FAILURES IN NATURAL LANGUAGE SYSTEMS: APPLICATIONS TO DATA BASE QUERY SYSTEMS

Eric Mays
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104

ABSTRACT

A significant class of failures in interactions with data base query systems are attributable to misconceptions or incomplete knowledge regarding the domain of discourse on the part of the user. This paper describes several types of user failures, namely, intensional failures of presumptions. These failures are distinguished from extensional failures of presumptions since they are dependent on the structure rather than the contents of the data base. A knowledge representation has been developed for the recognition of intensional failures that are due to the assumption of non-existent relationships between entities. Several other intensional failures which depend on more sophisticated knowledge representations are also discussed. Appropriate forms of corrective behavior are outlined which would enable the user to formulate queries directed to the solution of his/her particular task and compatible with the knowledge organization.

I INTRODUCTION

An important aspect of natural language interaction with intelligent systems is the ability to deal constructively with failure. Failures can be viewed as being of two types. One can be ascribed to a lack of syntactic, semantic, or pragmatic coverage by the system. This will be termed system failure, and manifests itself in the inability of the system to assign an interpretation to the user's input. Recent work has been done in responding to these types of failures, see for example, Weischedel and Black [8], and Kwasny and Sondheimer [3]. A second class of failures may be termed user failures. A user failure arises when his/her beliefs about the domain of discourse diverge from those of the system.**

To avoid confusion, a clear distinction should be made between failures and errors. An error occurs when the system's response to an input is incorrect. Errors generally manifest themselves as incorrect resolution of ambiguities in word sense or modifier placement. These errors would usually be detected by the user when presented with a paraphrase that differs in a meaningful way from the original input [6]. More serious errors result from incorrect coding of domain knowledge, and are often undetectable by the user.

This paper concerns itself with the recognition and correction of user failures in natural language data base query systems -- in particular, failures that arise from the user's beliefs about the structure, rather than the content, of the data base. The data base model that has been implemented for the recognition and correction of simple user failures about the data base structure is presented. Several other failures which depend on more sophisticated knowledge representation are also discussed.

II PRESUPPOSITION AND PRESUMPTION

The linguistic notion of presupposition provides a formal basis for the inference of a significant class of user beliefs. There is a less restrictive notion, presumption, which allows the inference of larger class of user beliefs, namely, that knowledge which the user must assume when posing a question.

A presupposition is a proposition that is entailed by all the direct answers of a question.*** A presumption is either a presupposition or it is a proposition that is entailed by all but one of the direct answers of a question [2]. Hence, presupposition is a stronger version of presumption, and a presupposition is a presumption by definition. For example, question (1a) has several direct answers such as, "John", "Sue", etc., and, of course, "no one". Proposition (1b) is entailed by all the direct answers to (1a) except the last one, i.e., "no

---

*This work is partially supported by a grant from the National Science Foundation, NSF-MCS 79-08401.

**Some user beliefs regarding the domain of discourse are implicitly encoded in questions posed to the system. The beliefs held by the system are explicit in its knowledge representation, either procedurally or declaratively.

***The complete definition of presupposition includes the condition that the negation of a question, direct answer pair entails the presupposition.

one". Therefore, (1b) is a presumption of (1a). Proposition (1d) is a presupposition of (1c), since it is entailed by all of the question's direct answers.

(1a) Which faculty members teach CSE110?
(1b) Faculty members teach CSE110.
(1c) When does John take CSE110?
(1d) John takes CSE110.

Presumptions can be classified on the basis of what is asserted -- i.e., an "intensional" statement about the structure of the data base or an "extensional" statement about its contents. Thus an extensional failure of a presumption occurs based on the current contents of the data base, while an intensional failure occurs based on the structure or organization. For example, question (2a) presumes propositions (2b), (2c), and (2d). Presumption (2b) is subject to intensional failure if the data base does not allow for the relation "teach" to hold between "faculty" and "course" entities. An extensional failure of presumption (2b) would occur if the data base did not contain any "faculty member" that "teaches" a "course". Also note that the truth of (2b) is a pre-condition for the truth of (2c).

(2a) Which faculty members teach CSE110?
(2b) Faculty members teach courses.
(2c) Faculty members teach CSE110.
(2d) CSE110 is a course.

Although a presumption which fails intensionally will of necessity fail extensionally, it is important to differentiate between them, since an intensional failure that occurs will occur consistently for a given data base structure, whereas extensional failure is a transitory function of the current contents of the data base. This is not meant to imply that a data base structure is not subject to change. However, such a change usually represents a fundamental modification of the organization of the enterprise that is modelled. One can observe that structural modifications occur over long periods of time (many months to years, for example), while the data base contents are subject to change over relatively shorter periods of time (hourly, daily, or monthly, for example).

