# A NEW RESTAURANT GUIDE CONVERSATIONAL SYSTEM: ISSUES IN RAPID PROTOTYPING FOR SPECIALIZED DOMAINS[1]


*Stephanie Seneff and Joseph Polifroni*

Spoken Language Systems Group
Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139 USA
http://www.sls.lcs.mit.edu



## ABSTRACT

Over the past year, we have begun the development of a new restaurant guide domain, DINEX, which utilizes all of the same core technology as our other GALAXY systems [2]. The domain server has been adapted from our CityGuide domain within our GALAXY framework, using the same mechanisms to locate places on the map or to give directions. The system has information on several features for restaurants, such as hours, cuisine, parking availability, and an on-line menu. DINEX knows over 450 restaurants in the Boston area, and can also provide directions from subway stops or well-known landmarks. DINEX makes use of a relational database derived from several on-line sources, including a nationally known travel guide. This paper particularly emphasizes the language tools we have developed to construct the relational database semi-automatically and to derive from user queries the appropriate SQL queries for accessing the database.


## 1. INTRODUCTION

In the past three years, the Spoken Language Systems Group at MIT has been developing a framework for conversational systems which we call GALAXY. Among the new subdomains we have introduced within the past year is DINEX, a restaurant guide for the Boston area. DINEX utilizes all of the same core technology as our other GALAXY systems, including SUMMIT for speech recognition [3], TINA for language understanding [4] and GENESIS for language generation [1]. The domain server is an adapted version of our GALAXY CityGuide domain server, and uses the same mechanisms to display places on the map or to give directions.

In porting to this new domain, we had available the GALAXY framework with its asociated language tools, and a collection of potential information sources mostly in electronically readable but nonstandardized formats. This paper focuses mainly on issues involved in creating and accessing a database from this assortment of on-line sources. Our sources included an on-line restaurant guide for the Boston area (RG), as well as computer-readable versions of a nationally known travel guide (TG) and the NYNEX Yellow Pages (YP). A key new development is that the database is represented in a relational format, and the information is accessed using SQL. In order to produce the relational database, we used TINA [4] to parse raw text into semantic frames, and then automatically filled in fields in the database based on the semantic frame analysis. To decode the relational database, we used GENESIS [1] to produce SQL queries directly from semantic frames, treating it as a standard language generation problem. One nice consequence of having a relational database accessible via SQL was that it was very easy to extend the capabilities of the system to handle more complex linguistic phenomena such as disjunction and negation.


[1] This research was supported by a research contract from NYNEX Science & Technology, Inc.


## 2. OVERVIEW

DINEX has information on several features for restaurants, including, but not limited to, the name, address, telephone number, hours, cuisine, parking availability, ambience, whether reservations/credit cards are accepted, price range, handicap accessibility, and on-line menus. Many of these features are not available for all of the restaurants, but as the on-line sources become more extensive we will update our database accordingly. The hours are encoded in a separate table, which provides the hours as a function of meal and day, so that users can ask specific questions such as, "When are they open for dinner on Saturday?" The menu option links directly into the Web through a Web Browser, with only the URL being entered into the relational database.

The TG contained useful information about a number of hotels and museums in the Boston area, so we decided to extend the system's knowledge domain to take advantage of this information. We created a separate *hotel* table that could provide information about the availability of various amenities such as health club, pool, game room, business center, etc. A separate *museum* table contains such information as admission fees, handicapped access, and hours.

We felt it would be helpful if users could obtain directions to the places of interest, but we needed to have available a convenient reference starting point. To this end, we decided to link into the subway system, by providing most of the subway stations as available reference points. We also added a set of well-known landmarks, including universities, fast food establishments, and prominent buildings such as churches or corporate headquarters. Users can find restaurants or hotels "close to" any of these landmarks or they can get directions from the "closest subway station," for example.

We have selected an initial vocabulary of around 1,200 words, which includes the names of all of the restaurants in the RG, as well as the names of all hotels known by the TG. We have also obtained over 2,000 training sentences for the language models, which were a compilation of sentences adapted from the CityGuide domain, augmented with sentences that we constructed ourselves, reflecting the system's expanded knowledge domain. All of these training sentences can be parsed by TINA. We are currently in the process of collecting data from naive users who access the system in a "wizard" mode. Although a preliminary recognizer exists for the domain, we need to train it on a large body of relevant user data before its performance will be adequate for non-wizard mode data collection.

The system uses the same multimodal discourse module that is used for our other GALAXY domains. A powerful feature is that users can select a particular restaurant from a displayed list or a map location, either verbally, as in "give me directions to the *fourth* one," or through mouse clicks. A follow-up question can refer to such preselected items pronominally, as in "What are *their* hours?" The discourse module is beyond the scope of this paper, but interested readers are referred to [5] for further details.

