Explaining Conclusions from Diverse Knowledge Sources

J. William Murdock[*], Deborah L. McGuinness[+], Paulo Pinheiro da Silva[^,1], Chris Welty[*], and David Ferrucci[*]

[*]IBM Watson Research Center
19 Skyline Drive
Hawthorn, NY 10532
[+]Knowledge Systems, Artificial Intelligence Laboratory
Stanford University
Stanford, CA 94305
[^]Department of Computer Science
The University of Texas at El Paso
500 W University Ave
El Paso TX 79968-0518

Abstract

The ubiquitous non-semantic web includes a vast array of unstructured information such as HTML documents. The semantic web provides more structured knowledge such as hand-built ontologies and semantically aware databases. To leverage the full power of both the semantic and non-semantic portions of the web, software systems need to be able to reason over both kinds of information. Systems that use both structured and unstructured information face a significant challenge when trying to convince a user to believe their results: the sources *and* the kinds of reasoning that are applied to the sources are radically different in their nature and their reliability. Our work aims at explaining conclusions derived from a combination of structured and unstructured sources. We present our solution that provides an infrastructure capable of encoding justifications for conclusions in a single format. This integration provides an end-to-end description of the knowledge derivation process including access to text or HTML documents, descriptions of the analytic processes used for extraction, as well as descriptions of the ontologies and many kinds of information manipulation processes, including standard deduction. We produce unified traces of extraction and deduction processes in the Proof Markup Language (PML), an OWL-based formalism for encoding provenance for inferred information. We provide a browser for exploring PML and thus enabling a user to understand how some conclusion was reached.

Introduction

It has been recognized since at least the early days of expert systems research that systems should be able to provide information about how answers were obtained if users are expected to understand, trust, and use conclusions. In these early systems, conclusions may have been obtained by using sound inference procedures applied to knowledge bases of logical statements that were hand coded by experts. Under these conditions, the knowledge bases may have contained correct and trustworthy information and the reasoners may have been correct. The information about the

---

[1] This work was done while at the Knowledge Systems, Artificial Intelligence Laboratory at Stanford University.

answer generation process typically focused on the derivation path, and it was typically referred to as an explanation. Sometimes the explanations included some limited information about facts from the knowledge bases. Sometimes there was additional focus on taking the information concerning the derivation path and making it more understandable to the end user.

Modern semantic web systems also require this kind of support, however now they also have additional needs. There are two characteristics that our work addresses.:

1. Semantic web systems can have **different kinds of information** that form the basis of their reasoning, e.g., unstructured HTML, manually generated OWL ontologies, RDF stores, etc.
2. Semantic web systems that use different kinds of information will need to use **different kinds of processing** to manipulate that information.

In other words, semantic web systems may use *distributed* knowledge bases constructed by *different organizations* from *many sources* using *multiple reasoning components*.

Systems that process input information in the form of HTML and text typically operate in two phases. First, they *extract* logical statements from the text automatically or semi-automatically. Next those logical statements are combined with existing structured knowledge (if any) to form a knowledge-base used for additional reasoning.

Information extraction techniques are known to produce conclusions that are not sound. In an integrated system in which such statements are input directly into a knowledge base, from which reasoning may derive further incorrect information, there is an increased need to provide thorough and integrated explanations; they need to have access to the raw sources of information and its meta information (recency, authoritativeness, etc.) and they need to provide insight into how the knowledge base statements were obtained.

In this paper, we describe a solution infrastructure that provides meta-information for integrated Natural Language Processing / Knowledge Base systems that includes the sources of information (including documents, passages, linguistic markup, semi-structured and structured data- and knowledge-bases), the nature of the information (documents, annotations, facts), the epistemological status (extracted, derived, asserted), and the sources (people, articles, automated reasoning components, text extraction components). This meta-information, or *knowledge provenance*, is integrated with our explanation infrastructure so that conclusions can be traced to their sources along a derivation path.

