Exposing Extracted Knowledge Supporting Answers*

Deborah L. McGuinness[1] **Paulo Pinheiro da Silva**[1] **J. William Murdock**[2] **David Ferrucci**[2]
[1] Knowledge Systems, Artificial Intelligence Lab, Stanford University, Stanford, CA USA
{dlm, pp}@ksl.stanford.edu
[2] IBM Watson Research Center, 19 Skyline Dr., Hawthorne, NY USA
{murdock, ferrucci}@us.ibm.com

Abstract

As more systems rely on knowledge bases built from automatic and semi-automatic methods, it is becoming more important to provide solutions that not only answer questions but also provide information describing how the answers were obtained. We aim to make answers more useful that rely on the combination of knowledge bases, some of which may have been built as the result of extraction processes and others may have been hand constructed from reliable sources. Our solution provides an infrastructure capable of encoding justifications for answers in a single format. This provides an end-to-end description of the knowledge derivation process including access to the raw text documents, descriptions of the text analytic processes used for extraction, as well descriptions of the ontologies and many kinds of information manipulation processes, including standard deduction. We have implemented our solution and we are using it in several sponsored projects.

1 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 answers. In these early systems, answers 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 been viewed to typically contain correct and trustworthy information and the reasoners may have been viewed to be correct. The information about answer generation process typically focused on the derivation path and it was typically referred to as an explanation of the answer. Sometimes the explanations also included some limited information about the facts from the knowledge bases used to obtain answers. Sometimes there was additional focus on taking the information concerning the derivation path and making it more understandable to the end user.

Today's systems still require this kind of support for question answering systems, however now they also have additional needs. Today's knowledge bases are quite diverse. They may be distributed, thus potentially being created by multiple sources at different times using varying methods applied to a wide variety of information input. Some of the information input may be in the form of natural language text and text analytic technique may be used to automatically generate logical statements from the raw text. These logical statements may then provide a portion of or be an entire knowledge base from which some question answering systems may obtain answers.

Since some of the extraction techniques use heuristics or statistical methods, they are known to produce some conclusions that are not sound. Since these statements may be input directly into the knowledge base, it is now the case that some of the statements in the knowledge base, even if they were derived originally from authoritative natural language source material, may be questionable. These systems now need enhanced support that includes access to the knowledge base statements used to obtain conclusions. These systems also need information about how those statements came to be included in the knowledge base and what raw source was used (potentially by text extraction engines) to generate the facts.

Sometimes time may be critical and applications will not have the luxury of waiting for source material to be vetted before it is used. Thus, it could be the case that some of the raw source material may not be reliable. Thus explanations that expose raw sources and any meta-information known about the sources (such as authoritativeness, recency, etc.) become important as well.

Consider an example where an analyst may be interested in knowing where a particular individual's office(s) are. One or more documents may be available that have information that contains the name of the individual that the analyst is interested in. One document may be an ontology containing an axiom stating that an individual owning an organization may have an office at the organization. Another document may be a knowledge base stating that the individual of interest is the owner of an organization. From information in

* This work is partially funded by the Advanced Research and Development Activity under the Novel Intelligence from Massive Data (NIMD) program.

these structured documents, a system may be able to determine information about individuals, their ownership of businesses, locations of those businesses, and the potential locations of their businesses. However, the system may be unable to determine any useful information about individuals if the second document is an unstructured document, i.e., raw text, instead of a knowledge base.

Text extractors may be run over text documents to identify relationships between individuals and businesses and generate structured facts that may be used by reasoners. Conclusions about these office locations may then need to be inspected to determine how recent the information was that was used, how reliable the original documents were, what techniques were used to extract information from the text documents (and their reliability), what techniques were used to identify that two individuals were the same person (if this was used), and any inferences that were made with knowledge base statements – such as the inference that people typically have offices at businesses they own.

