Knowledge Acquisition using Opencyc

Kiran N Kumar and **Suresh Deivasigamani**
School of Informatics and computer science
Indiana University
Bloomington, Indiana, 47405

Abstract

Knowledge acquisition is the process of acquiring knowledge from sources. There are many varied techniques to populate ontologies. This paper explores techniques in knowledge acquisition from user dialog using opencyc. We look at techniques to constrain the relationship when entering concepts and developed a system which allows the user to interact with opencyc without any knowledge of ontologies or opencyc.

Introduction

Knowledge acquisition is the process of converting inherent knowledge from a given source into a representable and understandable format for the machine. The process has come a long way from MYCIN(Shortliffe 1974) which relied on encoding the knowledge by hand in the form of rules and working with doctors by asking them a series of questions to determine the correct diagnosis.

The most widely used method for knowledge acquisition today is to use ontologies which represent complex real world relationships in the form of IS-A and HAS-A hierarchies.Ontologies can also be used to represent knowledge in other forms(Brewster et al. 2004). These ontologies are created from human knowledge by representing the relationship using a formalism which converts the knowledge from human form to machine form. The problems with representing in human is the ambiguity associated with the language or representation, we can associate different words to different meanings. This is the case with images as well as shown by the Rorschach inkblot test. In essence we use a association map for the system which can represent the association a human uses to identify and reason about objects and events.

The idea of learning from analogy is not old, it forms the basis learning in infants, children and adults alike. The problem can be divided up into finding analogies and using them appropriately(Winston 1980). Finding analogies is not hard part, we can all find some object or event sequence analogous to a concept, but using them appropriately is really hard, because we can form analogies of a single object for different concepts. If we take the example of an apple, it can analogous to a fruit which is edible or a computer manufacturer, deciding this is entirely in context which forms the problem. To solve this problem Doug Lenat devised the Cyc architecture which has grow into the opencyc platform for semantic knowledge usage and discovery.

OpenCyc

The opencyc architecture consists of various elements all of which form the basis for knowledge acquisition and the reasoning process associated with it. The knowledge base, Cyc KB consists of more than 3.2 million assertions describing more than 280,00 concepts and including more than 12,000 interrelating predicates(Matuszek et al. 2005). Then there is inference engine which forms the core of the reasoning process, it can take in assertions and assign truth values to it. These truth values are absolute but can encompass meta-assertions which help aid reasoning through augmentation.(Lenat 1995)

The cyc KB has an hierarchy of knowledge in its knowledge base, starting from microtheories, a set of assertions to describe a domain or topic. The microtheories can have shared assertion which means there could be concepts shared across domains but each of which has a specific contextual meaning in that domain. Along with these we have predicates which are used to calculate truth values of sentences and functions which can be used to form sentences.

The reason we are choosing opencyc as our knowledge base and inference engine for knowledge acquisition is precisely owing to the fact that we have a huge knowledge base which is already established and we can build on this knowledge base, which highlights the learning by analogy perspective. The usage of a lisp like scheme to store the knowledge also helps aid knowledge acquisition.

Related Work

There are many applications which perform knowledge acquisition in a specific domain and a few that also look at

common sense acquisition such as openmind. Openmind also looks at the approach of knowledge acquisition in a similar way, seeking user input to form assertions on knowledge contained in the knowledge base. To learn about new concepts, it asks the user to enter something which you would find or use on a known concept. Openmind gives users queries to extend its knowledge base.(Chklovski and Gil 2005)

This approach is good, but it does not allow rapid expansion of the knowledge base and forms contained structures of knowledge. Openmind also uses the IS-A relationship and has many internal relationships representations. This forms constrains on the knowledge that can be captured as opposed to maintaining the consistency of the knowledge. The user is never able to express new concepts or relationships among concepts which may be locally contained because of a internal representation of the relationship information.

Knowledge acquisition process

Opencyc allows anyone to enter knowledge into the system without associated information, we add an restriction which allows us to build a compact knowledge base based on the analogy of relationships among concepts. Opencyc supports two types of concepts - IS-A and GenLs. The IS-A as the name suggests represents the instance of a analogous object, in which we are saying object A is analogous to object B. On the other hand, the GenLs is a generalization relationship and represents the containment in an analogous object, in which we are saying object A is $containedin$ analogous object B.

Creating a concept

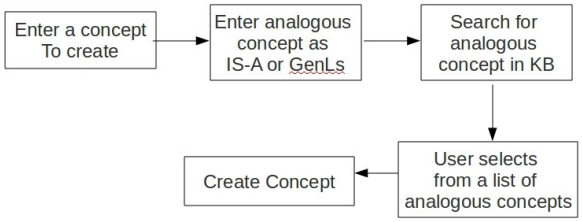

Figure 1: Creating a concept

We aid the creation of a concept by asking the user an analogous object. The problem here is the internal representation of CYC has its own naming mechanism and the user is generally not aware of this internal representation. We aid the process by querying the knowledge base for the related concept, and finding items that are similar to the name specified. The user is then asked to select from a list of functions that relate semantically to the concept intended by the user. If the user does not find an analogous concept, he will be required to create the foundational concept first. This follows from the premise that before you learn about a complex about you need to learn the fundamentals underlying that concept.