This paper is not addressing the issue of presentation techniques for knowledge provenance that may include abstractions and dialogues, and thus is not about explanation in the traditional sense. The primary contributions of the paper are the *uniform framework* that provides the basis for explanations over a much broader range of systems than any known previous work, and a view of *extraction as inference* [Ferrucci, 2004] that allows the integration of proof-based explanations with the field of text analytics.

Solution Architecture

Our solution relies on integration work between research on unstructured and structured information. The primary integration work is between two foundational components: The Unstructured Information Management Architecture (UIMA) and

the Inference Web (IW). UIMA is a framework for integrating software components that analyze unstructured information such as text [Ferrucci and Lally, 2004]. IW is a framework for explaining systems that manipulate structured information and now unstructured information [McGuinness and Pinheiro da Silva, 2004]. We have developed new capabilities supporting the combination of IW and UIMA, enabling the former to present explanations of analysis performed within the latter.

UIMA

UIMA provides an infrastructure for integrating analysis components. The components use a declarative formalism. The specifications are hierarchical, i.e., aggregate components may be constructed out of a combination of primitive components and/or other aggregate components. At each level of the component hierarchy, the specification describes input requirements and output capabilities using a simple ontology. By describing analysis systems in terms of inputs and outputs at multiple levels of abstraction, UIMA provides an effective and convenient starting point for explaining analysis.

   To support explanation, UIMA now provides a scalable repository for storing the final results of the knowledge extraction processes. This repository is known as the EKDB (Extracted Knowledge Database). The EKDB stores not only the content of the extracted knowledge (i.e., the set of entities and relations that the analysis system concluded from the corpus) but also some intermediate analysis results (such as assigning types to spans of text) and links among the intermediate and final results.

Inference Web

Inference Web provides an infrastructure for providing explanations from distributed hybrid reasoning systems. It utilizes a proof Interlingua – the Proof Markup Language (PML) [Pinheiro da Silva, McGuinness, Fikes, 2004] to encode justifications of information manipulations. It also provides numerous services for manipulating PML documents. It includes a browser for viewing information manipulation traces, an abstractor for rewriting PML documents so that the low level machine-oriented proofs can be transformed into higher level human-oriented explanations, an explainer to interact with users by presenting explanations and corresponding follow-up questions, a registrar[McGuinness, et al., 2005] for storing and maintaining proof related meta-information, and new search and trust [McGuinness, et al, 2006, Zaihrayeu, et al, 2005] components. It also includes services for helping systems to generate PML, check PML documents for valid applications of inferences[Pinheiro da Silva, et al., 2005], and services for automatic registration of sources and meta-information.

Text Analytic Information Manipulations

Our explanation solution framework uses a proof interlingua to encode justifications of answers. We can view all information manipulation steps as a kind of inference. One contribution of our work is the design and specification of a taxonomy of text analytic processes and tasks that can be viewed as inferences.

   We generated a taxonomy motivated by the need to describe and explain the dominant extraction tasks in UIMA, without overloading the system with more information than would be useful. One key was to generate a taxonomy that is adequate to accurately describe extraction task functionalities and simultaneously

abstract enough to be able to hide details of the tasks from end users. Another key was to support explanations to end users of the integrated system, not authors of software components debugging their products.

First we will describe the taxonomy and later we will discuss issues related to its granularity, size, reusability, and extensibility.

We divided text extraction into three primitive areas: annotation, coreference, and integration. We will describe each briefly and provide examples of a few tasks used in a later example.   Annotation tasks make assertions about spans of text that recognize a type or argument.   Annotation inferences include:

1) **Entity Recognition:** determines that some span of text refers to an entity of a specified type. For example, a component could take the sentence "Joseph Gradgrind is the owner of Gradgrind Foods" and conclude that characters 0 to 16 of that sentence refer to some entity of type *Person*.

2) **Relation Recognition**: assigns a relation type to a span (e.g., a sentence describes a relation of type *Owner*).

3) **Relation Annotation Argument Identification**: determines and assigns values to the roles of a relation (e.g., a particular person is a participant in a given ownership relation instance).

Coreference inferences utilize annotation inferences and further identify that multiple text spans actually refer to the same entity or relation.

