coming years, an environmental official said Tuesday in an unusually optimistic assessment.

Zhang Lijun, vice minister of the State Environmental Protection Administration, said that while the amount of ammonia and nitrates in waterways increased in 2006 and overall air quality declined, pollution control facilities and stepped-up enforcement would have an impact.

"It is true that last year the total pollutants discharged was still rising," Zhang said. But he pointed out that the increase in the amount of pollutants in 2006 was smaller than the increase in the previous year.

"I’m confident that in this year, the total pollutants discharged will come to a turning point," Zhang said, adding he expected a decrease in the amount of pollutants discharged, although he did not provide a specific figure.
A.6 Sets of clusters in the Full Associations and Combined Associations interfaces for the three news stories
<table>
<thead>
<tr>
<th>Keywords (Stories about Nicolas Sarkozy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>law police protests</td>
</tr>
<tr>
<td>law students unions</td>
</tr>
<tr>
<td>police violence youths</td>
</tr>
<tr>
<td>law police protests students</td>
</tr>
<tr>
<td>constitution police protests referendum</td>
</tr>
<tr>
<td>unions violence youths</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locations (Stories about Nicolas Sarkozy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>uk</td>
</tr>
<tr>
<td>europe</td>
</tr>
<tr>
<td>united states uk</td>
</tr>
<tr>
<td>italy germany uk</td>
</tr>
<tr>
<td>iraq london uk</td>
</tr>
<tr>
<td>europe germany uk</td>
</tr>
<tr>
<td>uk london united states iraq</td>
</tr>
<tr>
<td>uk germany europe london</td>
</tr>
<tr>
<td>europe germany iraq london</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Persons (Stories about Nicolas Sarkozy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>angela merkel blair chirac</td>
</tr>
<tr>
<td>blair bush francois bayrou</td>
</tr>
<tr>
<td>ilan halimi villepin chirac</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keywords + Locations (Stories about Nicolas Sarkozy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>london uk europe</td>
</tr>
<tr>
<td>germany uk</td>
</tr>
<tr>
<td>united states uk</td>
</tr>
<tr>
<td>law students protests</td>
</tr>
<tr>
<td>protests unions police</td>
</tr>
<tr>
<td>europe italy iraq</td>
</tr>
<tr>
<td>candidate germany constitution</td>
</tr>
<tr>
<td>youths germany violence</td>
</tr>
<tr>
<td>uk europe london</td>
</tr>
<tr>
<td>candidate italy united states</td>
</tr>
<tr>
<td>police protests students law</td>
</tr>
<tr>
<td>europe constitution germany uk</td>
</tr>
<tr>
<td>united states europe uk germany</td>
</tr>
</tbody>
</table>

Table A.3: First part of the *Full Association* set of clusters for stories about Nicolas Sarkozy.
Table A.4: Second part of the *Full Association* set of clusters for stories about Nicolas Sarkozy.
<table>
<thead>
<tr>
<th>All Associations</th>
<th>Stories about Nicolas Sarkozy</th>
</tr>
</thead>
<tbody>
<tr>
<td>bush chirac</td>
<td></td>
</tr>
<tr>
<td>hadi yahmid villepin</td>
<td></td>
</tr>
<tr>
<td>constitution referendum</td>
<td></td>
</tr>
<tr>
<td>law police 7 unions students</td>
<td></td>
</tr>
<tr>
<td>violence protests</td>
<td></td>
</tr>
<tr>
<td>germany Blair</td>
<td></td>
</tr>
<tr>
<td>candidate uk</td>
<td></td>
</tr>
<tr>
<td>iraq united states</td>
<td></td>
</tr>
<tr>
<td>francois bayrou italy</td>
<td></td>
</tr>
<tr>
<td>angela merkel blair chirac</td>
<td></td>
</tr>
<tr>
<td>blair bush francois bayrou</td>
<td></td>
</tr>
<tr>
<td>ilan halimi villepin chirac</td>
<td></td>
</tr>
<tr>
<td>law police protests 3 law students unions 8 police violence youths</td>
<td></td>
</tr>
<tr>
<td>chirac referendum constitution</td>
<td></td>
</tr>
<tr>
<td>blair germany europe</td>
<td></td>
</tr>
<tr>
<td>uk iraq united states</td>
<td></td>
</tr>
<tr>
<td>candidate marseille europe</td>
<td></td>
</tr>
<tr>
<td>law police protests students</td>
<td></td>
</tr>
<tr>
<td>referendum constitution chirac united states</td>
<td></td>
</tr>
<tr>
<td>europa chirac germany constitution</td>
<td></td>
</tr>
<tr>
<td>students law protests chirac police</td>
<td></td>
</tr>
<tr>
<td>europe united states chirac bush angela merkel</td>
<td></td>
</tr>
</tbody>
</table>

Table A.5: Combined Association set of clusters for stories about Nicolas Sarkozy.

