Computers in Human Behavior 38 (2014) 196–200

Contents lists available at ScienceDirect

Computers in Human Behavior

journal homepage: www.elsevier.com/locate/comphumbeh

How does an intelligent agent infer and translate?

Faria Nassiri-Mofakham *

Department of Information Technology Engineering, Faculty of Computer Engineering, University of Isfahan, Hezar Jerib Avenue, Isfahan 81746-73441, Iran

ARTICLE INFO

Article history:
Available online 30 June 2014

Keywords:
Software intelligent agent
Logical inference
Translation

ABSTRACT

Recently, websites employ online guides to help the users exploring required materials and information. The guides are presented through exchanging online questions and answers. For a foreign language visitor, website tour guides not only need to provide background and justification for the argument, but also they are better to translate the interaction. This paper presents an automated and intelligent software agent that can answer the questions logically. Although people can somehow simply reason and argument in their daily life, the nature of the humans' reasoning is generally complex and nontrivial. To make the inference and reasoning automated, the agent is armed with first-order logic in artificial intelligence. This enables the agent to understand and answer questions. Implementation of the complex process and the results are shown through a simple example. In addition, to make the agent more trustable and user-friendly, the intermediary inference and justification steps are translated in the user's language.

1. Introduction

Intelligent agents recently play an important role in problem solving. They can represent internal knowledge of humans and reason inside a domain-specific environment (Russell & Norvig, 2010). Conversational agents are to help users in finding their needs in worldwide systems. Most of them have been "chatbots" as either a website accessibility tool or an entertainment. Eliza and Alice-bots for socially interactive purposes in a non-judgmental manner answer the user by constructing another question. SGT Star and MAX are two other agents which employ animation of an avatar to make human-like conversations in military recruit and museum guide, respectively (Rubin, Chen, & Thorimbert, 2010). The other intelligent web agents help users with translating texts. Such agents can teach themselves to improve their translation performance by learning or interacting humans or other agents (Goutte, Cancedda, Dymetman, & Foster, 2009; Turchi, De Bie, & Cristianini, 2012). Computational story telling agents in automatic text generation have also been in consideration for many years. Automatic Novelwriter, Tale-spin, Universe, Minstrel, Josep, and Storybook are examples of these systems (Faas, 2002). ActAffAct is a work-in-progress that creatively is authoring stories using a framework based on a real-time questionnaire evaluation of authors' preferences for affective character creation within appropriate varieties of personalities (Rank, Hoffmann, Struck, Spierling, & Petta, 2012). Making agents more communicative and humanlike have also drawn studies to build robots with multimodal behaviour using learning-based speech, gaze, and gesture in a narration task (Huang & Mutlu, 2014). A few other researchers have also considered website navigations by interacting with users and text comprehension (Rank et al., 2012; Turchi et al., 2012) or cognitive modelling and mining of user's navigation patterns to predict when the user decide to click on a link or leave a page (Belk, Papatheocharous, Germanakos, & Samaras, 2013; Fu & Pirolli, 2007). In these researches, most applications answer the question using a searchable text-based interactive FAQ list or a database and in a human-like appearance or behaviour (Fig. 1). This paper shows how more complicated questions can be asked from an agent so that it requires inferring and reasoning through information provided by a few sentences rather than one database entry. It uses inference techniques and logical knowledge representation. In addition, users need to decide when to trust answers and trace proof. Explaining justifications shows the user where the information came from and how they were derived (Durkin, 1994). To do that, the intermediary inference steps are translated in the user's language to make the agent more trustable and user-friendly.

Overview of this Article. After reviewing a few concepts in Section 2, the Logical Translator Inference Agent is presented in Section 3. Research findings and implications of the study are discussed in Section 4. Section 5 then concludes.

* Tel.: +98(0)31 3793 4510; fax: +98(0)31 3793 4038.
E-mail addresses: fnasiri@eng.ui.ac.ir, fnasirimofakham@yahoo.com
URL: http://eng.ui.ac.ir/~fnasiri/

http://dx.doi.org/10.1016/j.chb.2014.05.019
0747-5632/

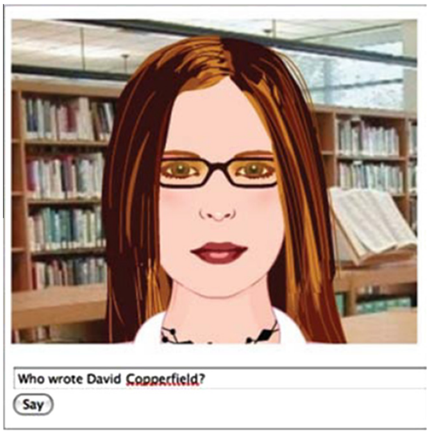

Fig. 1. OCLC virtual librarian (adapted from Rubin et al. (2010).

