Epistemic Structures of Interrogative Domains

By

Cameron A. Hughes

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Computing and Information Systems

Youngstown State University

May 2008

Epistemic Structures of Interrogative Domains

Cameron A. Hughes

Signature:

_____

Cameron A. Hughes Student Date

Approvals:

_____

Dr. Alina Lazar, Thesis Advisor Date

_____

Dr. John Sullins, Committee Member Date

_____

Dr. Yong Zhang, Committee Member Date

_____

Peter J. Kasvinsky, Dean of School of Graduate Studies & Research Date

Abstract

At Ctest Laboratories we are exploring the notion of automated conversion of the semi-structured text to an epistemic structure suitable for deductive inference. In this paper we will develop an epistemic structured representation for electronic transcripts of interrogative domains. We propose that knowledge which is typically not visible to keyword search or string matching, can be readily extracted from the an electronic transcript when it is given an appropriate epistemic structure. We introduce an Epistemic Structure E_s and a process for converting a semi-structured transcript from and interrogative domain to E_s. In this paper we restrict our discussion and analysis to transcripts that have been stored as semi-structured text. In particular we are interested in any knowledge that can be deduced by an interrogative agent from the content of an electronic transcript. Further we develop the notion of an interrogative agent that relies on epistemic justification as a condition for knowledge.

Acknowledgements

I would like to give a shout out to Dr. Alina Lazar, Dr. John Sullins, Dr. Yong Zhang and Dr. Eugene Santos without which this thesis would not have been possible.

Table of Contents

Introduction.....10

1 Why Epistemology.....10

1.1 The Classical Tripartite Analysis.....12

1.2 Why the Tripartite Definition?.....13

1.3 Which Discrete Structures.....13

2 Interrogative Agents and what they Consider Possible.....16

2.1 Semantics Of Agency for Interrogative Agents.....20

2.2 But What Does The Interrogative Agent Know..... 23

2.3 Closed World Assumption.....25

3 A Closer Look at the Epistemic Structure.....26

3.1 a *priori* and a *posteriori* Propositions.....28

3.2 Justification in E_S 31

3.3 Modeling Belief in E_S.....34

3.4 Non-monotonic Justifications.....34

3.5 Knowledge Space K_s of an Interrogative Agent 35

4 Propositional Knowledge in Interrogative Domains.....36

5 The NOFAQS Project.....37

5.1 Background.....42

5.2 The Corpus.....42

5.3 Model Theoretic Semantic of Transcript.....43

5.4 Important Target Predicates.....44

5.5 An Interrogative Agent.....46

5.6 The Method.....47

5.7 Frames for G_1 and G_2.....48

5.8 Q&A Pair Contingent Propositions.....50

5.9 Discussion.....52

Conclusion.....53

Future Work.....55

References.....56

List Of Figures

1. Nwana's Classic Breakdown of Agent Types..... 19

2. An Interrogative Agent is intersection of Q&A systems,

epistemic logic, information extraction systems and agents..... 20

3. A Simplified Structure of An Interrogative Agent..... 22

4. Epistemic Justification Taxonomy..... 33

List of Tables

1. Analytic vs. Synthetic Propositions..... 29

2. Classification of Epistemic Justification..... 31

3. Simple Question Classifications..... 44

4. Classes and Sample Relations..... 48

Introduction

At Ctest Laboratories we are exploring the notion of automated conversion of the semi-structured text to an epistemic structure suitable for deductive inference. In this paper we will develop an epistemic structured representation for electronic transcripts of interrogative domains. We propose that knowledge which is typically not visible to keyword search or string matching, can be readily extracted from the an electronic transcript when it is given an appropriate epistemic structure. We introduce an Epistemic Structure E_s and a process for converting a semi-structured transcript from an interrogative domain to E_S. In this paper we restrict our discussion and analysis to transcripts that have been stored as semi-structured text. In particular we are interested in any knowledge that can be deduced by an interrogative agent from the content of an electronic transcript. Further we develop the notion of an interrogative agent that relies on epistemic justification as a condition for knowledge.

1 Why Epistemology?

Epistemology is the study of knowledge [1]. The goal of Epistemology as a field is to answer the questions: What is knowledge? How is knowledge acquired? Can we know anything? Are there limits to what we can know? How can we determine the quality of knowledge? What are the sources of knowledge? What differentiates knowledge from other things that we hold mentally? How can we determine when something counts as knowledge? At what point can we say that an agent has

knowledge? Epistemology examines the relationships between what is true and what we accept as true[2]. Investigations of the subject and structure of knowledge can be traced back to Plato (c. 427-c. 347 B.C) in his Theaetetus and Aristotle's Posterior Analytics (384-322 B.C). In addition to the primary questions of Epistemology, the pursuit is often divided on the types of knowledge. There are typically three divisions:

- Propositional Knowledge
- Procedural Knowledge
- Interrogative Knowledge

We rely on Rescher [3] for our use and interpretation of interrogative knowledge. Here we will limit our discussion of Epistemology to its treatment of propositional knowledge. In particular we are interested in the Tripartite Analysis of Knowledge. We direct our attention to the Tripartite Analysis because it invigorates and provides the primary foundation for the computer based epistemic structure that we introduce later. Although the Tripartite Analysis has short comings and has been thoroughly criticized , [4] for the basic attack on the Tripartite deconstruction, it is well suited as a spring board in our development of an epistemic representation for software agents.

In the context of this paper, the possible worlds, and interrogative domains that our software agents confront along with the closed world of assumption that they rely on are completely predetermined, logically constrained, and in most cases deductively defined. Therefore the Gettier problem with the Tripartite Analysis is of little concern to us here.

Further, in those rare incidences where the Gettier attacks may have merit for our application we select defeasibility as our fourth condition [5]. The augmentation of the Tripartite Analysis that Keith Lehrer and Thomas D. Paxson [5] layout in their "Knowledge : Undefeated Justified True Belief" provides enough cover for our software agent's epistemic representation to be safe from Gettier. We shall have much to say about possible worlds and closed world assumptions when we discuss the logical foundations for our epistemic structure, but first let's take a closer look at the Tripartite Analysis of Knowledge.

1.1 The Classical Tripartite Analysis

The basic Tripartite Analysis of Knowledge says that knowledge is justified , true, belief. This analysis dates back to Plato, and its treatment has received much discussion for the last 500 years[6]. We ask that the reader make a mental note of the definition of knowledge as justified true belief because the components in our software based epistemic structure map directly to this definition. In fact our epistemic structure is logically tripartite with respect to functions of the individual components. Given that we have an individual **A**, and a proposition **P** and some justification(s) **J,** the classical analysis can be easily represented as
follows:

A Knows **P** *iff*

P is true

A believes that **P** is true

A is justified in believing that **P** is True

That is **A** can only be said to know something that is true, that he believes to be true, and that he is justified in believing.

1.2 Why the Tripartite Definition?

We are encouraged by this definition for two primary reasons: First, it can be simply stated and thereby gives us a compass that helps us to find our way during the formulation of an epistemic structure. Second, the Tripartite Analysis lends itself easily to a discrete representation. In particular the artifacts of the Tripartite Analysis fit nicely into First Order Logic (FOL), Graph Theory and Set theory. This is important because the type of software structures and notions discussed in this paper are best described as discrete structures. Since both the artifacts of the Tripartite analysis and our epistemic structure can be represented using discrete structures, we can use discrete structures as our bridge from epistemological subject matter to knowledge representation for software-based agents. We use software-based agents to negotiate explicit and implicit knowledge from our interrogative domains.

1.3 Which Discrete Structures?

The discrete structures that have our primary attention in this paper are the S and K systems which form the basics of modern day modal logic [7,8]. In particular the S5 system and Kripke structures. While we will have occasion to draw from some of the basic notions of Set theory, and Graph theory, we are especially interested in the use of Kripke structures as a formalization of Possible Worlds Model. The possible worlds

model is central to our discussion because our epistemic structure is in part a deconstruction and clarification of what it means for a world to be possible to a software agent.

"The intuitive idea behind possible-worlds models is that besides the true state of affairs, there are a number of other possible states of affairs or worlds. Given his current information an agent may not be able to tell which of a number of possible worlds describes the actual state of affairs" [9]. An agent is said to know a fact ϕ if ϕ true at all the worlds he considers possible. Our epistemic structure is concerned with and enumerates "what an agent considers possible". To bring possible worlds into focus will require a little notation and a description of a Kripke structure[7].

