The Value Of Ontologies

Erik Eilerts, Dirk Ourston

Austin Info Systems, Inc.
301 Camp Craft Road
Austin, Texas 78746
eilerts@ausinfo.com, ourstond@ausinfo.com
http://www.ausinfo.com/kbdt

Abstract

Austin Info Systems (AIS) is developing the Open Source Automated Link Analysis Tool (OSALAT), an application that uses ontologies to help search open source repositories and processes the search results. The use of ontologies has greatly benefited OSALAT in a variety of ways, including the decoding the information contained within the open source documents. This document discusses the use of ontologies by OSALAT and then proposes some extensions to support Lessons Learned systems.

Introduction

Austin Info Systems (AIS) is developing OSALAT, an application that uses WordNet-like ontologies (Fellbaum 1998) to help search open source repositories and processes the search results. The use of ontologies is necessary as OSALAT must attempt to decode the information contained within the open source documents, extracting data such as times, people, and places. The ontologies have proven their usefulness for a variety of OSALAT tasks.

The motivation for the OSALAT system is to provide a better tool for intelligence analysts to use to extract data from open information sources. The analysts work can range from extracting information about well-known terrorists (military) to finding details about a competitor's product (commercial). OSALAT supports this by providing a set of tools for searching open sources, parsing relevant documents for storage in a repository and retrieving documents from the repository based on search criteria. Additional tools in the suite include a document clusterer, a key phrase extractor, a link analyzer, and a graphical interface.

This document discusses the use of ontologies by OSALAT and then proposes some extensions to support Lessons Learned systems.

Use of Ontologies in OSALAT

The first tool developed in support of OSALAT was the KB Editor, a tool to browse and edit ontologies. To populate the KB Editor, WordNet and several military thesauri were imported. The KB Editor differs from a knowledge base system in that its functionality is restricted to little more expressiveness than is found in WordNet. More details on the KB Editor are presented in (Eilerts, Lossau, and York 1999).

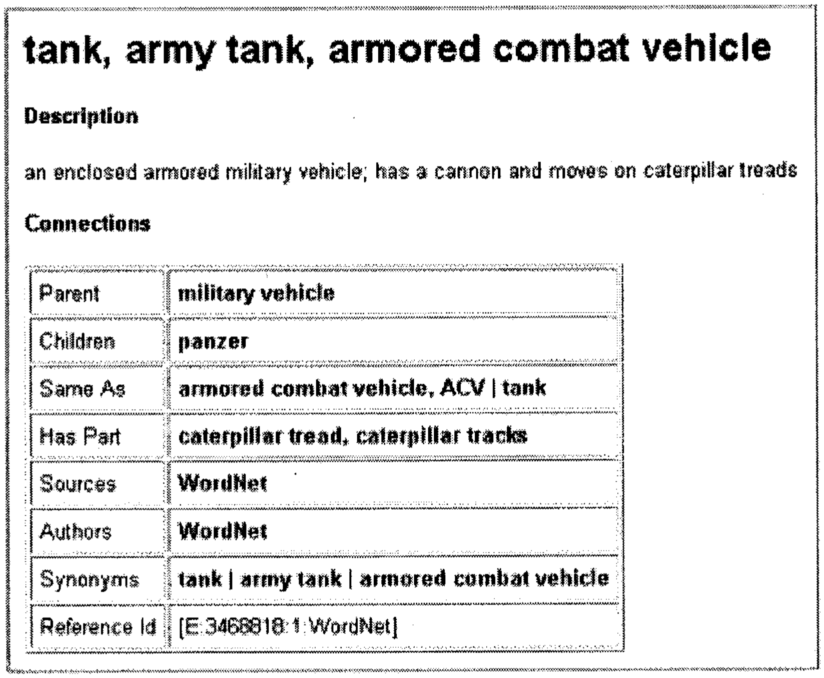
The KB Editor ontologies became the foundation for several, subsequent tools. The IIR Search tool uses the ontologies to help the user refine search criteria. For example:

1. The user enters the search phrase "tank"
2. The ontologies are searched, finding entries for an army tank and a tank of water
3. The alternate choices for tank are presented to the user as potential "concepts" for search
4. The user decides which of these concepts is the one they are actually searching for, say army tank
5. The search tool uses the selection to further refine the user's search.

For example, the query could be constructed as: +army tank +military tank –water tank, where military tank is added since it's a synonym of army tank and water tank is negated since it was a concept that the user didn't select. This query can be further expanded using relationships between these concepts and other concepts in the ontology.

Indexing of documents is a more complicated task. Documents are first analyzed to extract their text and tag their parts of speech (nouns, verbs, ...). The next step is to cluster nouns into noun phrases. The last step is to match these segments (nouns, noun phrases, verbs, etc.) with concepts in the ontologies. As each segment could potentially match multiple concepts in a given ontology (see step 2 above), the challenge is to select the appropriate concept automatically with little user intervention. Indexing can be done *actively,* by selecting a set of documents and indexing as described above, or *passively,* by indexing on the fly as search documents are returned to the user. This is an ongoing area of investigation.

