
Usable Browsers for Ontological Knowledge Acquisition

Alicia Tribble

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
atribble@cs.cmu.edu

Carolyn Rosé

Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
cprose@cs.cmu.edu

Abstract

In this paper we compare the usability of several presentation formats for ontological knowledge of events. The goal is to support further work in knowledge acquisition from informants who are not necessarily experienced with knowledge representations. This work investigates the question: How can we present detailed ontological information to such informants, in a format that is easy to understand, modify, and augment? We compare three formats: two commonly-used diagram styles and one lisp-like list of knowledge axioms. Ongoing work on this topic will expand the investigation into a study of the role of natural language in knowledge acquisition.

Keywords

knowledge display, knowledge acquisition, natural language

ACM Classification Keywords

I1.2.4. Artificial Intelligence: Knowledge Representation Formalisms and Methods. H5.m. Information interfaces and presentation: Miscellaneous.

Introduction

Knowledge acquisition (KA, or acquisition) is a bottleneck for development of knowledge-based systems. These systems are important for tasks including machine translation [7], speech recognition [11], and information extraction [8], but building a large knowledge base (KB) requires many hours of

labor by knowledge engineers. Tools that address the acquisition problem can make knowledge-based systems easier to build. They also decrease the effort of adding new domain coverage to an existing knowledge-based system. In order to make such tools successful, we must consider the usability of their design.

In this paper we compare the usability of several presentation formats for *ontological knowledge*, which generally means a large set of semantic concepts along with relations that connect concepts to each other. This work investigates the questions: How can we present detailed ontological information to informants who are not KB experts? What presentation format is easiest to understand and modify? Our long-term goal is to investigate the role of natural language (NL) in a system for knowledge acquisition. The project as a whole is informed by existing graphical interfaces for KA and also by the computational demands of analyzing NL into a formal knowledge representation (KR). The full system for knowledge acquisition should include feedback to the user regarding the current contents of a knowledge base. The experiments presented in this paper fit into the larger work-in-progress by comparing methods for presenting this type of feedback to an informant.

Background and Related Work

Typically, a knowledge base is accessed through a formal *query language* for expressing queries and statements. A knowledge engineer must learn the syntax of the query language, and when s/he adds knowledge to a KB, s/he must impose the structure of this language on her/his ideas before teaching them to the machine.

Graphical tools

Some KA systems have used graphical representations to improve on this paradigm (SHAKEN [3] and WebOnto [4], among others). A user sees a graph of existing KB knowledge, and s/he modifies it as necessary to capture a new meaning. Experiments have shown this technique to be usable and useful for acquiring knowledge [1].

Natural language tools

Natural language (NL) provides another elegant alternative for knowledge entry. If facts can be entered in English, for example, then any speaker of English may "teach" a computer some of what s/he knows. It also allows us to acquire a vocabulary to associate with any new knowledge structure. This vocabulary is crucial for tasks where concepts from the knowledge base need to be identified later in text (e.g. information extraction). Several acquisition systems have explored NL-like interfaces such as syntactic templates and controlled language. These include Ontolingua [6], OpenMind [10], and Learner [2], among others.

Knowledge Representation

Each of the acquisition tools mentioned above stores knowledge in its own knowledge base, using its own formalism or representation. In this project we use the Scone knowledge representation system [5] as the underlying form for the KB. Scone is a semantic network, where concepts are *elements* in a graph. Links between concepts are graph edges, called *relations*. Scone offers the additional support of built-in inference mechanisms that can check whether a statement is true in the current knowledge base, and can perform hypothetical reasoning using *contexts*.

Problem and Hypothesis

In this paper, we explore the task of asking a user to verify knowledge that has already been entered. The experimental question we ask is, are some presentation styles for ontological knowledge more useful than others for verification by non-expert users? Our hypothesis is that one of the 2-D network structures which is commonly used to represent knowledge bases graphically [9] will be easiest to understand and use. This structure is described below as *Style-1*.

Experiments

We compare three display formats: *Style-1*, *Style-2*, and *Style-3*. *Style-1* is a graphical display of KB concepts as nodes in a network structure, with KB links between concepts displayed as labeled edges. *Style-2* is a graphical list-style display of KB concepts as documents, with sub-folders representing links. *Style-3* is a textual list of concepts and links in 3-ary expressions: (concept, link, concept). Examples are given below.

