

A Novel Approach Towards Incorporating Context Processing Capabilities in NLIDB System

Arjun R. Akula, Rajeev Sangal, Radhika Mamidi

Language Technologies Research Center,
IIT Hyderabad, India.

arjunreddy.akula@research.iiit.ac.in,
{sangal, radhika.mamidi}@iiit.ac.in

Abstract

This paper presents a novel approach to categorize, model and identify contextual information in natural language interface to database (NLIDB) systems. The interactions between user and system are categorized and modeled based on the way in which the contextual information is utilized in the interactions. A relationship schema among the responses (user and system responses) is proposed. We present a novel method to identify contextual information in one specific type of user-system interaction. We report on results of experiments with the university related queries.

1 Introduction

Natural Language Interface to Database (NLIDB) systems allow the users to query databases in a natural language (Androustopoulos et al., 1995; Meng and Wang, 2001; Popescu et al., 2003; Stratica et al., 2005; Li et al., 2005; Giordani, 2008; Giordani and Moschitti, 2009; Gupta et al., 2012). Although NLIDB systems are able to answer a wide range of natural language queries (NL queries), they are not used much in commercial applications. One of the main reasons for the less acceptance of these systems in real-time applications is that they lack robust context processing capabilities (Bertomeu et al., 2006). Currently there is very little work which explicitly aims to investigate the role of context processing capabilities in NLIDB systems. However, the importance of context processing capabilities has been explored extensively in Question Answering systems (Chai and Jin, 2004; Kato et al., 2004; Kirschner and Bernardi, 2007; Negri and Kouylekov, 2007; Kirschner and Bernardi, 2010).

Users often fail to express their intention (information need) in a single NL query (user re-

sponse) (Bertomeu et al., 2006). Hence to answer a sequence of related NL queries, NLIDB systems should keep track of contextual information. NLIDB systems which do not use contextual information (non-contextual NLIDB) fail to completely capture the user's intention.

U1: How many third year students have registered for Robotics course in Monsoon 2011?

S1: 12

U2: How many got 'A' grade?

S2: 2

U3: Show the name of the professor who guides the student 'Newton'

S3: Prof. Einstein

U4: Did he teach any course?

S4: No

Figure 1: An example of context based user-system interaction

For example, let us consider a user-system interaction shown in Figure 1. User responses are represented as U1, U2, etc. and system responses are represented as S1, S2, etc. In this example, to interpret U2, information present in the preceding query U1 is needed. That means information present in U1 is the contextual information for U2. Query U3 does not depend on the information present in preceding queries. Semester name 'Monsoon 2011' present in U1 and the professor name 'Einstein' present in S3 are needed to interpret U4.

1.1 Background

In a semantic template based non-contextual NLIDB system (Gupta et al., 2012), the main stages involved in extracting answers (system's response) from the database are shown in Figure 2. At the syntactic analysis stage, the linguistic information is extracted from the NL query. At the semantic analysis stage, entities, attributes and the

values to these attributes are identified by using the output of the syntactic analysis module and semantic templates. At the query processing stage, entities identified in the semantic stage are mapped onto the domain conceptual model based on an entity relationship graph (ER graph) and a shortest path in the ER graph connecting them is computed. SQL (Structured Query Language) query is generated using the path obtained and the SQL query is later executed to produce results.

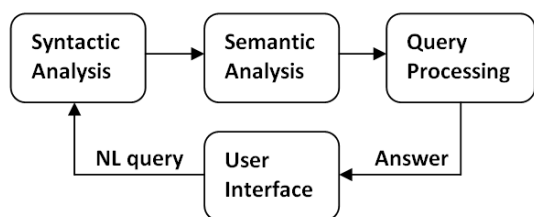


Figure 2: NLIDB system without context processing capabilities

In our approach, the context processing capabilities are incorporated into a non-contextual NLIDB system without disturbing the internal functioning of the existing modules of the system as shown in Figure 3.

