UNDERSTANDING REFERRING EXPRESSIONS IN A PERSON-MACHINE SPOKEN DIALOGUE

Claudia Pateras, Gregory Dudek and Renato De Mori

School of Computer Science, McGill University
3480 University, Montreal, Québec, Canada H3A 2A7
e-mail: cdp@cs.mcgill.ca, dudek@cs.mcgill.ca, renato@cs.mcgill.ca

ABSTRACT

In the domain of mobile robotic task execution under dialogue control, a primary goal is to identify the task target which is specified by a natural language description. A number of concepts are expressed in the user spoken language by vague terms like "the big box" and "very close to the door". We use fuzzy logic to map these vague terms onto the quantitative data collected by system sensors. Fuzziness may cause uncertainty in interpretation and, in particular, in understanding references. This uncertainty is abated by collecting additional information through queries to the user and autonomous sensing. Entropy is used to select the queries having the greatest discriminatory power among referent candidates. In addition, we examine the trade-off between querying, sensing and uncertainty. A framework to deal with each of these issues has been developed and will be presented in the following.

1. NATURAL LANGUAGE UNDERSTANDING

In order to facilitate person-machine interaction, several computer systems are using techniques to understand natural language, either spoken or written. Unlike most Natural Language Understanding (NLU) systems, in which recognized keywords and sentence structures are mapped to database entities and queries respectively, our system can handle words that are not directly related to database entries. An example of words of this type are qualifiers such as "small", "blue", and "far", which have imprecise definitions. These terms are often used to describe an object which cannot be referred to by a unique label. Consider an office environment in which a mobile robot has the capability to fetch and transport various objects. These objects may be so numerous that it would be unrealistic to assume that each can be referred to by a distinct name or label. Furthermore, it would be unnatural and impractical for humans to use unique labels for every article. A more reasonable assumption is that each object can be uniquely described by a finite set of qualifiers some of which may be relational. A system designed to interact with humans must be able to deal with such imprecise terms and identify the object being referred to by the human. The work described here assumes that user commands will contain a form of referring expression, or description, that designates the target of the command task. The system must first identify the referent in order to execute the task.

As sentence interpretation is performed on the output of a system for Automatic Speech Recognition (ASR), it is also important to learn, from a training corpus, the relation between the targets of the command task and the expressions used by the user. Interpretation does not depend only on a spoken sentence, but also on the dialogue history, the robot environment and the focus. Automatic learning along the lines of [KM94] involving history and focus information is used for this purpose.

2. DATA REPRESENTATION AND MAPPING

For the purposes of this work, we assume a mobile robot that is able to navigate in a partially known environment and able to use sensors to observe or compute attributes. We use the term attributes to refer to the various properties (derived or directly observable) of objects. Typical properties that can be sensed with real robots include volume, shape, colour, reflectance, height, and 3-D pose (position and orientation) [MD94, Bro85, BB82, Kro87]. The system has access to a Short Term Memory (STM) containing the attribute values of world objects in the form of numerical data as obtained from the sensors. In contrast, the user's representation and description of the environment will involve subjective and context-sensitive qualifiers. Thus, to enable the system to identify the referent of the user's description, we must perform a mapping between the qualitative labels and the quantitative stored data.

To address the mapping problem we propose to use fuzzy logic [Zad73]. This formalism maps continuous data into a finite number of categories. Each category has an associated fuzzy membership function which assigns a degree of membership into the category to a set of continuous values. To apply fuzzy logic to our problem, we assume that attributes are tied to an underlying continuous space. This space is then discretized according to the number and types of labels that users will associate with the attribute. We can therefore associate a membership function with each label for a given attribute. For example, the volume attribute may have the labels "big", "medium" and "small" associated to it. An object with a given volume measurement will have a different score for each of the three labels. This allows us to assign a score to an object for each qualifier mentioned in the user's description. These objects, or referent candidates, can be ordered according to the combination of the degrees of membership or possibility values. A

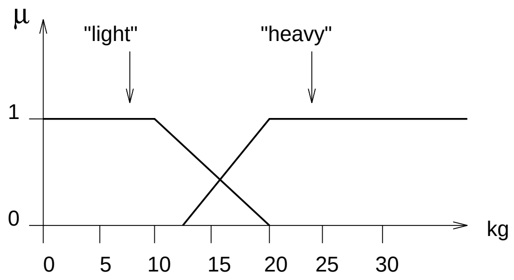


Figure 1: Fuzzy membership functions for the attribute *weight*

single object will be selected as the referent provided that it is the only candidate satisfying a decision criterion.

