Relational Recognition for Information Extraction in Free Text Documents

Erik J. Larson
University of Texas at Austin
Innovation, Creativity, and Capital Institute
elarson@icc.utexas.edu

Todd C. Hughes
Lockheed Martin
Advanced Technology Laboratories
thughes@atl.lmco.com

Abstract

Information extraction tools provide an important means for distilling content from free text documents, and knowledge-based tools provide an important means for automatically reasoning over statements expressed as well-formed tuples. A number of techniques deliver reliable extraction of entities, less reliable extraction of relations, and poor extraction on entity-entity-relation tuples. However, tuple extraction is needed to bridge the gap between free text and knowledge-based applications. We describe an information extraction system and experiment that demonstrates accurate tuple extraction in a selected domain.

Introduction

Information extraction technology is among the most critical weapons in the arsenal for the intelligence analyst. These tools enable the processing of free text documents at a volume humans cannot achieve. With petabytes of documents waiting to be searched, these tools are not merely a convenience but a necessity in the discovery of critical intelligence for homeland security.

In addition, research in ontology-based reasoning tools shows promise in their ability to automate significant portions of intelligence analysis processes. Importantly, most of these reasoners operate over knowledge bases comprised of statements expressed as well-defined (relation-entity-entity) tuples. The RDF knowledge representation format is now emerging as the standard format for expressing these tuples.

The information extraction research community has given considerable attention to the task of named entity recognition, thanks in part to events such as the NIST Text Retrieval Conference (TREC) (Vorhees 2003) and Message Understanding Conference (MUC) (Chinchor 1998). This research area has been rigorously studied, and a number of commercial products are now available that perform entity extraction.

However, a collection of extracted entities is not a knowledge base and is therefore of limited use to ontology-based reasoning tools. Further, very little semantic understanding is gained from summaries of extracted entities alone. Knowing that John Smith is a person and Lockheed Martin is a company is of little use to automated reasoners if relational information is not also available: `(promotedBy JohnSmith LockheedMartin)` and `(terminatedBy JohnSmith LockheedMartin)` give radically different pictures of John Smith and his connection to Lockheed Martin.

Currently information extraction technology cannot interoperate with these reasoning tools because they extract only entities but not the relations between them. The task of creating syntactically well-formed tuples from the output of entity extractors is largely a manual, onerous, and error-prone task that only impedes intelligence analysis. Clearly, technology for automating the generation of tuples is needed.

Relational Recognition

Currently information extraction technology cannot interoperate with these reasoning tools because they extract only entities but not the relations between them. The task of creating syntactically well-formed tuples from the output of entity extractors is largely a manual, onerous, and error-prone task that only impedes intelligence analysis. Clearly, technology for automating the generation of tuples is needed.

However, the automatic extraction of both entities *and* relationships from free text is still a relatively unexplored research area. Notably, MUC now includes a Template Relation (TR) track that addresses the recognition of entities and relations between them. However, accuracy rates on the TR track have notoriously been among the lowest.

Yet the analysis of natural language texts for the purposes of automated semantic representation and reasoning clearly requires relational recognition techniques.

Relational recognition techniques typically exploit part of speech and syntactic parse information in addition to extraction rules or templates. Challenges such as the disambiguation of word senses has made full syntactic parses of natural language sentences a difficult and ongoing research problem. Information extraction techniques for relational recognition often resort to "partial parse" strategies of noun, verb, and prepositional phrase chunks that yield valuable information about how entities may be related to each other. In addition, the existence now of large annotated corpora such as the Penn Treebank's *Wall Street Journal* corpus have made it possible to use statistical machine learning techniques to learn syntactic chunks and other phenomena, and even relational patterns in free text to some limited extent.

The research on relational recognition conducted at the University of Texas at Austin with the support of Lockheed Martin Advanced Technology Laboratories has resulted in a prototype that uses statistical machine learning techniques to automatically extract entities and relations between entities in free text. The system was conceived and developed specifically to bridge the performance gap between named entity and relational recognition. It uses a pipeline architecture that performs a sequence of processing steps to tokenize, tag, and parse sentences, then extracting their relevant features to identify entities and relations. A post-process filters duplicate tuples and unnecessary entity occurrences such as acronyms, and performs other procedures that return a minimal complete assertion set given the input.

Protoype Relation Recognition System

To provide a proof-of-concept of the relation recognition system, our group has developed a Java-based application that provides a graphical interface for the loading of a document or set of documents in a directory, running them through the information extractor, displaying the assertions that result, allowing some editing of the assertions, and providing persistent storage in the form of the Jena2 RDF knowledge base. These can be queried using either the standard RDQL language or an inference model that computes subsumption and other RDF-supported inferences. The application, when run in batch mode and completely automatically, populates a KB with assertions from free text input, and consequently can be used to rapidly speed up database population for the purposes of knowledge sharing and reasoning of a domain or domains of interest to intelligence analysts.