## 3. CREATING THE DATABASE TABLES

Early in the development of DINEX, we made the decision to use a relational database for accessing the data. We felt that the information about any given database entry would be sufficiently complex that the most efficient way of retrieving it would be one that enabled us to create multiple tables and join those tables as needed via SQL queries. Information about hours alone for restaurants requires that we keep track of opening and closing times for all three meals for all seven days of the week. This information must be evaluated numerically and combined with cuisine and area information to allow the system to handle complex queries such as, "Are there any Mexican restaurants in Back Bay or in the Faneuil Hall area open for dinner on Saturday night after 10 P.M.?"

The database tables were derived mainly from three knowledge sources mentioned previously, the RG, the TG, and the YP. These data were available to us as raw text. Thus it was necessary to convert the relevant data into a standardized tabular format, organizing information about individual instances of restaurants, hotels, etc. As a first step, we converted raw data to an intermediate format that was written to a file as sets of [key:value] pairs. This gave us a format that was not only easier to read and check for errors, but also easier to deal with in subsequent processing steps. The keys include categories such as name, address, location, cuisine, ambience, hours, entrees, price, etc., and the values range from a single word to several sentences, or could be in some coded format such as "mo-sa, 8am-10pm." In addition, we found that simple string searches could turn up other categories of interest, such as amenities available in hotels (e.g., pools, business centers, etc.) or types of museums (e.g., art, archaeological, etc.).

A second processing stage converts the [key:value] files directly into a set of relational database tables, sometimes joining files derived from different sources. In cases where the value needed to be more fully decoded, we used TINA to create an intermediate semantic frame, and then filled the relational database fields from this semantic frame. TINA was particularly useful for regularizing cuisine, address, and hours. When appropriate, we used GENESIS to paraphrase these frames into well-formed speakable responses.

**TEXT:** *"mo-we 11a-10p; th-sa 11a-11p"*
**FRAME:**

```
[ clause: hours
   weekday: mo
   end_weekday: we
   from_time: 1100
   to_time: 2200
   and: [ clause: hours
         weekday: th
         end_weekday: sa
         from_time: 1100
         to_time: 2300 ] ]
```

**DATABASE ENTRIES:**

| Day | mo tu we; | th fr sa; |
|---|---|---|
| Open | 1100; | 1100; |
| Close | 2200; | 2300; |
| Meal | lu di; | lu di; |
| Hours | 11A.M.-10P.M. | 11A.M.-11P.M. |

**Figure 1:** Example of semantic frame and database entries resulting from parsing a text string identifying the hours of a restaurant. When unspecified in the text, meals are inferred from the time intervals.

Addresses were often specified as a straightforward sequence of number, street_name, and street_type, (e.g., "545 Main Street"), but could also be given in alternative forms, such as an intersection of two streets (e.g., "the corner of Boylston and Exeter"), a street name and a nearby landmark ("Newbury Street near the Common") or something more complicated, such as "just off Moody, across from Iguana Cantina." We were able to write a small number of parse rules to cover the typical variations, and then automatically create a semantic frame which provides the information sorted into appropriate categories, such as *street_type* and *area*, for inclusion in a relational database table. At the same time, an automatically generated paraphrase of the frame would give an entry for the "descriptive_address," which could be provided when the user asked directly for the restaurant's location. We exploited augmentations with neighborhood/area names such as "the North End" or "Harvard Square," when available.

The most powerful use of TINA for regularizing text data was in enrolling the information about opening and closing times for restaurants into an hours table. In many cases, the information is incomplete – for example, the entry may specify only that the restaurant is open for breakfast, without giving the actual hours, or it may simply specify the hours of the restaurant (9:00 A.M. - 1:00 A.M.) without specifying individual meals. In many cases, the restaurant has differ-

| Source | Count | Supplement | Overlap |
|---|---|---|---|
| RG | 440 Restaurants | YP | 315 (72%) |
| TG | 260 Restaurants | RG | 100 (40%) |
| YP | 159 Hotels | TG | 36 (23%) |

**Table 1:** Some overlap statistics on our knowledge sources.

ent hours for different days, and separate opening and closing times for each meal. Finally, the restaurant may have no information on hours at all for a given meal/day, which is distinct from being closed during that interval. We developed a mechanism that would deal correctly with all of these situations. The original data for hours was typically parsed into a frame with multiple clauses, and each clause was decoded as one or several rows in the *hours* table. An example of such a text entry, along with the associated semantic frame, is given in Figure 1. The entries in the "Hours" row are obtained by paraphrasing the individual clauses in the semantic frame.

While the information available in the RG for hours was originally quite incomplete, we have gradually filled it in for many more restaurants through other sources, including the TG and direct inquiries. We use a simple textual format to specify hours, automatically recreating the database to incorporate new information.