<table>
<thead>
<tr>
<th>Keywords (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>competitors documentation regulators</td>
</tr>
<tr>
<td>fine hearing regulators</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locations (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>brussels california europe</td>
</tr>
<tr>
<td>brussels california europe new york</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Persons (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ballmer kroes todd</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keywords + Locations (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>consumers europe versions</td>
</tr>
<tr>
<td>brussels hearing fine</td>
</tr>
<tr>
<td>brussels documentation competitors</td>
</tr>
<tr>
<td>ruling brussels rivals</td>
</tr>
<tr>
<td>europe new york california</td>
</tr>
<tr>
<td>brussels fine hearing regulators</td>
</tr>
<tr>
<td>rivals ruling luxembourg fine 1 brussels california competitors europe new york</td>
</tr>
</tbody>
</table>

Table A.6: First part of the Full Association set of clusters for stories about Microsoft.
Table A.7: Second part of the *Full Association* set of clusters for stories about Microsoft.

<table>
<thead>
<tr>
<th>Keywords + Persons (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>competitors regulators todd</td>
</tr>
<tr>
<td>fine hearing rivals</td>
</tr>
<tr>
<td>horacio_gutierrez consumers versions</td>
</tr>
<tr>
<td>todd ballmer kroes</td>
</tr>
<tr>
<td>ruling barrett documentation</td>
</tr>
<tr>
<td>smith gates ruling</td>
</tr>
<tr>
<td>ballmer kroes regulators todd</td>
</tr>
<tr>
<td>fine hearing regulators smith</td>
</tr>
<tr>
<td>rivals documentation ruling smith</td>
</tr>
<tr>
<td>competitors versions consumers horacio_gutierrez</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keywords + Locations + Persons (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>brussels smith fine</td>
</tr>
<tr>
<td>ruling todd brussels</td>
</tr>
<tr>
<td>brussels documentation competitors</td>
</tr>
<tr>
<td>ballmer regulators todd</td>
</tr>
<tr>
<td>europe versions consumers</td>
</tr>
<tr>
<td>luxembourg fine hearing</td>
</tr>
<tr>
<td>brussels smith fine hearing</td>
</tr>
<tr>
<td>brussels ruling todd europe</td>
</tr>
<tr>
<td>regulators todd ballmer kroes</td>
</tr>
<tr>
<td>competitors ballmer brussels todd regulators</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locations + Persons (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>brussels europe todd</td>
</tr>
<tr>
<td>brussels barrett smith</td>
</tr>
<tr>
<td>todd gates las_vegas</td>
</tr>
<tr>
<td>horacio_gutierrez ballmer kroes</td>
</tr>
</tbody>
</table>

Table A.8: *Combined Association* set of clusters for stories about Microsoft.

