On the Training of Mental Reasoning: Searching the Works of Literature

Boris Galitsky

KnowledgeTrail, Inc.
9 Charles Str
Natick, MA 01760
bgalitsky@knowledge-trail.com

Abstract

We build the formalism and design the training software for reasoning about mental states and actions of literature characters, initiated by asking questions in natural language. The students (general audience, as well as children with various mental disorders) are encouraged to inquiry about intentions and desires, knowledge and beliefs, pretending and deceiving of the literature characters. Works of literature are identified based on the patterns of mental interaction between the characters. The role of suggested system in training the decision-making skills is discussed.

Introduction

In this paper, we address the issue of search for a new and emergent kind of data: works of literature (WOL) and explore its application in education and mental rehabilitation. The methodology and abstraction of such search are very different from those for database querying, keyword-based search of relevant portions of text, and search for the data of various modalities (speech, image, video etc.). Clearly, the search that is based on mental attributes is expected to be semantically accented: using just the author or title name is trivial. Also, using temporal (historical) and geographical circumstances of the characters reduces WOL search to the relatively simple querying against the relational database of WOL parameters.

Focusing on the mental component of WOL plots is rewarding from the prospective of building the compact and closed (in terms of reasoning) vertical natural language (NL) question-answering (Q/A) domain. It is very important that a user is aware of the lexical units and knowledge that is encoded in a domain to ensure the robust and accurate Q/A system. Division of the commonsense knowledge into mental and non-mental (physical) components introduces a strict and explicit boundary between the "allowed" and "not allowed" questions, that is a key to success of NL Q/A application in the field of education (Galitsky 2000).

Training of mental reasoning is important to develop such professional skills as negotiation, resolution of conflicts, argumentation, creative and fast decision-making. Using the literature domain for such training takes advantages of the variety of plots, appealing and entertaining environment and rather complex mental states of literature characters. We believe such kind of training is essential for business, military, legal, psychological and other professional fields, which require rapid orientation and reaction in emergent situations with inconsistent goals and beliefs of opponents and customers.

In our previous studies, we have built the game-like environment of virtual mental world (Galitsky 2001) and question-answering system for mental rehabilitation (Galitsky 2000a). Training of mental reasoning is an important way of developing emotional and intellectual capabilities of children and adults with various mental disorders (Baron-Cohen 1998). In this paper, we continue the design and implementation (for educational purposes) the interactive environment based on the mental states of WOL characters, developing the reasoning capabilities of the system. In recent years, there was a lot of attention to the formal background of reasoning about mental states and actions (consult for example Fagin et al. 1995, Wooldridge 2000), in particular, the BDI (belief-desire-intention) multiagent architecture. However, reasoning about mental states of software and human agents has recently started to find a variety of applications in the educational area. It is worth particularly mentioning such area as the simulation of human agent

The knowledge base, Q/A access to it and multiagent mental simulator software is developed that allows a trainee to speculate about WOL characters and "intelligently" access WOLs without explicit mentioning of its title or its author's name. In this study, we present the multiagent mental simulator (Section 4) that extends the match of mental formulas under question-answering. The overall training system architecture is presented in Section 2 and the system demonstration - in Section 3.

1. WOL database: mental states of literature characters

In this section, we introduce basic and derived mental entities and discuss their role in the creation of the WOL knowledge base. We have built the library (database) of

WOLs that includes the manually extracted mental states of their characters. As many WOLs were collected as it was necessary to represent the totality of mental states, encoded by logical formulas of the certain complexity (below four, see Galitsky 2000a). Mental formulas are expressed in a second-order predicate language, extended by the metapredicates for mental states and actions. We develop the metaprogramming technique to provide the required knowledge representation and reasoning capabilities, based on PROLOG (see, for example, Hill & Lloyd 1989).

The basic mental states are *intention* (subsumes goals and desires), *knowledge* and *belief*. The difference between belief and knowledge is that an agent is capable of changing and revising beliefs, but knowledge is only subject to acquisition. Almost any mental state or action entity can be defined within these basic mental states after adding an arbitrary predicate for a physical state or action (Galitsky 1998). Some mental entities (particularly, expressing emotions) cannot be formally derived using the basis above; however, this basis introduces the classes of equivalence with respect to the decision concerning the fixed set of actions (physical and mental). In other words, basic and derived mental states are the factorization of cognitive and emotional states relatively to the resultant physical or mental action in the multiagent settings.