The YP mainly provides geo-coded data (latitude and longitude) for many of the restaurants, hotels, and museums, allowing us to position them on the map and give travel directions. The TG provided a lengthy description for many restaurants, hotels, and museums. Instead of filling a huge entry in an SQL database, we set up these descriptions as separate HTML formatted files, and provided a field in the SQL database specifying the URL for the corresponding description file. When a restaurant provides an on-line menu, we include a pointer to its Web page in this file. When the user asks for "more information" about a particular place, she gets a display of this HTML document via a Web browser. The files and pointers are created completely automatically.

Some summary statistics on the amount of overlap of data in our various resources is given in Table 1. We defined our initial set of 440 restaurants based on the RG. 315 (72%) of these restaurants were also present in the YP, whereas only 100 of them were also in the TG. We were disappointed with the small overlap between the RG and the TG, because the TG contained a wealth of additional information for the restaurants that *were* present. The TG contained an additional 160 restaurants which were absent from the RG. These restaurants have not yet been incorporated into our system. The TG was an excellent source for additional information about hotels, although it covered fewer than a quarter of the hotels known to the YP.

## 4. GENERATING SQL QUERIES

Our GENESIS system was originally developed to generate paraphrases and responses in multiple *natural* languages [1]. Even though SQL is not a natural language, it was relatively straightforward to generate well-formed SQL queries from semantic frames using the same mechanism that is used for natural languages. We did however need to modify our generation functionality to allow a few special features that are a consequence of the artificial nature of SQL as a language.

Messages

| | |
|---|---|
| display | select distinct name, lat, lon, ..., city :TOPIC |
| serve_hours | :PRED [ and meal like '% :TOPIC %' ] |
| town | and :QUANT city = ' :NAME ' |
| np-on_street | :NP and street_name = ' :TOPIC ' |

Vocabulary

| | |
|---|---|
| university | "from landmarks where type = 'university' " |
| open_after | "and close_time >" |
| cafe | "from restaurants [and name like '%Cafe%']" |

**Figure 2:** Example entries from the vocabulary and messages files for GENESIS SQL generation from semantic frames. Segments delimited by square brackets are later moved to the end of the query.

**Utt:** IS WILD GINGER BISTRO OPEN FOR LUNCH ON SUNDAY

**Semantic Frame:**

```
[ clause: truth
    topic: [restaurant
        pred: [place_name
            topic: Wild Ginger Bistro] ]
    pred: [open
        pred: [on_day
            pred: [day_of_week topic: Sunday] ] ]
    pred: [with_meal
        topic: [meal :name lunch]]]
```

**Query:**

```
select distinct name, lat, lon, street_number, street_name,
street_type, tel, city, hours from restaurants, hours where
name like '%Wild Ginger Bistro%' and meal like '%l%' and
day like '%su%' and hours_id = rest_id
```

**Figure 3:** A sentence and its associated semantic frame and SQL query, as produced through GENESIS.

Generation is accomplished by a recursive algorithm that traverses the hierarchical semantic frame, constructing and ordering substrings of the query associated with each topic and predicate structure. In attempting to use GENESIS for SQL generation, the main problem we encountered was that individual units in the semantic frame sometimes needed to generate disjoint pieces of the query. Fortunately, a set of filtering clauses for SQL queries (such as "and x = 'string' ") could be organized in any order, as long as they were all *after* the "select" sequence. We solved this problem by implementing a mechanism invoked in a post-processing step, whereby any expressions that are surrounded by brackets are excised and moved to the end of the query.

GENESIS is controlled by a vocabulary file and a messages file that are specific to each language. Sample entries from these files for SQL, shown in Figure 2, are suggestive of the simplicity of the technique. Portions of the specification delimited by brackets are to be moved to the end of the query in a postprocessing step. An example sentence with the associated semantic frame and SQL query is shown in Figure 3.

## 5. EXAMPLES

Figure 4 shows the user window controlled by the GALAXY client. The user had just asked for restaurants in Chinatown that *don't* serve

Chinese food. The new query appears in the top window, and immediately below is the paraphrase, representing the utterance's interpretation in context. The response lists the cuisines for the five non Chinese restaurants in Chinatown, which are also displayed on the map along with numerical identification, although the reproduction quality obscures that feature. The user could now click on an item on the map or in the list and ask a follow-up question about the selected item.

