DARPA's High Performance Knowledge Base (HPKB) Program

Robert MacGregor
USC/Information Sciences Institute
macgregor@isi.edu

Deborah L. McGuinness
Stanford University
dlm@ksl.stanford.edu

The goal of DARPA's HPKB program is to demonstrate the practicality and utility of constructing and reasoning with very large knowledge bases [Cohen et al 98]. The contractors participating in this initiative were tasked to construct convincing applications in specific military domains.

The applications were and are constructed around ontologies and knowledge bases managed by general-purpose knowledge representation (KR) systems. The KR systems in use are Cyc (Cycorp), Ontolingua and ATP (Stanford), Ocelot and SNARK (SRI), Loom and PowerLoom (USC/ISI) and SME (NWU). These KR systems all support very expressive languages, and they have similar context mechanisms for subdividing knowledge. All of the systems except Loom have been extended during the program by special-purpose reasoners designed to accelerate and/or broaden the scope of their deductive processors. A variety of translators exist for translating between different pairs among these systems.

A high percentage of the program is devoted to manual construction of knowledge bases. Additional factual knowledge has been acquired using text extraction, "knowledge slurping", and extraction from structured textual sources such as the CIA World Fact Book.

Originally, the program was intended to support research into probabilistic reasoning as well as deductive reasoning. The difficulty of constructing sizable bases of probabilistic knowledge caused most of these efforts to migrate elsewhere. However, other kinds of reasoners are being employed in various applications, including analogical reasoning and geospatial reasoning (NWU), the EXPECT system (ISI), which reasons with quasi-declarative procedures, case-based reasoning (ISI), and qualitative simulation (Stanford and NWU).

The guiding force behind the applications are challenge problems that specify the domain of application, knowledge sources, example questions, and at testing time, test questions. An innovation of this program is the use of "parameterized questions" (PQs). A PQ is a structured English question that contains phrases that will be altered at testing time along pre-specified dimensions. Thus, each PQ represents a set of questions. For example, the general question: How is <term1> {different from/like} <term2> was instantiated as "How is a terrorist group's interest in increase prestige and influence different from a criminal organization's interest in increase prestige and influence? (The answer is they have different goals and targets – terrorists seek to increase prestige among their supporters to bolster their cause while criminal organizations target other criminal organizations in order to increase their chances of monopolizing their power and control in the illicit sector.) Each application builder has adopted a number of PQs, and is tailoring their knowledge bases and sometimes extending their underlying reasoning systems to reason competently with any questions generable by those PQs.

At the end of Year one, various developers had produced applications that exhibited impressive reasoning capabilities. However, there was little in the way of instrumentation for measuring or evaluating how the knowledge in the KBs contributed to each system's performance. This was partially remedied by post-hoc analysis of the applications. For Year 2, more careful mechanisms have been established for evaluating the applications and extensive metrics are being collected.

The manual effort expended to produce the ontologies and knowledge bases has been considerable. To the extent that this knowledge is *reusable*, this effort might be considered justifiable. Some of the instrumentation introduced into the systems during Year 2 is aimed at trying to quantify the amount of Year 1 knowledge that is reused in the Year 2 applications. Some experiments attempt to measure use of knowledge high up in an ontology relative to the use of more specific knowledge lower down in the hierarchies; a demonstrable use of higher-level (more abstract, more general) knowledge is seen as another demonstration of reuse.

In Year 2 some additional "critical component experiments" were introduced as well. These were aimed at extracting knowledge and also at merging knowledge bases. The text extraction critical component experiment aims to build frames from text input. The merging experiment takes two knowledge bases and uses Stanford's tools to support a knowledge engineer in a task of combining knowledge bases – thus helping identify when two terms should be merged into one or have links relating the two objects.

The follow-on program to HPKB will be called the Rapid Knowledge Formulation (RKF) program. That program will target the construction of tools that enable domain experts who are not versed in logic or other KR-related technology to construct usable knowledge bases. If RKF is successful, the cost of constructing knowledge bases will be reduced significantly, enabling the routine construction of large knowledge based applications.

References

[Cohen et al 98] P. Cohen, R. Schrag, E. Jones, A. Pease, A. Lin, B. Starr, D. Gunning, M. Burke., The DARPA High-Performance Knowledge Bases project, in *AI Magazine,* 19(4).