For example, the entities *inform, deceive, explain, forgive*, etc. can be expressed via *want-know-believe* basis. Conversely, the entity *fear* is neither basic nor derivable entity; however, it is equivalent to *not want* relatively to the potentially implied physical action (e.g. to run away). The difference between *fear*, *not want* and *prefer to avoid* is in the "degree of influence" of the mental entity; therefore, we can ignore this difference having the explicit degree for each entity within an equivalence class.

Analysis of our WOL knowledge base (Galitsky 2001) allows us to make the following conclusion. Usually, the main plot of a WOL deals with the development of human mental attributes, expressible via the basic and derived mental predicates. A single mental state expresses the very essence of a particular WOL for the small forms (a verse, a story, a sketch, etc.). When one considers a novel, a poem, a drama, etc., which has a more complex nature, then there is set of individual plots can be revealed. Each of these plots is depicting its own structure of mental states that is not necessarily unique. Taken all together, they have the highly complex forms, appropriate to identify the WOL. The sets of mental formulas are sufficient to identify a WOL.

2. System architecture

We enumerate the tasks that have to be implemented for the literature search system based on reasoning about mental attributes of characters (Fig. 1):

1) Understanding a natural language query or statement (see Galitsky 2000b for details). This unit converts an NL expression in a formalized one (mental formula), using mental metapredicates and generic predicates for physical states and actions.
2) Domain representation in the form of semantic headers (Galitsky 2000b), where mental formulas are assigned to the textual representation (abstract) of WOLs.
3) Multiagent mental simulator that builds the hypothetical mental states, which follow one mentioned in the query. These generated hypothetical mental states will be searched against WOL knowledge base together with the query representation in unit 5. We present mental simulator in details in Section 4.
4) Synthesis of all well-written mental formulas in the given vocabulary of basic and derived mental entities.
5) Matching the mental formula, obtained for a query against mental formulas, associated with WOLs. We use the approximate match in case of failure of the direct match.
6) Synthesis of canonical NL sentence based on mental formula to verify if the query was properly understood

Fig.1 presents the interaction between the respective components 1)-6) of the WOL search system. Suggested system architecture allows two functioning options: WOL search and extension of WOL knowledge base. When a user wishes to add a new WOL to the current knowledge base (Table 3), semantic headers are automatically build by unit 1 and are multiplied for semantically different phrasings by unit 3.

Rather complex semantic analysis (unit 1) is required for exact representation of input query: all the logical connectives have to be properly handled. Unit 3 provides the better coverage of the WOL domain, deductively linking mental formula for a query with mental formulas for WOLs. Unit 4 is necessary because the traditional axioms for knowledge and belief (see, for example, Fagin et al 1995, Wooldridge 2000) are insufficient to handle the totality of all mental formulas, representing the real-life situations. We developed the algorithm to extract the realizable mental formula from the totality of all well-written mental formulas, represented via metapredicates. In addition, introduction of the classes of equality of mental formulas are required for the approximate match of mental formulas (unit 5) that is also inconsistent with the traditional formalizations of reasoning about knowledge and belief. NL synthesis of mental expression (Unit 6) is helpful for the verification of the system's deduction. A trainee needs this component to verify that he/she was understood by the system correctly before starting to evaluate the answer. NL synthesis in such strictly limited domain as mental expression is straightforward and does not require special considerations. We mention that Units 1,2 and 5 are developed as a commercial product for advising in financial domains (Galitsky 2000b). However, semantic rules for the analysis of mental formulas require specific (more advanced) machinery for complex embedded expressions and metapredicate substitutions.