In this paper, we will describe a solution infrastructure that provides meta-information about raw text sources (and appropriate portions) used to generate knowledge bases. The solution also provides access to meta-information about hand constructed (or semi-automatically constructed knowledge bases). This is integrated with our explanation infrastructure that provides access to information concerning the reasoning path used to obtain answers and information manipulation techniques and systems. We call this information containing the meta-information about sources, information manipulators (text analytic engines or reasoners), and the derivation paths knowledge provenance. We distinguish this from one traditional sense of explanation since this paper is not addressing the issue of presentation techniques for this information that may include abstractions and dialogues[1]. We will introduce the solution architecture and describe its main components. We will also present an example of the system in use and include a discussion of the benefits focusing on the new capability of supporting systems that answer questions from knowledge obtained partially or completely from unstructured text.

2 Solution Architecture

Our solution relies on two 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 answers from question answering systems that manipulate structured information, and now unstructured information [McGuinness and Pinheiro da Silva, 2004]. We have developed new capabilities that allow us to combine IW and UIMA, enabling the former to present explanations of analysis performed within the latter.

[1] Presentation of knowledge provenance is addressed in our other related work omitted from this document for blind reviewing needs.

2.1 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 formal 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.

UIMA 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 were indicated in the corpus) but also some intermediate analysis results (such as assigning types to spans of text) and links among the intermediate and final results.

2.1 Inference Web

Inference Web provides an infrastructure for providing explanations from distributed hybrid question answering 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, and a registrar for storing and maintaining proof related meta-information. It also includes services for helping question answering systems to generate PML, check PML documents for valid applications of inferences, and services for automatic registration of sources and meta-information.

Inference Web provides the foundation for distributed and extensible explanations and UIMA provides the foundation for text analytics.

2.3 Integration

There are two key technical areas that required research in order to allow Inference Web to provide explanations of results produced within UIMA. The first involves specifying a taxonomy of tasks that are performed during the extraction of knowledge from text. The second involves a software component that takes a UIMA EKDB as input and produces a PML document as an output. The content produced by that software component is based on the taxonomy of extraction tasks.

We generated a taxonomy motivated by the need to describe and explain nine types of extraction tasks. A forthcoming publication provides an in depth discussion of these

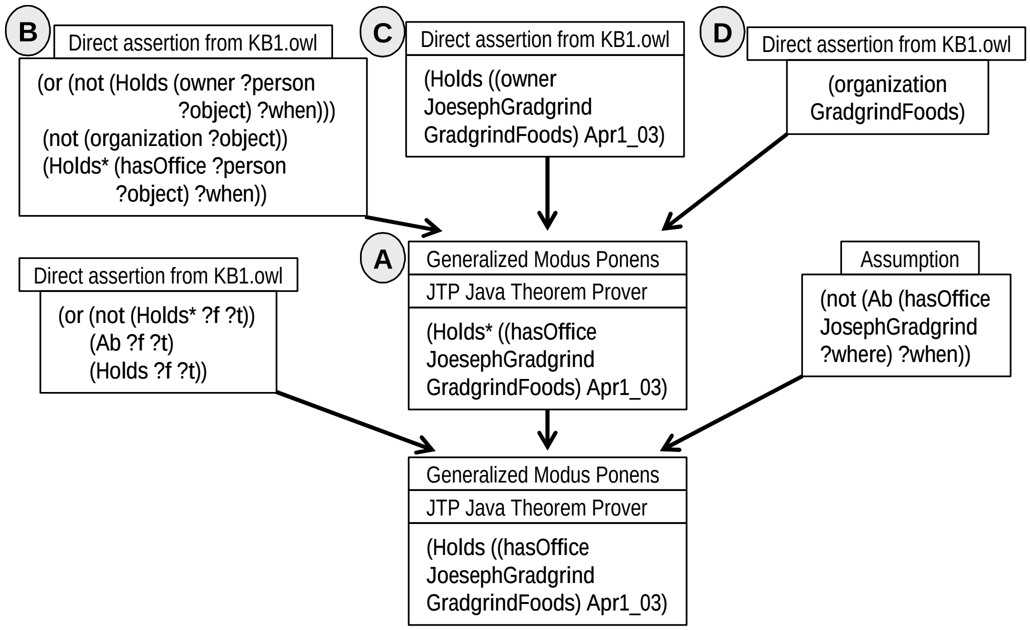

Figure 1: Question answering from knowledge base.