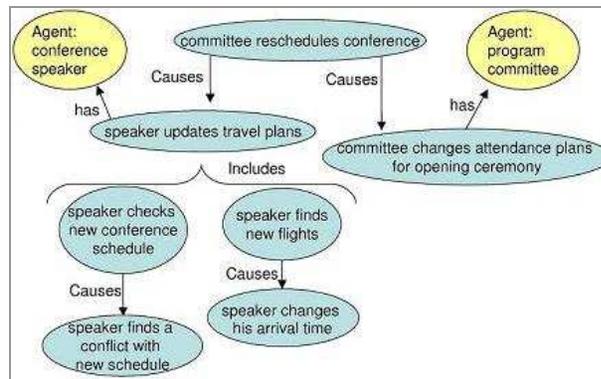


figure 1. Example figure using *Style-1*

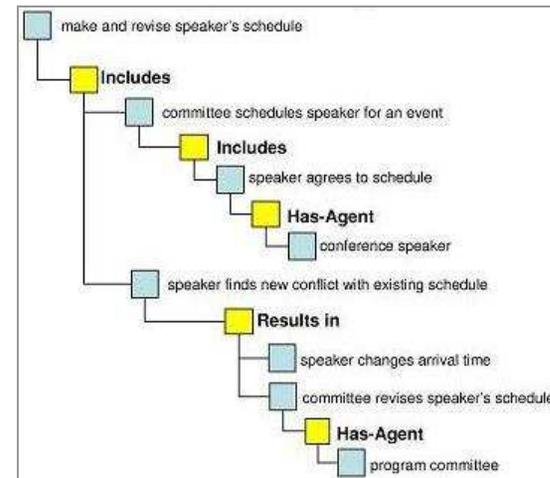


figure 2. Example figure using *Style-2*

```

(set event1 "negotiate presentation time for speaker")
(set event2 "committee and speaker agree on a schedule")
(set event3 "committee changes conference schedule")
(set event4 "speaker checks new schedule constraints")
(set event5 "speaker checks flight times")
(set event6 "speaker changes arrival time")
(set event7 "committee adjusts conference schedule")
(set event8 "committee confirms new schedule with speaker")

(event1 includes-subevents event2, event3, event6, event7)
(event3 has-preconditions event-2)
(event3 has-results event6)
(event6 includes-subevents event4, event5)
(event6 has-results event7)
(event7 has-results event8)
  
```

figure 3. Example figure using *Style-3*

Experimental design

The domain of the experiment is conference planning, represented by email communications among conference organizers. To prepare the experiment, we first selected three brief pieces of text from a corpus of emails and identified the events and agents appearing in each one. We then encoded these events in the Scone knowledge representation, along with the relations between them. This step produced three small, separate knowledge bases which we refer to as *Text-1*, *Text-2*, and *Text-3*.

Next we produced three figures for each piece of text, one in each of the display formats. This allowed us to compare the display formats while controlling for sources of bias that may have occurred in any one of the texts alone. The result is nine pairings of texts with presentation styles, shown here as a Cartesian product: (Text-1, Text-2, Text-3) X (Style-1, Style-2, Style-3)

These Text-Style pairings are the basis of the experiment. We create a unique diagram for each pairing, using the knowledge dictated by *Text-N* and the presentation format dictated by the *Style-M*. The diagrams are arranged, with control for ordering effects, into three different versions of a user study. We assigned 3 participants randomly to each version using a Latin square design with Style and Text as within-subject factors. This design is shown in table 1.

	Task 1	Task 2	Task 3
Subjects 1-3	Text-1,Style-1	Text-2,Style-2	Text-3,Style-3
Subjects 4-6	Text-3,Style-2	Text-1,Style-3	Text-2,Style-1
Subjects 7-9	Text-2,Style-3	Text3,Style-1	Text-1,Style-2

table 1. Latin square organization of the user studies

Participants in the study were given a series of exercises related to each diagram they saw. Exercises included interpretation questions (*What events cause other events in the diagram?*) and questions that involve modifying the diagram (*Modify the figure so that event1 causes event2*). We took objective measurements of usability by counting the number of correct answers and the number of correct modifications. We also took subjective measurements: participants rated their own confidence in their answers, and scored each diagram on a scale of 1 to 4 for being "easy to understand".

Results

We found the results of the experiment to be surprising. Our intuition favored Style-1, the graphical network style. As the tables below indicate, there was no strong evidence that this style of diagram was preferred by participants. In fact, the textual list style (Style-3) appeared to be as easy for participants in several respects. The study was small, and the statistical analysis revealed no significant effects, even with paired t-tests. But our intuition is that the preference for text, even dense Lisp-like text, indicates that natural language will be very relevant for the overall task of acquisition. That intuition must be verified in ongoing work.

