Towards Knowledge Acquisition from Information Extraction

Chris Welty and J. William Murdock

IBM Watson Research Center
Hawthorne, NY 10532
{welty, murdock}@us.ibm.com

Abstract. In our research to use information extraction to help populate the semantic web, we have encountered significant obstacles to interoperability between the technologies. We believe these obstacles to be endemic to the basic paradigms, and not quirks of the specific implementations we have worked with. In particular, we identify five dimensions of interoperability that must be addressed to successfully populate semantic web knowledge bases from information extraction systems that are *suitable for reasoning.* We call the task of transforming IE data into knowledge-bases *knowledge integration,* and briefly present a framework called KITE in which we are exploring these dimensions. Finally, we report on the initial results of an experiment in which the knowledge integration process uses the deeper semantics of OWL ontologies to improve the precision of relation extraction from text.

Keywords: Information Extraction, Applications of OWL DL Reasoning

1 Introduction

Ontologies describe the kinds of phenomena (e.g., people, places, events, relationships, etc.) that can exist. Reasoning systems typically rely on ontologies to provide extensive formal semantics that enable the systems to draw complex conclusions or identify unintended models. In contrast, systems that extract information from text (as well as other unstructured sources such as audio, images, and video) typically use much lighter-weight ontologies to encode their results, because those systems are generally *not* designed to enable complex reasoning.

We are working on a project that is exploring the use of large-scale information extraction from text to address the "knowledge acquisition bottleneck" in populating large knowledge-bases. This is by no means a new idea, however our focus is less on theoretical properties of NLP or KR systems in general, and more on the realities of these technologies *today*, and how they can be used together. In particular, we have focused on state-of-the art text extraction components, many of which consistently rank in the top three at competitions such as ACE (Luo, et al, 2004) and TREC (Chu-Carroll, et al, 2005), that have been embedded in the open-source Unstructured

Information Management Architecture (UIMA) (Ferrucci & Lally, 2004), and used to populate semantic-web knowledge-bases.

This, too, is not a particularly new idea; recent systems based on GATE (e.g. (Popov, et al 2004)) have been exploring the production of large RDF repositories from text. In our project, however, we are specifically focused on the *nature of the data* produced by information extraction techniques, and its *suitability for reasoning.* Most systems that we have come across (see the related work section) do not perform reasoning (or perform at best the most simplistic reasoning) over the extracted knowledge stored in RDF, as the data is either too large or too imprecise. This has led many potential adopters of semantic web technology, as well as many people in the information extraction community, to question the value of the semantic web (at least for this purpose). We believe this community can be important in helping drive adoption of the semantic web.

In this paper we will discuss our general approach to generating OWL knowledge-bases from text, present some of the major obstacles to using these knowledge-bases with OWL- and RDF-based reasoners, and describe some solutions we have used. Our research is not in information extraction, ontologies, nor reasoning, but in their combination. Our primary goal is to raise awareness of the real problems presented by trying to use these technologies together, and while we present some solutions, the problems are far from solved and require a lot more attention by the community.

2 Related Work

Research on extraction of formal knowledge from text (e.g., Dill, et al. 2003) typically assumes that text analytics are written for the ontology that the knowledge should be encoded in. Building extraction directly on formal ontologies is particularly valuable when the extraction is intended to construct or modify the original ontology (Maynard, Yankova, et al. 2005; Cimiano & Völker, 2005). However, there is a substantial cost to requiring text analytics to be consistent with formal ontology languages. There are many existing systems that extract entities and relations from text using informal ontologies that make minimal semantic commitments (e.g., Marsh, 1998; Byrd & Ravin, 1999; Liddy, 2000; Miller, et al., 2001; Doddington, et al., 2004). These systems use these informal ontologies because those ontologies are relatively consistent with the ambiguous ways concepts are expressed in human language and are well-suited for their intended applications (e.g., document search, content browsing). However, those ontologies are not well-suited to applications that require complex inference.

Work on so-called *ontology-based* information extraction, such as compete in the ACE program, (e.g. (Cunningham, 2005), (Bontcheva, 2004)) and other semantic-web approaches like (Maynard, 2005), (Maynard, et al, 2005), and (Popov, et al, 2004), focus on directly populating small ontologies that have a rich and well-thought out semantics, but very little if any formally specified semantics (e.g. using axioms). The ontologies are extensively described in English, and the results are apparently used mainly for evaluation and search, not to enable reasoning. Our work differs in that we

provide an explicit knowledge integration step that allows us to populate fully axiomatized ontologies from information extraction.

