A First Experimental Demonstration of Massive Knowledge Infusion*

Loizos Michael and **Leslie G. Valiant**
School of Engineering and Applied Sciences
Harvard University, Cambridge, MA 02138, U.S.A.
loizos@eecs.harvard.edu and valiant@seas.harvard.edu

Abstract

A central goal of Artificial Intelligence is to create systems that embody commonsense knowledge in a reliable enough form that it can be used for reasoning in novel situations. Knowledge Infusion is an approach to this problem in which the commonsense knowledge is acquired by learning. In this paper we report on experiments on a corpus of a half million sentences of natural language text that test whether commonsense knowledge can be usefully acquired through this approach.
We examine the task of predicting a deleted word from the remainder of a sentence for some 268 target words. As baseline we consider how well this task can be performed using learned rules based on the words within a fixed distance of the target word and their parts of speech. This captures an approach that has been previously demonstrated to be highly successful for a variety of natural language tasks. We then go on to learn from the corpus rules that embody commonsense knowledge, additional to the knowledge used in the baseline case. We show that chaining learned commonsense rules together leads to measurable improvements in prediction performance on our task as compared with the baseline. This is apparently the first experimental demonstration that commonsense knowledge can be learned from natural inputs on a massive scale reliably enough that chaining the learned rules is efficacious for reasoning.

Introduction

Knowledge Infusion is a particular approach to the problem of knowledge acquisition in intelligent systems (Valiant 2006). Its aim is to make systems possible that acquire knowledge on a large scale by learning, and then use it robustly for reasoning. The theory offers quantitative guarantees on the accuracy of the reasoning given certain assumptions about the learnability of the knowledge and the adequacy of the available data.

In this paper we describe an implementation of this approach for natural language data, and report on experiments that show that the approach provides quantifiable benefits. The experiments are performed on a natural language corpus of a half million sentences (Graff 1995). We measure performance on a natural language task of the following form: Given a word A and a sentence B from which the identity of one target word is hidden, determine whether the hidden target word is A. We are interested in the power of learned knowledge about the world, rather than natural language constructs themselves. For this reason we use information from an automatic parser, to identify, for example, the subject and object of a verb, and hence derive an instance where a certain relation holds in the world. This tagging information is then given with the sentence as part of the input during both the induction phase and the evaluation phase.

We have chosen performance on a natural language task as a vehicle for demonstrating the acquisition of real-world knowledge, both because such data is plentiful, but also because such tasks have been widely researched, so that there is some baseline for comparison. In general, surprisingly good quantitative performance has been found with simple statistical methods (e.g., n-gram methods (Manning and Schütze 1999)). Even better performances can be obtained using learning methods that look at the words (and possibly their parts of speech) within a fixed distance of the target word in the sentence or in the syntax tree (Even-Zohar and Roth 2000). We believe that the success of such syntax-oriented methods can be largely accounted for by the high volume of statistical evidence that can be made available to such algorithms, while keeping the number of features involved in the learning process moderate.

We seek to learn real-world knowledge from the natural language text. In particular, we wish to learn about all the concepts, or at least those that occur frequently enough in the corpus, in terms of all the others. In this setting the number of examples that corroborate the relationship between any pair of concepts may be few, and only limited statistical evidence for any one may be available even in an enormous corpus. Thus, the engineering problem to overcome is to design a system in which the performance gain that might derive from the extra knowledge gathered when compared with the more syntactic methods mentioned above, is not completely canceled by the noise introduced by the relative sparsity of the data that embodies this extra knowledge.

Knowledge Representation

Central to any approach for handling knowledge is the representation employed. For real-world knowledge, such as that encoded in natural language text, a relational representation is appropriate, naturally accommodating the representation

*This work was supported by the NSF grant CCF-04-27129.

of entities of interest and relations that hold amongst those. Beyond expressivity, the chosen representation should be one that is well suited for the efficient acquisition and manipulation of knowledge, and where these two processes can be carried out in a principled manner. We adopt in this work the formalism of Robust Logic (Valiant 2000), which has been explicitly designed with these goals in mind. We briefly review the main components of this formalism next.

We consider a fixed set $\mathcal{R}$ of *relations* R_i of arities $\alpha(i) \leq \alpha$, for some maximum arity α. A *scene* consists of n generic token objects $\mathcal{T} = \{t_1, \ldots, t_n\}$, and a vector that states for every relation $R_i \in \mathcal{R}$, and every one of its $n^{\alpha(i)}$ bindings over the tokens $\mathcal{T}$, whether R_i holds for that binding. The input examples to our learning and reasoning systems will be such scenes.

We define a *schema (of relations)* $\mathcal{Q}$ as a finite set of expressions composed of a finite conjunction of members of $\mathcal{R}$ each with some quantification. One such schema is

$$\{R_1(w), \quad \exists v : R_2(w, v), \quad \exists v : R_3(v) \wedge R_4(w, v), \\ \exists v_1 \exists v_2 : R_5(w) \wedge R_6(v_1) \wedge R_7(w, v_1, v_2)\}.$$

Expressions in a schema contain both quantified variables, denoted by v's, and free variables, denoted by w's. Expressions in a schema can be thought of as corresponding to composite features, derived from the primitive features defined by relations in $\mathcal{R}$.

Given the truth-values of all bindings of all relations in $\mathcal{R}$ as determined by a scene, the scene determines a truth-value for each binding of the free variables of each expression in a schema $\mathcal{Q}$. In this context, we view each member of the schema $\mathcal{Q}$ as an *independently quantified expression*, meaning that for the various expressions in $\mathcal{Q}$ and their binding of free variables, the quantification of the bound variables are independent. A propositional learning algorithm can then be applied on the determined boolean vector, and all its properties, such as sample bounds, attribute-efficiency and resilience to errors will be inherited in the relational domain.

The relational learning algorithm thus obtained induces *rules* of the form

$$body(w_1, \ldots, w_{\alpha(i)}) \equiv R_i(w_1, \ldots, w_{\alpha(i)}),$$

where $body$ is some (efficiently evaluatable) function over those expressions in a schema of relations that have the same set of free variables $w_1, \ldots, w_{\alpha(i)}$ as R_i, and all free variables are assumed to be universally quantified over the entire rule. In particular, the efficiently evaluatable function class of which $body$ is a member is that of linear threshold functions. For each binding of these free variables, the resulting binding of $body$ accurately (in the PAC sense (Valiant 1984)) predicts the value of the corresponding binding of R_i. This soundness guarantee allows the principled chaining of induced rules for reasoning.

Knowledge Acquisition and Manipulation

A relational rule

$$body(w_1, \ldots, w_{\alpha(i)}) \equiv R_i(w_1, \ldots, w_{\alpha(i)})$$

is interpreted under the PAC semantics of Robust Logic as stating that the rule is true *for every* binding of the free variables $w_1, \ldots, w_{\alpha(i)}$. This interpretation dictates how the rule is to be learned (i.e., how the body of the rule is to be determined), and how the rule is to be reasoned with (i.e., how predictions on the rule's head are to be made).

Constructing Learning Examples

When learning a rule for a relation R_i, one seeks to identify an expression, here a threshold function, over the expressions in the schema $\mathcal{Q}$ to act as the rule's body. The actual learning part is carried out by standard propositional learning algorithms (CCG-UIUC 2005; Littlestone 1988). We explain here how the relational scenes and rules can be appropriately manipulated by such propositional algorithms.

Every binding $R_i(t_{j_1}, \ldots, t_{j_{\alpha(i)}})$ of the relation R_i encountered to be *true* or *false* in a scene is taken to indicate that the scene provides, respectively, positive or negative evidence on what a rule predicting R_i should look like.

