OWL as a Target for Information Extraction Systems

Clay Fink, Tim Finin, James Mayfield and Christine Piatko

Johns Hopkins University Applied Physics Laboratory and the
Human Language Technology Center of Excellence

Abstract. Current information extraction systems can do a good job of discovering entities, relations and events in natural language text. The traditional output of such systems is XML, with the ACE Pilot Format (APF) schema as a common target. We are developing a system that will take the output of an information extraction system as APF documents and directly populate a knowledge base with the information extracted. We report on an initial OWL ontology that covers the APF schema, a simple program to convert a set of APF documents to RDF data and a demonstration system build with Exhibit to view the results.

Keywords. Information extraction, knowledge base, Semantic Web, OWL, ACE, APF, XML

1 Introduction

State of the art information extraction systems are capable of doing a good job of identifying and extracting useful information in documents, including references to entities (e.g., people, organizations, places), relationships between entities (e.g., employer, president of, located at) and events (e.g., the death of a person, the merger of two companies). The output of most current information extraction systems is an XML document, which is mapped into a relational database or data structures in an application. The Calais system developed by ClearForest [ref] for Reuters is a notable exception; it produces results in RDF. For many uses, this is adequate, for others we would like to map the extracted features into instance level data in a knowledge base.

Much of the research on information extraction systems has been done as part of the annual Automatic Content Extraction (ACE) program that has been run by the US National Institute for Standards and Technology since 1997. This program aims to support the development of automatic content extraction technology to support automatic processing of human language in speech and text form. Over the years, the ACE program has developed a complex XML DTD that applications can use to encode their output. An application is required to accept a text document, such as a newswire article, and produce an XML document using the ACE Pilot Format (APF) DTD. This document identifies the entities, relations and events found in the text along with the strings ("mentions") associated with them, identified by their offset positions within the text. In addition, various linguistic features are included in the representation. The applications that participate in an ACE evaluation are expected to process a collection of documents, perhaps including 10,000 documents, and are "scored" by how well their answers agree with the "ground truth", which is specified by human judgments made by trained assessors.

We conducted a pilot study with three objectives. The first was to develop an **ACE OWL Ontology** (AOO) that could represent, in an intuitive and natural way, the information covered in the APF DTD. Additional desiderata included extensibility and interoperability. We wanted an ontology that would go beyond the simple APF schema and support richer and more articulated semantic models as information extraction systems mature. We also wanted AOO to support interoperability and integration with other widely used data and knowledge resources. The second objective was to explore how system **evaluation** could be done in a scenario where both the system's output and ground truth are given as a populated knowledge base (e.g., a collection of instances). A final objective was to demonstrate how the information in a knowledge base could be used to support future extraction tasks and support them in ways that were better than could be done by information stored in a more traditional database.

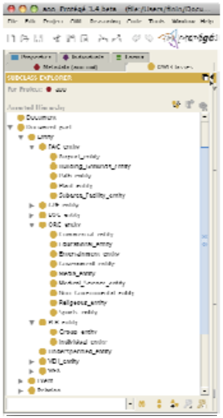


Fig 1. The AOO ontology is based on the ACE Pilot Format DTD widely used by information extraction systems.