Our emphasis actually makes our work similar to work in semantic integration or schema matching (e.g., Milo & Zohar, 1998; Noy & Musen, 2001), which typically focuses on finding very simple (e.g., one-to-one) mappings among terms in ontologies. Schema matching is useful when the ontologies are large and complex, so that these mappings, while individually simple, are numerous and challenging to find. Our work however focuses on the opposite circumstance: We assume that the ontologies are small and manageable enough that one can find the correspondences manually and that the mappings may be more complex (conditional, many-to-many, etc.) than an automated matching system can handle.

Schema-matching technologies have typically been used when the applications that the source and target ontologies were designed for are identical or at least quite similar; e.g., matching one e-commerce database schema to another. In those cases, the assumption that individual mappings will tend to be very simple can be valid; since the designers of the ontologies had the same basic purpose in mind. Mapping extracted information into formal reasoning ontologies does not have this characteristic; these applications are radically different and tend to lead to radically different conceptualizations of basic content. For these sorts of differences, it is not feasible to restrict the mappings between terms to be sufficiently simple and obvious enough that they can be discovered by state-of-the-art fully-automated matching techniques.

We use in our work components implemented within the Unstructured Information Mangagement Architecture (UIMA). UIMA is an open-source middleware platform for integrating components that analyze unstructured sources such as text documents. UIMA-based systems define “type systems” (i.e., ontologies with extremely limited semantic commitments) to specify the kinds of information that they manipulate (Götz & Suhre, 2004). UIMA type systems include no more than a single-inheritance type/subtype hierarchy, thus to do substantive reasoning over the results of UIMA-based extraction, one needs to convert results into a more expressive representation.

3 Generating RDF from Text

The context of our application deserves some attention, as our results are somewhat dependent on the assumptions that arise from it. First of all, we are taking the approach that analytics are more expensive to produce than ontologies. This presumes, of course, that we are talking about smaller, lightweight ontologies of no more than 100 classes and 100 object properties, which makes sense if they are to be populated from text analysis, as typical information extraction ontologies are extremely small. Analytics are available in reusable components that can be embedded in frameworks like UIMA, in which they are composed into larger aggregate analysis engines. The individual components overlap to varying degrees in the types of entities and relations they discover, and in the cases of overlap, need to have their results combined. While this has in general been shown to improve overall precision and recall, it does create interesting anomalies in the non-overlapping types

of data (which we will discuss below). The individual analytic components we treat as black boxes, their operation is for the most part functional (producing the same output for the same input). Ontologies therefore are custom built to suit particular application needs, whereas analytics are reused and composed off the shelf. Our experiences are that this characterizes hundreds, if not thousands, of users today looking to populate their part of the semantic web from textual sources. These users are in the medical domain, national and business intelligence, compliance, etc., and many have resources to fund research in the area. The work described here was funded jointly by IBM and the U.S. Government.

3.1 Text to Knowledge Pipeline

In our evaluation prototype, we produce knowledge-bases from text in a pipeline that proceeds through several stages:

Keyword Indexing. The simplest and most scalable processing is the generation of an inverted index to support keyword search. Although techniques such as link analysis, query expansion, etc., can offer minor improvements, this approach is generally very low in precision. In addition to its current established usage, we consider the function of keyword search to be *domain corpus production.* We employ recall-improving techniques such as query expansion to reduce the size of the target corpus to the scale required by the next stage of processing (information extraction) – this is typically 1-2 orders of magnitude.

Information Extraction. Information extraction (IE) in general can be viewed as the analysis of unstructured information to assign labels (or *annotations*) that carry some semantics to regions of the data. The canonical example would be to label the text "George Bush" with *Person*. The field has advanced considerably since these beginnings, and are well represented by the ACE program (Doddington, et al, 2004), participants in which produce annotations for entities (*Person, Organization*, etc.), relations (*partOf, citizenOf,* etc.), and coreference analysis. While almost any kind of information processing can be folded into an information extraction view, in our system, IE components play the role of providing relatively shallow processing in order to be scalable. In particular, this stage limits itself to processing data in documents, and performs the same analysis on each document independently. As a result, IE processing scales linearly with the size of the domain corpus.

Coreference Across Documents. The annotations produced in the IE stage are used as input to *corpus-level processing*, the most important to our purposes of which is coreference analysis – the identification of individual entities that are mentioned (and annotated) in multiple places. Many of our IE components produce coreference analysis within documents, but connecting these results across the entire corpus clearly requires processing that can collect information across the documents, and thus will typically scale at a polynomial rate. In our experience, the most critical properties of co-reference are recognition of aliases and nicknames, common spelling variations of names (especially in other languages), common diminutives, abbreviations, etc. This is a wide-open research area that requires significant attention.

