Semantic Specificity in Spoken Dialogue Requests

Ben Hixon
Hunter College of The
City University of New York
New York, NY, USA
shixon@hunter.cuny.edu

Rebecca J. Passonneau
Columbia University
New York, NY, USA
becky@cs.columbia.edu

Susan L. Epstein
Hunter College, and
The Graduate Center of The
City University of New York
New York, NY, USA
susan.epstein@hunter.cuny.edu

Abstract

Ambiguous or open-ended requests to a dialogue system result in more complex dialogues. We present a semantic-specificity metric to gauge this complexity for dialogue systems that access a relational database. An experiment where a simulated user makes requests to a dialogue system shows that semantic specificity correlates with dialogue length.

1 Introduction

A dialogue system (DS) and its users have asymmetric knowledge. The DS has access to knowledge the user is not privy to, and the user has intentions that the DS attempts to recognize. When the user's intentions are difficult for her to specify fully, the user and DS must collaborate to formulate the intention. The thesis of this work is that a DS can assess the specificity of its knowledge with respect to the user intentions it is designed to address. Our principal result is that, for a DS that queries a relational database, measures of the ambiguity of database attributes can be used both to assess the scope of the DS's task and to guide its dialogue strategy. To demonstrate our thesis, we have developed a *semantic specificity* metric applicable to any DS that queries a relational database. This metric measures the degree to which one or more attributes can uniquely specify an item in the database. Attributes whose values are more often ambiguous have lower semantic specificity.

CheckItOut is a book request DS that references a copy of the catalogue at the Heiskell Braille and Talking Book Library with its 71,166 books (Epstein et al., In Press). We focus on three book attributes: AUTHOR, TITLE and CALL NUMBER. Only the latter is guaranteed to identify a unique book. Of the 64,907 distinct TITLE values, a large majority return a unique book (N=59,236; 91.3%). Of the 28,045 distinct AUTHOR values, about two thirds return a unique book (N=17,980; 64.1%).

Query return size	Distinct TITLE values	Distinct AUTHOR values
1	59236	17980
2	5234	4377
3	345	1771
...		
10	2	168
...		
184	–	1
Total	64907	28045

Table 1: When used as a query, many TITLE values return unique books, but AUTHOR values are less specific.

To compare the specificity of TITLE and AUTHOR, we calculated *query return size*, the number of distinct books in the Heiskell database returned by each possible attribute value. Table 1 tallies how many attribute values have the same query return size. TITLE partitions the books into 10 subsets, where the two most ambiguous TITLE values, *Collected Stories* and *Sanctuary*, each return 10 distinct books. AUTHOR produces 89 subsets; its most ambiguous value, Louis L'Amour, returns 184 distinct books. Clearly, TITLE has higher specificity than AUTHOR.

After a survey of related work, this paper defines a semantic specificity metric that is a weighted sum of

Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL), pages 257–260,
Seoul, South Korea, 5-6 July 2012. ©2012 Association for Computational Linguistics

the number of query return sizes for one or more attributes. We show through simulation that dialogue length varies with semantic specificity for a DS with a simple system-initiative dialogue strategy.

2 Related Work

Little work has been reported on measures of the relationship between dialogue complexity and the semantic structure of a DS application's database. Zadrozny (1995) proposes Q-Complexity, which roughly corresponds to vocabulary size, and is essentially the number of questions that can be asked about a database. Pollard and Bierman (2000) describe a similar measure that considers the number of bits required to distinguish every object, attribute, and relationship in the semantic space.

Gorin et al. (2000) distinguish between semantic and linguistic complexity of calls to a spoken DS. Semantic complexity is measured by inheritance relations between call types, the number of type labels per call, and how often calls are routed to human agents. Linguistic complexity is measured by utterance length, vocabulary size and perplexity.

Popescu et al. (2003) identify a class of "semantically tractable" natural language questions that can be mapped to an SQL query to return the question's unique correct answer. Ambiguous questions with multiple correct answers are not considered semantically tractable. Polifroni and Walker (2008) address how to present informative options to users who are exploring a database, for example, to choose a restaurant. When a query returns many options, their system summarizes the return using attribute value pairs shared by many of the members.

3 Semantic Specificity

The database queried by a DS can be regarded as the system's knowledge. Consequently, the semantic structure of the database and the way it is populated constrain the requests the system can address and how much information the user must provide. Intuitively, Table 1 shows that TITLE has a higher semantic specificity than AUTHOR. Our goal is to quantify the query ambiguity engendered by the instantiation of any database table.

Often a user does not know in advance which combination of attribute values uniquely communicates her intent to the system. In addition, the DS does not know what the user wants until it has offered an item that the user confirms, whether explicitly or implicitly. The remainder of this section defines the specificity of individual and multiple attributes with respect to a set of database instances.