Ranking of documents takes a document and a search criteria and returns a score that indicates how well the document meets the search criteria. This process begins by converting the document into concepts. Since it is unlikely that a document's concepts will exactly match the concepts of the search criteria, relationships between concepts are explored. For example, in a search for "tank", a document may have a reference to a Panzer, which is a type of army tank, but no specific mention of the word tank. By following the parent/child relationship between army tank and Panzer, the document will get a better score because of this relationship, while a typical keyword search may have rejected the document outright for not having a specific reference to "tank". Figure 1 shows a picture of the army tank concept in the KB Editor and its connection to Panzer.



tank, army tank, armored combat vehicle

Description

an enclosed armored military vehicle; has a cannon and moves on caterpillar treads

Connections

Parent	military vehicle
Children	panzer
Same As	armored combat vehicle, ACV | tank
Has Part	caterpillar tread, caterpillar tracks
Sources	WordNet
Authors	WordNet
Synonyms	tank | army tank | armored combat vehicle
Reference Id	[E:3468818:1:WordNet]

Figure 1: A Concept for Army Tank

The tools described in this section make up the core of the OSALAT application.

OSALAT Example

A typical OSALAT usage might be:

- The user enters a query.
- The query tool helps the user to refine their query into specific concepts
- A search tool converts the concepts into search engine specific query strings and sends the query to a variety of information sources, such as Alta Vista, Excite, etc.
- The ranking tool is used to score the hits extracted from the information sources and reject spurious hits.
- The user specifies which of these final hits are of most interest. These documents are stored in a local repository for future retrieval. They are in indexed in the repository using the indexing tool.

Discussion

The interesting aspect in the development of these tools is the role that the ontologies play. Every aspect of the tool is somehow influenced by their use.

Several reasons for this include:

- Ontologies define relationships between concepts, providing much more understanding than simple keywords
- Concepts are defined using a collection of synonyms, making it easier to identify relevant terms, even if they are not exactly the same as the input search term.
- Ontologies can be specialized into different subject areas, giving the users a mechanism to handle domain-specific issues.
- The distance between concepts in an ontology is a valuable metric of which no similar value can be computed from keywords.

Lessons Learned Extensions

A Lessons Learned (LL) system is one that captures and stores experiential knowledge for later reuse. OSALAT can be extended to support LL systems in several ways.

A principal task would be to extract information from documents for insertion into Lessons Learned systems. For example, CALL (the Center for Army Lessons Learned, http://call.army.mil) has over 100,000 lessons learned documents covering all aspects of the US Army. These lessons are developed using a bottom-up learning strategy similar to that described in (van Heijst 1996). All of these documents are free text with few attributes encoded. The development of a LL system to encapsulate all of this information will be a difficult task to do manually. OSALAT could be used as a preprocessor to automatically extract and codify data from these documents for future insertion into an Army LL system.

Another possible task for OSALAT is intelligent searching of LL systems. Since lessons may be stored in free text documents, retrieving the appropriate lessons requires a search of both the text and the attributes stored by the LL system. As a first step, OSALAT can be used to help narrow the focus of the user's query. It can also be used to convert queries into application specific attributes.

Lastly, OSALAT itself can function as a LL system. The user can use the search tool to find relevant lessons learned documents. These documents can be indexed and stored in a repository. Later, another search can be performed on the repository to retrieve the relevant documents.

These tasks provide examples of how OSALAT's use of ontologies can be extended to benefit to Lessons Learned systems.

Conclusions

OSALAT has centered its functionality around the use of ontologies. This has proven to be beneficial as the ontologies drive many of OSALAT's tasks. Ontologies can also benefit Lessons Learned systems at least in the areas described by the OSALAT Lessons Learned extensions, and likely in many other areas as well.

Acknowledgements

The work reported on here was supported in part by a CECOM C2SID Phase II SBIR, contract number DAAB07-98-C-D027.

References

B. Chandrasekaran, John R. Josephson, and V. Richard Benjamins. What Are Ontologies, and Why Do We Need Them? IEEE Intelligent Systems, Vol. 14, No. 1, January/February 1999.

Eilerts, E., Lossau, K., and York, C. 1999. Knowledge Base Discovery Tool. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, 904-905. Boston, Massachusetts: MIT Press.

Fellbaum, C. ed. 1998. *WordNet: An Electronic Lexical Database.* Cambridge, Mass.: MIT Press.

Mayk, et.al. 1998 A Knowledge Based Doctrine Tool for Command and Control. In Proceedings of the Command and Control Research and Technology Symposium.

Andre Valente, Thomas Russ, Robert MacGregor, and William Swartout. Building and (Re)Using an Ontology of Air-Campaign Planning. IEEE Intelligent Systems, Vol. 14, No.1, January/February 1999.

van Heijst, G., van der Spek, R., & Kruizinga, E. (1996). Organizing Corporate Memories. In Proceedings of the 10th Banff Workshop on Knowledge Acquisition for Knowledge-Based Systems, 1996. Available online: http://ksi.cpsc.ucalgary.ca/KAW/KAW96/KAW96Proc.html