Knowledge Integration. Although it is not required, the data produced in the first three stages of our system are all based on the same underlying format (discussed in Ferrucci&Lally, 2004), which is a simple extension of an OO programming model with a tight programmatic API and a loose semantics (that is, the semantics of a data model can be interpreted by software as the programmers choose). The process of mapping the information from the previous stages into OWL is analogous to the general problem of semantic integration (schema matching, ontology alignment, etc.) with some additional challenges, which we discuss below. We call this stage knowledge integration. The result of knowledge integration, an OWL knowledge-base that can be viewed as a graph, provides the ability to use OWL-based reasoning to perform more sophisticated *deductive search*. For example, we can express axioms of spatial or temporal containment in OWL, and conclude obvious (but nevertheless implicit) results, such as a person in Paris is also in France.

3.2 Knowledge Integration Challenges

Knowledge Integration is analogous to semantic integration. The basic problem is to align the type system of the analytic components with the ontology of the reasoning components (see the beginning of this section for a discussion of why they are not the same), such that the data produced by the analytic components can "instantiate" the ontology. Knowledge integration is difficult however, and to our knowledge is not often attempted, due to the vastly different requirements, and different communities, on each side. As a result, what seems on the surface to be a natural connection – producing structured representations from unstructured information and then reasoning over those structures – turns out to be a difficult challenge. Below we list the five dimensions of interoperability we have identified and brief notes on how we are addressing them:

Precision. Formal reasoning systems are notoriously intolerant of errors, and IE systems are notoriously prone to producing them. This is probably the most fundamental problem in putting them together. In particular, logical reasoning becomes meaningless in the face of contradiction, most inference engines will prove any statement to be true if the knowledge-base is inconsistent to begin with. Although improving precision is an obvious approach to this problem, we take it as a given that IE processes will never be perfect, and furthermore even in the presence of perfect IE, data sources can contradict each other intentionally (e.g. reports from CNN and the pre-war Iraqi News Agency), and instead we focus on making the reasoning systems more tolerant of errorful data. Our simplest technique is to perform limited reasoning such as semantic constraints that can be checked rapidly, and that in our evaluations we find to be indicative of IE errors and not intended contradictions. We discuss this further below.

Recall. Imperfect recall is another significant obstacle to interoperability. The amount of knowledge we typically get from documents is quite small compared to what a human might produce from the same document. The reasoning system is, therefore, crippled by major gaps in the input. Using inference can actually help improve recall, however it is a different sense than is typically used in IE measurements. Recall measurements are based on comparison to a

"ground truth" (i.e. a human annotated corpus), in which implicit information does not appear. For example, in the sentence "Joe arrived in Paris", we would not expect a test corpus to include the relationship that Joe arrived in France, yet this inferred information clearly increases the recall.

Relationships. Simple IE systems that produce type annotations (such as Person, Organization, etc.) are not of much use as input to a reasoning system. These end up in a knowledge base as assertions that something is an instance of something else. There is very little reasoning that can be done with only that information. In order for reasoning to produce useful results, we need relationships to be extracted as well. For example, there is not much to conclude from the sentence, "Joe was in Paris," if all that was produced was that "Joe" is a person and "Paris" is a place. In this case, a located-in relation would be useful as well, as simple spatial containment axioms plus basic world knowledge (e.g. that Paris is in France) would allow a reasoner to conclude that Joe was in France. We use a number of IE components that produce relations over text, however the state-of-the-art in relation extraction is very poor on precision and recall.

Annotations vs. Entities. In our experience, relation annotation by itself creates another problem. Every relation annotation creates a tuple whose elements are the spans of text that participate in the relation, and thus do not appear in other relations. This severely limits the usefulness of reasoning, since the elements of the relation tuples are the *mentions* not the entities. For example, from the sentences, "Joe was in Paris. Fred was in Paris, too," relation annotation would produce two tuples, however the elements of the tuples are not the strings, "Joe", "Fred", and "Paris", but the regions containing those strings in the original text, and as a result we have four elements identified by their position in text, *not* by their contents. Thus the first and second occurrences of "Paris" are different elements, and we could not conclude in a reasoner that, e.g. Joe and Fred are in the same place. In fact, without connecting these two mentions of Paris (both within and across documents), we end up with a large list of unconnected relation tuples. We address this problem with coreference analysis, and although we do not discuss it in this paper, *coreference analysis is an essential task in populating knowledge-bases from text.* In particular, consider that the output of knowledge integration is a graph – the graph without coreference analysis would be a disconnected set of connected pairs.