3.1 Specificity for Single Attributes

When a user requests information about one or more entities, the request can map to many more database instances than intended. Let I be a set of instances (rows) in a database relation, and let α be an attribute of I with values V that occur in I. Denote by $q(v, \alpha)$ the *query return size* for $v \in V$ on α, the number of instances of I returned by the query $\alpha = v$. Whenever $q(v, \alpha) = 1$, the query returns exactly one instance in I; attributes with more such values have higher specificity. If $q(v, \alpha) = 1$ for every v, then α is *maximally specific* with respect to I.

Let Q_α be the set of d_α distinct query return sizes $q(v, \alpha)$ returned on I. We call Q_α the *query return size partition* for α. Q_α induces a partition of V into subsets $V_j, j \in Q_\alpha$ such that a query on every value in a given subset returns the same number of instances. Table 1 shows two such partitions. We now define the specificity $S(\alpha, I)$ of attribute α with respect to I as a weighted sum of the sizes of the subsets in the partition induced by α, normalized by $|I|$, the number of instances in I:

$$S(\alpha, I) = \frac{1}{|I|} \sum_{j \in Q_\alpha} w(j) \cdot |V_j| \qquad (1)$$

The weight function w in (1) addresses the number of distinct values in each subset of Q_α. A larger query return size indicates a more ambiguous attribute, one less able to distinguish among instances in I. To produce specificity values in the range $[0, 1]$, $w(j)$ should decrease as j increases, but not penalize any query that returns a single instance, that is, $w(1) = 1$. The faster w decreases, the more it penalizes an ambiguous attribute. Here we take as w the inverse of the query return size, $w(j) = \frac{1}{j}$.

For our CheckItOut example, equation (1) scores TITLE's specificity as 0.871 and AUTHOR's specificity much lower, at 0.300. This matches our intuition. The third attribute with which a user can order a book, CALL NUMBER, was designed as a primary key and so has a perfect specificity of 1.000.

3.2 Specificity for Multiple Attributes

The specificity of a set $\beta = \{\alpha_1, \alpha_2, ..., \alpha_k\}$ of k attributes on a set of instances I measures to what degree a *combination* (one value for each attribute in β) specifies a restricted set of instances in I. Let V be the combinations for β that occur in I, and let $q(v, \beta)$ be the query return size for $v \in V$. Then Q_β, the set of d_β distinct query return sizes, induces a partition on V into subsets $V_j, j \in Q_\beta$ where combinations in the same subset return the same number of instances. We take $w(j,k) = \frac{1}{j^k}$ to penalize ambiguity more heavily when there are more attributes. Then the specificity of β with respect to I is

$$S(\beta, I) = \frac{1}{|I|} \sum_{j \in Q_\beta} w(j,k) \cdot |V_j| \qquad (2)$$

Using this equation, the specificity of $\beta = \{\text{TITLE}, \text{AUTHOR}\}$ is 0.880. Interestingly, this is not much higher than the 0.871 TITLE specificity alone, which indicates that, in this particular database instantiation, AUTHOR has little ability to disambiguate a TITLE query. This is because many "books" in the Heiskell catalog appear in two formats, Braille and audio. This duplication creates an ambiguity that is better resolved by prompting the user for CALL NUMBER or FORMAT. In some cases, a value for FORMAT might still result in ambiguity; for example, different recorded readers produce different audio versions of the same title and author. In contrast, the large difference between AUTHOR's very low specificity (0.300) and that of $\{\text{TITLE}, \text{AUTHOR}\}$ (0.880) suggests that, given an ambiguous author, it would in general be a good strategy for the DS to then prompt the user for the title.

Because specificity is a function of a database instantiation, specificity can be used to guide dialogue strategy. For the books in Heiskell's catalogue that cannot be uniquely identified by AUTHOR and TITLE alone, it can be determined *a priori* that some book requests cannot be disambiguated without additional attribute values.

4 Specificity in Simulated Dialogues

A DS faced with an ambiguous query should enter a disambiguation subdialogue. It can either offer a list of matching instances that must be individually confirmed or rejected by the user, or indicate to the user that an attribute value is ambiguous and request additional information. In general, a less specific user request should increase the dialogue length.

We tested this hypothesis with a simulated user that interacted with the CheckItOut book-order DS through a text interface. As in our work where human subjects called CheckItOut, a new scenario for each call included "patron" identification data and a list of four books with their authors, titles, and call numbers. In each dialogue, the simulator received a scenario, the DS confirmed the simulator's "identity," and then the simulator requested the books.

The simulator uses a rule-based dialogue strategy with some parameters, such as persistence in re-ordering the same book, that can be set or randomized to represent a set of real users more realistically. For this experiment, the simulator was set to request books only by author, and to be willing to engage in a disambiguation subdialogue of arbitrary length. When the system cannot uniquely identify a requested book, it begins a disambiguation subdialogue, an example of which is shown in Figure 1. To avoid addressing information presentation issues such as those explored in (Polifroni and Walker, 2008), CheckItOut followed a simple strategy of offering each next candidate book in a query return, and user studies with CheckItOut restricted query return size to a maximum of three books. For the simulations, we expect an inverse relationship between specificity and dialogue length.