By way of illustration, assume that $\mathcal{R}$ contains the relations *human*, *gathering*, *hold* of arities 1, 1, and 2, respectively, and that $\mathcal{Q}$ contains, amongst others, the expressions *human*(w_1), $\exists v_1$: *hold*(v_1, w_2), and $\exists v_2$: *holds*$(w_3, v_2) \wedge$ *gathering*(v_2). Consider now the relational rule

$$(\exists v_2 : \textit{holds}(w_1, v_2) \wedge \textit{gathering}(v_2)) \equiv \textit{human}(w_1),$$

and a scene that determines that the bindings *human*(t_3), *gathering*(t_4), *gathering*(t_7), *holds*(t_3, t_7), *holds*(t_8, t_4) of the relations that appear in the rule are assigned the value *true*, while all remaining bindings of these relations are assigned the value *false*. For the binding that maps w_1 to t_3 and v_2 to t_7, the scene states that $\exists v_2$: *holds*$(w_1, v_2) \wedge$ *gathering*(v_2) is *true* and *human*(w_1) is *true* — this provides evidence in support of the relational rule. For the binding that maps w_1 to t_8 and v_2 to t_4, the scene states that $\exists v_2 :$ *holds*$(w_1, v_2) \wedge$ *gathering*(v_2) is *true* and *human*(w_1) is *false* — this provides evidence against the relational rule.

Such cases where scenes simultaneously provide evidence in support and against a relational rule are permissible, and in fact natural. They could arise, for instance, if the tokens t_3, t_4, t_7, t_8 correspond, respectively, to the entities Alice, press conference, barbecue party, government; whether this information is represented in the scene is inconsequential for our argument. The scene, then, implies that Alice (a human) is holding a barbecue party (a gathering), which agrees with the relational rule that each entity that is holding a gathering is a human. However, the scene also implies that the government (not a human) is holding a press conference (a gathering), which disagrees with the relational rule. Both pieces of evidence are valid, and should be taken into account.

Drawing Conclusions

Learned relational rules are interpreted as equivalences. Indeed, the approach taken by Knowledge Infusion is that of inducing all knowledge characterizing a target relation $R_i(w_1, \ldots, w_{\alpha(i)})$ that is implicit in the available training data. Unlike the case of implicational rules, equivalence rules support both positive and negative predictions.

The process of making predictions by applying learned rules on scenes is carried out in the following natural manner. The body $body(w_1, \ldots, w_{\alpha(i)})$ of a rule is evaluated on a given scene. For each of the bindings of the free variables in the rule for which the body of the rule evaluates to

true/false, the rule predicts that its head $R_i(w_1, \ldots, w_{\alpha(i)})$ is also *true/false*. Clearly, multiple conclusions may be drawn from a single relational rule so applied to a scene. It is also possible that a scene does not provide enough information for the body of a rule to be evaluated for certain bindings. For those bindings, no prediction is made.

Assume, as above, that we have the relational rule

$$(\exists v_2 : \textit{holds}(w_1, v_2) \wedge \textit{gathering}(v_2)) \equiv \textit{human}(w_1).$$

Assume also that a scene determines that the bindings $\textit{gathering}(t_7)$, $\textit{holds}(t_3, t_7)$ are *true*, that $\textit{holds}(t_8, t_4)$ is *false*, and that all remaining bindings are unknown. The rule predicts, then, that $\textit{human}(t_3)$ is *true*, but makes no prediction on the truth-value of $\textit{human}(t_8)$. Indeed, note that although entity t_8 does not hold t_4, nothing is known about whether it holds something else, say t_5, that could be a gathering. The scene does not provide enough information to determine whether t_8 is human or not.

Unlike the case of learning, where it is possible to reduce the relational case to a propositional one, such a reduction is not apparently possible for reasoning. A reasoner capable of dealing with variables, bindings, and relations, is necessary.

Related Work

Because of its central role in building intelligent agents, the acquisition and manipulation of commonsense knowledge has been investigated in numerous works in the literature.

Previous automated approaches to knowledge acquisition generally differ form our approach in that the extracted knowledge is in the form of facts (e.g., the KnowItAll Project (Etzioni et al. 2005)), as opposed to commonsense rules that embody some generalization. Non-automated approaches to knowledge acquisition generally rely on humans to provide knowledge that is either in some raw form not designed for reasoning (e.g., the Open Mind Initiative (Stork 1999)), or if it is in a computer-readable form (e.g., the Cyc Project (Lenat 1995)) it is not designed for error tolerance.

Robust Logic is designed to make computationally feasible both learning and reasoning for an expressive a class as possible. Inductive Logic Programming (Muggleton 1991) is an alternative approach to relational learning, which, however, does not place the same emphasis on algorithmic efficiency, or on a principled common semantics for learning and reasoning. Though more expressive within logic, ILP is restricted to logic, and offers no comparable performance guarantees. Although standard software packages for ILP have been employed for various natural language tasks in the past, these efforts have focused on rather small corpora comprised of only a few hundred sentences, due to scalability considerations (Page and Srinivasan 2003; Specia et al. 2006). By contrast, our experiments are carried out on corpora that are several orders of magnitude larger, and beyond the capabilities of such software.

In the particular application of Knowledge Infusion explored in this work, we seek to identify rules that generalize facts encoded in text. Liakata (2004) has studied this problem in a limited domain with a small set of learned concepts, aiming to establish that natural rules can be extracted. We are interested in the learning of knowledge on a more massive scale, and its subsequent application to recover information implicit in text. Our goal is related to those of Reading the Web (Mitchell 2005) and Machine Reading (Etzioni, Banko, and Cafarella 2006). Yet, the more ambitious goal of our work is to develop a general system that acquires and manipulates knowledge in a principled and medium-independent way.

The extraction of information implicit in text is, to some extent, related to the task of recognizing textual entailment (Dagan, Glickman, and Magnini 2005). Unlike our goal of acquiring knowledge of what holds in some underlying reality, textual entailment seeks to establish whether some statement is implied by another (irrespectively of their objective truth), and measures success subjectively against a human gold standard. One approach encodes sentences in a logical form, and employs a theorem prover to check for logical implication between them (Bos and Markert 2005). Another approach produces an abstraction hierarchy of syntactic and semantic information found in sentences, and seeks to determine whether the second sentence subsumes the first one (de Salvo Braz et al. 2005). Some of the machine learning techniques used to produce the abstraction hierarchy are also employed in our approach (CCG-UIUC 2006). Amongst the most successful approaches, and the one closest to our experimental setting, is that of recognizing textual entailment by employing knowledge induced from a large corpus (Hickl et al. 2006). The fundamental difference between textual entailment and our approach is that the former is a classification task, that of checking whether one sentence is implied by another, while ours is a generation task, that of deriving rules that can be applied to arbitrary situations.

In a different context, some work (Even-Zohar and Roth 2000) has focused more on learning computer-readable rules that are subsequently applied to disambiguate pairs of words in previously unseen text. More recent work (Punyakanok et al. 2005) has also examined how such rules interact with each other through known domain constraints, and how taking these constraints into account can enhance performance. Knowledge Infusion goes further, in that it considers how rules interact with each other through chaining: Robust Logic provides semantics for reasoning by chaining rules and our experimental results seek to quantify the benefits so obtained.