4) **Entity Identification**: determines that a set of entity annotations refer to a particular instance.

5) **Relation Identification**: determines that a set of relation annotations refer to a particular relation instance.

6) **Extracted Entity Classification**: determines that a particular coreferenced entity has a particular type. (e.g., the type of the entity referred to by "Gradgrind" is Person).

Knowledge integration inferences include mapping inferences providing access to provenance.

7) **Entity Mapping**: determines that an entity instance in the KB is derived from a set of entities and relation instances.

8) **Relation Mapping**: determines that a relationship in the target KB is derived from a set of entity and relation instances.

9) **Target Entity Classification**: determines that an entity instance is an instance of an entity type in the target ontology.

We have registered these inferences in the IW registry and we use these information manipulation steps to explain all of the UIMA components used in our prototype system, which provides intelligence analyst support for analyzing documents and evaluating results of text statements.

Text Analytic Manipulation Descriptions

We use our taxonomy of text analytic manipulations in declarative descriptions encoding what was done to generate the extracted knowledge bases. UIMA generates a large extracted knowledge database containing its conclusions. We needed to take

that as input (potentially augmented) and generate interoperable proof descriptions (a PML document) as an output.

The software component that produces PML documents for UIMA-based analysis processes begins with a specified result from a specified EKDB (e.g., *JosephGradgrind* is the *Owner* of *GradgrindFoods*). It follows the links in the EKDB from that conclusion back to the intermediate results and raw input that led to it. From these intermediate results, it is able to produce inference steps encoded in PML that refer to the corresponding tasks in the taxonomy. For example, if the EKDB records that characters 0 to 16 of some sentence were labeled as a *Person* and that this labeling was identified as specifying an occurrence of *JosephGradgrind* then the component would create an **Entity Recognition** inference step in PML for that labeling as well as coreference step for the result that the labeling is an occurrence of *JosephGradgrind*.

Example in Action

Figure 1 provides an example showing how our new end-to-end explanation infrastructure can provide explanations annotated with meta-information using knowledge bases that may contain facts extracted by UIMA analytics from raw text. This example is similar to, but simpler than the examples produced by our system. In the example, the system is attempting to determine who manages some aspect of Mississippi law enforcement and safety data infrastructure. The answer is derived by a combination of the JTP theorem prover and a set of extraction components. The original sources for the proof include a press release (http://www.ibm.com/industries/government/doc/content/news/pressrelease/1107628109.html) and a knowledge base containing some direct assertions. The format shown in Figure 1 is approximately the same format that the Inference Web Browser uses to present proofs.

The initial data (i.e., the nodes in Figure 1 that have no parents) include assertions from a knowledge base, KB1.owl and a sentence from the press release. A fact asserted in the KB is that the Mississippi Automated System Project (*MASProject1*) manages some Mississippi data infrastructure (*MissDataInfrastructure1*), as stated in node (C). An axiom asserted from the KB and encoded in node (B) says that management is transitive.

Node (D) in the figure concludes that *MJAllen1* is the manager of *MASProject1*. This result was derived by a knowledge extraction process that began with a passage in a press release. The process involved the consecutive use of three UIMA-compliant components. IBM EAnnotator [Ando, 2004] assigns entity types to text spans in the document, i.e., it produces entity annotations. An IBM relation recognizer determines relates those spans via a *managerOf* relation. Finally, IBM Coreference concludes that those annotations correspond to particular entities (*MJAllen1* and *MASProject1*) and a relationship between them (*managerOf*). From (B), (C), and (D), the reasoner can deduce (A), that *MJAllen1* is the manager of *MissDataInfrastructure1*. Some end-users may only be interested in that result, but others may wish to see the full derivation of the result from the KB and the raw text.