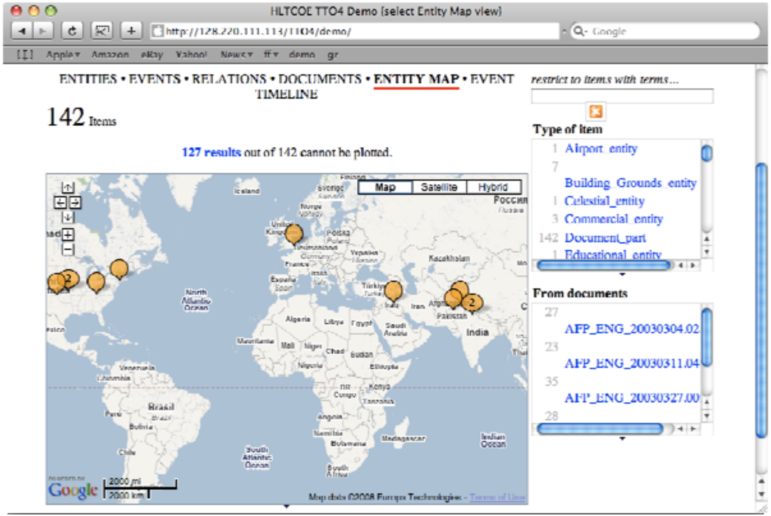
2 The Ace OWL Ontology

Our initial approach was to design an OWL ontology that could directly represent the information in the APF DTD [ref] but to do so in ways that exploit the features and strengths of RDF and OWL. Our initial AOO design [ref] contained 165 classes and 63 properties. We expressed the information, explicit and implicit, in the APF DTD used in the 2005 ACE evaluation and extended it to cover the cross-document entity resolution task added in the ACE 2008 evaluation. In our design, we tried to model things in ways we thought natural in OWL DL, resulting in an ontology with AL-HIF(D) expressivity.

The APF DTD has features intended to capture the semantics of the domain without directly expressing them. The key semantic classes are entities, events, relations, time expressions and mentions. The first three have a type and subtype attributes. For example, the entity class includes the types for facilities, geopolitical entities, locations, organizations, persons, vehicles and weapons. Each type has multiple subtypes. Organizations, for example, can be commercial, educational or one of seven additional subtypes. The DTD is able to specify that an object can only have one type and one subtype, but is not able to state that a set of subtypes must be associated with a given type. It is possible, for example, to specify that an object is of type *geopolitical entity* and subtype *celestial object*. OWL, of course, makes it easy to capture constraints between types and subtypes.

3 Evaluating a KB

Evaluating the output of a system designed to populate a knowledge base is more complex that evaluating one whose output is in the form of an XML document or content for database tables. This is true even when the output is restricted to be at the instance level and includes no content that extends or enriches the knowledge base's schema. The primary difference between the two evaluation scenarios is that while extraction output has a single right answer, in a KB system there may be many, even an infinite number, of sets of KB instances that count as a valid way to populate the KB.


HLTCOE TTO4 Demo [select Entity Map view]
http://128.220.111.113/TT04/demo/
Google
Apple▾ Amazon eBay Yahoo! News▾ ff▾ demo gr
ENTITIES • EVENTS • RELATIONS • DOCUMENTS • ENTITY MAP • EVENT TIMELINE
142 Items
Map
Satellite
Hybrid
Google
2000 mi
2000 km
Map data ©2008 Europa Technologies - Terms of Use
restrict to items with terms...
Type of item
From documents

127 results out of 142 cannot be plotted.

1	Airport_entity
7	Building_Grounds_entity
1	Celestial_entity
3	Commercial_entity
142	Document_part
1	Educational_entity

27	AFP_ENG_20030304.02
23	AFP_ENG_20030311.04
35	AFP_ENG_20030327.00
28	

Fig 2. We used the Simile sweet of tools, including Exhibit, to demonstrate the potential to quickly build a Web application using the RDF form of the information extracted from text.

We define the abstract KB evaluation task as follows. Given a fixed KB schema and a set of instances that populate it (a combination of any pre-existing instances from the KB plus the extracted instances), evaluate the resulting KB model for accuracy. The evaluation consists of two parts. The first is to ensure that sentences that should be in the model are in fact in the model, either explicitly or through inference using the given KB schema. The second part is to show that the model does not contain false assertions or extraneous sentences for which there is no support or derivation from the original text sources (e.g., facts that should not be derivable from the input). The evaluation task is greatly simplified if we can ensure that the KB model will be finite and if there is always a clearly defined minimal model. By making reasonable restrictions on the subset of OWL used to define the KB schema, we will be able to enforce these conditions. We plan to use the RDF delta approach described in [ref] as the foundation of our evaluation methodology. This approach proposes a comparison function for RDF KBs, Δ_{dc} (Delta dense+closure) that produces a semantically sound and compact characterization of the difference between two RDF knowledge bases.

4 Demonstration prototype

The demonstration prototype was based on MIT's Simile tool set. We used the Babel tool to convert a N3 serialization of the KB to JSON. The resulting JSON was used as the input to the Exhibit tool which gave us different views of the data. One view was a hierarchal view of the data by class. Another was a timeline showing the temporal relationships between event entities. The third view was a map showing the position of geographic entities in the KB in relation to each other.