Figure 1 depicts two sample fuzzy membership functions for the weight attribute which are associated with the linguistic labels "light" and "heavy". A weight measurement (in kg's) would be assigned a degree of membership into both label categories. The measurement would be mapped to the label assigning the highest fuzzy value. As the goal of using fuzzy logic is to emulate human decision patterns, we propose to design the membership functions by training them with data collected from a group of human subjects. Thus, for a given label, a possibility of 1 could be assigned to all values for which a group of "experts" agree in assigning the label, a possibility of 0 can be assigned to values for which there is a unanimous consensus for *not* assigning the label, and a reasonable interpolation can be performed in between [DM83].

In our prototype implementation we use the attribute classes of size, shape, colour, height and material. Each of these is associated to a list of qualifiers that the user may choose to describe world objects. For example, the shape attribute is linked to the terms "round", "square" and "cylindrical" among others. The types of objects in our model environment are those that may typically be found in an office. Such items include: chairs, desks, boxes and books. To interact with the environment, our system is interfaced with a mobile robot controller/simulator that includes an RWI B-12 robot using video and sonar sensing [DJ93].

3. TRADE-OFF BETWEEN QUERIES AND SENSING

To identify a referent, the system parses the referring expression and uses its knowledge of the environment to rate each object with respect to its possibility of fitting the description. Difficulties arise when the system is faced with incomplete environment data preventing the scoring of each potential candidate for every qualifier in the referring expression. In order to compare all candidates, they must be rated with respect to the same attributes. The system is faced with three alternatives: 1) select the best candidate based on existing knowledge; 2) question the user about properties that the system knows about but were not mentioned in the original referring expression; 3) use sensors to collect missing data to evaluate the candidates more extensively and thus make a more informed selection. Of course, there is a trade-off between these alternatives. In the first case, the choice is made rapidly at the expense of accuracy. That is, there is a greater uncertainty about the correctness of the chosen candidate. In the second case, we may improve accuracy at the cost of interacting with (or disturbing) the user. Finally, the third case enables the system to make the choice which is most faithful to the description provided by the user, while increasing the knowledge base in the STM, at the cost of time and resource expenditure. In general, the correct choice depends on the ease or advisability of interacting with the user as compared with the expected cost and benefit of operating autonomously. In the work reported here, we select the "user interaction cost", whereas the cost for autonomous operation is based on an estimate of the effort required to perform additional sensing:

Let α be the user interaction cost.
Let β be the autonomous operation cost,
where β is defined as:

$$\sum_{i \in \text{Candidates}} \mathit{path\_length}(i, i+1) + \mathit{sensing\_effort}(i)$$

If $\alpha < \beta$ then the system should query the user
for more information.
Else the system should collect more sensory data.

By changing the value of the first cost, we can vary the system's response and compare the accuracy of the selection as well as the time required to make the selection. Note that the methods mentioned here are used only when the system cannot uniquely identify the referent with current knowledge. Thus, measures must be taken to ensure "understanding". In dialogue theory, this type of repair is typically carried out via user questioning. We extend the technique here to include autonomous behaviour (i.e.: sensing) other than queries.

4. REQUESTING MORE INFORMATION

The system may opt to query the user for more information either as a result of the cost comparison outcome, or in order to obtain a greater discrimination between the candidates so that a single object stands out as the only possible referent. The queries consist of asking the user to provide a more extensive description so that a re-scoring of the candidates will result in a larger gap between the best score and the remaining ones. In questioning the user, the system must attempt to prune the search space as efficiently as possible which will limit the amount of user interaction required to obtain more data. This is a direct result of soliciting information which will most widely discriminate between the candidates. This, in turn, amounts to selecting a question about the attribute having the greatest variation of values represented among the candidates. This variation can be uncovered by computing the entropy for each attribute over all objects in the environment. The probabilities used in the entropy equation are based on the membership values for each qualifier associated to a given attribute. To illustrate, consider the *size* attribute which may be associated to the three qualifiers "small", "medium" and "large". The entropy value for this attribute would be computed as follows:

$$Entropy(size) = -\mu_{small}\log(\mu_{small}) - \mu_{medium}\log(\mu_{medium}) - \mu_{large}\log(\mu_{large})$$

where:

$$\mu_{small} = \frac{\hat{\mu}_{small}}{\hat{\mu}_{small} + \hat{\mu}_{medium} + \hat{\mu}_{large}}$$

$$\hat{\mu}_{small} = \sum_{c \in \text{Candidates}} fms(c) \ for \ \text{"}small\text{"}$$

where *fms* is a *fuzzy membership score.*

In the above equations, $\hat{\mu}_{small}$ represents the total degree to which the candidate objects can be referred to as "small", and μ_{small} reflects the proportion of "small" objects among all candidates.