Figure 2 shows a partial/ablated screen capture of Inference Web's WWW-based browser displaying a portion of the automatically-generated extraction proof for the assertion that Major Julian Allen is the director of the Mississippi Automated System Project. As you can see, the proof is a slightly more complicated version of the one described above. The Inference Web browser interface allows users to show and hide
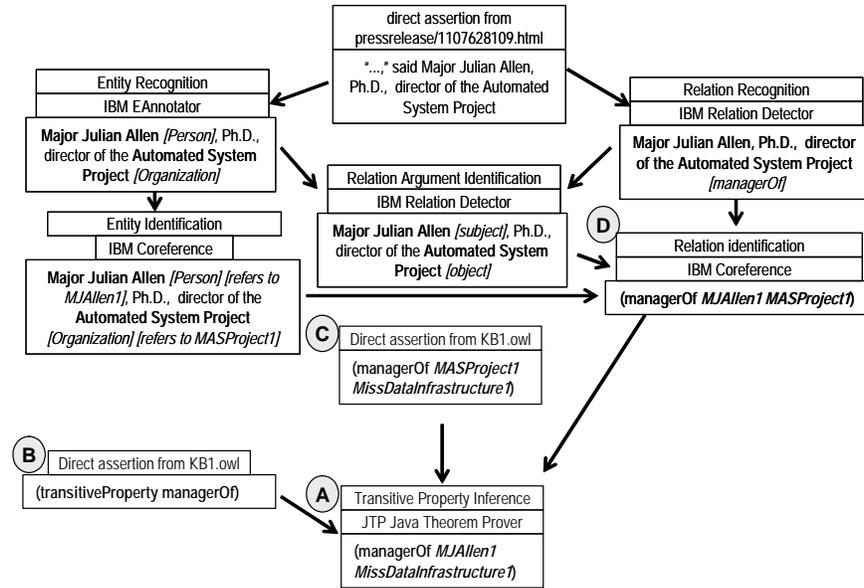
Figure1: Example of an integrated proof of extraction and reasoning.

individual steps in the proof in order to see the proof at varying levels of detail. The conclusion can also be explored in detail to determine for example that *uid184* refers to Major Julian Allen and *uid199* refers to the Mississippi Automated System Project. Additional summary views and follow-up options are available in the implemented system. Interested users can explore this proof at:

http://iw4.stanford.edu/iwbrowser/NodeSetBrowser?url=http%3A%2F%2Fiw4.stanford.edu%2Fproofs%2FMississippiAutomatedSystem%2Fns36.owl%23ns36

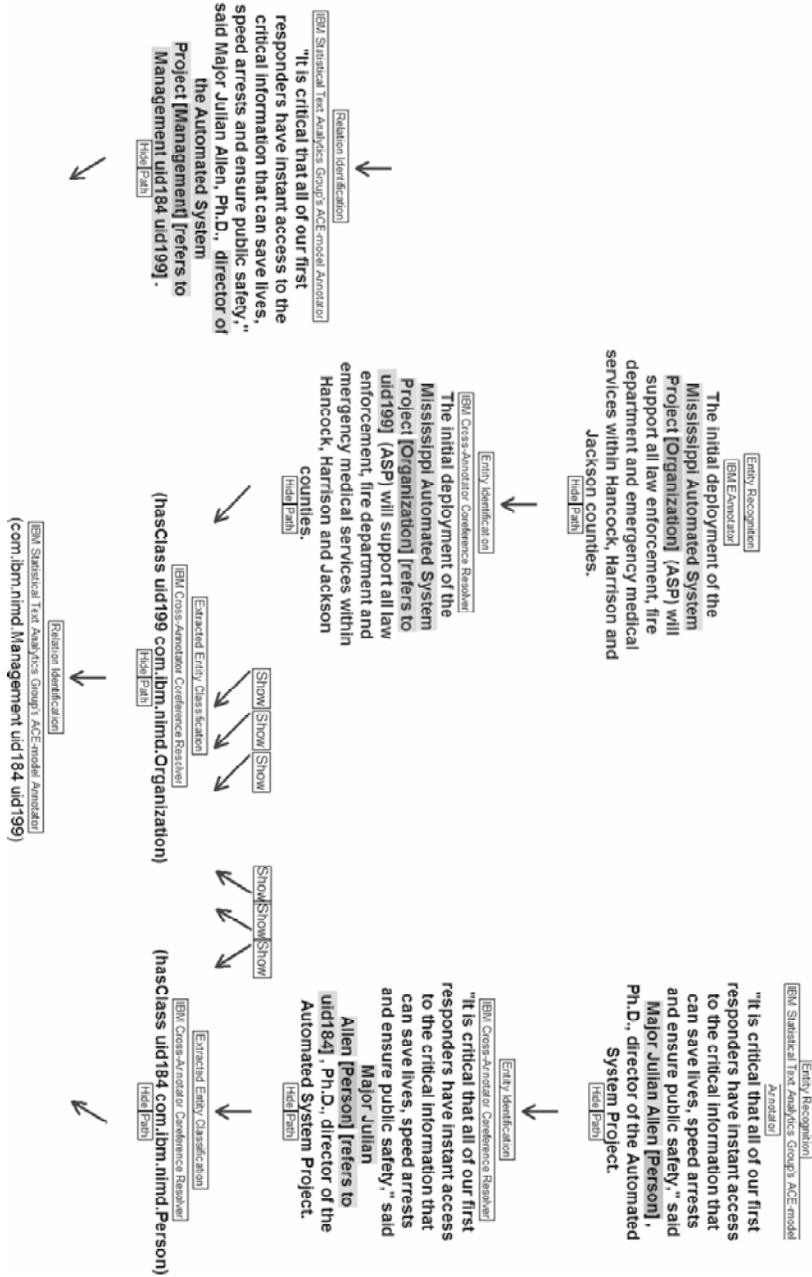The raw OWL for the final conclusion of that proof (which links to its antecedents, etc.) is at:

http://iw4.stanford.edu/proofs/MississippiAutomatedSystem/ns36.owl

Relation Identification | IBM Statistical Text Analytics Group's ACE-model Annotator | Hide Path

"It is critical that all of our first responders have instant access to the critical information that can save lives, speed arrests and ensure public safety," said Major Julian Allen, Ph.D., director of the Automated System Project [Management] [refers to Management uid184 uid199] .

Entity Recognition | IBM E-Annotator | Hide Path

The initial deployment of the Mississippi Automated System Project [Organization] (ASP) will support all law enforcement, fire department and emergency medical services within Hancock, Harrison and Jackson counties.

Entity Identification | IBM Cross-Annotator Coreference Resolver | Hide Path

The initial deployment of the Mississippi Automated System Project [Organization] [refers to uid199] (ASP) will support all law enforcement, fire department and emergency medical services within Hancock, Harrison and Jackson counties.

Extracted Entity Classification | IBM Cross-Annotator Coreference Resolver | Show Show Show | Hide Path

(hasClass uid199 com.ibm.nlimd.Organization)

Relation Identification | IBM Statistical Text Analytics Group's ACE-model Annotator

(com.ibm.nlimd.Management uid184 uid199)

Entity Recognition | IBM Statistical Text Analytics Group's ACE-model Annotator | Hide Path

"It is critical that all of our first responders have instant access to the critical information that can save lives, speed arrests and ensure public safety," said Major Julian Allen [Person], Ph.D., director of the Automated System Project.

Entity Identification | IBM Cross-Annotator Coreference Resolver | Hide Path

"It is critical that all of our first responders have instant access to the critical information that can save lives, speed arrests and ensure public safety," said Major Julian Allen [Person] [refers to uid184], Ph.D., director of the Automated System Project.

Extracted Entity Classification | IBM Cross-Annotator Coreference Resolver | Show Show Show | Hide Path

(hasClass uid184 com.ibm.nlimd.Person)

Figure 2: Partial Inference Web screen capture showing an extraction proof

Discussion

We are using a proof-oriented approach to provide the foundation for supporting explanation in a broad range of systems. Our work provides an encoding and infrastructure that allows explanations to include information beyond typical knowledge bases, for example, including unstructured portions of raw text used to generate knowledge base statements. Explanations can also point to knowledge bases that were used along with inference rules to generate conclusions. Utilizing Inference Web, we can also provide multiple views of the explanations, including source document summaries (what documents were used), KB summaries (what knowledge bases were used and what statements in those knowledge bases were used), summaries of trusted sources, assumption summaries, as well as information manipulation (deductive) summaries (what inference rules were used). The fact that the justification foundation is based on declarative specifications of information manipulation rules enables our work to be precise and extensible.