Although not statistically significant, we can see some trends in our results that can be explored in later work. Table 2 shows that participants spent the most time working with diagrams in Style-1, the graphical network style.

	Mean time spent, in seconds	Var	Stdev
Style-1	480.00	24300	155.88
Style-2	433.33	37600	193.91
Style-3	433.33	18700	136.75

table 2. Experimental time results

	Mean correctness	Var	Stdev
Style-1	65	0.23	0.48
Style-2	73	0.20	0.45
Style-3	68	0.22	0.47

table 3. Experimental correctness results

	Mean Response	Var	Stdev
Style-1	2.11	0.61	0.78
Style-2	1.88	0.36	0.60
Style-3	2.11	0.61	0.78

table 4. Results for "figure is easy to understand", on a scale of 1 (easy) to 4 (hard).

Table 5 shows that participants were the most confident in their answers when working with Style-1. They are slightly less confident in their answers when working with Style-3 and Style-2.

	Average Response	Var	Stdev
Style-1	1.81	0.54	0.73
Style-2	2.00	0.69	0.83
Style-3	2.18	1.15	1.07

table 5. Results for "confident about my answers", on a scale of 1 (confident) to 4 (unsure).

Analysis and Conclusions

General findings

Overall, objective measures indicated that participants are able to quickly and confidently work with Style-1, the conceptual graph. But the simplicity of this presentation format may be misleading, as indicated by lower correctness numbers for Style-1. In spite of this, participants spent more time on tasks involving Style-1 than either of the other two styles.

Unexpected responses

Some participants in the pre-study misunderstood the following question in the exercise: "give some other events that could have been included in *event-name...*". They gave novel activities, rather than interpreting the diagram. While this wasn't the intended reading of the question, it indicates that in this context people can easily -- perhaps too easily -- be prompted to give new, domain-related knowledge. This could be helpful for our ongoing work, described in the final section.

Future Work

Our ongoing work builds on these experimental results. We plan to ask participants to interact with an on-line

tool that prompts them for knowledge in English. Several different prompt styles will be tested:

- Syntactic templates (___ is a precondition of Going to the Airport)
- Prompts that ask for a list of short statements (*List 5 things that are preconditions of Going to the Airport*)
- Prompts for free text related to a given event (*Tell me more about Going to the Airport*)

After knowledge has been entered, the participant can see what the computer has learned, and provide feedback related to how well the system “understood” what s/he has said. Results from the experiments described here indicate that we should display this knowledge using *Style-2* or *Style-3*, for the most reliable results. This ongoing work addresses several research questions: What are the different types of structure that can be used to scaffold data entry? Can these structures be mixed and matched as appropriate for different task- and topic-domains? And finally, Do different types of structure affect different aspects of natural language?

Acknowledgements

This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. NBCHD030010.

Citations

[1] Barker, K. et.al. A knowledge acquisition tool for course of action analysis. In *Proc Innovative Applications of Artificial Intelligence Conference* (2003).

[2] Chklovski, Timothy. LEARNER: A System for Acquiring Commonsense Knowledge by Analogy. In *Proc K-CAP 2003* (2003).

[3] Clark, Peter et.al Knowledge entry as the graphical assembly of components. In *Proc International Conference on Knowledge Capture* (2001), 22-29.

[4] Domingue, J. Tadzebao and WebOnto: Discussing, browsing, and editing ontologies on the web. In *Proc 11th Knowledge Acquisition for Knowledge-Based Systems Workshop* (1988).

[5] Fahlman, S.E. *Scone User's Manual*, <http://www.cs.cmu.edu/~sef/scone/>.

[6] Farquhar, A. and Fikes, R. and Rice, J. The Ontolingua server: A tool for collaborative ontology construction. *International Journal of Human-Computer Studies*, 46 (1997), 707-727.

[7] Goodman, Kenneth and Nirenburg, Sergei, eds. *KBMT-89: A Case Study in Knowledge-based Machine Translation*. Morgan Kaufmann, San Mateo, CA, USA, 1992.

[8] Harabagiu, S. and Maiorano, S. Acquisition of linguistic patterns for knowledge-based information extraction. In *Proc LREC-2000* (2000).

[9] Mahalingam, K. *The Java Ontology Editor (JOE)*. Center for Information Technology, Electrical and Computer Engineering Department, University of South Carolina (1996).

[10] Singh, Push et.al. Open Mind common sense: knowledge acquisition from the general public. *Lecture Notes in Computer Science volume 2519*. Springer-Verlag, Heidelberg, Germany, 2002.

[11] Valin, Jean-Marc and Stork, David G. Open Mind speech recognition. In *Proc ASRU99* (1999).