Scalability. IE techniques scale far better than KR techniques, and as a result we also need to limit the amount of data that any reasoning component has to deal with. In our experience, documents provide an excellent and reliable heuristic for KB size, as well as for consistency. We have found that, excluding IE errors, in excess of 90% of the documents we process are internally consistent, and thus far all documents (we focus mainly on news articles, intelligence reports and abstracts) have been the basis of small enough KBs for any of our advanced reasoning systems. Still, document-based partitioning is inadequate for a lot of information gathering tasks that we have focused on, so a variety of incremental capabilities are required, as are efforts at more scalable reasoning.

We attempt to address these dimensions in a component-based framework for supporting knowledge integration, discussed in the next section. Due to space considerations we cannot discuss all five dimensions, and will focus mainly on

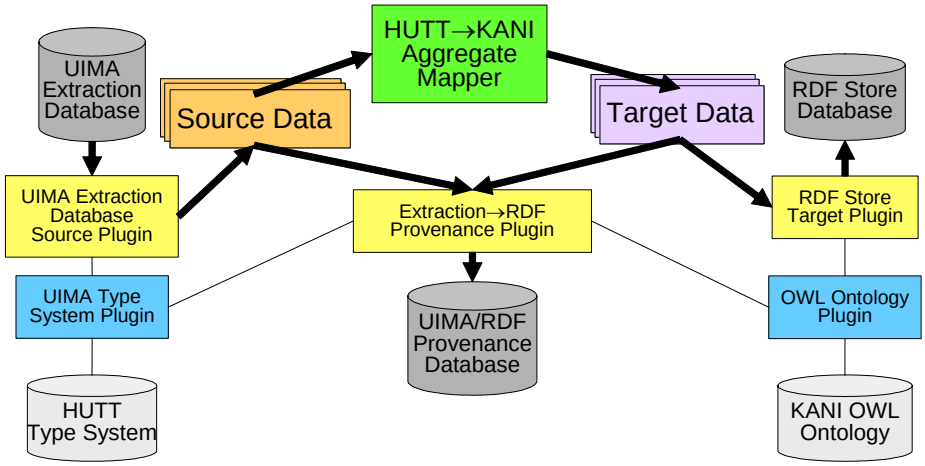

Figure 1: Example KITE-based application

experiments we have performed to use deeper semantics expressed in OWL-DL to improve precision.

4 Knowledge Integration and Transformation Engine (KITE)

KITE (Murdock & Welty, 2006) is a middleware platform for use by developers of knowledge integration applications. KITE consists of two major components:

- **KITE Core Framework**: Java interfaces, data structures, and a central control mechanism for mapping entities and relationships from one ontology to another.
- **KITE Commons**: A set of broadly applicable plugins that comply with the interfaces specified in the core framework.

A KITE-based integrator takes as input a *Source Repository* (e.g., a database, an RDF/XML file). Information in that repository is encoded in the *Source Ontology* (which is accessed via an *Ontology Language Plugin*). The *Source Plugin* reads from the source repository and outputs *Source Data* encoded in KITE data structures for instances and tuples. *Mapper Plugins* may be primitive or aggregate. Aggregate mapper plugins are composed of other (primitive or aggregate) mapper plugins. Primitive mapper plugins are Java objects that take *Source Data* as input and output *Target Data* (which consist of the same data structures, but are encoded in the *Target Ontology*). The *Target Plugin* writes that data to a *Target Repository* and the *Provenance Plugin* writes the mappings from source to target data into a *Provenance Repository*.

Figure 1 shows an example of a KITE-based knowledge integrator. Source data for this application is encoded in HUTT (Hierarchical Unified Type Taxonomy), a UIMA type system based on a variety of established information extraction

taxonomies (e.g., Doddington, et al., 2004; Sauri, Litman, et al., 2004). The output ontology for this application is the OWL ontology used in the KANI project (Fikes, Ferrucci, & Thurman, 2005).

The input data for the example application is stored in a database designed to contain UIMA extracted information. The KITE Commons includes a plugin (*UIMA Extraction Database Source Plugin*) that accesses this database and outputs KITE instances and tuples (*Source Data*). This source data is provided to an aggregate mapper composed of an assortment of both generic mappers from the KITE Commons and specialized mappers that were written for the HUTT to KANI integrator. These mappers output target data. That data is consumed by two plugins from the KITE Commons: the *RDF Store Target Plugin* writes the target data alone into a relational database for RDF triples, and the *Extraction → RDF Provenance Plugin* records (potentially complex) mappings from source data in the extraction database to target data in the RDF database; these mappings are stored in the *UIMA/RDF Provenance Database*.