Some form of chaining is employed in transformation-based learning (Brill 1993), although little emphasis has been placed on its formal analysis. According to this method, missing information on all features is initially completed heuristically, and rules are subsequently induced and applied to correct the initial predictions. Such rules encode knowledge about the structure of the wrong predictions of earlier rules, and not knowledge about the underlying reality that one ultimately seeks to discover. The strategy of forcing all features to obtain definite values fundamentally differs from our approach of making predictions only when it is justified for the rules to do so. Despite the success of this strategy in certain natural language tasks (e.g., part-of-speech tagging (Brill 1995), text chunking (Ramshaw and Marcus 1995), spelling correction (Mangu and Brill 1997)), completing all missing information is unrealistic in the experimental setting of Knowledge Infusion that we consider.

Because of the large number of features with missing information in each scene, making such predictions would incur huge computational costs.

Stracuzzi (2005) explores the logistics and memory management issues that arise during learning of multiple concepts. He concludes that rules should be learned in levels, and one should select carefully which rules to use for enhancing information at higher levels.

Scene Construction

Central to Knowledge Infusion is the construction of scenes, from which knowledge is acquired, and on which the knowledge is evaluated. Text corpora offer a natural and abundant source for obtaining scenes. The process through which this is achieved in our system is outlined in this section.

The scenes employed in the experiments reported are based on text taken from the *North American News Text Corpus* (Graff 1995), a series of plaintext newspaper articles. Six months worth of articles, comprising approximately a half million sentences, were employed. Each of the sentences was tagged by the *Semantic Role Labeler* (CCG-UIUC 2006), an automated tagging software. The result was a set of sentences, with each word tagged by its part of speech, and each verb associated with fragments of the sentence that correspond to the verb's arguments. A particular resulting sentence is illustrated in Figure 1.

The *Collins Head Rules* (Collins 1999, Appendix A) were applied on sentence fragments to identify their head words, permitting the association of each verb argument with typically a single word. Roughly, the head word of a sentence fragment captures the essence of the sentence fragment. Referring to Figure 1, for instance, the head words of the sentence fragments "the second warrant" and "a search that was overly broad and therefore illegal" are respectively "warrant" and "search". Using these head words one may then extract from the entire sentence the information that "warrant precipitated search", which summarizes the sentence.

Entity and Relation Identification

A scene is constructed for each sentence. With each word in the sentence we associate in the Robust Logic a token. The seventeen words in the sentence of Figure 1 are, thus, associated in order of appearance with the tokens $t_1, \ldots, t_{17}$. We consider a sentence as admitting two interpretations, one where the sentence itself is the reality of interest, and the other where the sentence simply describes some underlying scenario of interest. Depending on which of these two realities one considers, the token associated with each word admits a different interpretation. Consider, for instance, token t_{17} that is associated with the word ".". Under the first interpretation, then, token t_{17} corresponds to the entity "." as a *syntactic* element of a sentence; there is no entity under the second interpretation to which token t_{17} may meaningfully correspond. Indeed, this is consistent with the fact that the dot symbol in a sentence only serves as offering syntactic information, but not semantic information. Consider, now, token t_2 that is associated with the word "defense". In addition to the syntactic entity to which token t_2 corresponds under the first interpretation, one may easily identify a *semantic* entity to which token t_2 corresponds; in the context of the particular sentence considered, this semantic entity could be, for instance, a lawyer in some court of law. Again, this is consistent with the fact that nouns in a sentence offer both syntactic and semantic information.

A set of relations is extracted from each sentence, along with specific bindings. Since the size of the scene grows exponentially with the arity of the relations, here we restrict the maximum arity α to be 2. Although any constant value guarantees, in theory, a scene of size polynomial in the number of tokens, tractability is, in practice, compromised once arities are increased beyond 2 or 3. Three main categories of relations are defined.

The first category of relations considered is that of *word/pos instances*, which comprises unary relations derived from the word associated with each token, and its corresponding part of speech. Since the word "defense" is associated with token t_2, the instance $\mathit{defense}_{\mathit{wrd}}(t_2)$ becomes part of the constructed scene. A second unary relation corresponding to the part of speech of "defense", also holds on t_2; thus, $\mathit{NN}_{\mathit{pos}}(t_2)$ also becomes part of the scene constructed for this word/pos instance in the given sentence. Such instances offer both syntactic and semantic information.

For each verb in a sentence, we construct a second category of relations which can have any arity up to the maximum arity α, and which we call *verb instances*. The arguments of each such relation refer to head words of (some of) the arguments of the corresponding verb, that may be its subject, object, manner, actual occurrence of the verb in the sentence, etc. Consider, for instance, our earlier example where the words "warrant" and "search" were identified, respectively, as the subject and object of the verb "precipitate", while the word "precipitated" was identified as the actual occurrence of the verb in the sentence. According to our approach of assigning tokens, the verb's three arguments are associated, respectively, with the tokens t_6, t_9, and t_7. Since the maximum arity α is 2, the following verb instances are constructed:

$$\mathit{precipitate}_{\mathit{sbj}}(t_6), \quad \mathit{precipitate}_{\mathit{obj}}(t_9),$$
$$\mathit{precipitate}_{\mathit{vrb}}(t_7), \quad \mathit{precipitate}_{\mathit{sbj,obj}}(t_6, t_9),$$
$$\mathit{precipitate}_{\mathit{sbj,vrb}}(t_6, t_7), \quad \mathit{precipitate}_{\mathit{obj,vrb}}(t_9, t_7).$$

Verb instances offer purely semantic information, as they describe what holds in an underlying scenario according to the sentence's meaning.

Proximity instances, the third category of relations, arise from words within close distance of a certain other word in a sentence. For each of the former words and their part of speech, we construct a unary relation that holds on the latter word, and is annotated by their relative position, a number in the interval $[-3, +3]$. The sentence in Figure 1, for instance, gives rise to proximity instances that include the following:

$$\mathit{the}_{\mathit{wrd,+2}}(t_2), \quad \mathit{precipitate}_{\mathit{wrd,-2}}(t_9), \quad \mathit{JJ}_{\mathit{pos,-1}}(t_6)$$

In some sense, then, proximity instances resemble word/pos instances, but hold on the token associated with a nearby word. Unlike the other categories of instances, proximity instances are clearly syntactic. Such relations offer information relating to the medium in which information is encoded

Syntactic Information *Semantic Information*

```
sentence                                                                contend precipitate   be

(S1 (S (NP (DT The)                                                    (A0 +         +         +
           (NN defense))                                                   + A0)     +         +
       (VP (VBZ contends)                                               (V + V)     +         +
           (SBAR (S (NP (DT the)                                        (A1 +     (A0 +         +
                        (JJ second)                                        +         +         +
                        (NN warrant))                                      +         + A0)     +
                    (VP (VBD precipitated)                                 +       (V + V)     +
                        (NP (NP (DT a)                                     +     (A1 +     (A1 +
                                (NN search))                               +         +         + A1)
                            (SBAR (WHNP (WDT that))                        +         +   (R-A1 + R-A1)
                                  (S (VP (AUX was)                         +         +     (V + V)
                                         (ADJP (ADJP (RB overly)           +         +     (A2 +
                                                     (JJ broad))           +         +         +
                                               (CC and)                    +         +         +
                                               (ADJP (RB therefore)        +         +         +
                                                     (JJ illegal)))))))))) + A1)     + A1)     + A2)
       (. .)))                                                             +         +         +
```

Figure 1: The output of automated tagging of the sentence "The defense contends the second warrant precipitated a search that was overly broad and therefore illegal.". Syntactic information about the sentence is shown on the left. Semantic information about the verbs that occur in the sentence, along with the sentence fragments that correspond to each verb's arguments are shown on the right. Syntax tags beginning with `VB` and `NN` correspond, respectively, to verbs and nouns. The semantic tags `V`, `A0`, and `A1` indicate, respectively, the position of the verb in the sentence, the part of the sentence that corresponds to the verb's subject, and the part of the sentence that corresponds to the verb's object.