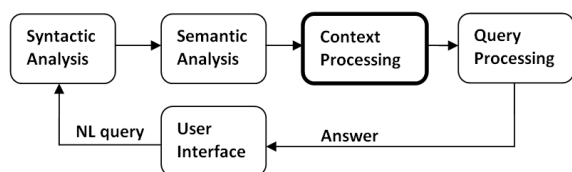


Figure 3: Context based NLIDB system

2 Related Work

Chai and Jin (2004) and Sun and Chai (2007) investigate the role of discourse modeling to track contextual information in interactive Question Answering systems. They analyzed the relations between user’s responses and proposed models based on centering theory to identify the contextual information. However, the above mentioned models fail to utilize system’s responses. Kirschner and Bernardi (2007) and Bernardi and Kirschner (2008) proposed models which utilize both user’s responses and system’s responses. But, in all these approaches, no attempt was made to understand the structure of user-system interactions. *We believe that understanding the structure of user-system interactions is the key to identifying an effective model to track contextual information.*

Bertomeu et al. (2006) made an attempt to understand the structure of user-system interactions. Along the lines of their work, we aim to identify models which reflect the underlying structure of user-system interactions. We propose three models based on the way in which the contextual information is utilized in the user-system interactions. Contextual information can sometimes be found beyond the immediate preceding responses (antecedents) as discussed in (Bertomeu et al., 2006). The approach proposed in this paper was able to identify contextual information present in such responses. Further, it was also able to identify the contextual information present in more than one antecedent.

The remainder of this paper is organized as follows. In section 3, we more precisely define the problem and introduce our terminology and notation conventions. In section 4, we categorize the interactions between user and system. We model the user-system interactions in section 5. We propose a relationship schema among the responses (user and system responses) in section 6. In section 7, using these relations, we present a novel method to identify contextual information for one of the models proposed in section 5. Finally, we present our experimental results in section 8 and conclude in Section 9.

3 Problem

Responses by both user and system in a user-system interaction can be grouped into a set based on the information shared among them. Each individual group is called ‘local contextual group’ (LCG) and the corresponding information (i.e. information present in every user response of that group) maintained by it is called ‘local contextual information’ (LCI) or ‘contextual information’. Given a user response, first we need to identify the LCG to which it belongs and then use the corresponding LCI to interpret the user response.

The following notation is used throughout this paper:

lc_i denotes the i^{th} LCG.

u_{kl} denotes the k^{th} user response and there are l LCGs just before this response is given by the user.

s_{kl} denotes the k^{th} system response and there are l LCGs just before this response is given by system.

For every user response u_{kl} , there will be a corresponding system response s_{kl} . We define the pair (u_{kl}, s_{kl}) as a **dialogue unit** d_{kl} .

The user response u_{kl} can either belong to any of the previous LCGs $lc_i = 1,2,3 \dots l$ or it can lead to the formation of new LCG lc_{l+1} . This is because the user can only either refer to the past information or can provide new information. User cannot refer to future local contexts (i.e. $i > l+1$).

The system response s_{kl} can only belong to any of the previous local contexts $lc_i = 1,2,3 \dots l$. It cannot belong to lc_{l+1} or any of the other future local contexts. This is because the system can only provide output for the past ($i \leq l$) user responses. Hence, only a user response can create a new LCG.

So there are two primary steps to identify contextual information of a user response u_{kl} : (a) To identify all the LCGs present in the interaction and (b) To find the corresponding LCG to which u_{kl} belongs.

4 User-System Interactions

Kato et al. (2004) categorized the interactions between user and system into two types: Browsing type and Gathering type. In our experiments, we found a similar and more finer categorization to be helpful for analyzing the interactions:

1) Strongly Coherent interaction: In this kind of interaction, the user interacts with the system with a topic in mind and a goal to achieve. In our experiments, we found that most of the responses in such an interaction are closely related with each other (section 8).

2) Coherent interaction: In this kind of interaction, the user only knows about the topic and he does not have any specific goal. Here, the responses may not be as closely related as in strongly coherent interactions.

3) Weakly Coherent interaction: In this kind of interaction, the user neither has a topic nor a goal. Most of the responses in this type of interaction may not be related with each other.

5 Modeling User-System Interaction

Depending on the way in which the contextual information can be utilized in the user-system interactions, we propose the following three models:

1) Linear Disjoint Model: In this model, the following three conditions hold true:

condition 1: u_{kl} can belong to only one LCG.

condition 2: $u_{kl} \in lc_i$ or $u_{kl} \in lc_{i+1}$, where $i = l$.

