Development of the HRL Route Navigation Dialogue System

Robert Belvin
HRL Laboatories, LLC
3011 Malibu Canyon Road
Malibu, CA 90265
310-317-5799
rsBelvin@hrl.com

Ron Burns
HRL Laboatories, LLC
3011 Malibu Canyon Road
Malibu, CA 90265
310-317-5445
rBurns@hrl.com

Cheryl Hein
HRL Laboatories, LLC
3011 Malibu Canyon Road
Malibu, CA 90265
310-317-5489
cheryl@hrl.com

ABSTRACT

In this paper we report on our work on a prototype route navigation dialogue system for use in a vehicle. The system delivers spoken turn-by-turn directions, and has been developed to accept naturally phrased navigation queries, as part of our overall effort to create an in-vehicle information system which delivers information as requested while placing minimal cognitive load on the driver.

Keywords

Dialogue Systems, Discourse, Navigation, NLP, Pragmatics, Dialogue Manager

1. INTRODUCTION

In this paper we report on our work on a spoken language navigation system which runs in real-time on a high-end laptop or PC, for use in a vehicle. We focus on issues in developing a system which can understand natural conversational queries and respond in such a way as to maximize ease of use for the driver. Because today's technology has the potential to deliver massive amounts of information to automobiles, it is crucial to deliver this information in such way that the driver's attention is not diverted from the primary task of safe driving. Our assumption has been that a dialogue system with a near-human conversational ability would place less of a cognitive load on the driver than one which behaves very differently than a human.

We have implemented a testbed on which to develop and evaluate driver interfaces to navigation systems. Our approach is multi-modal and the interface will include a head-up display, steering hub controls, and spoken language, though it is only the latter modality that we report on here. We first discuss our development phases, and after this we provide an overview of our implementation, emphasizing the natural language processing aspects and application interface to the map databases. Next we provide results of our initial evaluation of the system, and finally we draw conclusions and summarize plans for future work.

2. DEVELOPMENT PHASES

One can identify four distinct subproblems which must be solved for a navigation system: 1) the natural language navigation interface, 2) street name recognition, 3) the natural language destination entry interface given street name recognition, and 4) the map database interface. We have partitioned the problem and have phased our development to progressively implement solutions with increasing complexity.

Navigation system implementation is complicated by the potential of having a very large street name vocabulary with many unusual and uncommon pronunciations with significant variations across speakers. The appropriate name space is dynamic since it depends on the location of the vehicle.

Our initial system does not accept queries with proper street names. In addition, we assume separate destination entry and route planning systems, and that one or more routes have been loaded into the navigation system. The system relies on open dialogue to resolve the directions at any stage of the journey and may or may not use the Global Positioning System (GPS) to determine the progress along the route. By implementing this system first we could concentrate on the dialogue aspects of the navigation problem and also establish a baseline with which to compare our other implementations.

In the second phase we include a limited set of street names as part of the language model and lexicons. Initially we are using a predefined set of names with hand tuning of the pronunciations. Additional research is required to solve the street name recognition problem generally and automatically. We assume in-vehicle GPS and use a map matching system to determine the vehicle's position and if it is on-route. This phase includes development of the natural language components for destination entry and also broadens the scope of the navigation queries to include questions with and about street names. More distant plans include on-road route replanning, providing information to requests for specific street names or points of interest along the route, and traffic information and workarounds.

3. IMPLEMENTATION

Our implementation is based on the Galaxy-II system [6] from the Massachusetts Institute of Technology (MIT), which is the baseline for the Communicator program of the Defense Advance Research Projects Agency (DARPA). The architecture consists of a hub client that communicates, using a standard protocol, with a number of servers as shown in Figure 1. Each server generally implements a key system function including Speech Recognition, Frame Construction (language parsing), Context Tracking, Dialogue Management, Application Interface, Language Generation, and Text-to-speech.

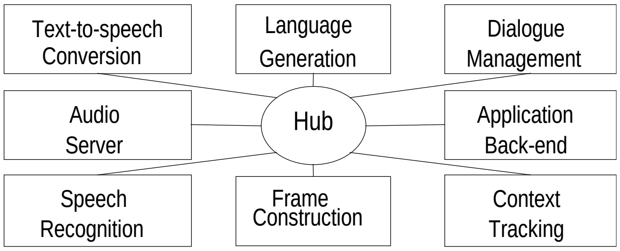

Figure 1. The client-server architecture of the MIT Galaxy-II is used to implement our navigation system testbed.