Kripke structures are used to delineate the notion of possible worlds and an agent's knowledge with respect to those possible worlds. A Kripke structure *M* for *n* agents **over** ϕ is a tuple:

(1) $M = (S, \pi, K_1, \ldots K_n)$

Where S is a set of possible worlds, $\pi$is an interpretation that associates with each possible world S a truth assignment, and K_i is a binary relation on S. Further, an agent *A* is said to know (K) a proposition ψ **if** ψ is true at all worlds that *A* considers possible. If we let $s \in S$ then we can formally state the agent's knowledge with respect to ψ :

(2) $(M,s) \models K_A \psi \leftrightarrow (M,t) \models \psi$ for all t such that $(s,t) \in K_A$

(2) provides a Kripke epistemic interpretation for an agent. The Kripke structure can be seen as a very simple model of the agent's knowledge space. It is our aim to summon our epistemic structure to bring flavor to the simple Kripke epistemology. The relation *(s,t)* in (2) captures the worlds the agent considers possible. The relation *(s,t)* forms the agent's epistemic alternatives [10] . But what is the justification for *(s,t)* ? What is the epistemic basis for *(s,t)?* In answer to these question we develop a justification for those alternatives using our epistemic structure. We deploy E_s as a tripartite decomposition of S as it relates to K. Thus we denote S:

(3) $$S \equiv E_s$$

Where E_s is used provide a more epistemological foundation for K_i as it relates to *M* and our interrogative domains. Our intention here is to provide an epistemic foundation [11] and epistemic justification[12] for the agents that are denoted in the Kripke structure. We propose an extended Kripke model for epistemic agents. In this extended mode an agent's knowledge is consistent with a *logical interpretation* of the traditional Tripartite Analysis. That is an epistemic agent *A* is said to know a proposition ψ *iff* :

(4) $(M,s) \models K_A \psi \leftrightarrow (M,t) \models \psi$ for all t such that $(s,t) \in K_A$

A believes ψ

A is justified in believing ψ

But before we can relate possible worlds to semi-structured domains and then to our

epistemic structure we will take a closer look at what we mean by an agent and why an epistemic foundation for our interrogative agents is necessary.

2 Interrogative Agents And What they Consider Possible

Typically a HTML or XML capable software browser is used to allow the user to view and search semi structured documents that have been encoded using HTML or XML. In the case of our interrogative domains, these include documents such as the transcripts of federal court cases and transcripts of congressional hearings . HTML or XML software browsers can be used to search these documents. The user who is interested in a particular content may try to search for particular documents or for particular areas within documents using keywords or key phrases. The browsers have search features that support keyword and key phrase search. In the case of the Internet and most intranets the user has access to search engines. These search engines typically allow the user to retrieve a set of documents based on keyword or key phrase matching. Once the documents are retrieved the user can use the HTML or XML browser to locate the content containing the keywords within the document. The documents returned to user are usually guaranteed to contain the keywords or phrase the user supplied.

We contrast our method of answering a user's query with that of the HTML or XML based browser/ search engine combination. In our scenario instead of returning a collection of documents that may or may not be relevant to the user we deploy an epistemic agent [13] to answer the user's query directly. Whereas with the browser/search engine combination the user determines the relevance and usefulness

based on reading or reviewing the documents returned, in our case the user relies on the knowledge of a software agent to understand the query presented and to provide a reliable response to the query. Our agent is a derivative of the classical Question and Answer (Q&A) systems. In a classical Q&A system a question or query is posed in natural language. A question analyzer deconstructs the question. A question categorizer classifies the question. This classification narrows down the search space for possible answers. The question is converted into a FOL representation or some other structure suited for query processing [34]. A set of candidate answers are extracted ranked and presented to the user[14]. The agent's justification is constrained to the context of the document. We restrict the context of our Q&A system to electronic transcripts from interrogative domains. This allows for coherent analysis for the agent [15] leading to knowledgeable answers presented to the user. Several questions immediately present themselves once a user relies on our epistemic agent to answer her query:

- How does the user know that the response delivered by the agent is reliable?
- Unless the agent's response is obvious how does the user know that the response is even relevant?
- How does the user know that the response delivered by the agent is complete or impartial?
- How does the user know that the response delivered by the agent is correct?
- How does the user know or trust what the agent knows ?

It is precisely these questions that our epistemic structure cautiously approaches and

hopes to appease. These questions are not at issue in the browser/search engine approach because the documents are available for the user to inspect directly. The user must synthesize what she wants from the documents. The user determines whether the documents are relevant or not. The user answers her own query by reviewing the documents returned. Our approach shifts the responsibility of extrapolating the answer from the user to the agent. Instead of relying solely on her knowledge in answering a query from the documents retrieved, the user is (*at least partially*) subject to the knowledge of the agent. We are convinced that our epistemic structure provides the initial ground work for a productive interaction between user and software agent with respect to transcript analysis.

In this paper we are concerned with the epistemology of *software agents* in a very narrow domain, in particular we look exclusively at interrogative agents over interrogative domains. Although a precise and agreed upon definition for what constitutes an *agent* is at the very least controversial [16], and at the time of this writing still elusive, we find the definition given by Michael Wooldridge[17] convenient:

"By the term agent, I mean an entity that acts upon the environment it inhabits. Agents are not merely observers of their environment, nor are they passive recipients of actions performed by other entities. Rather Agents are the active, purposeful originators of actions."

To Wooldridge's definition we add the notion of an *agency relationship* between the user and the agent. An *agency relationship* is present when one party(the principal) depends on another party (the agent) to undertake some task on the principal's behalf [18]

We distinguish our interrogative agent from the many software agent types. Figure 1 contains Nwana's[19] classic breakdown of agent types

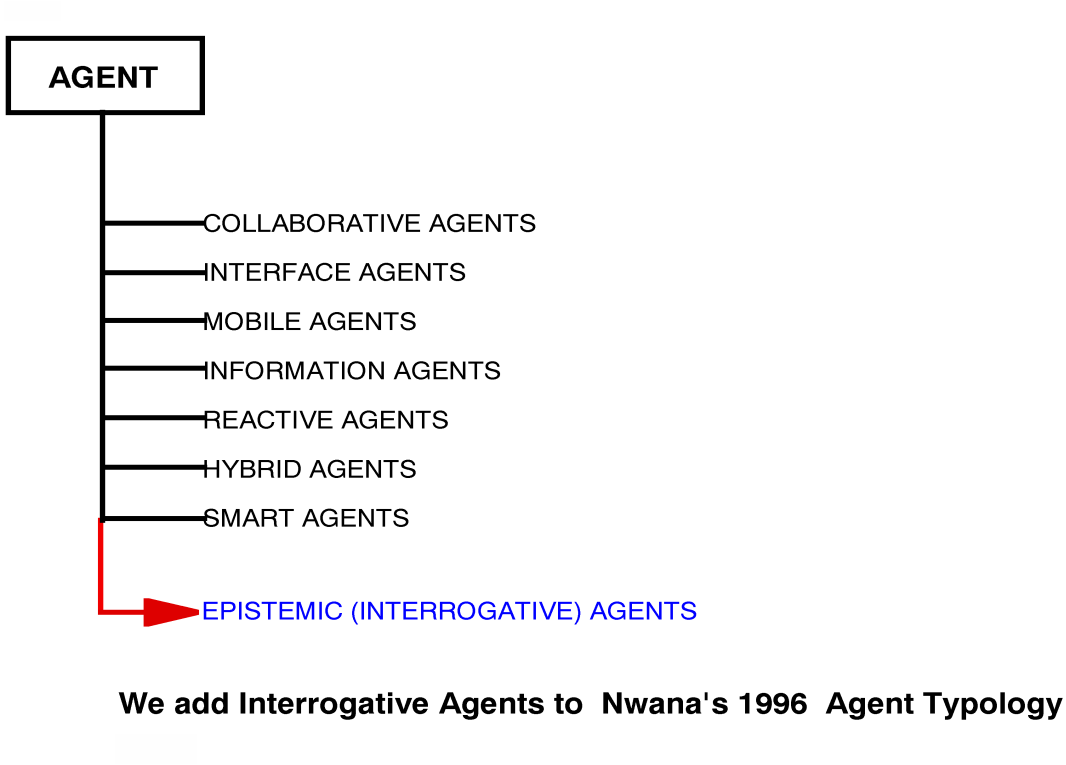

Figure 1. Nwana's classic breakdown of agent types.