(i.e., English text), and do not directly offer information on the underlying scenario described by a sentence. They are, however, known to be useful in many natural language tasks (Even-Zohar and Roth 2000; Roth and Yih 2001).

Each identified relation undergoes synonym clustering, and is replaced in the scene by its primary synonym according to its most common WordNet sense (Miller 1995).

Negative Instance Sampling

Knowledge encoded in natural language text is largely about properties of a domain that are *true*, and rarely about those that are *false*. In view of the need for negative learning instances, one could take the approach of treating all unknown information as *false*. For certain tasks, such as disambiguation of a pair of words (Even-Zohar and Roth 2000), this is appropriate. This approach is not, however, appropriate for Knowledge Infusion. A practical concern is that the constructed negative instances should not outnumber the available positive ones by several orders of magnitude. A more philosophical concern is that in Knowledge Infusion when a rule is applied it should not predict *false* every time something is unstated. Instead, we expect the unknown information to be *completed* in a manner that corresponds to the truth in the underlying domain of interest.

To construct negative instances for the relation R_{i_1} of arity m we employ a form of sampling. Whenever a positive instance of some other relation $R_{i_2}(t_{j_1}, \ldots, t_{j_m})$ of arity m is encountered, a negative instance $R_{i_1}(t_{j_1}, \ldots, t_{j_m})$ for R_{i_1} is constructed on the same tokens. The negative instance is probabilistically constructed according to the frequency of positive examples of R_{i_1} compared to that of other relations, in a manner that ensures that positive and negative instances for R_{i_1} are (on expectation) balanced. Hence, only few of the unknown instances of R_{i_1} are treated as *false*, and only when some other relation holds on the same tokens. In the scene derived from the sentence in Figure 1, thus, the binding $\mathit{precipitate}_{\mathit{sbj,obj}}(t_9, t_{13})$ is, with some probability, taken to be *false*, since the binding $\mathit{be}_{\mathit{sbj,obj}}(t_9, t_{13})$ is already known to be *true* in the scene.

Schemas and Target-Centered Quantification

The experiments reported all use the schema $\mathcal{Q}$ described below. The expressions in $\mathcal{Q}$ are based on conjunctions

$$R_{i_0}(t_{j_1}, \ldots, t_{j_{\alpha(i_0)}}) \wedge R_{i_1}(t_{j_1}) \wedge \ldots \wedge R_{i_{\alpha(i_0)}}(t_{j_{\alpha(i_0)}})$$

of a (unary or non-unary) relation R_{i_0} with a set of other unary relations R_{i_k} that hold on distinct tokens on which the former relation holds; call these conjunctions the *generating expressions*. Word/pos, verb, and proximity instances may all appear in a generating expression. With respect to the set of relations derived from the sentence in Figure 1, for instance, $\mathit{precipitate}_{\mathit{subj,obj}}(t_6, t_9) \wedge \mathit{warrant}_{\mathit{wrd}}(t_6) \wedge \mathit{RB}_{\mathit{pos,+3}}(t_9)$ and $\mathit{warrant}_{\mathit{wrd}}(t_6) \wedge \mathit{precipitate}_{\mathit{wrd,+1}}(t_6)$ are amongst the generating expressions constructed.

Once a generating expression is constructed, all of its possible existential quantifications are considered, and unquantified tokens are replaced with free variables. The set of all expressions so constructed comprise the schema $\mathcal{Q}$. Note, however, that only a subset of the quantifications is actually employed in any particular learning task, the subset depending on the target relation itself. If, for instance, the target relation is $\mathit{warrant}_{\mathit{wrd}}$, and the scene assigns a definite truth-value only to one binding of this relation, say

warrant$_{wrd}(t_6)$, then any expression in the schema Q resulting from a generating expression that contains tokens other than t_6, as is the case for $\exists v_1 :$ *precipitate*$_{subj,obj}(v_1, t_9)$, is not eligible to appear in the body of the rule being induced. By contrast, both $\exists v_2 :$ *precipitate*$_{subj,obj}(t_6, v_2)$ and $\exists v_1 \exists v_2 :$ *precipitate*$_{subj,obj}(v_1, v_2)$ are eligible.

To avoid explicitly constructing those expressions in Q not used in a given scene, we employ a *target-centered quantification* technique, which is applied during the training phase, and dynamically builds quantifications on a per target relation basis. For a given binding $R_i(t_{j_1}, \ldots, t_{j_{\alpha(i)}})$ of a target relation R_i, and a given generating expression

$$R_{i_0}(\ldots, t_{j_k}, \ldots, t_{\ell_1} \ldots, t_{\ell_m}) \wedge$$
$$\ldots \wedge R_{i_k}(t_{j_k}) \wedge \ldots \wedge R_{i'_1}(t_{\ell_1}) \wedge \ldots \wedge R_{i'_m}(t_{\ell_m}),$$

we existentially quantify each token t_ℓ not amongst the tokens on which R_i holds, so that different quantifiers imply different quantified tokens; this gives the expression

$$\exists v_1 \ldots \exists v_m : R_{i_0}(\ldots, t_{j_k}, \ldots, v_1, \ldots, v_m) \wedge$$
$$\ldots \wedge R_{i_k}(t_{j_k}) \wedge \ldots \wedge R_{i'_1}(v_1) \wedge \ldots \wedge R_{i'_m}(v_m).$$

The remaining tokens are replaced by distinct free variables, and the resulting expression becomes part of the dynamically constructed schema Q. Thus, for each fixed binding of a target relation, a generating expression is uniquely and efficiently quantified, in a manner that retains as much of its association with the binding of the target relation as possible. The resulting quantified expression is allowed to appear in the body of the rule being induced for the target relation only if the two share some unquantified token. This last requirement further reduces the number of expressions in Q, by eliminating expressions that convey no direct information for the given binding of the target relation. Thus, for the binding *warrant*$_{wrd}(t_6)$ of the target relation *warrant*$_{wrd}$, the expression $\exists v_2 :$ *precipitate*$_{subj,obj}(t_6, v_2)$ is the unique quantification of the generating expression *precipitate*$_{subj,obj}(t_6, t_9)$ that is eligible to appear in an induced rule for *warrant*$_{wrd}$.

Experimental Approach

The main goal of our experimental setting was that of establishing that commonsense knowledge can be learned from natural inputs on a massive scale reliably enough that chaining the learned rules is efficacious for reasoning. We discuss in this section the approach we have taken.

Primitive Rule Operations

The primitive blocks of our experimental approach are three rule-based tasks: induction, evaluation, and application.

In the *rule induction task*, one is given as input a target relation R_i, and a training set T of scenes. The scenes are sequentially fed to the relational learner implemented within our system. Relations are translated to appropriate propositions as described earlier, and a propositional Winnow-based learner (CCG-UIUC 2005) is invoked to produce a propositional rule, which is then mapped back to a relational rule K_i. This relational rule is the output of the induction task.

In the *rule evaluation task*, one is given as input a rule K_i with an associated head relation R_i, and a testing set E of scenes. Each binding of R_i is considered. For some of these bindings the scenes in E designate a truth-value for R_i. For each such case, the truth-value is recorded, and then obscured. The remainder of the scene is given as input to the rule K_i, and the rule makes a prediction for the obscured binding of R_i. The prediction is recorded and contrasted against the actual truth-value that was obscured. The process is repeated across all bindings in E. Recall, precision, and F-measure performance is computed for the rule. These performance values are the output of the evaluation task.