tasks and the level of granularity. While we do anticipate adding more to the taxonomy as we support explanations of a wider variety of text analytic components, it is an interesting finding that these nine satisfy the explanation requirements for the current components we are using in an intelligence project. For the purposes of this paper, we will focus on only a few tasks and will introduce them by example. The **Entity Recognition** task 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*. Similarly, the **Relation Recognition** task assigns a relation type to a span (e.g., that some sentence describes a relation of type *Owner*). Other tasks involve determining and assigning values to the roles of a relation (e.g., that a particular person is a participant in a given ownership relation instance), resolving coreference (e.g., determining that text spans that cover the strings "Joseph Gradgrind" and "Joe Gradgrind" are the same person), etc.

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 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*.

3 Example in Action

In this section we describe two scenarios in which a query answer is derived from extracted knowledge. We identify the knowledge provenance information in the example. In both scenarios, we ask the same question: "Where did Joseph Gradgrind have an office on April 1, 2003?" In both scenarios, the system answers "Joseph Gradgrind had an office at Gradgrind Foods on April 1, 2003." In one of the scenarios, the answer is derived from axioms and facts encoded in a single knowledge base where one of the facts in the KB was extracted from text. In the other, the extracted fact is no longer encoded in the knowledge based but is produced directly from raw text. The contrast between these two scenarios demonstrates some of the added value that can be provided by maintaining and presenting records of extraction processes.

3.1 Question Answering from Knowledge Bases

Figure 1 contains a proof supporting one answer for where Gradgrind has an office on the date in question. The answer was derived by Stanford's JTP theorem prover using a knowledge base containing some direct assertions, a typicality assumption, and a context reasoner. The format shown in Figure 1 is approximately the same format that the Inference Web Browser uses to present proofs.

The direct assertions in Figure 1 are from a single knowledge base, `KB1.owl`. One fact asserted in the KB is that Joseph Gradgrind was the owner of Gradgrind Foods on April 1, 2003, as stated in node (C). Another is that Gradgrind Foods is an organization, as stated in node (D). An axiom asserted from the KB and encoded in node (B) says that if a person owns a certain thing, and that thing happens to be an organization, then *provisionally*[2] that person has an office located at that organization. This axiom is stated as provisional since users may want to identify and expose abnormal situations [McCarthy, 1986], if any. Thus, typicality assumptions need to be explicitly assumed in order to derive statements that do not have provisional qualifiers associated with them. (Non-qualified statements are referred to as non-provisional.). For instance in our scenario, a JTP user explicitly assumed that no office-occupation involving Joseph Gradgrind is atypical. Thus, JTP answers that Joseph Gradgrind had an office at Gradgrind Foods on April 1, 2003, as stated in the conclusion of the proof displayed in Figure 1.

In the above example, the proof has full provenance information about the information presented to JTP; in the situation above, that information consists of `KB1.owl` and a typicality assumption provided by a JTP user. However, the proof does not provide any information about how the knowledge got into `KB1.owl` to begin with. In some cases, knowledge of this sort is extracted automatically from raw text. In those cases, it can be useful to add additional steps to the proof describing that extraction process, as described in the next subsection.

3.2 Question Answering From Knowledge Bases and Texts

Figure 2 shows a variation for part of the proof in Figure 1. The nodes labeled (A) in both Figures 1 and 2 are the same. Nodes (B) and (C) have been removed from this figure (to save space); the labels are left in to show where those nodes occur. The difference between the proofs is that the node (D) in Figure 1 was replaced by the node (D') in Figure 2. Nodes (D) and (D') have the same conclusion but have different justifications for that conclusion.