Note that in Figure 1 we add to Nwana's classification the notion of an epistemic agent. In particular, we add an epistemic agent that is restricted to interrogative domains. Nwana's classification in Figure 1 includes an interface agent and an information agent. Our interrogative agent could be considered a subset of the interface agent class and understood as a superset of the information agent class. In the later case we choose superset because the interrogative agent is capable of retrieving knowledge as well as information. We can group our interrogative agent with interface agents because our

interrogative agent provides a natural language interface to the user. However our interface agent is best understood as an epistemic agent that is formed by the intersection shown in Figure 2.

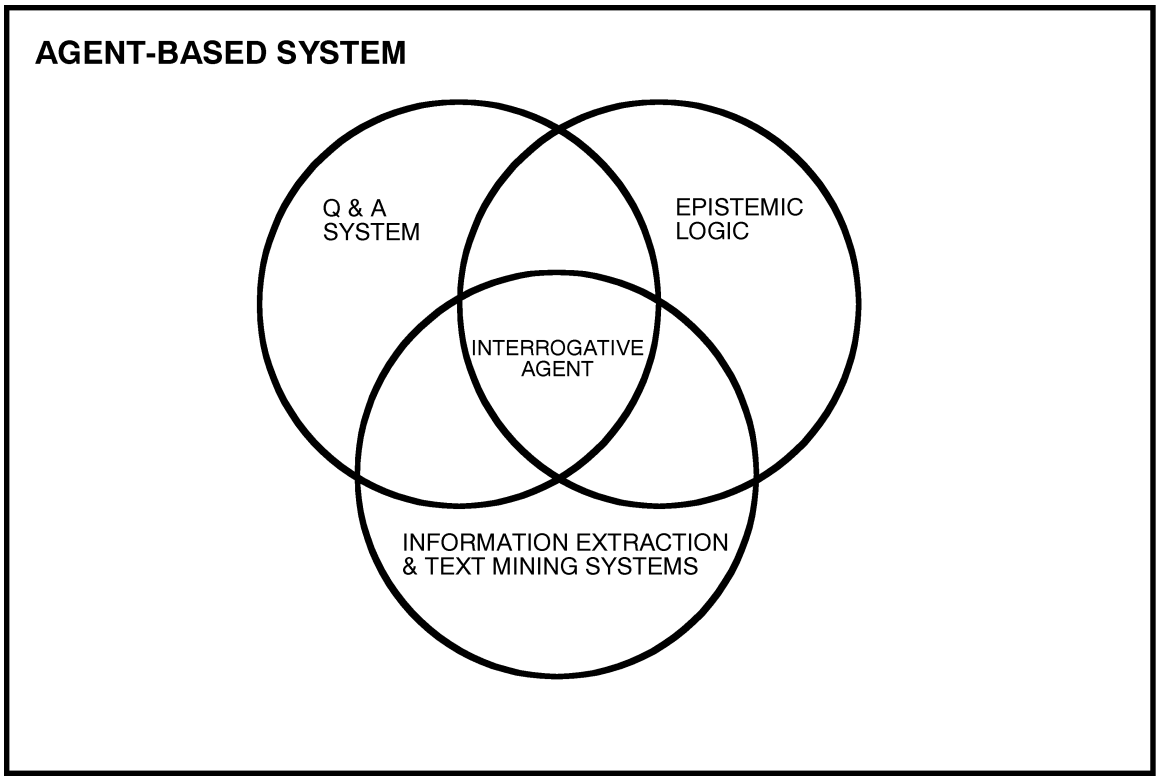

Figure 2. An Interrogative Agent is intersection of Q&A systems, epistemic logic, information extraction systems and agents.

2.1 Semantics Of Agency For Interrogative Agents

We now turn to the semantics of the the agency between our interrogative agent, the user, and our application of possible worlds. The interrogative agent interrogates a document on behalf of the user. From the agent's perspective the user's document or collection of

documents constitute the possible worlds that the agent will consider when responding to the user's query. That is, the user's document serves as a closed world of assumption for the agent. When the user makes a query the agent uses inference to search the document on behalf of the user. Because the implied propositions of the original document are not readily visible to the typical browser and keyword search engine, the interrogative agent's responses provide the user with a deeper level of analysis than a typical browser can. To get some idea of how agency is applied to the possible worlds that are contained within the original document we show a simplified interrogative agent structure and flow in Figure 3**.**

The structure for the interrogative agent in Figure 3. is adapted from Russell and Norvig's architecture for model-based reflex agents [20]. We start with the Russell and Norvig model and we replace their world states with the state of our agents knowledge and belief. In Figure 3 the questions serve as input or sensors for the agent and the agent's responses serve as output or actuators. The questions and answers put the agent in a feed back loop. The agent is situated in the interrogative domain the user's original document or collection of documents. Notice in Figure 3 that the user is modeled by *1..N* incomplete frames[21]. That is the user has some goal in mind prior performing a query

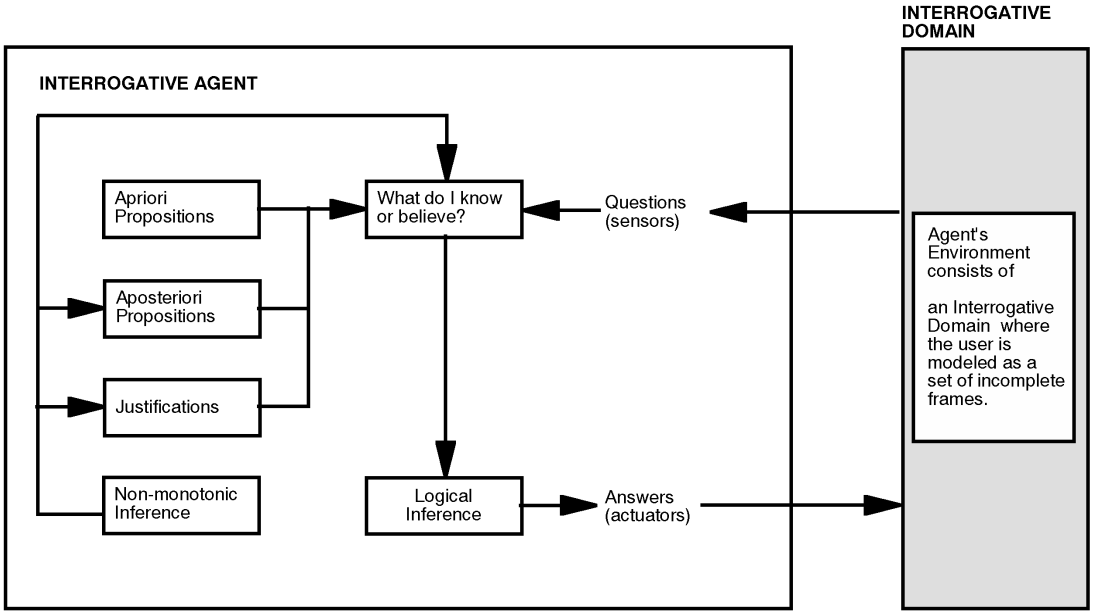

Figure 3. A Simplified Structure of An Interrogative Agent

against the corpus. That goal is part of some frame of reference for the user see [21] . The purpose of the query is to fill in or to complete some incomplete or partial frame. The user poses a query. A satisfactory response from the interrogative agent will serve to complete or partially complete the user's frame. This may cause the user to either invoke another frame[21] or to pose further queries within the context of the current frame. The interaction shown in Figure 3 illustrates that the the interrogative agent affects its environment by completing or partially completing frames and thereby guiding or informing the user's next question. In particular the environment of incomplete

frames are modified directly by application of the agents knowledge to the user's query.

2.2 But What Does The Interrogative Agent Know?

Since the an interrogative domain consists of a transcript consisting of questions and their corresponding answers our interrogative agent possesses two basic classifications of propositional knowledge, one explicit and the other implicit. The agent knows:

- The Questions and Answers that are in the transcript
- The Propositions that are entailed by combining the question and the answer using the rules of language and any propositions that can be deduced from those propositions.

In the first case the agent is able to simply report what questions and answers were found in the transcript. In the second case the agent is able to report what propositions follow from the questions and answers by applying the rules of language. For instance:

Question: Where was Essam Al Ridi last seen?

Answer: at the corner of Fifth and Olney.