In the *rule application task*, one is given as input a set of rules K, and a set S of scenes. Each triple comprised of a rule $K_i \in K$, a binding of the rule's head R_i, and a scene in S is considered in turn. In case the scene already assigns a truth-value to the binding of R_i on which a prediction is to be made, that truth-value is first obscured. The rule is then applied to the scene and makes a prediction, and the scene is updated with the prediction of the rule. Rules in K are applied in parallel in that their predictions are not visible to each other; only the input scenes are visible to the rules. The set of all enhanced scenes (after the application of all rules on all scenes in S) is the output of the application task.

The output of the rule application task can be used as input to another task of rule induction or rule evaluation. This allows one to investigate the effects of chaining rules, since the rule that is to be induced or evaluated on the output of the rule application task has access to the conclusions of previously applied rules, and is allowed to build on those. In this *rule chaining task*, the rules that were applied during the rule application task comprise the *first layer of rules*, whereas the single rule that is induced/evaluated during the rule induction/evaluation task comprises the *second layer of rules*.

To avoid any possible leakage of information between the two layers of rules in this chaining process, we modify the rule application task as follows: The head R_{i_0} of the rule that is involved in the second layer is also given as input to the rule application task; note that the head is known even if the rule has not been induced yet. Before the first layer of rules K is applied to scenes in S to draw conclusions, any information about R_{i_0} holding in the scene is obscured. Thus, rules K do not have access to the truth-value of any binding of R_{i_0}, and cannot encode any information about that truth-value in their predictions.

When enhancing scenes during the rule application task, several alternatives may be considered. Predictions that are in conflict with information already given in a scene may be either retained or discarded. In those cases where a rule's prediction is chosen to be added to a scene, we consider adding the prediction either on the actual target relation, or on a duplicate new target relation. For instance, the prediction on the binding *defense*$_{wrd}(t_2)$ may be either added in a scene as is, or as *defense*$^*_{wrd}(t_2)$. The latter approach introduces additional learning features for the second layer of rules, but does not affect the distribution of the original ones. We also investigate the case where only confident rule predictions are considered for inclusion in a scene.

Confident Rule Predictions

The option of considering only confident rule predictions amounts, in general, to artificially reducing the completeness of rules in favor of increasing their soundness. We employ this in the rule application task, but *not* on any of the rules involved in a rule evaluation task, whose performance is reported in our experiments.

To achieve an increase in predictive precision for rules K in a rule application task, one may first decide which of the given rules K are to be applied, and which are to remain inactive. To do so, each rule is assigned an overall confidence corresponding to *our* belief in the rule's predictive soundness; we call this confidence *external*. This confidence is based on the rule's empirically measured predictive soundness, when trained and tested through cross-validation. Note that all such training and testing is performed on the first half of the corpus. No access is permitted to the second half of the corpus, which is reserved for the final evaluation of our experimental investigation of extracting commonsense knowledge. Based on the external confidence assigned to each rule, only rules that exceed some specified *external confidence threshold* are chosen to be applied, ensuring that their positive and negative predictions are highly sound.

A second means of increasing the predictive precision is to decide which amongst the predictions of a given rule in K are to be incorporated in a scene, and which are not. Each rule is assigned a confidence on *each* of its individual applications, corresponding to *the rule's own* belief that a positive prediction will be indeed accurate; we call this confidence *internal*. Rules with bodies based on linear thresholds, as the ones we consider here, have a natural internal confidence indicator: the sum of the weights of the active features in a given scene, abbreviated henceforth as SWAF. Since a linear threshold rule makes a positive prediction when its SWAF is sufficiently large, it is natural to assume that the higher the rule's SWAF is, the more confident the rule is on the accuracy of making a positive prediction. This indicator is, however, insufficient as it does not meaningfully correspond to the likelihood that the prediction will be accurate.

We convert SWAFs into such likelihoods by establishing an empirical mapping between the two through cross-validation. The goal of the mapping is to identify when a given rule is internally confident with probability p that a particular positive prediction is accurate. For each application of the rule, we record the actual value that the scene determines for the target relation to be predicted, and the rule's SWAF. We then adjust the rule to make positive predictions only when its SWAF is above some value $v(p)$, computed to be the least value such that when the rule's SWAF is above $v(p)$, at least a percentage p of the actual values of the target attribute were *true* during cross-validation; these cases correspond to those where the rule correctly predicts that the value of the target attribute is *true*. $v(p)$ is the rule's *internal confidence threshold for achieving precision* p.

By standard arguments, it can be shown that the two approaches discussed above are provably predictive in the PAC sense, in the exact context in which they are employed — to identify highly sound rules, and highly precise predictions.

Learning and Reasoning Interplay

Our experimental tasks were designed to evaluate, among other things, various types of interactions between the learning and reasoning processes. Interactions differ on whether rules are applied in parallel or chained, and on whether chaining happens during the training phase or the testing phase. The different configurations discussed below consider different ways in which rule induction, rule evaluation, and rule application interact. Each experimental task takes as input a training set T_0 of scenes, a testing set E_0 of scenes, an enhancement set R_{enh} of relations, and a target relation R_{i_0}. In all cases, the output of the experimental task is a single rule K_{i_0} for predicting R_{i_0}, and the performance of that rule on a testing set of scenes that is determined by each task.

In a *type 00* experimental task, the relations in R_{enh} are ignored. The rule K_{i_0} is induced on the training set T_0, and evaluated on the testing set E_0.

In a *type 01* experimental task, a rule is induced on the training set T_0 for each relation in R_{enh}. The learned rules K_{enh} are applied on the testing set E_0 to obtain an enhanced testing set E_1. The rule K_{i_0} is induced on the training set T_0, and evaluated on the enhanced testing set E_1.

In a *type 10* experimental task, a rule is induced on the training set T_0 for each relation in R_{enh}. The learned rules K_{enh} are applied on the training set T_0 to obtain an enhanced training set T_1. The rule K_{i_0} is induced on the enhanced training set T_1, and evaluated on the testing set E_0.

In a *type 11* experimental task, a rule is induced on the training set T_0 for each relation in R_{enh}. The learned rules K_{enh} are applied on the training set T_0 and on the testing set E_0 to obtain an enhanced training set T_1 and an enhanced testing set E_1, respectively. The rule K_{i_0} is induced on the enhanced training set T_1, and evaluated on the enhanced testing set E_1.

We repeat that the performance of the rule K_{i_0} is the only piece of information reported in the experimental tasks. Rules in K_{enh} are only employed to enhance the information available to K_{i_0} during its induction and/or evaluation. Only externally confident rules in K_{enh} are used during the enhancement phase, and only their internally confident predictions are recorded. As discussed earlier, we exclude the possibility that any information on the relation R_{i_0} leaks into the predictions of K_{enh} by obscuring during the enhancement phase all information on expressions involving R_{i_0}.

Information on certain additional expressions is also obscured whenever any rule in $K_{enh} \cup \{K_{i_0}\}$ makes predictions. Recall that the relations $\mathcal{R}$ constructed for a scene are associated with certain words in some sentence, and that multiple relations might be associated with the same word. For instance, the relations *contend*$_{wrd}$, *contend*$_{sbj,obj}$, and *contend*$_{wrd,+2}$, corresponding, respectively, to a word/pos instance, a verb instance, and a proximity instance, are all associated with the word "contend", despite the fact that the relations are meant to encode different pieces of information regarding the syntax and semantics of the sentence. If any of these three relations is to be predicted through some rule, then all expressions in the employed schema that involve any of these relations are obscured.