Kaplan [2] has investigated the computation and correction of extensional failures of presumptions. The approach taken there involves accessing the contents of the data base to determine if a presumption has a non-empty extension. The remainder of this paper discusses several ways a presumption might be subject to intensional failure. These inferences are made from the structural information of the data base.

III DATA BASE MODEL

In order to recognize failures of presumptions concerning the structure of the data base, it is necessary to use a robust data model. The discussion here will assume a data base model similar to that proposed by Lee and Gerritsen [4], which incorporates the generalization dimension developed by Smith and Smith [7] into the entity-relationship model [1]. Basically, entities participate in relationships along two orthogonal dimensions, aggregation (among dissimilar entities) and generalization (among similar entities), as well as having attributes that assume values. As an example of this type of structure consider the data base model fragment for a typical university in figure 1. Entity sets are designated by ovals, aggregation relationships by diamonds, and generalization relationships by edges from the super-entity set to the sub-entity set. The parallel arcs denote mutual exclusion.

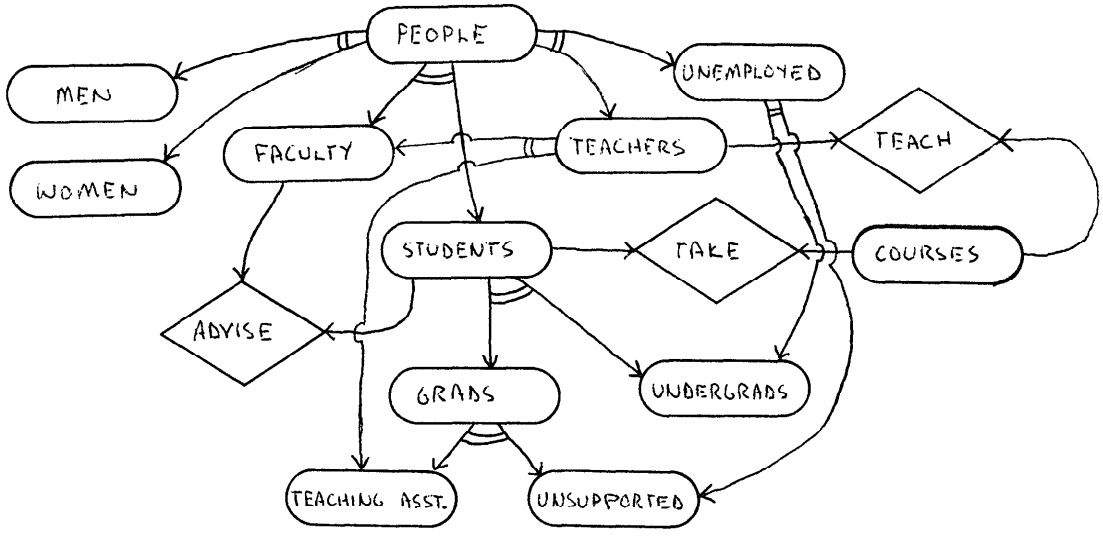

FIGURE 1

Mutual exclusion is used to infer the difference between "men that are also faculty" (a possibly non-empty set) and "men that are also women" (an empty set by definition), for example given figure 1. This distinction can be made by prohibiting the traversal of a path in the data model that includes two entity sets which are mutually exclusive. Furthermore, the path in the generalization dimension is restricted to "upward" traversals followed by "downward" traversals. An upward (downward) traversal is from a sub-entity (super-entity) set to a super-entity (sub-entity) set. This restriction is made to prevent over-specialization of an entity set when traversing downward edges. The set of inferences that can be made in the presence of this restriction is not overly constrained, since any two entity sets that have a common intersection (sub-entity set) will also have a common union (super-entity set).*

IV INTENSIONAL FAILURES

A. Non-existent Relationships

The most basic intensional failure that can occur is the presumption of a non-existent relationship between entity sets. In the university data base model fragment given above, such a failure occurs in the question "Which faculty take courses?". This question presumes that a "take" relationship could exist between "faculty" and "courses" entities. Since no such relationship can be established, that presumption has failed intensionally. Recognizing the failure is only part of the problem -- it is also useful to provide the user with related intensional knowledge. Given a relation R, entities X and Y, and a failed presumption (R X Y), salient intensional knowledge can be found by abstracting on R, X, or Y to create a new relation. For example, consider the following exchange:

Q: "Which faculty take courses?"
A: "I don't believe that faculty can take courses.
Faculty teach courses.
Students take courses."

A similar failure occurs in the presumption of a non-existent attribute of an entity set. For example, "What is the cost of all courses taught by teaching assistants?", incorrectly presumes that in this data base, "cost" is an attribute of "courses".