3.1 Speech Recognition

We use the latest MIT SUMMIT recognizer [8] using weighted finite-state transducers for the lexical access search. We have also "plugged in" alternate recognizers such as the Microsoft Speech SDK recognizer and the Sphinx [3] speech recognizer available as open source code from Carnegie Mellon University.

We are in the process of developing a large database of in-vehicle utterances collected in various car models under a wide range of road and other background noise conditions. This data collection will be carried out in two phases, the first of which is completed; phase two is underway. Limited speech data will result from the first phase and subtantial speech data (appropriate for training acoustic models to represent in-vehicle noise conditions and testing of recognition engines) will come out of the second phase, and will become available through our partners in this collection effort, CSLR at University of Colorado, Boulder [4]. In the meantime we are using the MIT JUPITER acoustic models. The performance is acceptable for our language and dialogue model development, but we refrain from presenting any detailed recognizer results here since they would not reflect fairly on optimized recognizer performance.

Our vocabulary consists of about 400 words without street names. We have an additional 600 street names gleaned from the Los Angeles area where we do much of our system evaluation. Baseforms for the vocabulary are derived from the PRONLEX dictionary from the Linguistic Data Consortium at the University of Pennsylvania. Extensive hand editing is needed especially for the street names. The MIT rule set is used for production of word graphs for the alternate pronunciation forms. We have derived a language model from a set of utterances that were initially generated based our best guess of the query space. As evaluation evolves, we modify the utterance list and retrain the language model. The language model uses classes and includes both bigram and trigram models.

3.2 Application Interface

We are building the application interface in several phases. Initially we are only answering queries about turns and distances during navigation. We obtain the database in two steps. First, we access a commercial map database using standard text I/O for destination entry and route planning. This produces a detailed set of instruction that includes many short segments such as on- and off-ramps. We filter this and rewrite the data to provide a set of natural driving instructions suitable for verbal communication. The result is a flat database, such as the one shown in Figure 2. This is loaded into to the system and used to formulate answers to the route queries. In the example in Figure 2 the estimated driving time is 45 minutes. Each row is a segment of the trip. The first and second columns code right, left, straight, and compass direction information. The third column is the segment length in miles and the last is the segment name.

```
HRL                 Airport            45
0  0   1.0   Malibu Canyon Road
0  2  13.3   PCH
1  1   3.4   East I-10 freeway
0  1   4.5   South I-405 freeway
3  0   0.3   Howard Hughes Parkway
0  1   1.7   Sepulveda Boulevard
0  4   0.4   Los Angeles International Airport
```

Figure 2: Example database used as source data to generate guidance instructions for the route between HRL Laboratories and the Los Angeles airport

A sample dialogue is shown in Figure 3 which illustrates the kind of responses the system can generate from a database such as that shown above, given navigation queries of the sort shown; this sample was drawn from our phase I user-system data logs:

S	Welcome to Navigator
U	*What routes are loaded?*
S	I have one set of directions loaded. It is for the route to the airport from HRL
U	*What's my first turn?*
S	From Malibu Canyon Road turn left onto Pacific Coast Highway
U	*And after that?*
S	From Pacific Coast Highway go straight onto East I-10 freeway
U	*How far is that turn?*
S	The distance is 14.3 miles
U	*My odometer reads fifteen point one miles.*
S	Your odometer reading is fifteen point one
U	*What is my next turn?*
S	From East I-10 freeway turn right onto South I-405 freeway
U	*How far is it from here?*
S	The distance is 2.6 miles
U	*How far is the airport from there?*
S	The distance is 6.7 miles
U	*Bye Bye.*
S	Goodbye. Have a nice day.

Figure 3. A navigation system dialog illustrating some sample queries and showing the responses derived by the dialogue manager based on the database of Figure 2.

Off-line construction of the global navigation database is not unrealistic since it could be done, at least in the near term, by a service organization such as OnStar from General Motors (GM). However as navigation systems become widely deployed, users will expect destination entry including real time route re-planning to be an integral part of system. We are developing a direct voice interface to the commercial map database that includes destination entry, route planning, and map matching using GPS data to determine if the vehicle is on-route or not.

During the destination entry phase street names need to be robustly recognized. We are currently working with a subset of street names in the Los Angeles area preloaded in the recognizer and language models. It is untenable to keep all of the street names in Los Angeles loaded in the recognizer simultaneously (there are around 16,000, including 8,000 base names), thus we are developing a method for dynamic loading of map names local to the vehicle position which we will report on in the near future.