Syntactic and Semantic Information

Orthogonally to how rules are induced, evaluated, and applied, we also examine the effect of the type of information that is made available to rules. Recall that scenes constructed from sentences provide two main types of information: syntactic and semantic. These correspond respectively to information about the sentence itself; and information about the underlying reality, or meaning, of the sentence.

In our experimental setting we employ the term *syntactic* information to mean that part of a scene that only contains word/pos and proximity instances. We employ the term *semantic* or *commonsense* information to mean that part of a scene that only contains word/pos and verb instances. This distinction gives rise to three categories of experiments, depending on which type of information is allowed: syntactic, semantic, or both. Thus, for instance, when inducing, evaluating, or applying rules, we may make only semantic information visible to the rules. Such rules capture, then, commonsense knowledge, since they encode information that relates semantic pieces of information (about the underlying reality). Note that the type of information that is visible to a rule within a given experimental task does not change across the rule's induction, evaluation, and application.

In the case of chaining rules, each of the two layers of rules may face different types of information. It is possible, for instance, for rules in the first layer to be induced and applied with access to semantic information only, whereas the single rule in the second layer is induced and evaluated with access to both semantic and syntactic information.

Experimental Parameters and Results

We report in this section experimental results that were obtained by applying the methodology described in this paper.

Experimental Parameters

The available part of the *North American News Text Corpus* (Graff 1995) was split into two equal halves, containing different sets of articles. The first half was further split and was used for parameter fitting during the design of the experiments. The entire first half was also used as a training set for the experiments reported, with the second half being used as a testing set. Each sentence gave rise to a scene, and the first and second halves of the corpus were used, respectively, to populate the training set T_0 and the testing set E_0. In particular, the two sets were constructed from sentences coming from different newspaper articles, and in fact, from newspaper articles coming from different and disjoint three month time periods, so that any implicit correlation between the two sets would be avoided. Each of the sets was processed as described earlier in this paper.

Various standard learning parameters were determined empirically. During induction tasks, the training set was sequentially fed to the learning algorithm 20 times to allow the induced rules to stabilize, and avoid any artifacts that the fixed ordering of the training set might have produced. Features that occurred rarely in the training set were pruned.

Target relations R_{i_0} were selected to be word instances that occurred more than 500 times and less than $10,000$ times in the training set T_0. 268 different such targets were chosen. In all experiments that employed chaining of rules, a common enhancement set R_{enh} of relations was used, which was obtained as follows: Rules for all 268 target relations R_{i_0} were trained under various sets of learning parameters, and various scene construction parameters. The verb instances that were part of the learned rules were assembled, and those occurring in some expression with a weight of more than 0.05 were selected to populate the enhancement target set R_{enh}. For efficiency reasons, the size of R_{enh} was reduced to the hundreds by retaining only the top 30% of the verb instances ordered according to their weight; this gave rise to 599 such verb instances. We emphasize that the relations in R_{enh} were selected based *not* on their occurrence frequency, but, rather, on their perceived usefulness for rules that predict at least one of the 268 target relations for at least one choice of learning parameters. This abductive approach allowed the size of the rule base K_{enh} associated with R_{enh} to remain relatively constrained compared to the simpler approach of learning rules for all verb instances found in any of the scenes in the training set T_0.

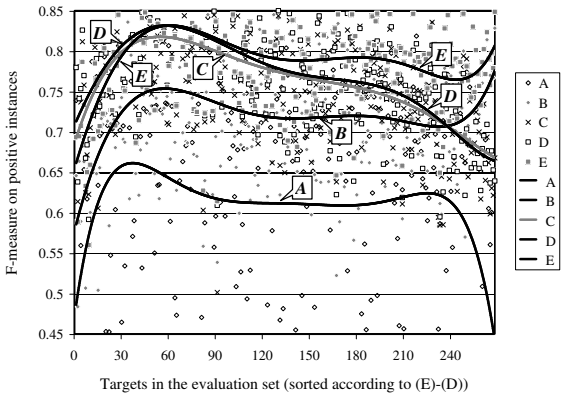
When inducing and applying rules K_{enh}, only semantic information was made visible and used. These are the rules in our experimental setting that we consider as the commonsense knowledge being extracted from the text corpus. Following the experimental tasks that we have already described, these commonsense rules were used to enhance scenes with extra predicted information that was not originally present in the scenes. Only rules in K_{enh} with external confidence 75% were employed to enhance the scenes T_0 and E_0; 200 rules satisfied this constraint. Only predictions with internal confidence 99.9% were recorded in the enhanced scenes T_1 and E_1. Recorded predictions assumed new names, and conflicts were not discarded, so as to facilitate the easier monitoring of the behavior of the system. Changing either of these choices did not seem to affect the reported end results.

Experimental Results

The four types $00, 01, 10, 11$ of experimental tasks were carried out with input $\langle T_0, E_0, R_{enh}, R_{i_0} \rangle$, for each of the 268 different choices of R_{i_0}. As anticipated, the performance on the experimental tasks of type 01 and 10 was on average significantly lower than the performance on the experimental tasks of type 00 and 11, because of the mismatch between the distributions for training and testing of the rule K_{i_0} predicting the target relation R_{i_0}. The performance on the latter two experimental tasks is illustrated in detail in Figure 2.

For any fixed target relation R_{i_0} (corresponding to a point on the x-axis in Figure 2), our experimental setting resembles the following scenario: An agent is presented with a training set of sentences, tagged with part of speech information, and with the each sentence's verbs (and their arguments) identified, and is allowed to induce a rule K_{i_0} for predicting whether the word associated with the target relation R_{i_0} is present in each sentence. Subsequently, the agent is presented with a new sentence from some testing set of sentences, tagged as before, but with some word obscured. The agent is asked to determine whether the obscured word is equal to the word associated with the target relation R_{i_0}; always replying "yes" (or "no") guarantees a 50% accuracy.

individual statistics	accuracy	recall	precision	F-measure
mean for case *(A)* ◊	66.17%	61.03%	72.67%	61.17%
mean for case *(B)* ♦	74.46%	66.24%	79.78%	71.98%
mean for case *(C)* ×	78.01%	71.65%	82.32%	76.18%
mean for case *(D)* □	78.65%	71.67%	83.79%	76.87%
mean for case *(E)* ■	80.24%	74.57%	84.32%	78.95%



statistics on *(B)*−*(A)*	accuracy	recall	precision	F-measure
mean (of all values)	8.29%	5.20%	7.11%	10.81%
trimmed mean (80%)	7.93%	4.37%	6.99%	8.62%
median (of all values)	7.46%	3.44%	7.30%	6.03%
average deviation	4.97%	21.81%	9.97%	12.30%
standard deviation	6.21%	26.11%	11.50%	15.96%
99% confidence interval	[7.31%, 9.26%]	[1.10%, 9.31%]	[5.30%, 8.91%]	[8.30%, 13.32%]
t-test on *(A)* and *(B)*	0.00000	0.00325	0.00000	0.00000
S_{evl} targets benefited	248 (92.5%)	143 (53.4%)	180 (67.2%)	204 (76.1%)

statistics on *(D)*−*(C)*	accuracy	recall	precision	F-measure
mean (of all values)	0.64%	0.02%	1.47%	0.69%
trimmed mean (80%)	0.78%	0.23%	1.35%	0.75%
median (of all values)	0.78%	0.53%	1.11%	0.81%
average deviation	1.18%	3.88%	2.48%	1.73%
standard deviation	2.02%	5.61%	3.31%	2.55%
99% confidence interval	[0.32%, 0.96%]	[−0.86%, 0.90%]	[0.95%, 1.99%]	[0.29%, 1.09%]
t-test on *(C)* and *(D)*	0.24873	0.98355	0.00770	0.29872
S_{evl} targets benefited	200 (74.6%)	148 (55.2%)	185 (69.0%)	184 (68.7%)

statistics on *(E)*−*(D)*	accuracy	recall	precision	F-measure
mean (of all values)	1.58%	2.90%	0.53%	2.08%
trimmed mean (80%)	1.35%	2.89%	0.20%	1.78%
median (of all values)	1.19%	2.39%	−0.04%	1.46%
average deviation	1.82%	4.52%	2.69%	2.58%
standard deviation	2.41%	6.30%	3.61%	3.51%
99% confidence interval	[1.20%, 1.96%]	[1.91%, 3.89%]	[−0.04%, 1.10%]	[1.53%, 2.63%]
t-test on *(D)* and *(E)*	0.00297	0.00076	0.26608	0.00082
S_{evl} targets benefited	195 (72.8%)	200 (74.6%)	131 (48.9%)	200 (74.6%)