This implies that user response can only either belong to the immediate previous LCG or it can form

a new LCG.

condition 3: All LCGs are disjoint.

This implies that responses belonging to a LCG can be interpreted without depending on the information present in responses belonging to any of the other LCGs.

For example, let us consider a Linear Disjoint interaction (i.e. user-system interaction which can be modeled by Linear Disjoint Model) shown in Figure 4. In this example, d_{10} , d_{20} belong to first LCG and d_{31} , d_{41} belong to second LCG. We can interpret the responses belonging to second LCG without depending on the information present in responses belonging to first LCG.

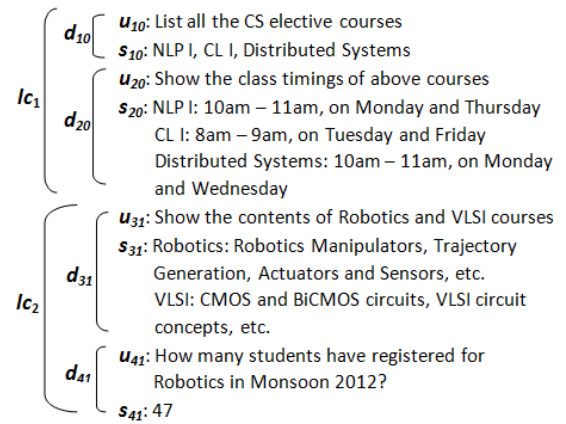


Figure 4: An example of Linear Disjoint interaction

2) Linear Coincident Model: In this model, condition 1 and condition 2 hold true. Condition 3 does not hold true if lc_i and lc_j are adjacent (i.e. $|j-i| = 1$). This implies that interpreting responses belonging to a LCG may need the information present in the responses belonging to its adjacent LCG.

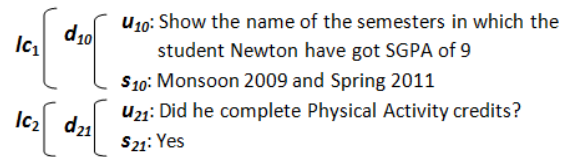


Figure 5: An example of Linear Coincident interaction

For example, let us consider an example of Linear Coincident interaction shown in Figure 5. In this example, d_{10} belong to first LCG and d_{21} belong to second LCG. u_{21} corefer the student name 'Newton' present in u_{10} . Hence interpreting the

responses belonging to second LCG needs the information present in responses belonging to first LCG.

It may also be noted that d_{10} and d_{21} may appear to belong to the same LCG but this is not so. If we assign both d_{10} and d_{21} to same LCG, then the information present in d_{10} would be used to interpret u_{21} . In this case u_{21} would be interpreted as ‘Did Newton complete Physical Activity credits in Monsoon 2009 and Spring 2011’. But the user’s intention is to know whether Newton completed Physical Activity credits or not (in any of the semesters). Hence both d_{10} and d_{21} cannot belong to same LCG.

3) Non-Linear Model: In this model, all the three conditions may or may not hold true. This implies that user response can belong to more than one LCG. Also interpreting responses belonging to a LCG may need the information present in the responses belonging to any of the other LCGs. Identification of contextual information in such interactions is very difficult compared to Linear Disjoint and Linear Coincident interactions. Complexity of contextual information present in various models is as follows:

Linear Disjoint Model < Linear Coincident Model < Non-Linear Model

6 Relationships between User Response and Dialogue Units

In a semantic template based non-contextual NLIDB system (Gupta et al., 2012), entities identified in semantic stage (explicit entities) are mapped onto the domain conceptual model based on an entity relationship graph (ER graph). A shortest path (sub-graph) in the ER graph connecting the explicit entities is computed. Implicit entities are the entities in the sub-graph which connect the explicit entities. For every user response, a sub-graph is generated. So for every dialogue unit, there exists a sub-graph.

Between user response (u_{kl}) and dialogue units (d_{ij} where $i < k$ and $j < l$), we define the following relationships based on their corresponding sub-graphs:

(1) **Strong Link:** u_{kl} and d_{ij} are said to be strongly linked if their sub-graphs satisfy the following three properties.

property 1: there is at least one explicit entity (e_f) in common.

property 2: there is at least one attribute (a_f) of the

entity e_f in common.

property 3: there is at least one value (v_f) to the attribute (a_f) in common.