Systems that access instances and triples from the RDF store can request traces of the information extraction and knowledge integration processes that created those instances and triples. The provenance database is able to return that information either as database entries or in the OWL-based Proof Markup Language, PML (Pinheiro da Silva, McGuinness & Fikes, 2006). Systems that perform additional reasoning over the extracted knowledge can provide integrated end-to-end PML traces that explain their conclusions as a combination of logical inferences from the RDF knowledge and extraction inferences used to obtain that knowledge from text (Murdock, et al., 2006).

The most complex mappers that were written for this application involve the handling of temporal information. The representation of time in HUTT is based on TimeML (Sauri & Littman, 2004), a language for marking up expressions of time in natural-language text. The representation of time in the KANI ontology is OWL-Time (Hobbs, 2004), a semantic web ontology. OWL-Time makes relatively subtle distinctions that are usually implicit in text (e.g., distinguishing between time intervals and time interval descriptions). Furthermore, OWL-Time has distinct properties to encode different aspects of a description of a time (year, month, day, hour, etc.). In contrast, TimeML does not encode a time and its expression separately, and uses a relatively compact normalized form to encode a full time description in a single string. These differences are motivated by the different applications that these ontologies were designed for; OWL-Time directly enables a wide variety of logical inferences about times, while TimeML provides a convenient and compact formalism for identifying, normalizing, and linking expressions of time in text. A generic mapping component that was expressive enough to handle the mapping between these two portions of the HUTT and KANI ontologies would be extremely complicated to develop and to use. However, many of the other terms in HUTT and KANI *are* handled easily by simple, generic mappers from the KITE Commons.

5 Improving Annotator Precision & Recall Using OWL

One particularly promising result of our knowledge integration efforts supported by the KITE framework has been using the kind of deep, axiomatic, semantics that OWL enables, to help improve precision and recall in the results. We present here our technique and a preliminary evaluation of its effectiveness with a large UIMA-based application that includes dozens of "off the shelf" analytic components run on a corpus of news articles.

5.1 Technique and Evaluation for Improving Precision

The most problematic kind of extraction produced by analytic components we have experienced is relation extraction. A common type of error we see in extracted relations is the violation of simple domain and range constraints. For example, in the following sentence:

> *In February 1993, US officials said that US President Bush's decision in September 1991 to withdraw tactical nuclear bombs, missiles and torpedoes from US Navy ships has caused the impetus for naval arms control to whither.*

our analytics extract an ownership relation in the underlined text between "nuclear" (annotated as a weapon), and "bombs" (also a weapon), which maps to a *ownerOf* relation in the ontology. The *ownerOf* relation has a restriction limiting the domain to *Person* or *Organization* or *GPE* and a disjointness constraint between each of these and *Weapon*.

Our approach is a simple one. During knowledge integration, we construct an intermediate knowledge base (in fact, a Jena model) consisting of only the mapped entities and their type information. Then, during the mapping process producing relations, we add resulting triples to this KB one at a time. With each addition, we run the KB through a consistency check using Pellet. If the KB is not consistent, we "drop" the triple, if it is consistent, we add the triple to the output of the transformation. Obviously this technique does not scale particularly well and is entirely dependent on the degree to which the ontology is axiomatized. In preliminary experiments, however, the technique appears promising and does quite well – offering a clear improvement in precision by dropping incorrect triples. We are still exploring how these results generalize, but we present here some concrete examples, analysis, and discussion:

Ontology. The ontology we tested consists of 56 classes and 62 object properties in the domain of nuclear arms proliferation. Other than a few specific classes for weapons, the classes and properties are fairly generic (people, places, facilities, etc.). All properties have global domain and range constraints, however some are not that restrictive. Five classes have local range constraints. Cardinality constraints are not of use in our domain. The most effort was spent assigning appropriate disjointness constraints, as these are key to the technique.

Analytics. Our analytics are 42 off-the-shelf components that were developed for other projects such as TREC and ACE, and that we aggregated using the composition capabilities of UIMA.

The merged type system contains 205 entity and 79 relation types; most of our analytic components overlap on common types such as *PERSON* and *ORGANIZATION*, etc., but each adds some unique functionality to the overall aggregate. We have special purpose components for arbitrating between conflicting annotation assignments and for computing co-reference across documents.