Figure 2: Empirical results for 268 sets of experiments. Each set of experiments involves inducing and evaluating a rule K_{i_0} for a given target relation R_{i_0}. Within each set of experiments, five performance points are reported, depending on the type of information that is available during the induction and evaluation of K_{i_0}: *(A)* only semantic information is visible to K_{i_0}; *(B)* as in *(A)*, but training and testing scenes are enhanced through commonsense rules K_{enh}; *(C)* only syntactic information is visible to K_{i_0}; *(D)* both syntactic and semantic information is visible to K_{i_0}; and *(E)* as in *(D)*, but training and testing scenes are enhanced through commonsense rules K_{enh}. The top left table shows the average performance for *(A)* through *(E)* across the 268 choices of R_{i_0}. The graph plots the F-measure performance (on the y-axis) for each of the 268 choices of R_{i_0} (on the x-axis); the curves show polynomials of degree six that minimize the squared-distance error from the corresponding data points. Statistics on the differences between three pairs of cases are shown on the right. Rows labelled "99% confidence interval" show intervals for the true mean of the difference assuming target relations R_{i_0} are uniformly drawn. Rows labelled "t-test on X and Y" show the probability of the empirical means of X and Y being this far apart under the null hypothesis that they come from the same distribution. The difference *(B)*−*(A)* shows the benefit of chaining when only semantic information is visible to K_{i_0}. The difference *(D)*−*(C)* suggests that cases *(C)* and *(D)* have comparable performance. The difference *(E)*−*(D)* shows the benefit of chaining when both syntactic and semantic information is visible to K_{i_0}. The 268 choices of R_{i_0} are sorted (on the x-axis) in the graph in order of increasing *(E)*−*(D)* difference.

Alternatively, and conceptually closer to our treatment of missing information, one may view the task as that of predicting whether a certain entity in the underlying scenario described by the sentence has a certain property, namely the property that corresponds to the target relation R_{i_0}.

In the experimental tasks of type 00 the agent attempts to reach a decision by applying the rule K_{i_0} that it has previously induced. Different cases are examined when the rule K_{i_0} is induced and evaluated, by allowing the rule access to only semantic (case *(A)*), only syntactic (case *(C)*), or both (case *(D)*) types of information in the sentence. In the experimental tasks of type 11 (cases *(B)* and *(E)*), the agent again attempts to reach a decision through the induction and application of the rule K_{i_0}, but it is first allowed to enhance the sentences by drawing conclusions through a set of commonsense rules K_{enh}. All commonsense rules K_{enh} are applied both before inducing K_{i_0} and before applying K_{i_0}. Of interest in our experimental setting is examining whether the application of these commonsense rules enhances the agent's ability to make correct predictions in the described scenario.

The experimental results indicate that enhancing scenes with commonsense conclusions is useful overall, leading to performance increases that are robust across various choices of R_{i_0}. We note, in particular, that *(D)* and *(E)* refer to experiments in which exactly the same information (namely, both syntactic and semantic) is visible to the rule K_{i_0}. They only differ in that an extra layer of rule application is allowed in *(E)*. Hence, we can deduce that in this instance reasoning with commonsense knowledge leads to an increase in performance. The purpose of *(C)* is to offer a baseline for comparison. It shows that a state-of-the-art technique (where only syntactic information is visible to the rule K_{i_0}) gives similar performance to *(D)*, and hence that the improvement shown for *(E)* over *(D)* is meaningful in the context of high performing systems. In other words, it is this improvement of *(E)* over *(D)* that we offer as evidence for usefulness of the rules extracted and hence the success of Knowledge Infusion. Note that with respect to F-measure, *(E)* improves performance in 200 of the 268 experiments, the average improvement over all experiments is 2.08%, and the 99% confidence interval for this mean is $[1.53\%, 2.63\%]$.

Cases *(A)* and *(B)*, included for illustrative purposes, show an analogous improvement obtained from reasoning with commonsense knowledge, when only semantic information is visible to the rule K_{i_0}.

For completeness we present part of a rule induced in the

second layer of a type 11 experimental task for predicting $\mathit{price}_{\mathit{wrd}}(w)$. Associated with each feature (i.e., member of Q) is its weight in the induced linear threshold, which has a threshold value of 1.

$$\exists v_1 : \mathit{lower}^*{}_{\mathit{sbj,obj}}(w, v_1) \wedge \mathit{demand}(v_1) \quad (0.514704)$$
$$\exists v_2 : \mathit{lower}^*{}_{\mathit{sbj,obj}}(v_2, w) \wedge \mathit{bargain}(v_2) \quad (1.088027)$$
$$\exists v_3 : \mathit{lower}^*{}_{\mathit{sbj,obj}}(v_3, w) \wedge \mathit{competition}(v_3) \quad (0.985423)$$

This rule snippet shows that price lowers demand, and is lowered by bargain and competition. Note that the rule would likely predict $\mathit{price}_{\mathit{wrd}}(w)$ on the basis of the truth of either of the last two features, but would need additional corroboration for the first one. Note also that in this example the verb instance $\mathit{lower}^*{}_{\mathit{sbj,obj}}$ is not present in the original scenes, but is added to the enhanced scenes during the application of the commonsense rules in K_{enh}. The rule presented above is able to exploit this predicted feature.

Evaluation

Although certain individual components of our experimental approach might have been used in previous studies, this is apparently the first demonstration of an actual system capable of extracting real-world knowledge in an automated manner and on such a massive scale as reported here.

Our NLP task is designed to test the hypothesis that commonsense knowledge has been learned, and is different from and not strictly comparable to previous NLP experiments. We learn one prediction rule for each target word to encapsulate the commonsense knowledge about it. The traditional NLP task most relevant to our setting is Word Sense Disambiguation (Agirre and Edmonds 2006), where rules predict which among a *pair* of words is missing. That is an easier task. One could in principle reproduce such results on our scale by considering the $267 \cdot 268 = 71556$ possible pairs of words. Our techniques may offer similar increases in performance in such experiments, but we would consider that a weaker demonstration of commonsense knowledge acquisition, even if it could be done with moderate computational cost.

We wish to emphasize that the system's design and implementation has been an engineering challenge that span several years. The system was built to handle noise and ambiguities in natural language text, and parsing errors that resulted from the use of NLP tools. The corpus that was employed was not specially prepared to identify sentences with useful information. Instead, for each target the rule was induced from about one thousand sentences found automatically among the half million sentences in the corpus. The design and implementation of the system were carried out with scalability in mind, often employing tailored algorithms and advanced data structures, heavy use of memoization, and explicit compression techniques for storing information in memory. On several occasions, the requirement for scalability necessitated that data be handled in a manner known to introduce additional noise, dealing with which was delegated to a noise-resilient learning algorithm.