S: NEXT BOOK?
U: Dan Simmons
S: DID YOU SAY DAN SIMMONS?
U: Yes
S: WOULD YOU LIKE ENDYMION BY THIS AUTHOR?
U: No
S: WOULD YOU LIKE THE CROOK FACTORY
BY THIS AUTHOR?
U: Yes

Figure 1: Sample disambiguation subdialogue.

We randomly sampled Heiskell's full set of 71,166 books to create five smaller instantiations of 1000 books each. We deliberately sampled at different frequencies within each subset of the original partition induced by AUTHOR, so that $S(\text{AUTHOR}, T)$ for instantiation T ranged from 0.3528 to 1.000. For each instantiation we simulated 25 dialogues. Conditions of relatively lower speci-

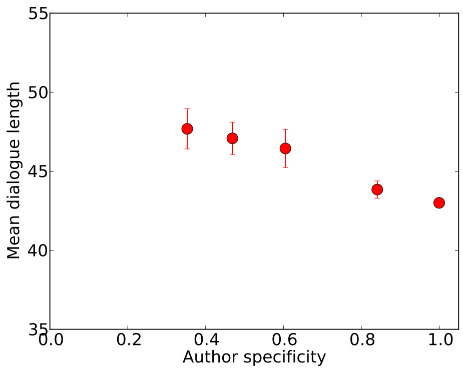

Figure 2: Dialogue length averaged across 25 simulated dialogues for each run of 5 different attribute specificity levels, shown with 95% confidence intervals.

ficity result in more dialogues like the one shown in Figure 1, with multiple turn exchanges where the DS offers the simulator different books by the requested author. As specificity approaches 1.0, the first book offered by the DS is more frequently the requested book, so no disambiguation is required, and the minimum dialogue length of 43 turns is achieved. Figure 2 compares the mean dialogue length for each sub-instantiation to its author specificity, and clearly shows that dialogue length increases as author specificity decreases. The error bars shrink as specificity increases because there is less variation in dialogue length when there are fewer candidate books for CheckItOut to offer.

5 Conclusion and Future Work

Semantic specificity has two important applications. Because it predicts how likely a value for a database attribute (or a combination for a set of attributes) is to return a single database instance, semantic specificity can help formulate subdialogues with a priority order in which the DS should prompt users for attributes. Because it is a predictor for dialogue length, semantic specificity could also be used to evaluate whether a DS dialogue strategy incurs the expected costs. Of course, many factors other than semantic specificity affect DS dialogue complexity, particularly the relation between users' utterances and the semantics of the database. In the examples given here, the way users refer to books corresponds directly to attribute values in the database. Other domains may require a more complex procedure to map between the semantics of the database and the semantics of natural language expressions.

Finally, how well semantic specificity with respect to a database instantiation predicts dialogue length depends in part on how closely the database attributes correspond to information that users can readily provide. Here, AUTHOR and TITLE are convenient both for users and for the database semantics. However, the maximally specific CALL NUMBER is often unknown to the user. For DSs where the database attributes differ from those that can be extracted from user utterances, we intend to explore enhanced or additional metrics to predict dialogue length and guide dialogue strategy.

Acknowledgments

We thank Julia Hirschberg for helpful comments, and Eric Schneider for help with the user simulator. National Science Foundation awards IIS-0745369, IIS-0744904 and IIS-084966 funded this project.

References

Susan L. Epstein, Rebecca J. Passonneau, Tiziana Ligorio, and Joshua Gordon. In Press. Data mining to support human-machine dialogue for autonomous agents. In *Proceedings of Agents and Data Mining Interaction (ADMI 2011)*. Springer-Verlag.

A. L. Gorin, J. H. Wright, G. Riccardi, A. Abella, and T. Alonso. 2000. Semantic information processing of spoken language. In *Proceedings of ATR Workshop on MultiLingual Speech Communication*, pages 13–16.

Joseph Polifroni and Marilyn Walker. 2008. Intensional summaries as cooperative responses in dialogue: Automation and evaluation. In *Proceedings of ACL-08: HLT*, pages 479–487, Columbus, Ohio, June. Association for Computational Linguistics.

Shannon Pollard and Alan W. Bierman. 2000. A measure of semantic complexity for natural language systems. In *NAACL-ANLP 2000 Workshop: Syntactic and Semantic Complexity in Natural Language Processing Systems*, pages 42–46, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ana-Maria Popescu, Oren Etzioni, and Henry Kautz. 2003. Towards a theory of natural language interfaces to databases. In *Proceedings of the 8th international conference on Intelligent user interfaces*, IUI '03, pages 149–157, New York, NY, USA. ACM.

Wlodek Zadrozny. 1995. Measuring semantic complexity. In Moshe Koppel, Eli Shamir, and Martin Golumbic, editors, *Proceedings of the Fourth Bar Ilan Symposium on Foundations of Artificial Intelligence (BISFAI 1995)*, pages 245–254.