Corpus. The corpus contains over 30K documents that average about a page in length. Most are news articles or summaries of news articles in the domain of interest. Due to the extensive cost of evaluation (which must be done by hand), the experiments were performed on 10, 41, and 378 documents. We report here the results of the 378 document test. On average our analytics produce 322 entity annotations and 21 relation annotations per document, and coreference merges an average of 15 annotations per entity and 1.8 annotations per relation. The KITE-based knowledge integrator maps those entities and relations into instances and tuples in the KB. For the 378 document corpus, the end result is a KB of 6281 individuals and 834 object property triples. These numbers clearly demonstrate the significance of recall in this process, only a fraction of the generated knowledge base is of any real use to the semantic web, more than 70% of the entities simply have a label and a type.

Results. Our technique dropped 67 (object property) triples of the 834 produced by the mapping process. Of the 67 dropped, 2 were actually correct and should not have been dropped (see the analysis below). This is a *relative* improvement in precision of 8.7%., which is considerably more than the difference between the first and fifth place competitors in the ACE competition relation extraction task (for which this scale is also appropriate). The cost of this improvement is high; the system without this check takes about 5 minutes to generate a KB from 378 documents, and with the reasoning check takes over an hour. There is a lot that can be done to improve this, however, and our precision improvement results are encouraging enough that we are exploring alternatives, such as a much more rapid heuristic consistency checker (Fokue, et al, 2006), partitioning the KB by document instead of checking global consistency, and others.

5.2 Analysis of Evaluation

Of the 67 triples we reported dropped, 2 should not have been dropped. A further 11 triples fell into a special category in which the triples themselves were correct, but the coreference resolution or type assignments for relation arguments were wrong, so a more robust solution would have been to amend the coreference or typing.

Many (but not all) of the correct filtering of incorrect relations is a result of the combination of multiple independent annotators to determine the type of an entity. An example of this occurred in the following phrase:

> *With the use of these pits, landmines, and guerrilla attacks, Khmer Rouge forces allegedly drove off the personnel sent to repair the road.*

One of our entity and relation annotators incorrectly determines that “Khmer Rouge” is a person who is the leader of the “forces.” However, the combination of annotators

concludes that "Khmer Rouge" is actually an organization. Since the OWL ontology indicates that an organization can't be the leader of another organization, this triple is correctly dropped.

The two erroneously dropped triples were due to a combination of weak typing of entities and errors in another relation that did not manifest as inconsistencies until the triple in question was added to the KB. For example, consider the phrase:

… of countries like Pakistan, India, Iran, and North Korea, who are building …

A comma between two geopolitical entities often indicates a subPlace relation (e.g., "Delhi, India"), and one of our annotators incorrectly extracts a subPlace relation between India and Iran. The cross-document coreference process is unable to authoritatively assign the "Country" label to the entity corresponding to "India", so it ends up as a GPE (geopolitical entity), a superclass of Country. The entity corresponding to "Iran", however, is correctly typed as a Country. In the ontology, there is a local range restriction on the *Country* class that prevents it from being a *subPlace* of another country. So, if the entity corresponding to "India" had been correctly labeled as a country, our technique would have dropped the "India subPlace Iran" relation when it was mapped, however since some countries are subplaces of GPEs (e.g. France subPlace EU), the weaker GPE assignment for India allows the erroneous triple through. By happenstance, a subsequent triple in the mapping process results from this passage,

… were seized by Indian authorities after a raid on a suspected weapons lab …

where our analytics correctly extract a *citizenOf* relation in the underlined text between "authorities" and "Indian", correctly coreference "Indian" with the entity for "India" in the previous passage, and correctly assign the type Person to the entity corresponding to "authorities". The ontology contains a global range restriction for the *citizenOf* relation to instances of Country. Since the erroneous *subPlace* triple added previously prevents India from being a country (since a country cannot be a subPlace of a country), adding this correct triple causes an inconsistent KB. This shows the technique has some order dependences, had these triples been added in a different order the correct one would have been dropped. Fortunately our initial results indicate these circumstances to be rare (2 erroneous drops out of 834 triples).

There were eleven examples of dropped triples where problems were actually in the assignment of types to the entities or in coreference, so that a better approach would have been to fix the type assignments or undo a coreference merge. For example:

… and the increase in organized criminal groups in the FSU and Eastern Europe.

In this case, the analytics produce a *basedIn* relation between "groups" and "FSU" in the underlined text, but multiple annotators disagree on the type of "FSU" (some correctly say GPE, some incorrectly say Organization), and the incorrect label (Organization) ends up winning. Overall our technique for combining annotations does improve precision, but like all IE techniques it isn't perfect, as in this case.