Figure 5 gives a sample dialogue, illustrating several features of the system. For the second utterance, the system determines the distance between the Prudential and all hotels with a pool, selecting those that are within a certain radius. The third sentence further constrains the set to those that also have a business center, reducing it to just two hotels. Utterance 4 demonstrates the mouse selection capability, where "closest" is now understood to mean "closest to the item selected." For utterance 5 the system displays a Web page containing a travel guide description of l'Espalier, along with a pointer to the on-line menu. For the final utterance the system uses a path finding algorithm to compute directions from the Sheraton to l'Espalier, speaking the directions and displaying the path on the map.

## 6. FUTURE PLANS

DINEX is still in the early stages of development. We have just begun to collect data in wizard mode, recording simultaneously both wideband and telephone quality speech, and keeping a detailed log of each dialogue. This process will be continued over the next several months, and these data and logs will be used to improve all aspects of the system. We would like to add the capability of dynamic vocabulary adaptation, so that the known names of restaurants, hotels, etc., could be adjusted to include those names that had been recently displayed to the user. We have tentative plans to produce a version of DINEX that operates without a display, which would provide a useful telephone-only service. It would be relatively straightforward to expand to other cities, such as New York City, particularly if appropriate on-line travel guides are available. We may also port the system to other languages, most probably to Spanish.

## 7. REFERENCES


1. J. Glass, J. Polifroni, and S. Seneff, "Multilingual Language Generation across Multiple Domains," *Proceedings, ICSLP*, pp. 983-986, Yokohama, Japan, 1994.
2. D. Goddeau, E. Brill, J. Glass, C. Pao, M. Phillips, J. Polifroni, S. Seneff, and V. Zue, "GALAXY: A Human-language Interface to On-line Travel Information," *Proceedings, ICSLP*, pp. 707-710, Yokohama, Japan, 1994.
3. M. Phillips and D. Goddeau, "Fast Match for Segment-based Large Vocabulary Continuous Speech Recognition," *Proceedings, ICSLP*, pp. 1359-1362, Yokohama, Japan, 1994.
4. S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61–86. 1992.
5. S. Seneff, D. Goddeau, C. Pao, and J. Polifroni, "Multimodal Discourse Modelling in a Multi-User Multi-Domain Environment," *These Proceedings*, 1996.


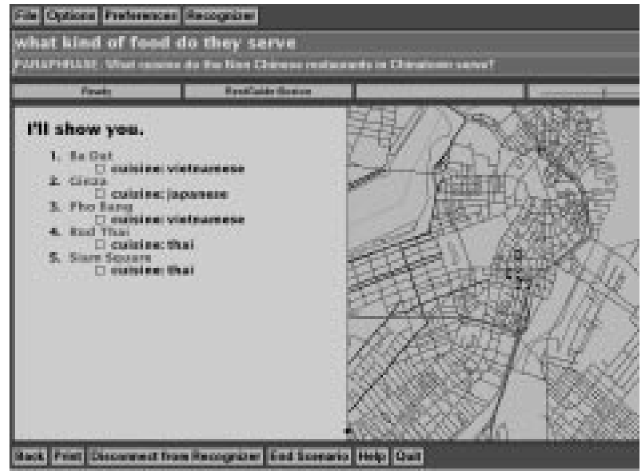


**Figure 4:** Example of the DINEX user interface. The user had previously asked for restaurants in Chinatown that do not serve Chinese food.

**1 U:** WHERE'S THE PRUDENTIAL BUILDING

**1 S:** *The Prudential building is at 800 Boylston Street.*
<MAP DISPLAY>

**2 U:** ARE THERE ANY HOTELS WITH A POOL NEAR THERE

**2 S:** *Here are the hotels near the Prudential that have a pool.*
1. Back Bay Hilton
2. Colonnade Hotel
...
8. Sheraton Boston Hotel & Towers

<MAP DISPLAY>

**3 U:** WHICH OF THESE HAS A BUSINESS CENTER

**3 S:** *Here are the hotels near the Prudential that have a pool and a business center.*
1. Marriott Copley Place
2. Sheraton Boston Hotel & Towers

<MAP DISPLAY>

**4 U:** <click on Sheraton Boston Hotel & Towers >
WHAT'S THE CLOSEST FRENCH RESTAURANT

**4 S:** *Here is the french restaurant closest to the Sheraton Boston Hotel and Towers.*
* L'Espalier <MAP DISPLAY>

**5 U:** WHAT DO YOU KNOW ABOUT THIS RESTAURANT

**5 S:** Click here if you'd like to see a menu: MENU
Here is what the Travel Guide says about L'Espalier.
<TRAVEL GUIDE DESCRIPTION>

**6 U:** HOW DO I GET THERE

**7 S:** *Starting from Sheraton Boston Hotel & Towers, follow the traffic on Dalton Street. Turn right on Boylston Street. Turn left on Gloucester Street. L'Espalier will be at 30 Gloucester Street on your left.*

**Figure 5:** Sample dialogue from the DINEX restaurant guide domain.