<table>
<thead>
<tr>
<th>All Associations (Stories about Microsoft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>competitors fine</td>
</tr>
<tr>
<td>hearing rivals</td>
</tr>
<tr>
<td>regulators ruling</td>
</tr>
<tr>
<td>brussels todd</td>
</tr>
<tr>
<td>kroes ballmer</td>
</tr>
<tr>
<td>europe consumers</td>
</tr>
<tr>
<td>smith luxembourg</td>
</tr>
<tr>
<td>san_francisco versions</td>
</tr>
<tr>
<td>california new_york</td>
</tr>
<tr>
<td>germany japan</td>
</tr>
<tr>
<td>las_vegas gates</td>
</tr>
<tr>
<td>competitors documentation regulators</td>
</tr>
<tr>
<td>fine hearing regulators</td>
</tr>
<tr>
<td>brussels smith fine</td>
</tr>
<tr>
<td>ballmer regulators todd</td>
</tr>
<tr>
<td>ruling rivals luxembourg 4 new_york</td>
</tr>
<tr>
<td>competitors europe</td>
</tr>
<tr>
<td>fine barrett kroes</td>
</tr>
<tr>
<td>brussels smith fine hearing</td>
</tr>
<tr>
<td>brussels ruling todd europe</td>
</tr>
<tr>
<td>ballmer fine kroes</td>
</tr>
<tr>
<td>competitors ballmer brussels todd regulators</td>
</tr>
<tr>
<td>brussels california competitors europe new_york</td>
</tr>
</tbody>
</table>
Table A.9: First part of the *Full Association* set of clusters for stories about China AND pollution.
<table>
<thead>
<tr>
<th>Keywords + Persons (Stories about China + pollution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>river spill water</td>
</tr>
<tr>
<td>greenhouse gas blair summit</td>
</tr>
<tr>
<td>chirac greenhouse gas change</td>
</tr>
<tr>
<td>bush climate change emissions</td>
</tr>
<tr>
<td>blair bush greenhouse gas summit</td>
</tr>
<tr>
<td>climate change emissions greenhouse gas stephanie tunmore</td>
</tr>
<tr>
<td>climate change gerald meehl john walsh tim barnett</td>
</tr>
<tr>
<td>drinking water river water water supplies</td>
</tr>
<tr>
<td>blair chirac climate change greenhouse gas summit</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Keywords + Locations + Persons (Stories about China + pollution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>climate change kyoto united states</td>
</tr>
<tr>
<td>united states summit blair</td>
</tr>
<tr>
<td>kyoto emissions japan</td>
</tr>
<tr>
<td>bush kyoto greenhouse gas</td>
</tr>
<tr>
<td>chirac greenhouse gas india</td>
</tr>
<tr>
<td>river russia spill</td>
</tr>
<tr>
<td>drinking water spill water</td>
</tr>
<tr>
<td>russia united states emissions climate change</td>
</tr>
<tr>
<td>india kyoto united states japan</td>
</tr>
<tr>
<td>spill river water supplies water</td>
</tr>
<tr>
<td>kyoto bush blair summit</td>
</tr>
<tr>
<td>japan kyoto united states greenhouse gas india</td>
</tr>
<tr>
<td>russia river spill water water supplies</td>
</tr>
<tr>
<td>kyoto emissions climate change united states bush</td>
</tr>
<tr>
<td>blair bush climate change emissions united states</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locations + Persons (Stories about China + pollution)</th>
</tr>
</thead>
<tbody>
<tr>
<td>united states britain blair</td>
</tr>
<tr>
<td>india russia united states</td>
</tr>
<tr>
<td>india japan kyoto</td>
</tr>
<tr>
<td>chirac tokyo japan</td>
</tr>
<tr>
<td>bush britain europe</td>
</tr>
<tr>
<td>wen united states shanghai</td>
</tr>
<tr>
<td>gerald meehl john walsh tim barnett united states</td>
</tr>
<tr>
<td>india japan united states kyoto</td>
</tr>
<tr>
<td>chirac united states blair</td>
</tr>
<tr>
<td>bush tokyo united states blair</td>
</tr>
</tbody>
</table>