This approach not only captures human analogy but develops very knowledgeable fine grained knowledge bases, which makes the task of inference and semantic understanding very easy. There are definitely drawbacks to this approach, the user could easily select a very general concept to relate, but considering the objective is to represent the human analogy to the system we overlook this situation of a malicious intent user for now.

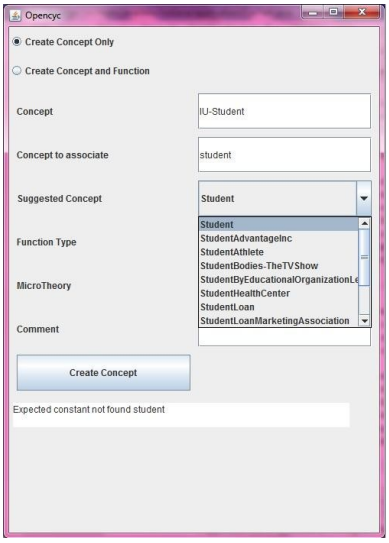

Figure 2: The UI for creating a constant - The user enters the concept to create and types the related concept, if present the concept is created, if not the drop down is populated with choices for the user.

Creating a function

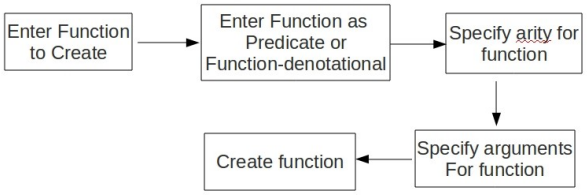

Figure 3: Creating a function

There are two ways we can create terms to use in sentences represented by logic, predicates and functions. As explained before predicates are used to calculate truth value of sentences and functions are used to build up complex sentences. These two features allow very expressive language representations and enable a user to create events and other related functions.

The predicates and functions both have the same format, they have a said number of arguments and the type for each argument, both of which are optional. This gives us powerful predicates and functions which can restrict the user from using it to represent malicious concepts or allow them to express ambiguous concepts relating to more than one domain.

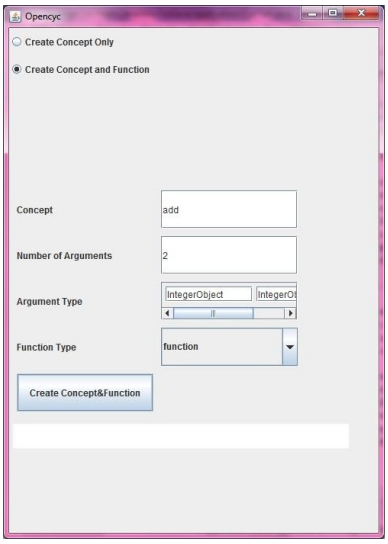

Figure 4: Creating a function

Opencyc provides various tools to integrate this feature and we can also perform many checks, apart from what the inference engine does, to maintain integrity of knowledge base.

Future Work

Opencyc provides no features for NL processing now. We intend to work on embedding a feature rich POS tagger and semantic parser with the opencyc KB and enable automatic learning of concepts. The information in opencyc could become decrepit over time and needs to be updated like the current president or the existence of Berlin wall. Opencyc allows you to kill a concept and add a new one but not update one. We intend to work providing an update feature which would have to work on confirm the update either by trusted sources or human user.

Conclusion

We came across a few issues with the knowledge engineering process, to represent a concept across microtheories is difficult and forming associations in terms of a known concept has a few scalability issues, adding the semantic parser would provide a definite relief to the retrieval process of analogous terms. There are many inference checks carried out when adding knowledge which make the knowledge acquisition a difficult process for a user. Integrating with an inference engine and proving a usable interface is an attractive option for knowledge acquisition from dialog.

References

Brewster, C.; O'Hara, K.; Fuller, S.; Wilks, Y.; Franconi, E.; Musen, M. A.; Ellman, J.; and Shum, S. B. 2004. Knowledge representation with ontologies: The present and future. *IEEE Intelligent Systems* 19:72–81.

Chklovski, T., and Gil, Y. 2005. Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In *Proceedings of the 3rd international conference on Knowledge capture*, K-CAP '05, 35–42. New York, NY, USA: ACM.

Lenat, D. B. 1995. Cyc: a large-scale investment in knowledge infrastructure. *Commun. ACM* 38:33–38.

Matuszek, C.; Witbrock, M.; Kahlert, R. C.; Cabral, J.; Schneider, D.; Shah, P.; and Lenat, D. 2005. Searching for common sense: populating cyc™ from the web. In *Proceedings of the 20th national conference on Artificial intelligence - Volume 3*, 1430–1435. AAAI Press.

Shortliffe, E. H. 1974. A rule-based computer program for advising physicians regarding antimicrobial therapy selection. In *Proceedings of the 1974 annual ACM conference - Volume 2*, ACM '74, 739–739. New York, NY, USA: ACM.

Winston, P. H. 1980. Learning and reasoning by analogy. *Commun. ACM* 23:689–703.