Entailed Proposition: Essam Al Ridi was last seen at the corner of Fifth and Olney

It should be noted that the entailed proposition is not easily available to a keyword search. Further, once the interrogative agent has deduced the entailed proposition the agent could respond to a query such as:

User: Was Essam Al Ridi ever on Fifth and Olney?

Agent: Yes.

Because our interrogative agent's knowledge is based on its epistemic DNA (Deductive Nuclear Architecture) , the agent can only provide responses that are the result of the rules of language, inferential implication and material implication. To bring the notion of an epistemic DNA into focus we will add a little notation to what has been previously given. In particular we let φ and ψ be propositions or formulas that can be deduced from S where $S \equiv E_s$. Also let T be an electronic transcript where $T \in E_s$. If φ is valid then agent A knows φ. That is, A has knowledge of valid formulas. If agent A knows φ and $\varphi \vdash \psi$ then agent A knows ψ . This suggests that the agent's knowledge is closed under logical implication. Further if agent A knows φ and $\varphi \leftrightarrow \psi$ then agent A knows ψ . But characterizing the knowledge of the agent in this way leaves us with the problem of *logical omniscience.* That is our agent is purported to know every proposition that is a logical consequence of a question and answer pair from T and every proposition that is a logically implied from those propositions! This clearly presents a problem for a resource bounded agent. We will show how our epistemic structure handles logical omniscience later. Now, we further characterize the knowledge of our interrogative agent as:

(5)

- $\models \varphi \rightarrow \psi \Rightarrow \models K\varphi \rightarrow K\psi$ (Closed under valid implication)
- $(K\varphi \wedge K\psi) \rightarrow K(\varphi \wedge \psi)$ (Closed under conjunction)
- $K\varphi \rightarrow \neg K \neg \varphi$ (Knowledge is Consistent)
- $K(K\varphi \rightarrow \varphi)$ (Agent believes nothing false)
- $K\varphi \rightarrow K(\varphi \quad \psi)$ (Weakning of Knowledge)

from S5 and KD45 epistemic formulas see [10,9,11] . To this characterization of our agent's knowledge we also add the caveat that the interrogative agent operates under closed world of assumption.

2.3 Closed World Assumption

The worlds that the agent consider possible come exclusively from the questions and answers that are explicitly statement in the transcript and from the propositions that are entailed from combining the question and corresponding answer. Our agent considers anything that it does not know or consider possible false. This is part of the notion of closed world assumption [22]. In our case the the electronic transcript or collection of transcripts is a closed world and represents the complete universe that the agent has access to. Although this may seem like a limitation for our interrogative agent, it is exactly the condition for knowledge that we are attempting to capture. The browser keyword search and the keyword or phrase search capabilities of most search engines are restricted to the content of the documents that are being searched. Our interrogative agent is no more restricted than a browser's keyword search or that of the typical search engine. Our interrogative agent deploys a form of Information Extraction[23] that is restricted to deduction, inferential implication, and material implication. This restriction is part of the contract that the agent has with the user. The user can rely on the fact that any response that the agent gives is something that can be deduced from the current transcript. That is, there is no “expert” metaphor associated with our interrogative agent. The agent is capable returning responses that are a simple

matter of deductive and material implication. This makes our interrogative agent weak because it cannot induce, or abduce anything new from *T*. Further agent *A's* knowledge is deductive weak [3]. If we let $\Box\varphi$ denote the case where φ is contingently true or possibly true, the weak deductively principal leaves us with:

(6) **if** $KA\,\Box\varphi$ and $\varphi \vdash \psi$ then $KA\,\Box\psi$.

In essence the agent is only able to report what is either explicitly or implicitly (and currently) present in the transcript , with the caveat that given that the content of the transcript is true the entailed propositions are true. The advantage that our interrogative agent has over typical keyword searches is that it uses the epistemic structure of the transcript to make implicit but (*logically entailed)* propositions visible to the user. Therefore in this case the *semantics of agency* is a function of deductive inference on behalf of the user against an electronic transcript. The epistemic structure E_s of the transcript is tantamount to what the interrogative agent will consider possible worlds. Using the Kripke structure *M* to capture the relationship between the transcript and the Agent, and understanding what our agent can and can't know, we are now able to characterize the epistemology of our interrogative agent from a closer inspection of the structure E_s.

3. **A Closer Look at** the Epistemic Structure $\mathbf{E_S}$

The structure E_s is a knowledge structure. That is, its purpose is to hold or store the

knowledge of an epistemic agent; in our case an interrogative agent. The propositional knowledge of the interrogative agent consists of the worlds that it considers possible. The worlds that it considers possible are taken from *1..N* electronic Transcripts (T) from some interrogative domain (d) .

For example, suppose we take a transcript from one of the thousands of U.S. congressional hearings that have been archived in electronic form, in particular a transcript that has been stored as semi-structured text. This transcript will contain the questions and answers that were given during the course of the hearing. Our interrogative agent's propositional knowledge consists precisely of those questions, their corresponding answers, and the propositions that are entailed by combining the questions with their corresponding answers. We can now describe the knowledge transfer function θ.

Let t = {set of questions and corresponding answers from the transcript}

Let ρ = {set of propositions, questions, and answers that the agent knows}

then we have:

(7) $\theta: t \rightarrow \rho$

Where θ is an injective mapping from t to ρ and $|\rho| > |t|$. The transfer function θ serves to populate the agent's knowledge. The epistemology of our interrogative agent and the semantics of its agency with respect to the user can now be stated in terms of E_s

Let $\mathbf{E}_S$ be the structure:

(8) $\mathbf{E}_S = < G_1, G_2, J_S, V_c, F>$

G_1 is a graph (V,E). V is a non empty set of apriori propositions, models, and question and answer pairs. E is a non empty set of relations between the elements of V.

G_2 is a graph(V,E) . V is a set of posteriori propositions and E is a set of relations between the elements of V.

G_1 and G_2 contain the statements, models , propositions that the agent considers possible. In addition to these, the questions that were asked and the answers that were given are also stored in G_1

3.1 a priori and a posteriori Propositions

The propositions that are stored in G_1 and G_2 are derived from the rules of language, deductive implication , and are analytic in nature. The a priori and posteriori designations are an important part of controversies in most discussion on Epistemology[24,4,1]. Table 1 shows our usage of analytic propositions that are a priori and a posteriori versus synthetic propositions

Table 1. Analytic vs Synthetic Propositions

Proposition Classification	***A priori***	***A posteriori***
Analytic	S believes P not because of experience but because P is true by definition and its truth can be discovered by reason alone.	S believes P because S has discovered the truth of P through experience although P is true by definition and experience was not necessary.
Synthetic	S believes P not because of experience but because P is has been proven to be true by the experientially by others.	S believes p because S has discovered the truth of P through experience, and P is not true by definition.

It is important to note that the nature of the propositions contained in G_1 and G_2 are the reason that our Interrogative agent uses weak deduction. Recall:

(6) **if** $KA\,\Box\varphi$ and $\varphi \vdash \psi$ then $KA\ \Box\,\psi$

describes the a scenario where our agent's propositions are only possibly or contingently true. It is here where we hope to redeem the simple Kripke based agent by adding the color of epistemic justification to the agent's knowledge. Consider the Kripke statement of what an agent knows from (2):

$$(M,s) \models K_A \psi \leftrightarrow (M,t) \models \psi \text{ for all t such that } (s,t) \in K_A$$

The relation *(s,t)* is captured by propositions that are related and contingently true in our interrogative domains. This means that the possible worlds *(s,t)* that an

interrogative agent will consider are derived from collections of related and contingently true propositions that deductively follow from the transcript. The nature of the interrogative domains under consideration such as transcripts from court hearings, congressional hearings, etc. are full of propositions that are made which are not necessarily true. The interrogative agent will believe (accept as true) any proposition so long as the agent has no reason to reject it. In other words *A* will accept ψ as true long as the agent has justification for ψ .