After computing the entropy for every attribute, the one having the highest value will be the one for which the user will be requested to supply the referent's value. This calculation and questioning continues until a single candidate is reliably selected by the decision criterion. This unique candidate is then presented to the user for verification.

5. VERIFICATION AND CORRECTION

Once the system has made its selection, the chosen candidate must be subjected to an acceptance procedure. This procedure has two outcomes: either the user acknowledges that the chosen candidate is in fact the referent, or else rejects it, in which case both system and user must collaborate in the system's identification of the referent. Typically, when the user rejects the system's selection, he/she will be expected to indicate the discrepancy between the selection and the referent by stressing the attribute values that discriminate between both [CWG86]. Using this new data and those gathered during the search process, the system will attempt to correct the error and correctly identify the referent.

6. SYSDEM DESIGN

Figure 2 illustrates the architecture of our prototype system. The arrows indicate the flow of information between the constituent parts. Each module is responsible for an independent subtask in the referent identification process. At the heart of the system lies the Dialogue Manager which coordinates the data exchanges between the components as well as those between the system and user. The Short Term Memory (STM) and Knowledge Base (KB) contain task-relevant information, environment object representations and general world knowledge. These components are updated through the collection of sensory data as well as

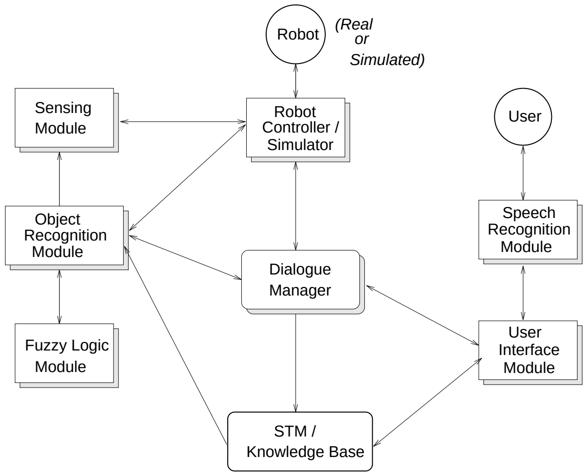


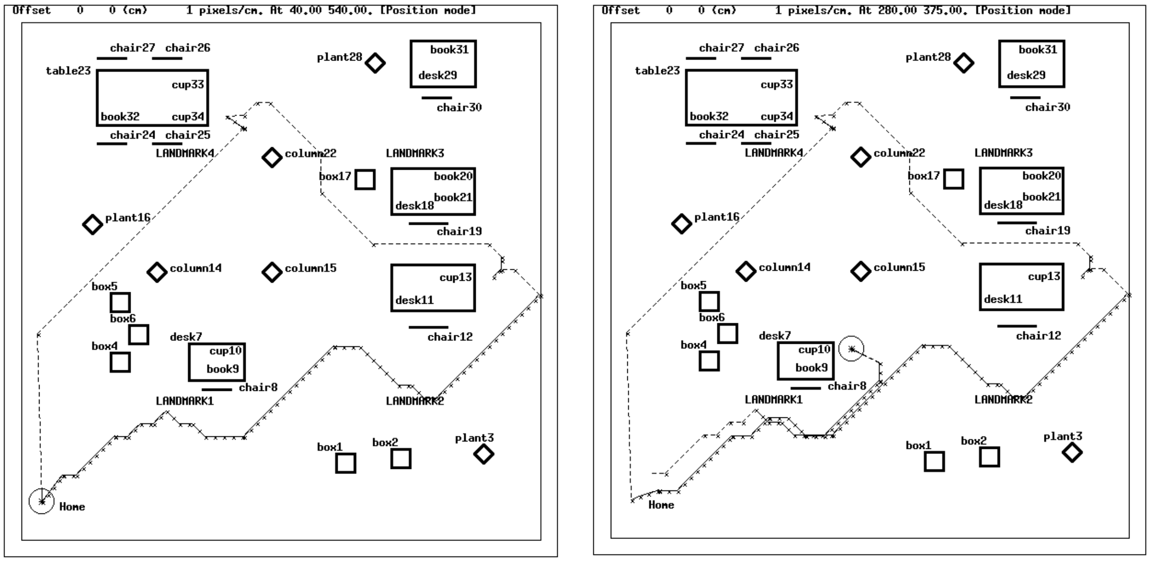
Figure 2: System Architecture

user-proived information. The Speech Recognition Module (SRM) [DMKGS95] receives the spoken command as input and derives sequences of words $W_1, W_2, \ldots, W_n$ taken from the system's vocabulary, V while assigning each sequence a probability of having been produced by the acoustic signal. The word string with the highest probability is selected and communicated to the User Interface Module (UIM). The UIM uses an expert system tool [Cul89] to extract the keywords from the SRM output which include the task name, the target object type and the object qualifiers (e.g., "Please *get* the *big, blue box*"). These keywords are then provided to the Object Recognition Module (ORM) which uses the information in the STM and KB to select the environment object which best matches the descriptive information conveyed by the keywords. To accomplish this task, the ORM uses data provided by the Fuzzy Logic Module (FLM) and the Sensing Module (SM). The FLM is responsible for the mapping between natural language descriptors and sensory data. It also calculates the attribute entropies as these are based on fuzzy membership values as explained in Section 4. Finally, the SM manages the system sensors and determines which objects adn their corresponding attribute values that must be measured.

7. EXPERIMENTAL RESULTS