2. Logical agent

When a user is given the ability to check the process of inference to derive a deduction, the answer seems more acceptable for him or her (Durkin, 1994). There are different methods of inference in which logical representation of the knowledge can be processed forward by providing a few facts and rules or backward from a goal, the question (Durkin, 1994; Russell & Norvig, 2010). In this section, these two techniques are shortly described. In Section 3, they are employed to show how the process can be done automatically.

2.1.1. Proof by resolution

Resolution is an important technique that intelligent agents employ for knowledge representation in either propositional logic (PL) or first-order logic (FOL). The importance of these techniques also returns to convertibility of FOL to PL sentences. Resolution along with standard logical equivalences gives the agent the ability for proving theorems (Russell & Norvig, 2010).

2.1.2. Standard logical equivalences

The agent employs standards defined for logical sentences. A few of the equivalences are given below (for a complete list, please see Chapters 7 and 8 in Artificial Intelligence book (Russell & Norvig, 2010)). In the following inference rules, "and" (conjunction), "or" (disjunction), and "implication" are represented by "^", "v", and "→", respectively.

Implication Elimination: $a \rightarrow b$ is logically equivalent to $\sim a$ v b, in which $\sim a$ means negation of literal a.

And-Elimination: From a conjunction ($a \wedge b$), any of the conjuncts (i.e., a and b) can be inferred.

2.1.3. The resolution inference rule

Two clauses can be resolved, if they contain complementary literals; that is, one literal is negation of the other. Resolution proves the question by deriving the empty clause when it adds negation of the question to the knowledge base (KB). Propositional or first-order *factoring* reduces two literals to one if they are identical or unifiable (Russell & Norvig, 2010). Resolution is the best way for computers to think automatically about proving things. Resolution Inference Rule says that if we know something in the form of "a or b", and we know "not b or c", then we can conclude "a or c". First, all sentences should be converted to conjunctive normal form (CNF) in which a CNF is a conjunction of disjunctive clauses (a disjunctive clause is disjunction of literals). Next, we are going to do a proof by contradiction.

Resolution refutation is applied as follows:

1. Convert all sentences to CNF.
2. Negate the desired conclusion (converted to CNF).
3. Apply resolution rule until either.
   a. Derive false (a contradiction).
   b. Cannot apply any more.

That is, we assert that the thing that we are trying to prove is false, and then we try to derive a contradiction. So we negate the desired conclusion and convert it to CNF. And we add each of these clauses as a premise of our proof, as well. Then we apply the Resolution Rule until either we can derive false.

For example, suppose we are given "a or b", "a implies c" and "b implies c". We would like to conclude c from these three known things. The first four rows in Table 1 show the assumptions. We can apply resolution to rows 1 and 2, and get "b v c" by resolving away a. Similarly, we can take rows 3 and 4, resolve away c, and get "not b". By resolving away b in rows 5 and 6, we get c. And finally, resolving away c in rows 4 and 7, we get the empty clause, which is false. This means that $\sim c$ could not be given, and c is then proved by contradiction.

Table 1
Resolution inference.

Suppose		
1	a v b	
2	a v b	
3	$b \rightarrow c$	
Prove c		
Step	Formula	Derivation
1	a v b	Given
2	$\sim a$ v c	Given
3	$\sim b$ v c	Given
4	$\sim c$	Negated conclusion
5	b v c	1, 2
6	$\sim b$	3,4
7	c	5,6
8	Empty (=false)	4,7

2.2. Backward chaining inference

Backward chaining is a goal-directed reasoning for answering a question (Durkin, 1994; Russell & Norvig, 2010). It works backward from the query and finds those implications in the knowledge base whose conclusion is this query. If all premises of one of those implications can be proved true, the query is then true. That is, it requires reaching a set of known facts. The idea of this inference strategy is to check whether a particular fact a is true. That is, given the fact a to be proven,

1. See if a is already in the KB. If so, return TRUE.
2. Find all implications, I, whose conclusion matches a.
3. Recursively establish the premises of all i in I via backward chaining.

3. Logical translator inferencing agent

The knowledge base of the agent is created from an English story. The agent is asked a question from the story in English. The Agent needs to infer and answer logically in Persian. To answer the question, the agent logically uses a few sentences of its knowledge base and asserts a few intermediary implications in KB. The agent translates each sentence either employed or inferred in