To bring this notion in to focus, lets break down the Kripke structure *M* a little further. Suppose S is a set of worlds and *T* is our transcript and S $\in$ *T*. Suppose that **R** is a relation on S such **R** $\subseteq$ W= (S $\times$ S), then **R** captures the relationships between propositions ρ that are consequences of the questions and answers in *T*. So we have:

(9) T $\vdash \rho$, T $\models \Box \rho$

We can more clearly state what the interrogative agent considers possible. If (s,t) $\in$ W then (s,t) $\in$ R *iff* the agent does not have any reason to reject *s* or *t*. Further *s* and *t* are qualified as T $\vdash$ s , T $\models \Box$ s, T $\vdash$ t, T $\models \Box$ t

Therefore the notion of belief for an interrogative agent is represented by what the agent accepts as true[12] and according to our logical tripartite analysis the agent cannot know something that he doesn't believe. The addition of belief and justification to Kripke structure provides a more epistemic foundation for the notion of

'agent knowledge' under the Kripke model.

3.2 Justification in E_S

J_S, is the set of justifications and challenges for those propositions contained in G_1 and G_2 .For our interrogative agent we use epistemic justification based on the source of the knowledge[6]. Figure 4 situates our justification in the context of the taxonomy of epistemic justifications. In this case the only source of our agent's propositional knowledge is the transcript taken from the interrogative domain. Our epistemic justification is of the deductive variety[2] see Table 2 presents two classifications for epistemic justification. We claim a deductive epistemic justification based on the source of our agent's knowledge. However we acknowledge that a transcript from an interrogative domain is built around contingent truth.

Table 2. Classification of Epistemic Justification

Epistemic Justification	***Description***
Deductive	When a proposition logically entails what it justifies. When p as a consequence must be true if the antecedents $p_1 ... p_n$ that it is inferred from are true. When a proposition has both inferential implication and material implication. When a proposition is based on logical truths or mathematical truths.
Inductive	Justification based on experience. Empirically derived propositions. Arguments from authority, causal induction, statistical syllogism, hypothetical induction, causal elimination arguments from analogy and enumerative induction.

Any propositions that the agent considers are deduced as opposed to induced. If the transcript is reliable and true then any consequences that follow from the transcript are reliable and true. That is:

(10) $T \vdash \rho$, $T \models \rho$

Note in Figure 4 that our epistemic justification follows the negative branch. This means that we can state our Agent's justification as follows:

A's belief that φ is justified a *priori* if and only if *A's* justification for the belief that φ does not depend on experience.

Only propositions that have *epistemic justification* are considered as inclusion into the worlds the agent considers possible see [24] .

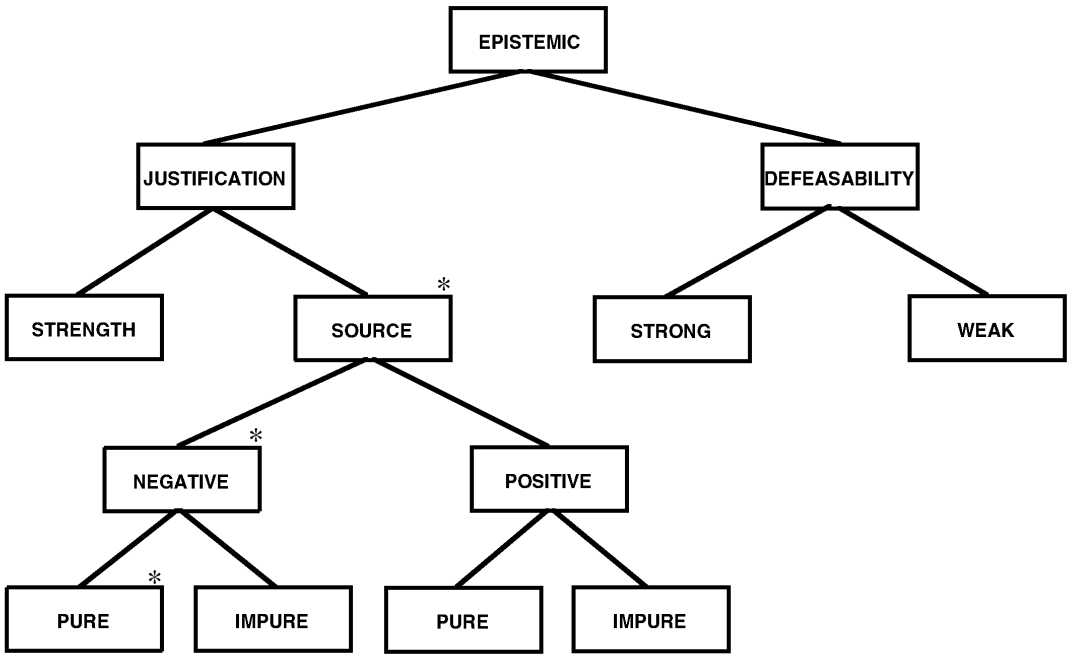

Figure 4. Epistemic Justification Taxonomy

Each proposition from G_1 is initially considered unchallenged by the agent and the agent accepts it as contingently true until a proposition that has been deduced challenges it. Once a proposition has been challenged, both the original proposition and the challenger are considered tainted. For each of those propositions the agent will acknowledge that they are a consequence from the original question and answer set, but the agent does not have any commitment to them. This means that if either *s* or *t* is tainted then (s,t) will not be in the worlds the agent considers possible and the agent will not have any commitment to them. However the user can still retrieve them with certain

caveats in place. That is the agent can return to the user those propositions that it considers tainted. If (s,t) is not tainted then $t \in J_S$ as a justification. If(s,t) is tainted then *t* and *s* are marked as tainted and are stored as elements of J_S

3.3 Modeling Belief in $\mathbf{E_s}$

An interrogative agent *A's* beliefs are measured by what *A* accepts as true. A accepts φ as true as long as A is not aware of any ψ that defeats φ. G_1 and G_2 contain the a *priori* and a *posteriori* propositions that *A* is aware of. V_c in $\mathbf{E}_S$ represents the commitment of the agent for each $\varphi \in G_1 \cup G_2$. if φ has not been challenged then V_c will contain a 2 for φ otherwise V_c will contain a 1 for φ.

3.4 Non monotonic Justifications

F is a non monotonic function over $\mathbf{E}_S$ that updates V_c for $\forall\varphi \in \{T \vdash \rho , T \nvdash \Box \rho\}$ [20]. The agent's commitment to a proposition is a simple two valued logic. If the proposition has not been challenged then the agent is committed to it. If the proposition has been challenged then the agent will only acknowledge that the proposition is an implicit or explicit consequence of the transcript . This suggests that an agent can move from belief to non belief for any proposition under consideration. However the reverse is not the case. Once a proposition is challenged it will remain challenged for that transcript.

3.5 Knowledge Space K_s of an Interrogative Agent

An interrogative agent's knowledge may consist of multiple domains. Each domain is represented by an epistemic structure. Where $\mathbf{E_S}$ is our epistemic structure.

Let $d = \{ E_{s1}, E_{s2}, E_{s3} \ldots E_{sn}\}$

Where *d* is a set of epistemic structures representing a particular domain, or a collection of domains

Then we have:

$$K_S = \bigcup_{i=1}^{N} d$$

where K_s is the total knowledge space of the agent and K_s is a set union of the domains.

Let: $\mathbf{F} = \{ m \mid K_s \models m\}$

Where **F** is the set of all propositions/models that are entailed from K_s

and where m are implemented as frames F_R

Let : $\Delta = (a, F_R)$, $a \varepsilon \{ a \mid a$ is an attribute of F_R in $F\}$ where F_R is some Frame(s) in F

and Δ is a relation on a, F_R and a is an attribute.

We have:

oav(a,V, Δ) is true *iff* there is a relation (a, F_R) where a is an attribute of F_R and V is the value of a.

The model (Q_s, P,w) is true *iff* P is the set of {attribute,value pairs} |= Q_s and w is the set of terms in Q_s

4 Propositional Knowledge in Interrogative Domains

Public Government Hearings, Criminal and Civil Trials, Interrogations, Personality tests are examples of what we call interrogative domains. We call these interrogative domains because the primary content of these domains are questions and answers. In each of these domains there is either an implicit or explicit search for the truth. At the very minimum there is a search for information and in the best case scenario knowledge is gained. In some cases the parties involved in the question and answer process are motivated to withhold information or knowledge. In some cases parties involved in the process are simply unaware of important information, or knowledge that they may have. Sometimes the goal in an interrogative domain is to substantiate facts, information, or knowledge that is available. In other cases the goal is to discover information or knowledge that not readily available.