Therefore we end up with an organization being *basedIn* an organization, and the ontology requires organizations to be *basedIn* GPEs, and specifies that GPEs and Organizations are disjoint.

It is somewhat debatable whether dropping this triple is a mistake – clearly it would be *better* to fix the type, but the entity corresponding to "FSU", as presented to the KB, is an organization and cannot be the object of a *basedIn* relation. Thus the KB does end up cleaner without it.

5.3 Techniques for Improving Recall

Our initial motivation for combining IE with semantic technology in general was the possibility of improving information access beyond keyword-based approaches through inference. For example, in the passage "Joe arrived in Paris", no keyword search, nor search enhanced by semantic markup, would retrieve this passage in response to the query, "Who is in France?" Clearly with some world knowledge (that Paris is in France) and the ability to accurately recognize the relation in the passage (& query), we could employ reasoning to catch it.

OWL-DL is not particularly strong in its ability to perform the kinds of "A-box" reasoning that would be needed to make a significant improvement in this kind of recall. Other choices are RDF rules and translating the KBs into more expressive languages (like KIF). A semantic web rules language would obviously help here as well.

An interesting challenge is in measuring the impact of this kind of reasoning. It makes sense to call this an improvement in recall; in the simple example above clearly the passage in question contains an answer to the query, and clearly keyword search would not find it. However, it is a different sense of recall than is typically used in IE measurements. Recall measurements are based on comparison to a "ground truth" (i.e. a human annotated corpus), in which implicit information does not appear. In textual entailment (Dagan et al, 2005) the measurement problem is similar, however they address this in evaluations by always making the determination based on pairs of text passages. So we can show improvement in recall by selecting meaningful queries and determining if and how reasoning improves the recall for each query, but measuring recall improvements in the KB itself is more difficult.

6 Conclusions

In our research to use information extraction to help populate the semantic web, we have encountered significant obstacles to interoperability between the technologies. We believe these obstacles to be endemic to the basic paradigms, and not quirks of the specific implementations we have worked with. In particular, we identified five dimensions of interoperability that must be addressed to successfully populate semantic web knowledge bases from information extraction systems that are *suitable for reasoning.* We called the task of transforming IE data into knowledge-bases *knowledge integration*, and briefly presented a framework called KITE in which we are exploring these dimensions. Finally, we reported on the initial results of an

experiment in which the knowledge integration process used the deeper semantics of OWL ontologies to improve the precision of relation extraction from text. By adding a simplistic consistency-checking step, we showed an 8.7% relative improvement in precision over a very robust IE application without that checking.

This work is still in the beginning stages, but we do have results and conclusions, the most important of which is to address a long-standing problem that presents an obstacle to interoperability: being realistic. IE and NLP systems do not produce perfect output of the sort that KR systems deal with, and KR systems are not capable of handling the scale, precision, and recall that NLP and IE systems produce. These are not criticisms but realities. We cannot just sit back and wait for the two technologies to eventually meet, rather we must begin exploring how to realistically integrate them.

We should also point out that none of the implemented systems we used were baseline "strawman" systems, but reportedly state-of-the-art systems in each area. It is not our intention to advance research in information extraction nor in knowledge representation and reasoning, but rather in the combination of the two. We believe that the combination will be better than either individually, and have demonstrated one example of how this is so, using deeper semantics and reasoning to improve precision of relation extraction.

Acknowledgements

This work was supported in part by the ARDA/NIMD program.

References

K. Bontcheva. 2004. Open-source Tools for Creation, Maintenance, and Storage of Lexical Resources for Language Generation from Ontologies. *Fourth International Conference on Language Resources and Evaluation (LREC'2004)*. Lisbon, Portugal. 2004.

Roy Byrd & Yael Ravin. 1999. Identifying and Extracting Relations in Text. *4th International Conference on Applications of Natural Language to Information Systems* (NLDB). Klagenfurt, Austria.

Jennifer Chu-Carroll, Krzysztof Czuba, Pablo Duboue, and John Prager. 2005. IBM's PIQUANT II in TREC2005. *The Fourteenth Text REtrieval Conference (TREC 2005).*

Philipp Cimiano, Johanna Völker. 2005. Text2Onto - A Framework for Ontology Learning and Data-driven Change Discovery. *10th International Conference on Applications of Natural Language to Information Systems* (NLDB). Alicante, Spain.

Ido Dagan, Oren Glickman and Bernardo Magnini. The PASCAL Recognising Textual Entailment Challenge. *In Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, 2005.

Hamish Cunningham. 2005. Automatic Information Extraction. *Encyclopedia of Language and Linguistics, 2cnd ed.* Elsevier.