One contribution of our integration work is a more complete exposition of an integrated extraction and deduction process. The exposition of the appropriate portion(s) of original sources instead of or in addition to derived sources allows users to better evaluate the trustworthiness of answers. In our example, the answer was derived from KB1.owl in combination with a portion of the press release. The exposition of extraction rules helps focus the user's attention on the fact that the process may not be entirely based on sound rules. Our example proof uses the *Entity Recognition*, *Relation Recognition, and Relation Identification* rules (from extraction engines that may be unsound) in addition to *Transitive Property Inference* (from a theorem prover expected to be sound).

Another contribution of our work is the design and integrated use of a taxonomy of text analytic tasks along with rules describing tasks performed by other kinds of systems. The new work connecting to text analytic components provides the foundation for transparent integration of knowledge-based question answering systems with information retrieval and text analysis. Within the Inference Web framework, that now enables text analytic components to be integrated with theorem provers (such as Stanford's JTP, SRI's SNARK, etc.), expert systems (such as UFPE's JEOPS), information integrators (such as ISI's Prometheus), web service composition discovery services (such as Stanford's SDS), and task processing (such as SRI's SPARK). .

The work provides the possibility to interact more with applications that use automatic and semi-automatic methods to generate knowledge bases. In the past, most explanation systems have focused on knowledge bases that were carefully constructed by hand with authoritative data. As more reasoning systems rely on semi-automatic and automatic generation of knowledge support for understanding the question answering process becomes more critical. With our explainable text analytic platform, we can now expose imprecision in the knowledge base building process and help users understand and probe the system to make appropriate decisions. When imprecise methods are used, it becomes more critical to provide access to meta-information such as source, author, recency, etc. If users (humans and agents) can request this information along with the answer or filter answers based on this information, they can make more informed decisions about what information to rely on. Tools such as ours may be a key differentiator in situations such as those cited in

the Select Senate Committee Report on Iraq[2], where recommendations were made to provide judgments that are not overstated, that are supported by underlying intelligence, expose assumptions, and expose uncertainties in the judgments. We claim that our infrastructure provides the key to explanations that may be used with applications that use knowledge bases built manually, semi-automatically, or automatically by providing ways to filter, understand, and evaluate answers.

We have a prototype implementation of the integration between twelve UIMA text analytic components, the explanation system, and a theorem prover. We are exploring issues including granularity of inference and coverage. Our work is being used to explain answers in intelligence tasks in DTO's NIMD program. The explanations are available through the Inference Web interface and are also being exposed through a customized interface designed for analysts. We believe the work is reusable and extensible. The taxonomy of text analytic tasks has provided coverage adequate to explain the text analytic needs that arise from the intelligence tasks addressed to date in the program. Additionally, the taxonomy provides a level of abstraction that has been useful to date in explanations. This paper's contribution is the taxonomy and architecture. A preliminary evaluation of the explanation representation and reasoning infrastructure along with its services for intelligence analysts is described in [Cowell, et. al, 2006].

We provide access to meta-information associated with nodes in PML documents. Thus, if meta-information concerning confidence level, authoritativeness, recency, etc. is encoded, users will have an option of displaying it in explanation presentations and summaries. We have recently begun integration with algorithms for composing answer confidence levels from confidence levels associated with other sentences, such as in [Zaihrayeu *et al*., 2005, McGuinness et al., 2006]. We are integrating this work with social networks to provide a more complete solution to explaining and propagating trust information.

Finally, an interesting practical use of this work is the ability to use the inference web as a repository for information that is *hidden* from some resource-limited component, but may be needed later. One example of this is a general undo facility. In many of our components that need to process large amounts of data in memory, we do not have the resources to handle all the information leading to a particular conclusion, however on occasion we need that information, e.g. when conclusions are found to be incorrect and should be undone. Rather than keep that information in memory in all cases, we can load it back in from the inference web when needed.