One of the primary artifacts of a public hearing, criminal or civil trial,an interrogation, or personality test is the transcript. The transcript is a record of everything that was said. It captures all of the questions and all of the answers or in some cases (non answers). Many transcripts of public hearings, trials, interrogations, etc. are stored in electronic format for future reference. This format can range from entries in a unstructured text to semi-structured text such as Hypertext Markup Language(HTML) Extended Markup Language(XML) to highly structured database entries. For example there are HTML based standards and new XML based standards for digital transcripts generated by court recorders.

5 The *NOFAQS* Project

HTML based standards and the new XML based standards for digital transcripts generated by court recorders offer more search and analysis options than the traditional CAT (Computer Aided Transcription) technology. The LegalXml standards are promising opportunities for new methods of search for legal documents. The HTML and LegalXML standards allow judges, lawyers and other interested parties to analyze digital transcripts with additional and increasingly sophisticated search techniques. However, the search techniques employed are still largely restricted to keyword search of the digital transcripts and various probabilistic association techniques. Rather than keyword and association searches, we are interested in semantic and inference-based search. In this paper, a process for transforming the semi-structured XML/HTML version of the digital transcript to an epistemic structured representation suitable for semantic and inference-based analysis is explored. This representation allows us to search for implicit

knowledge. Implicit knowledge has higher visibility in this epistemic structure than it does in its semi-structured XML/HTML version. The epistemic structures presented in this paper are knowledge representation schemes used for constructing a model of the transcript domain.

We are interested in the idea of viewing the digital transcript as a knowledge space[25]. Where the arguments of the attorneys, the testimony of witnesses, and statements of defendants are converted from their semi-structured representation to epistemic structures that collectively form the knowledge space of the trial. Once the transcript is converted, queries can be posed that can be answered using semantic processing techniques[26] rather than probabilistic or keyword search techniques. The answers are ultimately derived from knowledge that is explicitly or implicitly given during the course of a trial. In particular, the questions and answers that occur during the examinations and cross-examinations by the attorneys of the witnesses and defendant(s), as well as valid opening arguments and certain classes of objections are all used as micro-sources for contingent truth. Here we use the notion of contingent truth as it is used in modal logic introduced by Lewis[8] and some of its variations formalized by Kripke[7] and Hintikka[27]. We also take advantage of Hintikka's formalization for statements that are simply believed to be true[4]. Many statements made during the course of a trial are only possibly true and therefore answers to queries have to be qualified. Reference [7] and [27] give formalization's for this kind of qualification. In the HTML/XML version of a legal transcript our candidates for contingent truth (e.g. questions, answers, examinations, cross examinations, etc.) are tagged. For instance, question and answer pairs are

conveniently coded as:

Q. Was Oren in fact a medical doctor?

A. Yes, he's a surgeon, doctor.

The **Q.** and **A.** are tags placed in the HTML that conveniently identify the start and stop of a question and answer pair. Together the question and the answer can imply one or more propositions. In this case the proposition:

Oren is a surgeon.

can be inferred. The truth of this proposition is subject to the credibility, integrity and possibly the belief of the person answering the question or in the trial. This is an example of what we mean by as *contingent truth*. Each island of contingent truth in the transcript is considered for acceptance as a node in a concept graph[28]. If it is accepted then it becomes a weighted part of the transcript's knowledge space which makes it more visible to a user's query. The proposition:

Oren is a surgeon.

is only available by making an inference from the combination of the question and answer pair and is therefore not visible to keyword queries of the semi-structured XML/HTML version of the transcript.

Reference[28] initially provided the most detailed treatment of conceptual graphs. We use this treatment of conceptual graph to facilitate our inference-based search to locate knowledge that can only be inferred through chains of implication. For instance the answer to the question:

Was the testimony of Jones impeached?

Is not explicitly given in the text of the transcript and can not be found by keyword or associative search techniques. Instead trial information is searched to determine whether Jones was a witness, or defendant. Then the appropriate examination, cross-examination information is searched to see what statements Jones has made. Then the statements of others in relation to the statements that Jones made are considered. The exhibits that relate to the statements that Jones has made are examined. Finally, the statements that Jones has made in relation to other statements that he has made are also considered. At that point the question of whether Jones testimony was impeached or not is answered. The search is done by traversing the nodes in the conceptual graphs.

The combination Q&A into inferred propositions and the building of a conceptual graph model for the transcript are parts of our five step process that transforms the semi-structure text into a epistemic structured representation. The five steps in the process are:

Step 1: Using the HTML/XML and other structured

rules of the transcript, convert the entire
transcript into a tagged corpus.

Step 2: Create a model theoretic semantic of
the corpus[29], where the models
are captured both as predicates and frames[21].

Step 3: Convert the Q&A pairs to propositions.

Step 4: Construct frames from the propositions generated in Step 2.

Step 5: Using the structures created in Steps 2, 3, 4
instantiate the structure that represents the
knowledge space of the transcript/trial.

While this is somewhat of a simplification, these five steps capture the basic idea behind the transformation of the original transcript. Predicate Calculus and First Order Logic (FOL) are used as the primary representations for our frames and propositions in Steps 2 through 5.

The two primary challenges discussed in this paper are:

- Exploiting the Q&A, opening and closing arguments, objections and other tagged patterns of legal transcripts to take advantage of explicit and implicit knowledge

- Deriving the epistemic structures for the knowledge space of the transcript

5.1 Background

Our transcript conversion process is part of the NOFAQS system. NOFAQS is an experimental system currently under development at Ctest Laboratories. NOFAQS is designed to answer questions and draw inferences from interrogative domains (e.g. interviews, congressional hearings, trials, surveys/polls, and interrogations). NOFAQS uses rational agents and Natural Language Processing (NLP) to aid users during the process of deep analysis of any corpus derived from any context that is question and answer intensive. In this paper we discuss how the system was used on a set of court room transcripts of a famous court case that lasted 76 days.

5.2 The Corpus

To realize our theoretical model $\mathbf{E}_S$ we have selected as experiment a corpus from the interrogative domain of court room transcripts.

The corpus was obtained from the Internet as a collection of Hypertext Mark up Language (HTML) files. It consisted of:

- 76 File formatted in standard HTML

  (one for each day of the trial)
- 25,979 Questions

- 25,930 Answers
- 461,938 Lines of domain relevant text
- 1,854,242 Domain relevant words

In this case the corpus is a collection of digital transcripts generated by a scopist and the process of Computer-Aided Transcription (CAT). A scopist edits a transcript translated by CAT software into the targeted language, correcting any mistakes and putting it into the appropriate standard format. The CAT system is computer equipment and software that translates stenographers notes into the targeted language, provides an editing system which allows translated text to be put into a final transcript form, and prints the transcript into the required format.

5.3 Model Theoretic Semantic of Transcript

Step 2 of our five step process generated a model theoretic semantic representation of the transcript. Here the model is described as: **M = (D,F)** where **M** is a model theoretic semantic representation of all the language that is contained in the trial corpus. **M** consists of a pair **(D,F)** where **D** is the *Domain* which is the set of people and things referenced in the corpus (e.g. defendants, jurors, attorneys, witnesses) plus the relations (e.g., lawyer(X,Y), trial_day(N), etc.) between those people and things. **F** is an *Interpretation Function* which maps everything in the language onto something in the domain [30]. **F** is implemented using the Lambda Calculus operator and - conversion. The initial model theoretic representation serves as part of the background knowledge for the Inductive Logic Programming (ILP) learning process. The results of

our ILP learning process augments the initial **M**.

5.4 Important Target Predicates

Inductive Logic Programming (ILP) techniques were used to help with the process of classifying question types[31] and in determining whether a given Q&A pair constituted a legitimate proposition. ILP is a form of programming that combines machine learning and logic programming[32]. We used ILP to learn two of our important target predicates:

question_classification(Q, Class)

ques_ans_classification(QA,Resolved)

The question_classification(Q,Class) predicate represents a learned program that when given a question Q returns Class, the classification of Q. The ques_ans_classification predicate is a learned program that when given a question and answer pair QA returns whether or not the question was actually answered. If the question was resolved then the question and answer pair can be considered as a candidate for contingent knowledge.

The first target predicate used simple question classification[13]. Table 3 contains the question classifications used.

Table 3. Simple Question Classifications

Class	Interrogative Indicator	Example

Person	who	You tell the jury who you raised money from and what the money was for?
Information	how, what	And how many helicopters did the group shoot at? What other weapons did you receive training in?
Explanation	why	Tell the jury why it is you chose to leave New York to go to Peshawar in Pakistan?
Location	where	Where did you get the $250,000 to buy the farm?
Temporal	when	When did those four people go back to the Sudan?
Yes/No	can, do, did, is, were, are, was, will, would, does, could, have	Can you tell the jury what happened?