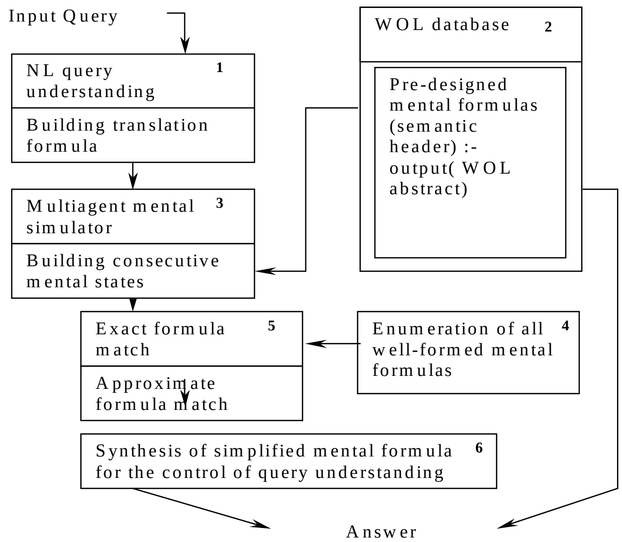

Figure 1. The units of the WOL search and mental reasoning system. NL query that includes mental states and action of WOL characters is converted into mental formula (1). Multiagent mental simulator (3) yields the set of mental formulas, associated with the query to extend the knowledge base search. Obtained formulas are matched (5) against the totality of prepared semantic headers (mental formulas) from the WOL database (2). If there is no semantic header in the domain knowledge component that satisfies the mental formula for a query, the approximate match is initiated. Using the enumeration of all well-formed mental formulas (4), the system finds the best approximation of the mental formula for a query that matches at least single semantic header (mental formula for an answer).

The special question-answering technique for the weakly structured domains has been developed to link the formal representation of a question with the formal expression of the essential idea of an answer. These expressions, enumerating the key mental states and actions of the WOL characters, are called the *semantic headers* of answers (Galitsky 2000b). The mode of knowledge base extension (automatic annotation), where a customer introduces an abstract of a plot and the system prepares it as an answer for the other customers, takes advantage of the flexibility properties of the semantic header technique.

3. Using the Literature Search System

The demo encourages the users (players, students) to demonstrate their knowledge of classical literature, from medieval to modern, asking questions about the mental states of the characters and compare the system results with your own imagination.

The system stimulates the trainees to extract the mental entities, which can be formalized, from the totality of features of literature characters. After an answer is obtained, it takes some efforts to verify its relevancy to the question. It takes a little variation in the mental expression to switch from one WOL to another (compare Table 2 and Table 3). More advanced users are offered the option of adding new WOL (Table 3).

For mental rehabilitation (particularly, autistic children) certain visualization aids are useful in addition to the WOL search system (Fig. 2, Galitsky, 2000a).

Figure 2. Autistic child learns the mental interaction with the characters (participants of the scene), using suggested system.

How would a person pretend to another person that she does not want that person to know something?
When would a character pretend about his intention to know something?
Why would a person want another person to pretend about what this other person want?
How can a person pretend that he does not understand that other person does not want?

Table 1. Sample questions for the literature search.

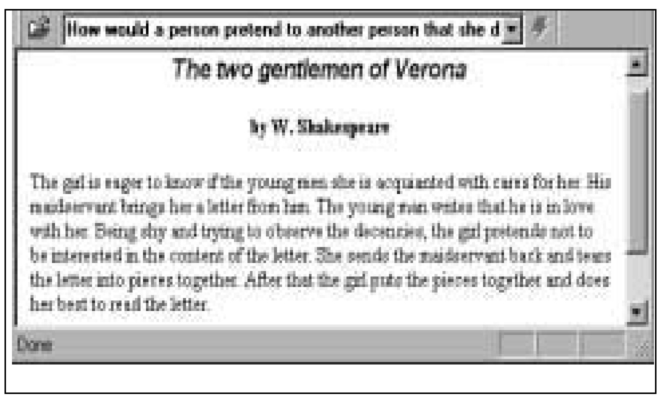

Table 2. Windows question-answering software outputs the answer for the first question above (Table 1). Formal representation for the first query looks as follows:
pretend(person, other_person,
not want(person,know(other_person, Something))).

<table>
<tr><td colspan="2">WOL search system allows a literature fan to extend the knowledge base with the new favorite story or novel and to specify the major ways of accessing it (asking about it). This toolkit processes the combination of the answer (an abstract of a story, introducing the heroes and their interactions) and a set of questions or statements (explicitly expressing the mental states these interactions are based on).</td></tr>
<tr><td>When does a person pretend about her intention to know something?