Stephen Dill, Nadav Eiron, David Gibson, Daniel Gruhl, R. Guha, Anant Jhingran, Tapas Kanungo, Sridhar Rajagopalan, Andrew Tomkins, John A. Tomlin, & Jason Y. Zien. 2003. SemTag and Seeker: Bootstrapping the semantic web via automated semantic annotation. *12th International World Wide Web Conference* (WWW), Budapest, Hungary.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, & Ralph Weischedel. 2004. Automatic Content Extraction (ACE) program - task definitions and performance measures. *Fourth International Conference on Language Resources and Evaluation* (LREC).

David Ferrucci & Adam Lally. 2004. UIMA: an architectural approach to unstructured information processing in the corporate research environment. *Natural Language Engineering* 10 (3/4): 327–348.

Richard Fikes, David Ferrucci, & David Thurman. 2005. Knowledge Associates for Novel Intelligence (KANI). *2005 International Conference on Intelligence Analysis* McClean, VA.

Achille Fokoue, Aaron Kershenbaum, Li Ma, Edith Schonberg and Kavitha Srinivas. 2006. The Summary Abox: Cutting Ontologies Down to Size. *Proceedings of the 5th International Semantic Web Conference*. Springer-Verlag.

T. Götz & O. Suhre. 2004. Design and implementation of the UIMA Common Analysis System. *IBM Systems Journal* 43 (3): 476-489.

Jerry R. Hobbs and Feng Pan. 2004. An OWL Ontology of Time. http://www.isi.edu/~pan/time/owl-time-july04.txt

Elizabeth D. Liddy. 2000. Text Mining. *Bulletin of American Society for Information Science & Technology*.

Xiaoqiang Luo, Abraham Ittycheriah, Hongyan Jing, Nanda Kambhatla, Salim Roukos: A Mention-Synchronous Coreference Resolution Algorithm Based On the Bell Tree. *ACL 2004*: 135-142.

Elaine Marsh. 1998. *TIPSTER information extraction evaluation: the MUC-7 workshop*.

D. Maynard. 2005. Benchmarking ontology-based annotation tools for the Semantic Web. AHM2005 Workshop "Text Mining, e-Research and Grid-enabled Language Technology", Nottingham, UK, 2005.

Diana Maynard, Milena Yankova, Alexandros Kourakis, and Antonis Kokossis. 2005. Ontology-based information extraction for market monitoring and technology watch. ESWC Workshop “End User Apects of the Semantic Web,” Heraklion, Crete, May, 2005.

Scott Miller, Sergey Bratus, Lance Ramshaw, Ralph Weischedel, Alex Zamanian. 2001. FactBrowser demonstration. *First international conference on Human language technology research* HLT '01.

T. Milo, S. Zohar. 1998. Using Schema Matching to Simplify Heterogeneous Data Translation. VLDB 98, August 1998.

J. William Murdock & Chris Welty. 2006. Obtaining Formal Knowledge from Informal Text Analysis. IBM Research Report RC23961.

J. William Murdock, Deborah L. McGuinness, Paulo Pinheiro da Silva, Christopher Welty, David Ferrucci. 2006. Explaining Conclusions from Diverse Knowledge Sources. *Proceedings of the 5th International Semantic Web Conference.* Springer-Verlag.

N. F. Noy & M. A. Musen. 2001. Anchor-PROMPT: Using Non-Local Context for Semantic Matching. *Workshop on Ontologies and Information Sharing*, Seattle, WA.

Paulo Pinheiro da Silva, Deborah L. McGuinness & Richard Fikes. A proof markup language for Semantic Web services. 2006. *Information Systems* 31(4-5): 381-395.

Borislav Popov, Atanas Kiryakov, Damyan Ognyanoff, Dimitar Manov, Angel Kirilov. 2004. KIM - A Semantic Platform for Information Extraction and Retrieval. *Journal of Natural Language Engineering*, 10(3-4): 375-392.

Roser Sauri, Jessica Littman, Robert Gaizauskas, Andrea Setzer, & James Pustejovsky. 2004. TimeML Annotation Guidelines, Version 1.1. http://www.cs.brandeis.edu/%7Ejamesp/arda/time/timeMLdocs/guidetest.pdf

Alexander Schutz and Paul Buitelaar. 2005. RelExt: A Tool for Relation Extraction from Text in Ontology Extension. *Proceedings of ISWC-05.*

Johanna Voelker, Denny Vrandecic, York Sure. 2005. Automatic Evaluation of Ontologies (AEON). In *Proceedings of ISWC-05.*