Creativity in Knowledge-bases

Debasis Mitra

Department of Computer Science
Florida Institute of Technology
dmitra@cs.fit.edu

Key Words: Computational creativity, Knowledge-base

Types of Creativity and Innovation

Human creativity is one of the most ill-understood concepts. There are two aspects of creativity – the creative process and the product of creation. During the process the creator often ignores how useful the product will be, rather the joy of creation drives the process. On the other hand the product is judged for creativeness by a relevant community at large, not necessarily by the creator. In case of innovation, distinction between these two aspects may not be so strong. Patent laws attempt to define the innovativeness of products. In the area of computational creativity we are interested in understanding and emulating the process to the extent that the product may be recognized as creative.

In some domains the amount of explicit knowledge needed to be coded may not be very significant (e.g., computer generated music). However, there are areas where sufficient amount of procedural as well as assertive knowledge is prerequisite to any creation. Scientific research is one of such areas. Our project tries to initiate studies in computational creativity in science. We are currently experimenting with a knowledge-base in physics.

Knowledge-bases and Operations

Primary operations on most knowledge bases (KBs) are: entering knowledge, modifying knowledge, and answering queries, similar to the operations in databases. There are different types of knowledge bases (KB). Most KBs are monolithic – every part of the KB is expected to have relationship with some other part. CYC™ is a good example of such a KB that attempts to build on common sense knowledge [Lenat and Guha, 1990]. Another type of KB uses rather abstract independent knowledge chunks or micro-theories that are instantiated and linked at the time of answering a query. AURA is such an experiment (http://www.ai.sri.com/project/aura) based on the KM knowledge representation language (Peter Clark and Bruce Porter. KM - The Knowledge Machine 2.0: *Users Manual*, University of Texas at Austin.).

The architectures of most current KBs are based on some variations of Frames. Description logic is sometimes used explicitly. Some KBs even uses quantification over the quantifiers for the sake of expressive brevity, thus getting beyond the realm of first order logic (e.g., CYC).

Possible Creative Behavior with KBs

The following behaviors may be considered creative while interacting with a KB.

Question generation: Even when a KB is not perfect in the sense that it is incapable of answering questions it should be able to, we have observed that it is possible to generate new questions from the KB. A typical KB does not provide such a mode of operation.

Suggestion on failure to answer queries: Some KBs are capable of providing explanation for its answer on an input query. This is considered an important feature for the credibility of a KB. However, when a KB fails to answer a question it is possible to diagnose where is the inconsistency in its reasoning or what additional information may make the query successful.

New concept learning: Concept learning is a holy grail of Knowledge representation research. In the past, studies in this area ushered into the new field of Machine learning. However, typical KBs still cannot learn new concepts from its interaction with its environment. During the process of reasoning (when it answers query or it is functioning under any operation) a system should recognize a worthy concept to be added to its KB.

Observation on data: Concept learning process may also involve data. Integration of KB and databases is rarely heard of [Johansson et al., 1996]. Datamining software typically uses procedures to work over data and does not have any capability to use pre-existing knowledge in the domain.

Introspection: Some computational creativity researchers consider a minimum amount of rudimentary self-awareness as a prerequisite to any creative software [Colton, 2008]. CYC encodes some explicit awareness of its KB and functionalities.

Methodology

Question Generation: A query typically involves a few asserted facts in the query (could be a null set in a trivial query) and a questioned topic that should be answered. The facts and the questioned topic are unified with the KB to find out the respective relations and then the relations are used to derive the value of the queried topic. A question generation mechanism may work in a similar way. Given a set of asserted facts, query generator can find out their relations with the other concepts in the KB and generate a question on a related topic (in the line of *inductive bias* generation [Cabral et al., 2005]).

Diagnosis: Query answering process often explores a few of the possible reasoning alternatives when a reasoning path fails because one or more slot values are not in the query (or in the KB, as the case may be). In case of such a failure the system could ask for the needed value in the query. Alternatively, it could “dig deeper” and see if a unpromising branch of the reasoning tree or a concept from unrelated area of the KB may provide the answer. This second alternative is often considered as a “Eureka” event when a human scientist finds an almost impossible answer.

Concept generation: Relational links between concepts within the KB are used by a reasoning algorithm. If there is a mechanism to recognize a temporarily produced chunk/subtree as useful new concept, then that may be properly abstracted and archived. Previous attempts to learn concepts in a KB (Porter et al., 1990) were driven by specific training.

Introspection: Meta-concepts about the KB may be considered as the basis of such introspection. Capability to learn, manipulate and use such meta-concepts in the reasoning process may make the latter more efficient and the system may be able to show unexpected creative behavior from this process.

Measurements of Success

One of the major challenges in computational creativity research is - how to measure success? One way is to let human judges decide if the product or output behavior of the system is considered creative enough. Another way may be to have some internal parameters. If a question is generated using a large number of concepts, then the process may be considered “difficult” and thus, the output as more creative (although all difficult tasks are not necessarily creative in nature). Similarly, the number of concepts used in answering a query may be used to justify the creativity of the output answer. A creativity-measuring indicator could be use of such introspection.

Experimental Set Up

We are currently using AURA knowledge base in high school physics to experiment with some of the ideas expressed in the previous sections. In the project HALO lead by SRI (http://www.projecthalo.com/), an AURA KB is being developed that is targeted to answer AP level questions on some selected topics in Physics. It is a nugget-based architecture where the independent abstract concepts are represented as Frames. Our current focus is to generate new questions over this KB. We will evaluate the difficulty level of a question using the number of concepts used. Difficulty level is one of the parameters to measure the usefulness and creativity of a question. We will also like to evaluate the usefulness of the questions from the domain experts and eventually cross-correlate human evaluation of creativeness of the questions with our internal metric of usefulness.

Conclusion

In this abstract we have outlined a few directions how creativity may be emulated while interacting with a KB. Possible methodologies to achieve those are also discussed here.

Acknowledgement: The work is supported by a NSF Grant IIS-0732566.

References

Bruce W. Porter and Ray Bareiss and Robert C. Holte (1990) "Concept Learning and Heuristic Classification in WeakTtheory Domains." *Artificial Intelligence* 45(1-2): 229-263.

Colton, S. (2008). “Creativity versus perception of creativity in computational systems.” *AAAI Tech Report SS-08-03* from Spring Symposium on *Creative Intelligent Systems* at Stanford, California.

Johansson, B., Shahsavar, N., Ahlfeldt, H., and Wigertz O. (1996) “Database and Knowledge Base Integration in Decision Support Systems.” In *Proc. AMIA Annual Fall Symp*osium: 249–253.

Lenat, D. B., and Guha, R. V. (1990) *Building Large Knowledge Systems: Representation and Interface in the Cyc Project*. Addison-Wesley Publishing, Reading, Massachusetts.

Cabral, J., Kahlert, R.C., Matuszek, C., Witbrock, M., Summers, B. (2005) “Converting Semantic Meta-Knowledge into Inductive Bias.” In *Proc. of the 15th InternationalConference on Inductive Logic Programming*, Bonn, Germany.