We have experimented with using a subset of street names as a filter list, and as a lookup list based on spelling the first few letters, to try to resolve the destination requested. If this fails, or if the trip is outside of the area from which names are loaded, we rely on more complete spelling to determine the destination. The origin for the route plan is generally implied since it is determined by the GPS position of the vehicle most of the time. Once the destination is determined it is straightforward to continuously re-plan the route based on the current vehicle position and thereby be able to provide remedial instruction if the driver departs from the route plan.

3.3 NL Analysis and Generation

The core NLP components in our system are a TINA [5] grammar, context tracking mechanisms, and the GENESIS language generation module. The TINA grammar includes both syntactic and semantic elements and we try to extract as much information as possible from the parse. The information is coded in a hierarchical frame (Figure 4a) as well as a flat key-value pair (Figure 4b). In addition to handcrafting this grammar, a set of rules was also developed for the TINA inheritance mechanism. These rules are applied during context tracking, after the parse, to incorporate information from the dialog history into phrases such as "and after that" and "how about my second turn," and are also used to incorporate modifications that are a result of dialogue management.

a) Parse frame

```
{c locate
 :domain "Nav"
 :pred {p locate_object
      :topic {q turn
          :quantifier "poss_pro"
          :pred {p ord
              :topic 2 }}}}
```

b) Key-values Pairs

```
:clause "locate" :locate_object "turn" :ORD 2
```

c) Reply Frame

```
{c speak_turn
  :topic {q turn
      :turn_direction "straight"
      :current_roadway "PCH"
          :new_roadway "East I-10 freeway"
  :domain :Nav" }
```

Figure 4. Example frames produced for the simple query "What's my second turn?"

As noted, we use the MIT GENESIS server for language generation. Again this module is rule driven and we developed the lexicon, templates and rewrite rules needed for the three ways we use GENESIS. We extract the key-value pairs (e.g. Figure 4b) from the TINA parse frame. The key values are used to help control the dialogue management as well as provide easy access to the variable values. We use GENESIS to produce the English reply string that is spoken by the synthesizer. The example frame in Figure 4c in conjunction with our rules generates the sentence "From Pacific Coast Highway turn straight onto East I-10 freeway" Lastly GENESIS is used to produce an SQL query string for database access. Templates and rewrite rules determine which form the output from GENESIS will take. Technically these three uses (key-value, reply string, and SQL) are just generation of different languages.

3.4 Dialogue Management

We have developed servers for dialog management and to control the application interface for database query. The hub architecture supports use of a control table to direct which server function is called. This is especially useful for dialogue management. The control table is specified by a set of rules using logic and arithmetic operations on the key-value pairs. A well-designed set of rules makes it far easier to visualize the flow and debug the dialogue logic. For example, when a control rule such as:

```
Clause "locate" !:from --> turn_from_here
```

fires on the key-value pairs (Figure 4b), the hub calls the turn manager function "turn_from_here". In this simplified case, we are assuming if there is no ":from" key, the request is to locate an object (i.e. "turn") relative to the vehicle's current position. In this case the function needs only to extract the value of the key ":ORD" and look up the data for the second turn in the database of Figure 2. This data is then written into the response frame, here called "speak_turn" and shown in Figure 4c. GENESIS uses this frame to generate the English language reply that is spoken by the synthesizer as described above.

In the examples shown here we communicate with the database from the dialogue manager by downloading a flat database such as that of Figure 2, perhaps via a data link to an off-board service organization such as OnStar. In cases where we access databases directly, we use a separate server for this function. Generally, communications between the dialogue manager and database servers are routed via the Hub.

Our dialogue manager has been designed to use GPS data when available (in which case GPS coordinates would also be a part of the database) or to use location information based on current odometer readings provided as input by drivers when GPS is not available. We use this latter method for demonstrating the system in a desktop setting, though we have also recently completed a utility for employing maps generated by our commercial navigation database, graphically displaying a driver's progress along an imaginary route. We are now employing this tool as part of our current iteration of system testing and revision.

3.4.1 Referential amigiuties in driver queries

The driver can query to determine turn or distance information relative to current vehicle position, relative to another turn or reference point in the database, or as an absolute reference into the route plan stored in the database. We have devoted considerable effort to dealing with ambiguities which may arise as a result of different ways users may be conceptualizing the route (that is, in absolute or relative terms), as well as the driver being at different points in the route, and at different points in the progression of a discourse segment. Queries such as "what's next?" can be ambiguous. Determining the correct interpretation requires consideration of the discourse history and the user's circumstances. For example, in the following dialog sequence (drawn from our data), there are at least two possible interpretations for "what is next?" in the third turn (U:user, S:system):[1]

```
   ---------------
   U:  what's my next turn
   S:  From Malibu Canyon Road turn left
       onto Pacific Coast Highway.
   ---------------
   U:  and after that
   S:  From Pacific Coast Highway go
       straight onto East I-10 freeway
   ---------------
→ U:  what's next
```

Figure 5. Sample dialog containing ambiguous "What is next".