The justification for node (D) in Figure 1 is that the conclusion is asserted in `KB1.owl`. The justification for the node (D') in Figure 2 is that GradgrindFoods is that the conclusion was derived by a knowledge extraction process involving the consecutive use of three UIMA-compliant components. IBM EAnnotator [Ando, 2004] concludes that a span (i.e., a segment of the original text `gradgrind.txt`) refers to some unspecified entity instance of type *organization* (in the extraction ontology); i.e., it produces an entity annotation. IBM Cross-Annotator Coreference Resolver determines that the entity annotation on "Gradgrind Foods" refers to the entity instance GradgrindFoods. Finally, IBM Cross-Document Coreference Resolver concludes that GradgrindFoods in the extracted KB has a type *organization* (based on the type assigned to the entity annotation).

[2] The concept of provisionality is indicated by the Holds* predicate and is defined by the rule (or (not (Holds* ?f ?t)) (Ab ?f ?t) (Holds ?f ?t)). That rule may be read as "if a fact provisionally holds at a time, then either that fact holds at that time, or that fact is abnormal at that time." This concept makes it possible to reason about axioms that are typically (but not always) valid.

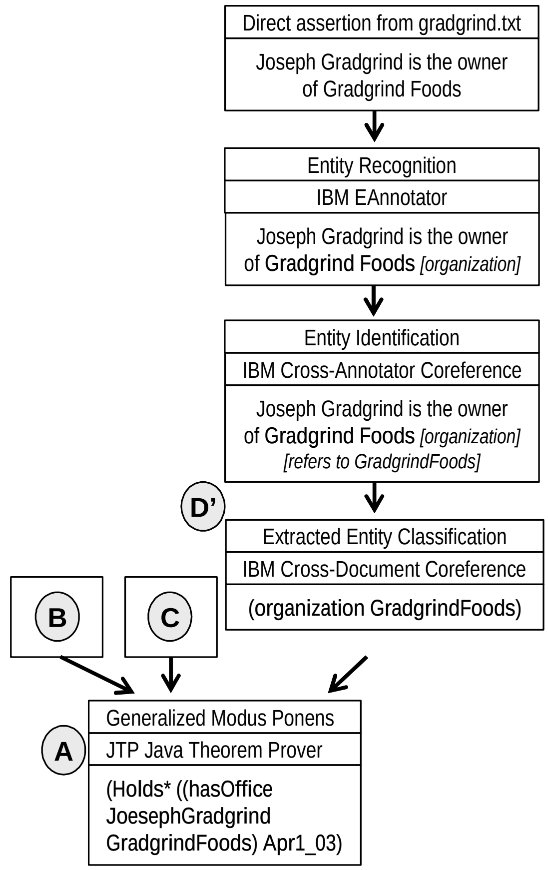

Figure 2: Question answering using the knowledge base and raw text.

4 Discussion

We are using a proof-oriented approach to provide the foundation for supporting explanation in a broad range of question answering systems. Our work provides an encoding and infrastructure that allows question answering system 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, such as `gradgrind.txt`, were used), KB summaries (what knowledge bases were used, such as `KB1.owl`, and what statements in those knowledge bases were used, such as GradgrindFoods is an organization in the first scenario), summaries of trusted sources, assumption summaries (what assumptions were used to obtain conclusions, such as Gradgrind's ownership is not atypical), as well as information manipulation (deductive) summaries (what inference rules were used, such as generalized modus ponens from JTP or entity extraction from UIMA). 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 work is the exposition of a more realistic picture of question answering processes. For instance, the exposition of original sources instead of or in addition to derived sources allows users to better evaluate the trustworthiness of answers. In the example in Figure 1, the answer was derived from `KB1.owl` in combination with the JTP user's assumptions. In Figure 2, the answer is additionally depends on `gradgrind.txt`. 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. For instance, the proof in Figure 2 uses the *Entity Recognition*, *Entity Identification* and *Extracted Entity Classification* rules in addition to *Generalized Modus Ponens*. An exposition of these components shows that the final answer is not entirely derived by a deductive inference engine. The original proof was entirely derived by JTP. The new proof is partially derived using JTP and partially derived using the three IBM extraction engines, all of which may use unsound inference.