Add to Knowledge</td><td>The Carriage of holly gifts
by P. Merimee
An old-aged king wants to learn from his secretary if the young girl he loves is faithful to him. The secretary is anxious to please the king…
Compile Knowledge base</td></tr>
<tr><td colspan="2">Domain extension code:
pretend(person, other_person, want(person, know(person,Smth))) <code>:-</code> do201.
do201:-output($The Carriage of holly gifts… $).
Domain is compiled. Ask a question to the updated domain
[ ] Ask Now you can ask the questions for the domain extension as well as for the base domain, varying the phrasings.</td></tr>
</table>

Table 3. User interface for extension the WOL knowledge base. In the second row, question(s) are specified on the left column, and the answer is in the right one.

4. Multiagent mental simulator

In this Section, we present the multiagent mental simulator that implements the extended BDI model for the complexity of mental formulas reaching four. The input of the multiagent simulator is an arbitrary (possibly, inconsistent) set of mental formulas. The output is the consecutive mental formulas, reflecting the states which are the results of the committed actions, chosen by the agents. The simulator can be viewed from the game-theoretic prospective in if the mutual beliefs of agents are absent (the trivial case).

To choose the best action, each agent considers each action he can currently possibly perform. For each such action, the agent assumes he has committed it and analyzes the consequences. They include the actions of other agents and various states, some of which the elected agent does not want to occur. The agents either decides to perform the action delivering the least unwanted state or action by another agent, or to do nothing. If there are multiple possible actions which do not lead, in the agent belief, to unwanted consequences, this agent either chooses the preferred action, if there is explicit preference predicate, or the action, which conjectures use the beliefs concerning the other agents in the least degree.

If our agent wants the particular action of the other agent, he can either do nothing or perform an action that is neither explicitly wanted nor unwanted, but is followed by the desired action of that other agent, in accordance to our agent's belief. Hence, the simplest predicate of the action choice is the following

```
whatToDo(MyAgent,PossibleActions ):-
 findall( PossibleAction,
  (want(MyAgent, PossibleAction),
  ownAction(MyAgent, PossibleAction),
  not(PossibleAction=(not NegPossibleAction)
), assume( PossibleAction ),
  not (want(MyAgent, not UnwantedState),
 UnwantedState ) ),PossibleActions).
```

For each `PossibleAction`, which our `Agent` wants, we check whether it is an action, `Agent` can perform himself, and that this is not a negation of some state. If this is a currently possible action, our `Agent` assumes that it has been committed. Then our `Agent` runs through all the unwanted states and verifies that his action has not lead to any of these states. Our `Agent` involves his own knowledge and belief only, ignoring the possible actions of the other agents.

To model the behavior of competing (or assisting) agents, our `Agent` uses the following predicate

```
otherAgentsToDo(Agent, AllAgentActions):-
 findall( AgentActions,
  (agent(OAgent), OAgent \= Agent,
   involve_knowledge(OAgent, Agent),
   whatToDo(OAgent, AgentActions),
   clean_knowledge(Agent)
   ), AllAgentActions).
```

To model the behavior of other agent `OAgent`, `Agent` involves his own knowledge and belief about `OAgent`'s intentions concerning actions and states. Furthermore, to build a prediction of `OAgent`'s action, `Agent` uses his own belief and knowledge about `OAgent`'s believe and knowledge. For all the other agents, selected `Agent` involves their knowledge and performs the same action choice procedure as she does for himself. As a result, she obtains the list of actions other agents would choose, from his (Agent's) prospective. To incorporate the predicted actions of the other agents in the own action choice, `Agent` employs the following predicate, which finally gives the optimal (from `Agent`'s viewpoint) `BestActions`.

```
whatToDoInConflict(MyAgent, BestActions) :-
 involve_knowledge(Agent, Agent),
        %MyAgent involves knowledge concerning himself
 findall( (PossibleAction,Score),
 (want(Agent, PossibleAction),
  ownAction(Agent, PossibleAction),
  not ( PossibleAction = (not
       NegPossibleAction)),
  assume( PossibleAction ),
  otherAgentsToDo(Agent, OtherActions),
 findall(Term, ( member(Term,
    OtherActions), assume(Term)),_),
 not (want(Agent, not UnwantedEvent),
 UnwantedEvent ),
 ), BestActions).
```