Notice that this query could be requesting information about the next turn from the driver's current position (i.e. the immediately approaching turn), or it could be requesting information about the *third* turn from the driver's current position, that is, the next turn from the most recently referred to turn. We will henceforth refer to these two interpretations as *next-from-here* and *next-after-that,* respectively.

The factor which *appears* to have the most influence on which interpretation is given to this utterance originates neither in the utterance itself nor in the preceding dialog, but is purely circumstantial, namely, how much time has passed since the last utterance. Our assumption has been that there is a kind of time-dependency factor in coherent discourses: while "what is next" is still within the scope of the preceding discourse context, it may (most likely will) be given the *next-after-that* interpretation. But after a certain length of time has elapsed, "what is next" cannot be interpreted as referring to some previously uttered instruction, but only as referring to the driver's current position. If we think of this in terms of the user's frame of reference for talking about their real or imagined location (we'll refer to this as the FROM value), then we could characterize this phenomenon as the value of FROM being reset to HERE in the absence of immediate discourse context.

Interpretations of *numbered* turn references (e.g. "what's my *second* turn") can also vary depending on another purely circumstantial factor, namely whether the driver is querying the system while preparing to begin the trip, or after she has begun driving. Some drivers will want to preview trip information before beginning to drive, and in this situation, interpretation of certain query types may differ from interpretation done during the trip. When the driver is querying the system before beginning to drive, she is more likely to conceive of and speak of the route in an absolute sense (cf. [7]). That is, the driver may conceive of the route as a fixed plan, wherein each turn and segment have a unique and constant order in a sequence. When conceiving of the route in this way, one may refer to turns by *number in the route,* rather than by *number relative to current position*. Although we have yet to gather real user data bearing on this question, our intuition is that once the trip is underway, especially once any significant distance has been traveled, if users do use numbered turn references at all, they will be much more likely to use them relative to their current position.

Queries of this type are, for practical purposes, only ambiguous once the user has begun the trip, but prior to the absolute numbered turn. Drivers are very unlikely to be asking about the second turn in the route once they have passed the second turn. Moreover, since people will generally only keep track of turn numbers in the range of 1-3, (give or take 1), numbered turn references will only be ambiguous prior to the third or fourth turn in the route (nobody is likely to be asking "what is the eighth turn in the route"). What is more, if the user asks a numbered turn query before beginning the trip, the system response will be the same, since the relative and absolute turn numbers will at that point coincide. Thus, the only time a true ambiguity must be handled by the system is the time after the trip is underway, and before the fourth turn. It is perhaps worth noting that if one looks at the overall query interpretation problem as entailing a determination of whether the user is asking a question relative to their current position, or some other position, then the absolute/relative distinction is just a special case that.

We have gone on the assumption that there are a substantial number of these ambiguous queries, not only for those of the "what's next" type, but also for some numbered turn requests, and for a class of distance query [1]. However, we have now carried out an experiment in which subjects interpreted such queries in a controlled setting, and the results indicate there is far less ambiguity in truly felicitous driver utterances than we originally hypothesized [2]. There probably will be some genuinely ambiguous queries, especially for a system which is not capable of detecting prosodic cues, however, we now are of the opinion that they will *not* comprise a significant percentage of driver queries. For the system which we describe herein, however, the control logic for queries of the type under discussion includes consideration of the temporal "reset" threshold discussed above, as indicated in the following table:

Table 1. Decision matrix showing some determining factors for interpreting "next" and numbered turn queries.

n-route	--	--	√	--	--	--	--
eset threshold eached	--	--	--	--	--		√
ext turn		√			√	√	√
umbered turn	n		n	n			
From here"	√	√					
After that"				√	√		
QL Turn number	c+n	c+1	n	r+n	r+1	r+1	c+1

√ = set **c** = current position
blank = not set **r** = most recently mentioned turn
n = number value **--** = irrelevant

[1] There is at least one further possible interpretation to "what is next?" here, at least if the proper prosodic features are present. If heavy emphasis is placed on "what," the query has a quasi *echo-question* interpretation, indicating either that the user did not hear, or else is surprised at the prior instruction and is asking for clarification or repetition.