Table A.10: Second part of the Full Association set of clusters for stories about China AND pollution.
<table>
<thead>
<tr>
<th>All Associations</th>
<th>Stories about China + pollution</th>
</tr>
</thead>
<tbody>
<tr>
<td>climate_change emissions</td>
<td></td>
</tr>
<tr>
<td>river spill</td>
<td></td>
</tr>
<tr>
<td>water_supplies water</td>
<td></td>
</tr>
<tr>
<td>bush united_states</td>
<td></td>
</tr>
<tr>
<td>greenhouse_gas kyoto</td>
<td></td>
</tr>
<tr>
<td>india chirac</td>
<td></td>
</tr>
<tr>
<td>summit britain</td>
<td></td>
</tr>
<tr>
<td>europe japan</td>
<td></td>
</tr>
<tr>
<td>john_walsh tim_barnett</td>
<td></td>
</tr>
<tr>
<td>stephanie_tunmore russia</td>
<td></td>
</tr>
<tr>
<td>climate_change emissions greenhouse_gas</td>
<td></td>
</tr>
<tr>
<td>drinking_water river_water</td>
<td></td>
</tr>
<tr>
<td>climate_change kyoto united_states</td>
<td></td>
</tr>
<tr>
<td>united_states summit blair</td>
<td></td>
</tr>
<tr>
<td>chirac greenhouse_gas india</td>
<td></td>
</tr>
<tr>
<td>river russia spill</td>
<td></td>
</tr>
<tr>
<td>japan chirac project</td>
<td></td>
</tr>
<tr>
<td>greenhouse_gas bush scott_mcclellan</td>
<td></td>
</tr>
<tr>
<td>climate_change europe britain</td>
<td></td>
</tr>
<tr>
<td>hong_kong shanghai project</td>
<td></td>
</tr>
<tr>
<td>gerald_meehl john_walsh tim_barnett</td>
<td></td>
</tr>
<tr>
<td>india kyoto united_states japan</td>
<td></td>
</tr>
<tr>
<td>climate_change gerald_meehl john_walsh tim_barnett</td>
<td></td>
</tr>
<tr>
<td>bush tokyo united_states shanghai</td>
<td></td>
</tr>
<tr>
<td>japan kyoto united_states greenhouse_gas india</td>
<td></td>
</tr>
<tr>
<td>russia river spill water water_supplies</td>
<td></td>
</tr>
<tr>
<td>kyoto emissions climate_change united_states bush</td>
<td></td>
</tr>
<tr>
<td>blair chirac climate_change greenhouse_gas summit</td>
<td></td>
</tr>
<tr>
<td>bush climate_change emissions india kyoto united_states</td>
<td></td>
</tr>
<tr>
<td>britain climate_change greenhouse_gas india russia summit united_states</td>
<td></td>
</tr>
</tbody>
</table>

Table A.11: Combined Association set of clusters for stories about China AND pollution.
A.7 Instructions/warm-up story used in news archive experiment
Scenario:
Imagine you are a reporter working for an international newswire agency. Your news editor has asked you to write a background for the wire service to support a breaking news story (see topic details overleaf). The background should be about 500 words long.

To begin this task you have run an initial search over a news archive using a standard search engine using the search terms listed (e.g. *earthquake AND Indonesia*).

This search has returned a set of 1000 documents. Your task is now to explore this collection using the interface provided and to find content that you believe will help you to prepare a good background to the story.

**Gathering Documents for Background:**
The goal of the task is to select 4 documents which you believe will help you to write the best possible background to the story.

As you explore the document collection you may copy and paste text from documents you think may be of potential use (please use the text editor of your choice). Remember to include the document header. You can then browse and review this list as you develop your ideas about the topic.

**Completing the task:**
When you feel you have found your 4 documents for the background, please copy and paste the title and header of your 4 documents into the box provided on the web page.

To complete the task, please answer the two feedback questions in the boxes provided overleaf.

**Before you begin:**
Notes on Browsing:
The interface provides a list of groupings of documents in the collection. These groupings are based on common patterns in document content.

- You may click on a grouping to view the associated document(s).

- You may go back to the groupings page at any time using the BACK arrow in the browser.

- Please note, the groupings are ranked based on a search of the top third of the document.

- Before selecting a grouping, we encourage you to read quickly through all the summaries to get an overview of what groupings of documents are available.
EARTHQUAKE ROCKS EASTERN INDONESIA

A magnitude-7.5 earthquake has rocked eastern Indonesia, the US Geological Survey said.

The quake struck in the Banda Sea, about 120 miles south of Ambon city, the USGS said on its website.

No immediate reports of injuries have been made, though some damage to property is to be expected, the USGS said.

AP
Bibliography


Lara D. Catledge and James E. Pitkow. Characterizing browsing strategies in the World-Wide Web. Computer Networks and


H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan. Gate: A framework and graphical development environment for robust nlp tools


B. Gaines. Transforming rules and trees into comprehensible knowledge structures, 1996b. URL citeseer.ist.psu.edu/gaines96transforming.html.


J. Han, Y. Huang, N. Cercone, and Y. Fu. Intelligent query answering by knowledge discovery techniques. IEEE Transactions on Knowledge and Data Engineering, 8(3):373–390, June, 1996.


Leila Kosseim, Stephane Beauregard, and Guy Lapalme. Using information extraction and natural language generation to answer E-mail. *Lecture Notes*