Another contribution of our work is the integrated use of a taxonomy of text analytic tasks along with rules describing tasks performed by other kinds of systems. There are several kinds of systems providing explanations encoded in PML. For example, in addition to text analytic components, the Inference Web currently supports explanations for 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). We are finding that a declarative specification of information manipulation steps provides a flexible and powerful foundation for explanations in much broader contexts. This integration allows our infrastructure to provide the basis for explaining answers from hybrid and distributed reasoning applications.

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 applications rely on semi-automatic and automatic generation of knowledge that may be relied on by reasoning systems, 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 (such as some used in extraction methods in Figure 2), it becomes more critical to provide access to meta-information such as source, author, recency, etc. If users (humans and agents) have the option to be presented with 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[3], 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. Our claim is 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.

One topic worth noting is support for confidence level reporting. Currently, 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*., 2004]. We are testing this work and integrating it with FOAF networks to provide a more complete solution to explaining and propagating trust information.

5 Related Work

There is a significant amount of existing work on building causal and/or explanatory representations of the *results* of text analysis (e.g., [Ram, 1994; Mahesh, *et al.*, 1994; Moldovan and Russ, 2001]). Representing analysis *processes* is less 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.

There is also a significant amount of existing work on supporting answer provenance. *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]. The exposition of 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 [Buchanan and Shortliffe, 1984, Swartout *et al*, 1991]. Inference Web continues that tradition and provides a method for declara-

[3] See conclusions 1 and 2 in http://intelligence.senate.gov/conclusions.pdf

tively 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.

6 Conclusion

It is becoming less common for question answering systems to be able to simply return answers without additionally being able to provide details about how the answers were produced and ultimately why the answers should be believed. As systems rely more on facts from knowledge bases that may have been built with semi-automatic or automatic methods potentially using 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. We have implemented our approach and are using it in several sponsored projects and are interested in additional users.

References

[Ando, 2004] Rie Ando. Exploiting Unannotated Corpora for Tagging and Chunking. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004.

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

[Buneman *et al*., 2001] Peter Buneman, Sanjeev Khanna and Wang-Chiew Tan. Why and Where: A Characterization of Data Provenance, In *Proceedings of 8th International Conference on Database Theory*, 316-330, 2001.

[Cui *et a*l. 2000] Yingwei Cui, Jennifer Widom and Janet L. Wiener, Tracing the Lineage of View Data in a Warehousing Environment, In *ACM Trans. on Database Systems*, 25(2), 179-227, 2000.

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

[Ferrucci and Lally, 2004] David Ferrucci and Adam Lally. UIMA by Example. *IBM Systems Journal* 43, No. 3, 455-475, 2004.

[McCarthy, 1986] John McCarthy. Applications of Circumscription to Formalizing Common-Sense Knowledge. *Artificial Intelligence* 28:89—118. 1986.

[McGuinness, 1996] Deborah L. McGuinness. *Explaining Reasoning in Description Logics*. Ph.D. Thesis, Rutgers University, 1996. Technical Report LCSR-TR-277.

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

[Pinheiro da Silva *et al.*, 2005] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Richard Fikes. A Proof Markup Language for Semantic Web Services. *Information Systems* (to appear)

[Pinheiro da Silva *et al.*, 2003] Paulo Pinheiro da Silva, Deborah L. McGuinness, and Rob McCool. Knowledge Knowledge Provenance Infrastructure. In *IEEE Data Engineering Bulletin* 26(4), 26-32, 2003

[Swartout, et al., 1991] William R. Swartout, Cecile Paris, and Johanna D. Moore. Explanations in Knowledge Systems: Design for Explainable Expert Systems. *IEEE Expert Systems*, 6:3, 58-64. 1991.

[Zaihrayeu, et al., 2004] Ilya Zaihrayeu, Paulo Pinheiro da Silva, and Deborah L. McGuinness. *IWTrust: Improving User Trust in Answers from the Web*. Technical Report DIT-04-086, Informatica e Telecomunicazione, University of Trento, Italy