For each `PossibleAction`, `Agent` assumes it has been committed. Then the other agents act, in accordance to `Agent`'s knowledge and beliefs, taking into account this `PossibleActions,` as if they have already seen what `Agent` has done. Finally, `Agent` verifies, whether all these action could lead to the unwanted consequences of her original `PossibleActions.` In case of the absence of unwanted states, `Agent` determines if there are any desired actions of the other agents, implied, to the best of `Agent` knowledge, by `PossibleAction.`

Now an `Agent` obtained the set of possible actions. The agent assumes that the other agents could also obtain this set. Correspondingly, these agents could make decisions concerning their actions assuming the `Agent` to perform either of his chosen actions. Therefore, the `Agent` takes this fact into account and reduces her available actions in accordance to reviewed possibilities of the other agents. Presented epistemic iterations ensure the optimality of the action choice.

The simulator helps to link the mental formula for a query with semantic headers in up to 30% of user's questions. Besides, using the simulator allows posing the implicit questions and ones with incomplete information. For instance, the question *What would a character do if she does not want another character to know that the third character does not want to do something?* may cause, for example, such action as *cheating* for the first agent and *pretending of not understanding* of the second one. As a result, the query above would be linked with semantic headers
cheat(A, B, want(A, know(B, not want(C, do(C, Smth))))) and *pretend(B, A, not understand(B, do(C, Smth))).* Note that the direct match is impossible.

For the detailed commented code of the simulator, the reader can consult
dimacs.rutgers.edu\~galitsky\MS\ma\Choice_action.html.

Conclusions

Literature search demo page was visited by a few thousand potential users of a wide variety of NL Q/A domains, including financial, legal, psychological etc., since 1999. Though the literature search was not considered as commercial as the other domains, available at
knowledge-trail.com website, it attracted the attention of the users, seeking an intellectual challenge. These users were interested enough to exhaust the majority of mental states to grasp the encoded knowledge base. For some works of literature, it took certain efforts to extract the assumed mental state and to understand the question-answer link. Our experience shows that it takes a few minutes for some users to associate a question with its answer that contains just a few sentence description of a mental state. Therefore, suggested system is a strong stimulator of thinking that is not an exhaustive search but reasoning in the very central meaning of that term.

Literature search domain was found by the users to be the most sensitive (in comparison with other domains) to the question semantics: slight deviation of meaning may lead to another answer (WOL). Certain users mentioned that using mental states is the preferred way of browsing WOLs with respect to conventional topics (as it is implemented, for example, at amazon.com).

Speculating about the mental states of literature characters has been demonstrated to be a novel educational area, appealing to adults as well as to children, interacting with the characters of scenes in NL. The high role of mental reasoning in the overall autistic rehabilitation process was discussed in Galitsky 2000a. Since the players are suggested to ask questions and share the literature knowledge, the system encourages the cooperation among the members of the trainees' community. Recognizing the questions and statements, involving the terms for mental states and actions only, we encourage the trainees to stay within a "pure" mental space and to increase the complexity of queries and statements. (We expect the system to handle them properly).

References

1. Baron-Cohen, S. 1998. Social and Pragmatic Deficits in Autism: Cognitive or Affective *Journal of Autism and Development Disabilities* 18:3.
2. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y. *Reasoning about knowledge*. MIT Press, Cambridge, MA, London, England (1995).
3. Wooldridge, M. 2000 Reasoning about Rational Agents. The MIT Press Cambridge MA London England.
4. Hill, P.M., Lloyd, J.W. 1989. Analysis of metaprograms, *in Abramson & Rogers (eds) Metaprogramming in Logic Programming*, MIT Press, pp. 23-52.
5. Galitsky, B. 1998. The formal scenario and metalanguage support means to reason with it. *DIMACS Tech. Report #98-27*, Rutgers University.
6. Galitsky, B. 2000a. Question-answering system for teaching autistic children to reason about mental states. *DIMACS Technical Report 2000-24*, Rutgers University, October 2000.
7. Galitsky, B. 2000b. Technique of Semantic Headers: a manual for knowledge engineers. *DIMACS Technical Report 2000-29*, Rutgers Univ.
8. Galitsky, B. 2001. Virtual Mental Space: Interaction with the characters of works of literature. *International Conference on Cognitive Technology* LNAI. Warwick Univ. UK 2001.