We chose the employment domain to demonstrate capability. We constructed a corpus consisting of 272 documents describing employment events taken from PRNewswire and other newswire sites. The documents were hand annotated according to a custom built XML schema similar to the MUC annotation format (i.e., ENAMEX elements are used for named entities). Relations are explicitly captured by specifying a relation tag ("RELEX") and a set of attributes with names indicating the type of entity and values indicating the ID number of that entity in the annotated document. Figure 1 shows an example of relational markup generated by our system.

```
<SENT>
<RELEX id="1001" type="EMPLOYMENT"
predicate="EMP1" person="18" org="22"
title="19" />
Prior to this position,
<ENAMEX id="18"
type="PERSON">Pellett</ENAMEX>
was
<ENAMEX id="19" type="TITLE">Director of
Operational Excellence</ENAMEX>
at the
<ENAMEX id="20"
type="LOCATION">Chester</ENAMEX>
site, which
<ENAMEX id="21"
type="ORGANIZATION">OMNOVA
Solutions</ENAMEX>
' former parent,
<ENAMEX id="22"
type="ORGANIZATION">GenCorp</ENAMEX>
, acquired from
<ENAMEX id="23"
type="ORGANIZATION">Sequa
Chemicals</ENAMEX>
in late
<TIMEX id="24" type="DATE">1998</TIMEX>
.
</SENT>
```

Figure 1: Automatically generated markup of free text sentence. The markup engine recognizes both entities and their relations with each other.

The ontology for the current system consists of 14 employment predicates: `Hire`, `Appoint`, `Terminate1`, `Terminate2`, `Terminate3`, `Report`, `Promote`, `Replace`, `Emp1`, `Emp2`, `Emp3`, `Found`, `Offer`, `Accept`. Each predicate has a set of minimum instantiation conditions (MICs) which determine the minimum number of arguments required to make a legal assertion with that predicate. For instance minimally an `Emp1` relation representing the "employed by" property must take an instance of class `Person` in the first argument position and an instance of `Organization` in the second argument position. Temporal information such date or end date as well as duration of employment, instantiates an `Emp2` or `Emp3`, respectively. Where there are specific morphological clues (verbal triggers such as "hired" or "appointed") that indicate a more specific relationship, the sentence and its constituents is mapped to the most specific predicate in the ontology that describes the relation (e.g., "Hire" and "Appoint"). If no additional information is known yet an employment relation has been recognized after processing, the default `Emp` relations are used.

Experimental Results

It was necessary to validate three different pieces of the system pipeline to draw any conclusions about relational recognition performance. First, to test named entity recognition we used the YamCha SVM package which uses TinySVM to implement SVM (Kudo and Matsumoto 2003). Feature selection was limited to capitalization, part of speech, and test for inclusion in word lists constructed from the training corpus. On 10-fold cross validation tests we achieved an fMeasure of 93% using 90% of the training corpus for training an SVM classifier and 10% for testing label predictions.

To test the relevance of a sentence (i.e., whether the sentence contains an employment relation given a specification of an ontology of employment concepts and relations) we used a decision tree based approach, Logistic Model Theory, implemented in the WEKA machine learning package (Witten and Frank 1999) and trained on features that exploit the inclusion of named entity information from the prior named entity recognition (NER) result. LMT tests produced an average fmeasure of 95% on 10 fold cross validation testing using the same 90/10 train/test split as in the NER testing.

Finally, to test the actual relational recognition performance we trained an SVM using as input sentences classified as relevant and with named entities recognized from NER processing. The SVM classifies each named entity in an employment relevant sentence as either a participant ("P") or non-participant in an employment relation from our ontology ("N"). The SVM produced a somewhat surprising fMeasure of 96% again on 90/10 splits from the corpus using 10 fold cross validation. Using actual NER and relevance results from unseen data resulted in only a modest drop in performance (to 93%).

Conclusion and Future Work

The prototype system described here begins to address the challenge of linking free text with knowledge-based applications. Its processing pipeline can be used to construct relations automatically within a domain of interest with the restriction that there is a single relation.

Future work will involve applying this pipeline to other domains of interest, such as financial transactions, commerce, communications, and terrorism. In addition, further research with multiple intra-sentential relations is currently underway with some promising additional transformations of the multi-relation problem into simple and well-understood classification techniques.

References

Chinchor, N. 1998. Overview of MUC-7/MET-2. In *Proceedings of the Seventh Message Understanding Conference* (MUC-7).

Kudo, T. and Matsumoto, Y. 2003. Fast Methods for Kernel-Based Text Analysis. *ACL 2003*: 24-31.

Witten, I. and Frank, E. 1999. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. San Francisco, CA: Morgan Kaufmann.

Voorhees, E. 2003. Overview of TREC 2003. *NIST Special Publication SP 500-255: The Twelfth Text Retrieval Conference (TREC 2003)*: 1-13.