The table is to be read column-by-column. Thus, the first column tells us that if we have a query with a numbered turn reference and a phrase which is semantically equivalent to “from here” (which is also the default), then the instruction number which will be requested (via SQL query) from the database is **current+number**.

4. EVALUATION

We are have implemented an initial system and are conducting ongoing evaluations and iterative enhancements as part of a second phase of effort. We are in advanced development on the second phase. We report here on some results of the first phase of our project.

Evaluation of a route guidance system is difficult because the majority of time is spent driving with only a periodic need for instructions. Therefore, for the purposes of developing the language and dialogue models we tried to expedite data collection by having dialogues in which the user simulated a trip by means of a more or less continuous conversation with the system. The position of the vehicle along the route was determined by the user providing odometer readings relative to the start of the trip. At each point the user would query the system and input a new odometer location along the route and continue the dialogue. While certainly not as meaningful as queries under normal driving condition, we did obtain good data for our recognizer language model and grammar coverage. In addition we could debug and tune our turn manager functions to make sure we were properly accessing the database and providing correct responses.

Each query essentially represents a single task and the most meaningful metric for this type of system seems to be the number of dialogue turns per correct response. By correct response we mean that the system provides the final answer versus providing a request to repeat or disambiguate the user query. We have accumulated several thousand utterances during dialogues that run around fifteen to twenty turns per session for a simple route like the one in Figure 2. About a third of the utterances are used to set the vehicle position via inputting odometer data.

We can also divide the dialogues into task oriented dialogues, where the user is trying to get helpful answers, and dialogues where the user is exploring the limits of the system. We find with the task oriented dialogues that the number of dialogue turns are about 15-20% greater that the number of correct responses and that the inital implementations even without street name recognition is a useful system.

5. SUMMARY AND FUTURE PLANS

We have reported on our initial implementation and results for an in-vehicle navigation system through the first phase of our system development effort and into the second phase. Full exploitation of the natural language interface is not fully completed in the first phase because we are still developing an operational in-vehicle navigation system to integrate with our dialogue system. The full interface, including destination entry, route planning, position tracking, and map matching will be available later this year. We have, however, developed most of the NL components needed for accessing the database functionality as it comes on-line. We plan to add other important functionality such as points-of-interest and traffic conditions as the project progresses.

Two other major elements need to be further explored to gain full system functionality. The first is recognizer robustness in the presence of in-vehicle noise during normal everyday use; the second is the street name recognition and pronunciation synthesis problem. Recognizer performance is being addressed by means of a full-scale data collection and corpora development project, in collaboration with GM and the Center for Spoken Language Research at the University of Colorado at Boulder. This work will provide the in-vehicle acoustic data needed to re-train the recognizer models as well as provide a database for developing noise-mitigation and speaker adaptation algorithms for improving recognizer performance. We are developing a method for dynamic loading of street names which we will report on in the near future.

Acknowledgments. This work was supported in part by a research contract from General Motors.

6. REFERENCES

[1] R. Belvin, R. Burns and C Hein "*What’s next*: A Case Study in the Multidimensionality of a Dialogue System," *Proceedings of ICSLP 2000*, Beijing, China, October 2000.

[2] A. Kessell and R. Belvin "Unambiguous Amiguous Questions: Pronominal Resolution in Human-to-Human Navigation," unpublished ms., HRL Laboratories, 2001.

[3] K-F. Lee, *Automatic Speech Recognition: The Development of the Sphinx System*, Kluwer, Boston, 1989.

[4] B. Pellom, W. Ward, J. Hansen, K. Hacioglu, J. Zhang, X. Yu, and S. Pradhan, "University of Colorado Dialog Systems for Travel and Navigation," these proceedings, 2001.

[5] S. Seneff. "TINA: A Natural Language System for Spoken Language Applications," *Computational Linguistics*, Vol. 18, No. 1, pp. 61-86, 1992.

[6] S. Seneff, E. Hurley, R. Lau, C. Pao, P. Schmidt and V. Zue, "Galaxy-II: A Reference Architecture For Conversational System Development," *Proc. ICSLP '98*, pp. 931-934, Sydney, Australia, November 1998.

[7] L. Suchman. *Plans and situated actions.* Cambridge University Press, Cambridge, 1987.

[8] V. Zue, S. Seneff, J. Glass, J. Polifroni, C. Pao, T. Hazen & L. Heatherington, “Jupiter: A Telephone-Based Conversational Interface for Weather Information,” *IEEE:Transactions on Speech and Audio Processing*, Vol. 8, No. 1, pp. 85-96, 2000.