(2) **Link:** u_{kl} and d_{ij} are said to be linked if property 1, property 2 are satisfied and property 3 is not satisfied.

(3) **Weak Link:** u_{kl} and d_{ij} are said to be weakly linked if none of the properties are satisfied i.e. they either have implicit entities in common or no entity in common.

For example, let us consider the user-system interaction shown in Figure 6. Here u_{20} and d_{10} are strongly linked because they both have common explicit entity ‘course’, common attribute ‘course name’ and common value ‘Database Systems’ to that attribute. Similarly, u_{41} and d_{31} are strongly linked. u_{52} and d_{41} are linked because they only have the entity ‘professor’ in common. u_{31} and d_{20} are weakly linked because they don’t have any explicit entity in common.

7 Identifying contextual information in Linear Disjoint Model

To use contextual information in a user-system interaction, we need to perform two primary steps. First, we need to identify all the LCGs present in the interaction. Then, given a user response, we need to find the corresponding LCG to which it belongs. In our approach, we perform these two steps simultaneously.

In Linear Disjoint Model, a user response can either belong to the immediate previous LCG or it can form a new LCG. Let the user response be u_{kl} . That means there are already l LCGs before user has given this response. Now we need to find whether u_{kl} belongs to lc_l or not.

Suppose if u_{kl} is assigned to the LCG lc_l , the corresponding contextual information is used to interpret u_{kl} . Otherwise, a new LCG lc_{l+1} is created and u_{kl} is assigned to lc_{l+1} .

We use the relationships between user responses and dialogue units to determine whether u_{kl} belongs to lc_l or not. The intuition behind using these relationships is given below:

1) If u_{kl} is *strongly linked* to **any** dialogue unit belonging to lc_l , then it indicates that the user might be referring to the information present in lc_l and hence u_{kl} is assigned to lc_l .

2) If u_{kl} is *linked* to **any** dialogue unit belonging to lc_l , then it indicates that user might be reducing focus on the information present in lc_l and hence

the system creates a new LCG lc_{l+1} and assigns u_{kl} to lc_{l+1} . Since reducing focus may not always lead to formation of new LCG, system confirms with user by asking some questions.

3) If u_{kl} is *weakly linked* to **any** dialogue unit belonging to lc_l , then it indicates that user might not be referring to the information present in lc_l and hence the information present in lc_l is not used as contextual information. A new LCG lc_{l+1} is created and u_{kl} is assigned to lc_{l+1} .

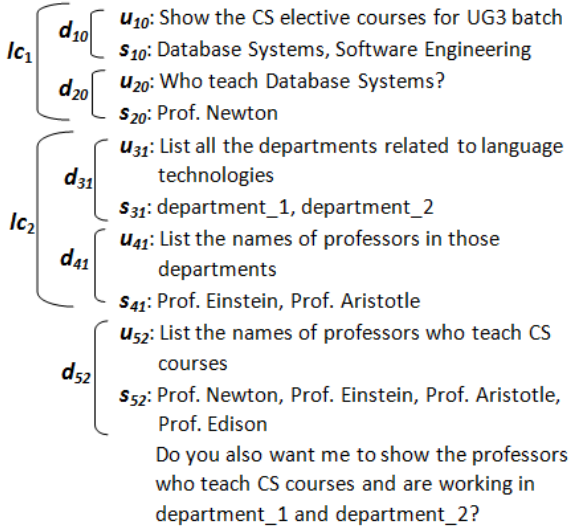


Figure 6: An example of Linear Disjoint model

For example, let us consider a Linear Disjoint interaction shown in Figure 6. Since u_{20} and d_{10} are strongly linked, we use the information present in d_{10} as the contextual information for u_{20} . Hence the output will be the names of professors who teach the course ‘Database Systems’ for UG3 batch.

As u_{31} and d_{20} are weakly linked, information present in d_{10} and d_{20} are not used as contextual information for u_{31} . Also a new LCG lc_2 is created and u_{31} is assigned to lc_2 . Similarly u_{41} uses information present in d_{31} as contextual information because they are strongly linked.