B. Inapplicable Functions

Intensional failures may also occur when attempting to apply a function on a domain. The question, "What is the average grade in CSE110?", will cause no processing problems provided grades are assigned over the real numbers. But if grades ranged from A to F, then the system should inform the user that averages can not be performed on character data. (Note that the clever system designer might trap this case and make numerical assignments to the letter grades.) A more significant aspect is the notion of a function to be meaningful over a particular domain. That is, certain operations, even though they might be applicable, may not be meaningful. An example would be "average social security number". The user who requested such a computation does not really understand what the data is supposed to represent. In such cases a short explanation regarding the function of the data would be appropriate. To achieve this type of behavior, of course, the data base model must be augmented to include type and functional information.

C. Higher Order Failures

The mutual exclusion operator allows the detection of a failure when the question specifies a restriction of an entity set by any two of its mutually exclusive sub-entity sets. For example, "Which teachers that advise students take courses?" presumes that there could be some "teachers" that are both "faculty" and "students". Since this situation could never arise, given the structure in figure 1, it should be communicated to the user as an intensional failure. If an exhaustiveness operator is incorporated as well, unnecessary restrictions of an entity set by disjunction of all of its exhaustive sub-entity sets can be detected. Although this would not constitute a failure, it does indicate that there is some misconception regarding the structure of the data base on the part of the user. If the sub-entity sets were known to be exhaustive by the user, there would be no reason to make the restriction. As an example, the addition of the fact that "grads" and "undergrads" were exhaustive sub-entity sets of "students" would cause this misconception to arise in the question "Which students are either grads or undergrads?". The following behavior would be desired in these cases:

Q: "Which teachers that advise students take courses?"
A: "Faculty advise students.
Students take courses.
I don't believe that a teacher can be both a faculty member and a student."

---

*See [5] for a more detailed description.

D. Data Currency

Some failures depend on the currency of the data. One such example occurs in a naval data base about ships, subs, and aircraft. The question "What is the position of the Kitty Hawk?" presumes that timely data is maintained. Actually, positions of friendly vessels are current, while those of enemy ships might be hopelessly out of date. In this case, the failures would be extensional since the last update of the attribute must be checked for currency. It may be the case that some data is current while other data is not. However, the update processing time lag from actual event occurence to capture in the data base might be sufficiently long that such presumptions might be subject to intensional failure. Thus the user could be made aware that current data was never available.

V CONCLUSION

In this paper we have discussed several kinds of failures of presumptions that depend on knowledge about the structure or organization of the data base. It is important to distinguish between structure and content, since there is a significant difference in the rate at which they change. When responding to intensional failures of presumptions, simply pointing out the failure is in most cases inadequate. The user must also be informed with regard to related knowledge about the structure of the data base in order to formulate queries directed at solving his/her particular problem. The technique described for recognizing intensional failures that are due to the presumption of non-existent relationships between entities and attributes of entities has been implemented. Further work is aimed at developing knowledge representations for temporal and functional information. We hope to eventually develop a general account of user failures in natural language query systems.

ACKNOWLEDGEMENTS

I would like to thank Peter Buneman, Aravind Joshi, Kathy McKeown, and Bonnie Webber for their valuable comments on an earlier draft of this paper.

REFERENCES

[1] Chen, P.P.S., "The Entity-Relationship Model -- Towards a Unified View of Data", ACM Transactions on Database Systems, Vol. 1, No. 1, March 1976.

[2] Kaplan, S.J., Cooperative Responses From a Portable Natural Language Data Base Query System, Ph.D. Dissertation, Computer and Information Science Department, University of Pennsylvania, Philadelphia, PA., 1979.

[3] Kwasny, S.C., and Sondheimer, N.K., "Ungrammaticality and Extra-Grammaticality in Natural Language Understanding Systems", Proceedings of the Conference of the Association for Computational Linguistics, La Jolla, CA., August 1979.

[4] Lee, R.M. and Gerritsen, R., "A Hybrid Representation for Database Semantics", Working Paper 78-01-01, Decision Sciences Department, University of Pennsylvania, 1978.

[5] Mays, E., "Correcting Misconceptions About Data Base Structure", Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence, Victoria, British Columbia, Canada, May 1980.

[6] McKeown, K., "Paraphrasing Using Given and New Information in a Question-Answer System", Proceedings of the Conference of the Association for Computational Linguistics, La Jolla, CA., August 1979.

[7] Smith, J.M. and Smith, D.C.P., "Database Abstractions: Aggregation and Generalization", ACM Transactions on Database Systems, Vol. 2, No. 2, June 1977.

[8] Weischedel, R.M., and Black, J., "Responding Intelligently to Unparseable Sentences", American Journal of Computational Linguistics, Vol. 6, No. 2, April-June 1980.