We now briefly demonstrate the working model of our system. The robot controller graphical interface depicting a sample office environment is shown in Figure 3. The Short Term Memory contains partial object representations that include the object's type (e.g., "box", "chair"), quantitative attribute values and spatial coordinates[1]. In addition, the STM contains a list of landmarks in the environment which are known to both user and system. These provide objective points of reference to which both agents can refer without the need for descriptive terms. The landmarks are useful to disambiguate descriptions which may plausibly apply to more than one environment object. Thus,

[1]Note that the label numbers in Figure 3 are solely for the benefit of the reader.



(a) Visiting and Sensing (b) After Identification of cup10

Figure 3: "Get the white paper cup" *(referent = cup10)*

rather than asking the user to supply the referent's value for a given attribute, the system may ask about the referent's proximity to a specific landmark. The distances are mapped to labels such as "near" and "far" and are included in the entropy calculation explained in Section 4.

Figure 3 represents an interaction in which the user has uttered the command: "Get the white paper cup" in reference to *cup10* on the diagram. The system identifies 4 candidates and rates them according to the information provided in the command. However, since the STM is lacking in the values for 2 of the candidates, it cannot rate them with respect to all attribute values – only those known for *all* candidates. Thus, the system resorts to either asking the user to supply other information that is known to the system for *all* candidates, or using sensors to collect missing data with which to obtain a more description-faithful scoring of the candidates. In the depicted case, sensing was rated more cost-effective than asking. Part(a) of the figure shows the path followed by the robot to collect the missing values for *cup13* and *cup34*. Part(b) depicts the robot in its new location after the correct referent identification.

8. REFERENCES

[BB82] D. Ballard and C. Brown. *Computer Vision.* Prentice Hall, Englewood Cliffs, N.J., 1982.

[Bro85] R. A. Brooks. *Visual Map Making for a Mobile Robot.* IEEE Computer Society, 1985.

[Cul89] Chris Culbert. *CLIPS Reference Manual.* A. I. Section, L. B. J. Space Center, July 1989.

[CWG86] H. H. Clark and D. Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986. also in [CMP90], pp. 463-493.

[DJ93] G. Dudek and M. Jenkin. A multi-layer distributed development environment for mobile robotics. In *Proc. of the International Conference on Intelligent Autonomous Systems (IAS-3)*, pages 542–550, Pittsburgh, PA, 1993.

[DM83] Renato De Mori. *Computer Models of Speech Using Fuzzy Algorithms.* Plemun Press, New York, 1983.

[DMKGS95] R. De Mori, R. Kuhn, M. Galler, and C. Snow. Speech recognition and understanding. In J. Liebowitz and D. S. Prerau, editors, *Worldwide Intelligent Systems*, chapter 8, pages 125–162. IOS Press, 1995.

[KM94] R. Kuhn and R. De Mori. The application of classification trees to natural language understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994. (to appear).

[Kro87] E. Krotkov. Focusing. *Internatonal Journal of Computer Vision*, 1:223–237, 1987.

[MD94] P. MacKenzie and G. Dudek. Precise positioning using model-based maps. In *Proc. of the International Conference on Robotics and Automation*, San Diego, CA, 1994. IEEE Press.

[Zad73] L. A. Zadeh. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on Systems, Man and Cybernetics*, SMC-3, no. 1:28–44, 1973.