As u_{52} and d_{41} are linked, the user might be reducing focus on the information present in lc_2 . Hence, u_{52} is interpreted without using the information present in lc_2 as contextual information and later system confirms with user by asking some questions.

8 Experiments and Discussions

We carried out experiments on university related queries. Using the existing non-contextual NLIDB

system (Gupta et al., 2012), we have developed 110 dialogues which cover a wide range of topics such as course registration, seminar talks, credit requirements and cultural events. Each dialogue contains a sequence of user and system responses (or turns). On an average, each dialogue contains about 12 responses, corresponding to a total of 1320 responses.

Out of these 110 dialogues, 40 dialogues are of strongly coherent type, 40 dialogues are of coherent type and 30 dialogues are of weakly coherent type. We found that 96.6% of these dialogues belong to Linear Disjoint Model and 3.4% of the dialogues belong to Linear Coincident Model. We did not find any dialogues belonging to Non-Linear Model. This indicates that the method proposed in this paper is sufficient to identify contextual information in most of the real-time interactions.

	Strong Links	Links	Weak Links
Strongly Coherent interaction	72.84%	22.89%	4.29%
Coherent interaction	46.25%	43.75%	10%
Weakly Coherent interaction	34.34%	41.34%	24.34%

Table 1: Average percentage of relationships observed in different types of interactions

Table 1 shows the average percentage of various relationships (proposed in section 6) observed in different types of interactions. In a strongly coherent interaction type, higher percentage of strong links are observed. This is consistent with the definition of strongly coherent interaction. In such an interaction, user interacts with the system with a topic in mind and a goal to achieve. At each stage of the interaction, the user tries to move closer to the goal. Hence, we can expect the user to construct a query (or response) using the information obtained from the previous queries. This also explains the presence of a very small percentage of weak links.

From the definition of coherent interaction type, one would expect a higher percentage of links than strong links. On the contrary, we found almost equal percentage of strong links and links. This is because in a coherent interaction, the user can ask about various details regarding a topic. We can call these details as short term goals (or temporary

goals). In contrast to strongly coherent interaction where user has a single goal (long term goal) to achieve, coherent interaction contains many short term goals.

User may not get an answer for every short term goal in a single query. Hence, we can expect the user to ask multiple queries (but these are much less than the total number of queries used to achieve long term goal) to achieve short term goals. The interaction corresponding to every short term goal have high percentage of strong links than links. Interactions corresponding to every two short term goals are expected to connect with either links or weak links. But since we have a fixed topic, we can expect higher probability for links to connect those short term goals. As there can be many short term goals, percentage of links will be also high.

In a weakly coherent interaction, higher percentage of weak links are observed compared to other two types of interactions. This is because the user neither has topic nor a goal to achieve. Hence, while interacting with the system, the user may randomly pick topics and ask various details related to those topics. Once a topic is chosen, the interaction can be viewed as a coherent interaction. Hence, we can see almost the same percentage of strong links and weak links. Notice that there is a higher probability for interactions with different topics to be weakly linked with each other. As a user may frequently change the topics, we can see the increase in the percentage of weak links.

Table 2 shows the average number of local contexts, average length of local context (i.e. total number of responses in each local context) observed in different types of interactions. As discussed earlier, in a strongly coherent interaction, the user has a fixed and a single goal to achieve. So, we can expect most of the queries to be related to each other. Hence, this type of interactions contain less number of local contexts and each local context has more responses.

Coherent interactions contain many short term goals and each short term goal is expected to contain less number of responses compared to the long term goals present in strongly coherent interactions. So this type of interactions contain comparatively more number of local contexts and smaller average length than strongly coherent interactions.

In weakly coherent interactions, user can change the topics very often and hence contain higher number of local contexts and least average length.

	Number	Length
Strongly Coherent interaction	2.2	4.67
Coherent interaction	3	1.88
Weakly Coherent interaction	3.83	1.43

Table 2: Average number and average length of local contexts observed in different types of interactions

We applied the method proposed in section 7 to 106 Linear Disjoint dialogues (which constitute 96.6% of the total dialogues). The results obtained are impressive. For each dialogue, we evaluated the percentage of the queries for which the corresponding contextual information has been identified correctly. The contextual information is identified with 100% accuracy for 78 dialogues i.e. our method successfully identified the appropriate context for every user response of those dialogues. The contextual information for 13 dialogues has been identified with 10 to 20% error. 9 dialogues are found with error greater than 40%.