The Simile tools were easy to use and we were able to quickly go from APF translation to providing views of the data that can give users the ability to validate and gain a better understanding of the co-reference results.

Deleted: ¶

4 Discussion

Converting the APF XML to OWL that instances the AOO ontology was straightforward. The initial version of the AOO is close to a one-to-one mapping from version 5.11 of the APF DTD. Some refactoring could be done in future version of the AOO to simplify the transformation but we need to be careful not to loose any of the semantics captured in the original DTD.

Deleted: 9

We used the Jena Java API [6], for RDF/OWL (version 2.5.4) to populate the instance documents. Entailments were initially generated using the Jena OWL Micro reasoner. This reasoner supports the RDFS axioms plus a small subset of the OWL axioms. We found that the default OWL reasoner and the OWL Mini reasoner were too computationally intensive even on a KB of less than 1000 entities. Since there was little added value provided by the entailments generated by the OWL Micro reasoner we opted for serializing the KB to N3 and generating the desired entailments using N3 and CWM. Using the Jena rule reasoner with a set of custom rules would be a possibility for future investigations.

The initial version of the KB was a N3 file serialized from a Jena model. In the future we would opt for using a triple store, possibly the Jena persistence mechanism or Oracle's triple store technology.

We have concerns about how this approach scales, both in the manipulation of the KB and in the use of the Exhibit visualization tools. The ACE 2008 evaluation will involve processing 10,000 documents and produce a knowledge base with five to ten million initial triples. These might easily increase ten-fold if the ontology is rich with axioms and reasoning is done promiscuously. It is not clear that the additional facts will be useful for any extant applications.

We can address this scaling problem in several ways. Some careful refactoring of the AOO ontology would help make the APF-OWL conversion easier by eliminating redundant concepts and relationships, as long as the refactoring preserves the DTD's semantics. That approach, however, is "lossy" and would prevent roundtrip conversions between APF and AOO documents. It is not clear to us if that is a major issue or not. Mention level information could also be segregated and stored in a separate KB or in a relational database to reduce the size of the KB.

A longer term concern is raised by the prospect of how best to use a language like OWL to represent the content of documents that contain contradictory information. Our current approach is straightforward, but ultimately problematic: we derive simple facts from a text and add them as assertions to a common knowledge base. If it extracts facts that are contradictory (e.g., "Pat is male; Pat is female."), even in two different documents, the knowledge base will contain contradictory facts. A natural way to address this in a logic-based representation system would be to introduce propositional attitudes, e.g., "Document 14972 asserts that Pat is male." Doing so in OWL, however, raises both theoretical and practical issues that we are not sure we are prepared to take on.

5 Conclusion

We designed an OWL ontology based on the ACE Pilot Format representation that is used by many information extraction systems. The ontology makes explicit constraints and axioms that heretofore were only implicit in the representation and allows the extracted entities, relations and events to be represented in RDF. We found that this facilitated manipulating the output and creating a simple mashup using the MIT Simile suite of tools.

References

[1] G. Doddington, A. Mitchell, M. Przybocki, L. Ramshaw, S. Strassel and R. Weischedel, The Automatic Content Extraction (ACE) Program--Tasks, Data, and Evaluation, Proceedings of the Fourth International Conference on Language Resources and Evaluation, pp. 837-840, May 2004

[2] Dimitris Zeginis, Yannis Tzitzikas, and Vassilis Christophides, On the Foundations of Computing Deltas Between RDF Models. ISWC/ASWC 2007: 637-651.

[3] ACE (Automatic Content Extraction) English Annotation Guidelines for Entities, Version 5.6.1 2005.05.23, Linguistic Data Consortium, University of Pennsylvania, 2005.

[4] E. Boschee, R. Weischedel and A. Zamanian, Automatic Information Extraction, Proceedings of the 2005 International Conference on Intelligence Analysis, McLean, VA, 2-4 May 2005.

[5] Rueters Calais Web Service, http://opencalais.mashery.com/

[6] B. McBride, Jena: a semantic Web toolkit, IEEE Internet Computing, v6m n6, pp. 55-59, 2002.

Deleted: 9