The target predicates:

question_classification(Q, Class)

ques_ans_classification(QA,Resolved)

are important to our process because they are used to build the a *posteriori* knowledge of the transcript (*Step 3*).

We also used ILP to learn viability of a question and answer pair as a candidate for contingent knowledge. In particular we use a variation of Shapiro's model inference system from Reference [32] and the basic FOIL algorithm to learn our ques_ans_classification (QA,Resolved) target. The two target predicates and basic NLP parsing was used to construct propositions in *Step 3* of our five step process.

In the NOFAQS system each rational agent A_i uses K_s as a search space in the resolution of a query posed by a user[33]. Each query is presented to A_i as either a complete interrogative sentence or a FOL query. A_i is a function implemented as:

5.5 An Interrogative Agent

function: SearchAgent returns response class

begin

FOL_v = parse(InterrogativeSentence)

FrameNode = map(FOL_V)

Response = search(K_s,FrameNode)

return Response

Here, FOL_v is a predicate calculus representation of the user's query and FrameNode is a partial frame that captures the attributes of the user's query. The search method of A_i selects a graph traversal search based on the type of FrameNode returned by the map method. This graph search is then applied to K_s Where K_s is given by:

$$K_s = \bigcup_{i=1}^{N} d$$

This results in a search by A_i of the concept nodes in G_1 and G_2. The response is then given a weight and certainty based on how G_1 and G_2 are mapped into C. So for a typical search in K_s we have:

Response = norm (A_i(S))

Where S is the user's interrogative sentence. When we decompose E_S for our legal transcript representation we have:

G_1 = {V,E} where **V** is a set of nodes that contain propositions and frames representing the non-testimony elements in the transcript (e.g. distribution of attorneys, opening & closing arguments, identity of judge, etc.) **E** is the set of relationships on set **V**.

G_2 = {V,E} where **V** is a set of nodes that contain propositions and frames representing the testimony elements of the transcript (e.g. questions, answers, direct examinations, objections). **E** is the set of relations on set **V.**

J = {X | X is non challenged proposition in G_1 that
provides weight to propositions from G_2}

C = V[G(m,n)] where V is a vector that represents the
agents level of commitment to propositions or
concept nodes found in G_1 or G_2

$F(E_S)$ is non monotonic function that manages the belief and justification of the agent as propositions are added to G_1 , G_2 or F

5.6 The Method

The 76 HTML files were considered as raw data that would require data cleaning, pruning, and noise removal. Further, the files consisted of semi-structured text that is not conducive to inferential analysis. In the second stage of processing, the semi-structured text required that it be transformed into its Model Theoretic Semantic (MTS) representation. This transformation produced 23 classes and 40 basic relations between

objects in the 23 classes. Table 2 contains samples of the 23 classes and 40 relations. So for **M**, where **M = (D,F)**, we have concepts such as *exhibits, cross examinations, witnesses* in **D** as well as relations such as *heard, knew,* and *observed*. The MTS representation was used as the baseline for the a posteriori knowledge. The a posteriori knowledge was taken primarily from the approximately 29,000 question and answer pairs that were given during the course of the trial. Its this a posteriori knowledge and its representation as FOL[34] that allows deep analysis against the trial corpus. While the question and answer pairs provided a legitimate source of contingent knowledge, we could not use them until they were classified[35].

Table 4. Classes and Sample Relations

Classes		**Relations**	
Attorneys	Trial	answered	asked
Defendants	Side-bar	did	do
Witnesses	Objections	said	tell
Arguments	Dates	told	examined
Testimony	Exhibits	objected	aware
Court	Examinations	recalled	recognize
Questions	Answers	saw	meet
Recesses	Adjournments	traveled	know
Counts (charges)	Jurors	knew	show
The record	Evidence	heard	hear

5.7 Frames for G_1 and G_2

The frames are schematic models of domain elements in the trial. For instance, the defense attorney frame contains slots and facets such as:

frame: DefenseAttorney

name:

is-a: Attorney

role: (lead or support)

default: lead

objections_sustained:

facet: if_needed execute (get_sustained)

objections_overruled:

facet: if_needed execute(get_overruled)

client:

facet: if_needed execute(get_client)

questions_on_direct: *Type-of- List*

end DefenseAttorney frame

There is at least one frame for each main domain element in the trial. Each frame consists of one or more slots. A slot may have one or more facets. The slots represent attributes of the frame. For instance, an attribute of an attorney is what type (either defense or prosecution). A facet represents some kind of special constraint or trigger, or processing for an attribute. For example, the fact that there was an objection during a cross examination may or may not be needed. The facet specifies what to do if it is needed. The slots for the frames for G_1 and G_2 typically require the values to be filled in from a priori and a posterior knowledge. This means prior to a user query, the frames are partially filled in. Once a query has been posed, the inference process fills in whatever slots are needed to answer the user's query.

5.8 Q&A Pair Contingent Propositions

The responses to the questions had to be classified as either answers or not answers before we could determine whether a Q&A pair was a candidate for contingent truth. The transcripts had many instances of witnesses and defendants that eluded answering the questions by such responses as: “I don't know”, “I don't remember”, “I can't recall”, etc. Other responses were unrelated to the questions. Before a Q&A pair could be considered as a modal proposition the negative responses had to be classified and filtered.

Three simple classifications of responses were used for our Q&A pairs:

- answered,
- not_answered
- answered_not_certain

The computer language Prolog was used for the hypothesis language and to represent the background knowledge in our ILP programming. Our target predicate:

ques_ans_classification(QA,Resolved) was implemented using the clause:

tuple([Q], [A], (X,Y))

where Q is list of words representing the question, A is the list of words representing the response and (X,Y) is the class pair representing the classification of the Q&A. X is the class of Q and Y is the class of A. tuple() is then implemented by hypothesis and answer predicates:

hypothesis(question([FrameGrammar]),

answer([FrameGrammar]), class(Q,A))

where question([FrameGrammar]) is a learned generalization of the Q**,** answer([FrameGrammar]) is a learned generalization of A, and class(Q,A) is a pair representing the class of **Q** and A. Each of the generalization of simple frame grammar representations are commonly used in natural language processing. For example:

Question:

Do you know what time of year it was Mr. X was arrested?

Answer:

I know it was during Ramadan, but I can't remember the day.

Here is the Prolog representation of the training set tuple and the learned target predicate:

```
tuple([do,you,know,what,time,of year...],
    [i,know,it,was,during,ramadan,but...],
    (yesno,answered_not_certain))
hypothesis(question([Av,Pn,V,Dpn,....]),
answer([PPn,V,C,Av,...,i,cant,remember]),
     (yesno,answered_not_certain))
```

The background predicates (knowledge) consisted of a lexicon of the parts of speech in English in horn clausal form, and a simple Phrase Structure Rule (PSR) grammar. For instance, the lexicon clauses took the form:

adjective(X).

interrogative_pronoun(X)

transitive_verb(X)

...

5.9 Discussion

The epistemic representation of the transcript provides the user with the ability to pose queries about the content of the transcript that can only be inferred. For instance we are able to ask questions such as:

- *Was John Smith's testimony impeached?*
- *Which witnesses were evasive in their answers?*
- *Were the opening and closing arguments supported by witness testimony?*
- *Did the prosecuting attorney lead the witness X on cross examination*

The answers to these types of questions evade keyword and associative search techniques. It would also be difficult to convert the semi-structured transcript into database form and expect answers to these kinds of questions. On the other hand graph search and traversal techniques are well suited for data that can be represented in predicate calculus or as Frames. This motivates the reasoning behind our epistemic representation approach to legal transcript analysis. The fact that the HTML/XML standards for legal transcripts provide tagged elements that make the question and answer analysis process feasible, differentiates Legal transcripts from other types of HTML/XML semi-structured documents. The tagging found in the original transcript

facilitates the automation of the conversion process in Step 1. As the new LegalXML standards are refined, the tagging will improve the mapping process between the a *priori* knowledge in the transcripts and the frames used for the concept nodes.