Related Work

The idea that information extraction can be used to provide valuable information to supplement the structured sources available on the semantic web is relatively well-established (e.g., [Dill, Eiron, et al. 2003; Maynard, Yankova, et al. 2005; Cimiano & Völker, 2005; Welty and Murdock, 2006]). However, relatively little work exists on explaining information extraction.

There is significant work concerning building causal and/or explanatory representations of text analysis *results* (e.g., [Ram, 1994; Mahesh, *et al.*, 1994; Moldovan and Russ, 2001]). However, representing analysis *processes* is less

---

[2] intelligence.senate.gov/conclusions.pdf (conclusions 1&2)

common. One system that does reason about text analysis processes is Meta-AQUA [Cox and Ram 1999], which generates explanations of reasoning failures in the domain of story understanding in order to facilitate automated learning. However, the tasks of interest in Meta-AQUA are ones such as retrieving scripts and predicting outcomes that are relevant to extracting implicit information from text. These tasks are complementary to the tasks we have modeled, which involve extracting information that is explicitly stated in text.

Significant work also exists concerning support for answer provenance. Work exists on *Knowledge provenance* including *source meta-information*, which is a description of the origin of a piece of knowledge, and *knowledge process information*, which is a description of the information manipulation process used to generate the answer [Pinheiro da Silva *et al*., 2003]. *Data provenance* and *data lineage,* the database community analog to knowledge provenance, typically includes both a description of the origin of the information and the process by which it arrived in the database [Buneman *et al*., 2001; Cui *et al*. 2000]. Our work focusing on including extracted knowledge includes enhanced provenance information and thus provides a more complete solution to problems for which users need provenance information.

Finally, there has been a long history of work on explanation, from communities such as expert systems [Davis, 1979; Buchanan and Shortliffe, 1984; Swartout *et al*, 1991] and case-based reasoning [Leake, 1992; Aleven and Ashley, 1996; Goel and Murdock, 1996]. Inference Web continues that tradition and provides a standards-based method for declaratively specifying the types of inference and information manipulation steps one is interested in explaining. The existing Inference Web registry contains a specification of many of the inference types needed for traditional theorem proving and expert system style deduction. Our work integrating Inference Web with UIMA extends the reach of the potential explanations since we provide an infrastructure that supports inclusion of knowledge bases built with extraction techniques.

Conclusion

It is generally not acceptable for semantic web systems to present conclusions without additionally being able to provide details about how those conclusions were produced and ultimately why they should be believed. As systems rely more on facts that may have been built with semi-automatic or automatic methods potentially using web sources that are unknown to users, techniques must be included for exposing information concerning sources and a broad range of information manipulation methods. Our work provides a solution to the problem where answers may rely on facts extracted from source text using text extraction techniques. The answers may also rely on information manipulation steps executed by reasoning engines. A set of information sources supporting answers can include raw text in addition to typical ontologies and knowledge bases. A set of information manipulators may include extractors in addition to theorem provers, information integrators, service composition discovery engines, or any other kind of manipulator able to encode justifications in the Proof Markup Language. A set of information manipulation rules may include extraction rules providing an infrastructure capable of explaining text analytic processes as well as standard deduction processes. Our solution bridges a gap between traditional reasoning engine-based solutions and text-analytic-based solutions. Our infrastructure is available for use and individual components such as

the taxonomy of inferences, text analytic components, registry, browsers, etc. may be used individually. We have implemented our approach and are using it in several sponsored projects and are interested in additional users.

Acknowledgements

References

Ando, R. 2004. Exploiting Unannotated Corpora for Tagging and Chunking. *Proc. of ACL.*

B.G. Buchanan and E. H. Shortliffe, editors, 1984. Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project. Addison-Wesley Publishing Company, Reading, Mass.

Buneman, P, Khanna, S., and Tan, W. 2001. Why and Where: A Characterization of Data Provenance. *Proc. of 8th International Conference on Database Theory.*

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.