The reasoning engine comprised the main bottleneck, and necessitated its implementation anew, despite also supporting the invocation of a Prolog engine. The reasoning engine tests hundreds of relational rules, for each of hundreds of groundings of their variables, and parses thousands of features in each rule's body to determine what conclusions to draw, repeating this on each of a half million scenes, and dealing with millions of distinct features overall. Such a massive scale reasoning task is far beyond the capabilities of Prolog engines in terms of memory and time usage. Inductive Logic Programming software, such as Aleph (Srinivasan 2004), is also excluded from consideration, since the ILP learning techniques involve exponential-time algorithms.

Significant effort was also put in the logistics of storing data shared across different experiments or corresponding to the system's final output. The reported experiments required tens of gigabytes of storage space for storing scenes, identified features, propositional examples, induced rules in propositional and relational forms, enhanced scenes, performance results for several metrics for each rule, and so on.

In closing, we note that the system supports numerous parameters not fully explored in the reported experiments. Parameterization is with respect to both the front-end component dealing with natural language text related processing, and the back-end medium-independent engines for learning and reasoning. The back-end component of the system offers a general tool for Knowledge Infusion, which can be employed in other experimental settings in place of existing general-purpose learning and reasoning software.

Conclusions

The development of automatic knowledge acquisition mechanisms for commonsense knowledge is clearly a worthy goal. A basic question addressed in this paper is a methodological one: how should progress in that pursuit be measured? The experimental design described in this paper offers one approach to this problem, and the experimental results successfully demonstrate the feasibility of the proposed approach on a massive scale real-world task.

The main conclusion that the experiments point to is that the chaining of learned rules, some of which express commonsense knowledge, results in increased performance for the prediction task we have described. We emphasize that every aspect of our experiment is designed to be scalable, so that similar experiments should be possible for larger data sets and more target words.

There are several promising avenues towards attempting to obtain larger performance improvements than we have reported here. Besides the obvious, larger corpora, more computational power and more elaborate algorithmic engineering, the following modifications may also lead to significant improvements: Coreference resolution may be incorporated. Scenes may be constructed from multiple sentences, rather than from a single sentence (which is particularly restrictive since when we obscure information related to the target in the scene enhancement stage, we are removing a significant fraction of the information). Learning algorithms other than the Winnow-based system we used can be tried, as may also other ways of producing negative examples, more expressive schemas, and more layers of reasoning.

Acknowledgments

The authors wish to thank Dan Roth and his group for providing SNoW and SRL, and technical support.

References

Agirre, E., and Edmonds, P., eds. 2006. *Word Sense Disambiguation: Algorithms and Applications*. New York, New York, U.S.A.: Springer-Verlag.

Bos, J., and Markert, K. 2005. Recognizing Textual Entailment with Logical Inference. In *Proc. of HLT/EMNLP'05*, 628–635.

Brill, E. D. 1993. *A Corpus-Based Approach to Language Learning*. Ph.D. Dissertation, Dept. of Comp. and Information Science, Univ. of Pennsylvania, U.S.A.

Brill, E. D. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part-of-Speech Tagging. *Computational Linguistics* 21(4):543–565.

CCG-UIUC. 2005. SNoW Learning Architecture. *Cognitive Computation Group, Univ. of Illinois at Urbana-Champaign* <http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=SNOW>.

CCG-UIUC. 2006. Semantic Role Labeler. *Cognitive Computation Group, Univ. of Illinois at Urbana-Champaign* <http://l2r.cs.uiuc.edu/~cogcomp/asoftware.php?skey=SRL>.

Collins, M. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. Dissertation, Dept. of Comp. and Information Science, Univ. of Pennsylvania, U.S.A.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The PASCAL Recognizing Textual Entailment Challenge. In *Proc. of RTE'05*, 1–8.

de Salvo Braz, R.; Girju, R.; Punyakanok, V.; Roth, D.; and Sammons, M. 2005. An Inference Model for Semantic Entailment in Natural Language. In *Proc. of AAAI'05*, 1043–1049.

Etzioni, O.; Banko, M.; and Cafarella, M. J. 2006. Machine Reading. In *Proc. of AAAI'06*, 1517–1519.

Etzioni, O.; Cafarella, M. J.; Downey, D.; Popescu, A.-M.; Shaked, T.; Soderland, S.; Weld, D. S.; and Yates, A. 2005. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *AIJ* 165(1):91–134.

Even-Zohar, Y., and Roth, D. 2000. A Classification Approach to Word Prediction. In *Proc. of NAACL'00*, 124–131.

Graff, D. 1995. North American News Text Corpus. In *Linguistic Data Consortium*, number LDC95T21. Philadelphia, Pennsylvania, U.S.A.: Univ. of Pennsylvania.

Hickl, A.; Williams, J.; Bensley, J.; Roberts, K.; Rink, B.; and Shi, Y. 2006. Recognizing Textual Entailment with LCCs Groundhog System. In *Proc. of RTE'06*, 80–85.

Lenat, D. B. 1995. CYC: A Large-Scale Investment in Knowledge Infrastructure. *CACM* 38(11):33–38.

Liakata, M. 2004. *Inducing Domain Theories*. Ph.D. Dissertation, Computational Linguistics Dept., Univ. of Oxford, U.K.

Littlestone, N. 1988. Learning Quickly when Irrelevant Attributes Abound: A New Linear-Threshold Algorithm. *Machine Learning* 2(4):285–318.

Mangu, L., and Brill, E. 1997. Automatic Rule Acquisition for Spelling Correction. In *Proc. of ICML'97*, 187–194.

Manning, C. D., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, Massachusetts, U.S.A.: MIT Press.

Miller, G. A. 1995. WordNet: A Lexical Database for English. *CACM* 38(11):39–41.

Mitchell, T. M. 2005. Reading the Web: A Breakthrough Goal for AI. *AI Magazine*.

Muggleton, S. H. 1991. Inductive Logic Programming. *New Generation Computing* 8(4):295–318.

Page, D., and Srinivasan, A. 2003. ILP: A Short Look Back and a Longer Look Forward. *JMLR* 4:415–430.

Punyakanok, V.; Roth, D.; Yih, W.; and Zimak, D. 2005. Learning and Inference over Constrained Output. In *Proc. of IJCAI'05*, 1124–1129.

Ramshaw, L. A., and Marcus, M. P. 1995. Text Chunking using Transformation-Based Learning. In *Proc. of WVLC'95*, 82–94.

Roth, D., and Yih, W. 2001. Relational Learning via Propositional Algorithms: An Information Extraction Case Study. In *Proc. of IJCAI'01*, 1257–1263.

Specia, L.; Srinivasan, A.; Ramakrishnan, G.; and das Graças Volpe Nunes, M. 2006. Word Sense Disambiguation Using Inductive Logic Programming. In *Proc. of ILP'06*, 409–423.

Srinivasan, A. 2004. The Aleph Manual. *Computing Laboratory, Oxford Univ., U.K.* <http://web2.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph>.

Stork, D. G. 1999. The Open Mind Initiative. *IEEE Expert Systems and Their Applications* 14(3):16–20.

Stracuzzi, D. J. 2005. Scalable Knowledge Acquisition Through Memory Organization. In *Proc. of AKRR'05*, 57–64.

Valiant, L. G. 1984. A Theory of the Learnable. *CACM* 27(11):1134–1142.

Valiant, L. G. 2000. Robust Logics. *AIJ* 117(2):231–253.

Valiant, L. G. 2006. Knowledge Infusion. In *Proc. of AAAI'06*, 1546–1551.