Conclusion

Exposing the epistemic structure of a semi-structured transcript allows the agent to apply graph traversal and FOL search techniques to the knowledge space of the transcript. Since the agent is dealing directly with the knowledge space, the agent's responses will be semantically and pragmatically related to the original query. This means that the agent's responses can be logically derived from the knowledge space. While providing meaningful and relevant answers to questions is highly desirable, generating the epistemic structure of a semi-structured transcript is computationally expensive. The original HTML or XML transcript must be completely restructured. This is usually not practical for the general case. The size of the original HTML or XML transcript is increased by a factor of 4. While the resulting format can be expressed in standard FOL or horn clause form, these forms are not immediately available to the many HTML and XML browsers in use. While most of the conversion from semi-structured to epistemic structured is automated, the process still requires manual intervention. In addition to these issues the generation of the epistemic structure is not instantaneous. The generation requires both time and space. However for some vertical applications and narrow segments of users, the cost of generating the epistemic structure of a semi-structured transcript is offset by the ability to perform deep analysis and get meaningful, relevant and accurate responses to a query.

There is at least some temptation to compare the type of knowledge space search discussed in this paper with keyword and probabilistic search used in many of today's common search engines. That temptation should be resisted. The keyword search engines can successfully process single words, or tokens. Our knowledge space search techniques require either complete interrogative sentences or FOL queries. There is also a shift in responsibility for answer resolution in the two approaches. In the approach discussed in this paper the interrogative agent is responsible for deriving the answer to the user's interrogative sentence. Conversely, in keyword and associative type search techniques the user has most of the responsibility in extracting the answer from the results returned from the search. Also the two techniques serve very different audiences with different goals and objectives. Keyword-type approaches are primarily interested in text extraction or resource retrieval, where as the epistemic representation is aimed at text interpretation , inference and knowledge representation of the semantics in the source document.

We address the problem of logical omniscience by making the agent's knowledge explicit. G_1 and G_2 contain the a priori and a posteriori propositions that the agent has to work with. If the agent has not yet deduced it then it is not under consideration. We use the semantic approach to Kripke structure[9] where $M = (S,\sigma)$ is a syntactic structure, and for each state *s* the function $\sigma(s)$ tells us which formulas are true at state s. We also rely on Montague-Scott structures[9] to help us dodge the problem of logical ominiscience. However, our epistemic structure is at this point a victim of the frame

problem[11]. This is do to the nature of non monotonic processing of *F*. Once the agent infers a proposition that challenges a currently unchallenged proposition every proposition that is a logical consequence of that proposition must be dealt with. But how do we identify every propositon that is a logical consequence? We save that battle for another adventure.

Future Work

Our process for converting interrogative sentences into FOL queries and then mapping those queries to the frames of our domain needs much refinement. First, FOL is only capable of representing a subset of the possible interrogative sentence forms that the user might make against a transcript. We've used inductive logic programming in attempts to learn a heuristic that can be used to bridge the gap between the variety of query forms and our FOL representation. But this process falls short and needs work. Second the process that matches FOL to the Frames of our domain degrades rapidly when presented with valid non-determinism among Frame selection. To offset these issues we will explore the addition of semantic headers and discourse representation structures to our FOL representation of the original interrogative sentence. Further, we will investigate the use of ILP against the Q&A pairs to determine if there are structures that can be used to supplement our frame-based representation in the concept nodes. In addition to this work, we are interested in automating the the conversion process from semi-structured to epistemic structured as much as possible. Lastly, we are not satisfied completely with our Montague-Scott and explicit enumeration of propositions as a solution to logical ominiscience. We are also haunted by the frame problem, something

that most defintely will be addressed.

References:

[1] M. Schlick "General Theory of Knowledge" Springer-Verlag 1974.

[2] P. K. Moser, D. H. Mulder, J.D. Trout "The Theory or Knowledge A Thematic Introduction" Oxford University Press, 1998.

[3] N. Rescher " Epistemic Logic A Survey of the Logic of Knowledge" University of Pittsburg Press, 2005.

[4] E. Gettier, "Is Justified True Belief Knowledge?," Analysis, Vol.23, pp. 121-23. 1963.

[5] Lehrer, Keith and Thomas D. Paxson, Jr. "Knowledge: Undefeated Justified True Belief", The Journal of Philosophy, 66.8 , pp. 225-237 .1969.

[6] I. Kant "The Critque of Pure Reason" , trans Norman Kemp Smith. New York: St. Martin, 1965.

[7] Kripke, S., "Semantical Considerations on Modal Logic," Acta Philosophica Fennica, pp. 16, 83-94, 1963.

[8] Lewis, C.I., "A Survey of Symbolic Logic," Berkeley: University of California Press , 1918.

[9] R. Fagin, J. Y. Halpern, Y. Moses, M Y. Vardi, "Reasoning About Knowledge" Massachusetts Institue of Technology, 1996.

[10] J. J. Meyer and W . Van der Hoek "Epistemic Logic for AI and Computer Science" Cambridge University Press, 1995.

[11] J Hintikka "Reasoning About Knowledge in Philosophy The Paradigm of Epistemic Logic" NSF Grant #1ST- 8310936

[12] R. Swinburne "Epistemic Justification" Oxford Press, 2001.

[13] D. Moldovan, C. Clark, S. Harabagiu " COGEX: A Logic Prover for Question Answering" Proceeding of HLT-NAACL Main Papers, pp. 87-93 2003.

[14] C. Kwok, O. Etzioni, D Weld "Scaling Question Answering to the Web" WWW10 May 1-5, 2001 Hong Kong. ACM 1-58113-348-0/01/0005.

[15] M. Sun and J Chai “Towards Intelligent QA Interfaces: Discourse Processing for Context Questions” IUI'06 January 29-Feburary 1, 2006, Sydney Australia. ACM 1-59593-287-9/06/0001.

[16] J. Bradshaw “Software Agents” AAAI Press/ MIT Press, 1997.

[17] M Wooldridge , “Reasoning About Rational Agents” The MIT Press, 2000.

[18] K. Eisenhardt “Agency Theory: An Assessment and Review” Academy of Management Review 14(1) pp. 57-74, 1989.

[19] H.S. Nawana “Software Agents: An Overview” Knowledge Engineering Review, 11(3): pp. 205-244

[20] S. Russell and P Norvig “Artificial Intelligence A Modern Approach” Pearson Education, 2003.

[21] M. Minsky, "A framework for representing knowledge," *The psychology of computer vision*. New York: McGraw-Hill, pp. 211-277, 1975.

[22] R. Reiter “ On Closed World Databases” In Logic and Databases Plenum Press, 1978.

[23] D. Harman “Overview of the First Trec Converence” ACM-SIGIR'93 6/93/Pittsburgh, PA, USA ACM 0-89791-605-0/93/0006/0036, 1993.

[24] A. Casullo, “A Priori Justification” Oxford University Press 2003.

[25] J.P. Doignon and J.C. Falmagne, "Knowledge Spaces," Hiedelberg; Springer ISBN-3-540-64501-2, 1999.

[26] S. Nirenburg, V. Raskin, "Ontological Semantics," MIT Press, 2004.

[27] Hintikka, J., “Individuals, Possible Worlds and Epistemic Logic,” Nous, pp. 1, 33-62, 1967.

[28] J. F. Sowa, "Semantics of Conceptual Graphs," *Proceeding of the 17th Annual Meeting of the Association for Computation Linguistics*, pp. 39-44, 1979.

[29] S.M. Brasil, B.B. Garcia, "Modeling Legal Reasoning in a Mathematical Environment through Model Theoretic Semantics," *ICAIL '03*. Edinburgh, Scotland, UK, 2003.

[30] M. Covington, "Natural Language Processing For Prolog Programmers," Prentice Hall, 1994.

[31] J. Cussens and S. Pulman, "Incorporating Linguistics Constraints into Inductive Logic Programming," *Proceedings of CoNLL-2000 and LLL-2000*, pp. 184-193. Lisbon, Portugal, 2000.

[32] F. Bergadano and D. Gunetti, "Inductive Logic Programming From Machine Learning to Software Engineering," MIT Press, 1996.

[33] G. Chen, J. Choi and C. Jo, "The New Approach to BDI Agent-Based Modeling," *ACM Symposium on Applied Computing '04*. Nicosia, Cyprus, 2004.

[34] P. Blackburn and J. Bos, "Representation and Inference for Natural Language," CSLI Publications, 2005.

[35] R. Fagin, J. Halbern and M. Vard, " Model Theoretic Analysis of Knowledge," *Journal of Association for Computing Machinery Vol 38. No 2*, 1991.