9 Conclusion

In this paper we categorized user-system interactions and then proposed three models (Linear Disjoint Model, Linear Coincident Model and Non-Linear Model) depending on the way in which the contextual information can be utilized in the interactions. We proposed a new relationship schema among the responses. Central in our approach is the use of these relationships to identify contextual information in Linear Disjoint interactions. Furthermore, we evaluated our approach on university related queries and the results confirm the viability of the proposed approach. In our corpus, we found that 96.6% of the total interactions are Linear Disjoint interactions. Hence the method proposed in this paper is sufficient to identify contextual information in most of the real-time interactions.

In the future, we plan to investigate how to identify the model of an interaction. We also intend to identify contextual information in Linear Coincident interactions and Non-Linear interactions.

References

- Ioannis Androutsopoulos, Graeme D Ritchie, and Peter Thanisch. 1995. Natural language interfaces to databases-an introduction. *arXiv preprint cmp-lg/9503016*.
- Raffaella Bernardi and Manuel Kirschner. 2008. Context modeling for iqa: the role of tasks and entities. In *Coling 2008: Proceedings of the workshop on Knowledge and Reasoning for Answering Questions*, pages 25–32. Association for Computational Linguistics.
- Núria Bertomeu, Hans Uszkoreit, Anette Frank, Hans-Ulrich Krieger, and Brigitte Jörg. 2006. Contextual phenomena and thematic relations in database qa dialogues: results from a wizard-of-oz experiment. In *Proceedings of the Interactive Question Answering Workshop at HLT-NAACL 2006*, pages 1–8. Association for Computational Linguistics.
- Joyce Y Chai and Rong Jin. 2004. Discourse structure for context question answering. In *Proceedings of the Workshop on Pragmatics of Question Answering at HLT-NAACL 2004*, pages 23–30.
- Alessandra Giordani and Alessandro Moschitti. 2009. Syntactic structural kernels for natural language interfaces to databases. In *Machine Learning and Knowledge Discovery in Databases*, pages 391–406. Springer.
- Alessandra Giordani. 2008. Mapping natural language into sql in a nlib. In *Natural Language and Information Systems*, pages 367–371. Springer.
- Abhijeet Gupta, Arjun Akula, Deepak Malladi, Puneeth Kukkadapu, Vinay Ainavolu, and Rajeev Sangal. 2012. A novel approach towards building a portable nlib system using the computational paninian grammar framework. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 93–96. IEEE.
- Tsuneaki Kato, Junichi Fukumoto, and Fumito Masui. 2004. Question answering challenge for information access dialogue-overview of ntcir-4 qac2 sub-task 3. In *Proceedings of the 5th NTCIR Workshop Meeting on Evaluation of Information Access Technologies*, pages 291–297.
- Manuel Kirschner and Raffaella Bernardi. 2007. An empirical view on iqa follow-up questions. In *Proc. of the 8th SIGdial Workshop on Discourse and Dialogue, Antwerp, Belgium*.
- Manuel Kirschner and Raffaella Bernardi. 2010. Towards an empirically motivated typology of follow-up questions: the role of dialogue context. In *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 322–331. Association for Computational Linguistics.
- Yunyao Li, Huahai Yang, and HV Jagadish. 2005. Nalix: an interactive natural language interface for querying xml. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 900–902. ACM.
- Xiaofeng Meng and Shan Wang. 2001. Nchiql: The chinese natural language interface to databases. In *Database and Expert Systems Applications*, pages 145–154. Springer.
- Matteo Negri and Milen Kouylekov. 2007. who are we talking about? tracking the referent in a question answering series. In *Anaphora: Analysis, Algorithms and Applications*, pages 167–178. Springer.
- 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*, pages 149–157. ACM.
- Niculae Stratica, Leila Kosseim, and Bipin C Desai. 2005. Using semantic templates for a natural language interface to the cindi virtual library. *Data & Knowledge Engineering*, 55(1):4–19.
- Mingyu Sun and Joyce Y Chai. 2007. Discourse processing for context question answering based on linguistic knowledge. *Knowledge-Based Systems*, 20(6):511–526.