Cowell, A., McGuinness, D., Varley, C. and Thurman, D. 2006. Knowledge-Worker Requirements for Next Generation Query Answering and Explanation Systems. *Proc. of the Workshop on Intelligent User Interfaces for Intelligence Analysis, International Conference on Intelligent User Interfaces* (IUI 2006), Sydney, Australia.

Cui, Y., Widom, J. and Wiener, J. 2000. Tracing the Lineage of View Data in a Warehousing Environment. *ACM Trans. on Database Systems*, 25(2), 179-227.

Davis, R. 1979. Interactive Transfer of Expertise: Acquisition of New Inference Rules. *Artificial Intelligence* 12(2):121-157.

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.

Ferrucci, D. 2004. Text Analysis as Formal Inference for the Purposes of Uniform Tracing and Explanation Generation. IBM Research Report RC23372.

Ferrucci, D. and Lally, A. 2004. UIMA by Example. *IBM Systems Journal* 43, No. 3, 455-475.

Goel, A. and Murdock, J. W. 1996. Meta-Cases: Explaining Case-Based Reasoning. Ashok K. Goel and J. William Murdock. *3rd European Workshop on Case-Based Reasoning*, Lausanne, Switzerland

Leake, D. 1992. *Evaluating Explanations: A Content Theory.* Lawrence Earlbaum.

McGuinness,D.1996. *Explaining Reasoning in Description Logics.* Ph.D. Thesis, Rutgers University. Tech Report LCSR-TR-277.

McGuinness, D. and Pinheiro da Silva, P.2004. Explaining Answers from the Semantic Web: The Inference Web Approach. *Journal of Web Semantics* 1(4):397-413.

Deborah L. McGuinness, Paulo Pinheiro da Silva, Cynthia Chang. IWBase: Provenance Metadata Infrastructure for Explaining and Trusting Answers from the Web. Technical Report KSL-04-07, Knowledge Systems Laboratory, Stanford University, USA, 2004.

McGuinness, D. Pinheiro da Silva, P., Murdock, J. W., Ferrucci, D., 2005. Exposing Extracted Knowledge Supporting Answers. Stanford Knowledge Systems Laboratory Technical

Report KSL-05-03, Knowledge Systems Laboratory, Stanford University, USA, 2005. http://www.ksl.stanford.edu/KSL_Abstracts/KSL-05-03.html.

McGuinness, D. L., Zeng, H., Pinheiro da Silva, P., Ding, L., Narayanan, D., and Bhaowal, M. Investigations into Trust for Collaborative Information Repositories: A Wikipedia Case Study. WWW2006 Workshop on the Models of Trust for the Web (MTW'06), Edinburgh, Scotland, May 22, 2006.

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.

Moldovan, D. I., Rus, V. 2001. Transformation of WordNet Glosses into Logic Forms. 14th International Florida Artificial Intelligence Research Society Conference, Key West, Florida, 459-463.

Pinheiro da Silva, P, McGuinness, D. and Fikes, R. 2006. A Proof Markup Language for Semantic Web Services. *Information Systems* 31(4-5): 381-395.

Pinheiro da Silva, P., Hayes, P., McGuinness, D. L., Fikes, R. E., and Deshwal. P. 2005 Towards Checking Hybrid Proofs. Technical Report KSL-05-01, Knowledge Systems Laboratory, Stanford University, USA.

Pinheiro da Silva, P., McGuinness, D., and McCool, R. 2003. Knowledge Provenance Infrastructure. *IEEE Data Engineering Bulletin* 26(4), 26-32.

Swartout, W., Paris, C. and Moore, J. 1991. Explanations in Knowledge Systems: Design for Explainable Expert Systems. *IEEE Expert Systems*, 6:3, 58-64.

Welty, C. and Murdock, J. W. 2006. Towards Knowledge Acquisition from Information Extraction. *5th International Semantic Web Conference*.

Zaihrayeu, I., Pinheiro da Silva, P., and McGuinness, D. 2005. IWTrust: Improving User Trust in Answers from the Web. *3rd Intl. Conference on Trust Management*.