Scaling Textual Inference to the Web

Stefan Schoenmackers, Oren Etzioni, and Daniel S. Weld
Turing Center
University of Washington
Computer Science and Engineering
Box 352350
Seattle, WA 98195, USA
stef,etzioni,weld@cs.washington.edu

Abstract

Most Web-based Q/A systems work by finding pages that contain an *explicit* answer to a question. These systems are helpless if the answer has to be inferred from multiple sentences, possibly on different pages. To solve this problem, we introduce the HOLMES system, which utilizes *textual inference* (TI) over tuples extracted from text.

Whereas previous work on TI (*e.g.*, the literature on textual entailment) has been applied to paragraph-sized texts, HOLMES utilizes knowledge-based model construction to scale TI to a corpus of 117 million Web pages. Given only a few minutes, HOLMES doubles recall for example queries in three disparate domains (geography, business, and nutrition). Importantly, HOLMES's runtime is *linear* in the size of its input corpus due to a surprising property of many textual relations in the Web corpus—they are "approximately" functional in a well-defined sense.

1 Introduction and Motivation

Numerous researchers have identified the Web as a rich source of answers to factual questions, *e.g.*, (Kwok et al., 2001; Brill et al., 2002), but often the desired information is not stated *explicitly* even in a textual corpus as massive as the Web. Consider the question "What vegetables help prevent osteoporosis?" Since there is likely no sentence on the Web directly stating "Kale prevents osteoporosis", a system must *infer* that kale is an answer by combining facts from multiple sentences, possibly from different pages, which justify that conclusion: *i.e.*, that kale is a vegetable, kale contains calcium, and calcium helps prevent osteoporosis.

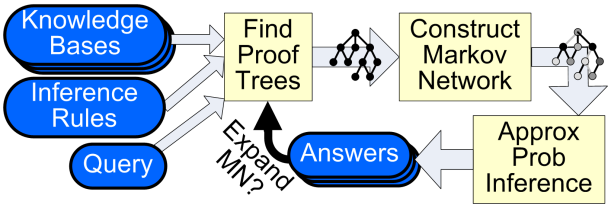

Figure 1: The architecture of HOLMES.

Textual Inference (TI) methods have advanced in recent years. For example, textual entailment techniques aim to determine whether one textual fragment (the *hypothesis*) follows from another (the *text*) (Dagan et al., 2005). While most TI researchers have focused on high-quality inferences from a small source text, we seek to utilize sizable chunks of the Web corpus as our source text. In order to do this, we must confront two major challenges. The first is uncertainty: TI is an imperfect process, particularly when applied to the Web corpus, hence probabilistic methods help to assess the confidence in inferences. The second challenge is scalability: how does inference time scale given increasingly large corpora as input?

1.1 HOLMES: A Scalable TI System

This paper describes HOLMES, an implemented system, which addresses both challenges by carrying out scalable, probabilistic inference over ground assertions extracted from the Web. The input to HOLMES is a conjunctive query, a set of inference rules expressed as Horn clauses, and large sets of ground assertions extracted from the Web, WordNet, and other knowledge bases. As shown in Figure 1, HOLMES chains backward from the query, using the inference rules to construct a forest of proof trees from the ground assertions. This forest is converted

into a Markov network (a form of Knowledge-Based Model Construction (KBMC) (Wellman et al., 1992)) and evaluated using approximate probabilistic inference. HOLMES operates in an *anytime* fashion — if desired it can keep iterating: searching for more proofs, and elaborating the Markov network.

HOLMES makes some important simplifying assumptions. Specifically, we use simple ground tuples to represent extracted assertions (*e.g.*, `contains(kale, calcium)`). Syntactic problems (*e.g.*, anaphora, relative clauses) and semantic challenges (*e.g.*, quantification, counterfactuals, temporal qualification) are delegated to the extraction system or simply ignored. This paper focuses on scalability for this subset of the TI task.

1.2 Summary of Experimental Results

We tested HOLMES on 183 million distinct ground assertions extracted from the Web by the TEXTRUNNER system (Banko et al., 2007), coupled with 159 thousand ground assertions from WordNet (Miller et al., 1990), and a compact set of hand-coded inference rules. Given a total of 55 to 145 seconds, HOLMES was able to produce high-quality inferences that doubled the number of answers to example queries in three disparate domains: geography, business, and nutrition.

We also evaluated how the speed of HOLMES scaled with the size of its input corpus. In the general case, logical inference over a Horn theory (needed in order to produce the probabilistic network) is polynomial in the number of ground assertions, and hence in the size of the textual corpus.[1] Unfortunately, this is prohibitive, since even low-order polynomial growth is fatal on a 117 million-page corpus, let alone the full Web.

1.3 Why HOLMES Scales Linearly

Fortunately, the Web's long tail works in our favor. The relations we extract from text are *approximately pseudo-functional* (APF), as we formalize in Section 3, and this property leads to runtime that scales *linearly* with the corpus. To see the underlying intuition, consider the APF relation denoted by the phrase "is married to;" most of the time it maps a person's name to a small number of spousal names so this relation is APF. Section 3 shows why this APF property ensures linear scaling, and Section 4 demonstrates linear scaling in practice.

[1] In fact, it is P-complete — as hard as any polynomial-time problem.

2 An Overview of HOLMES

HOLMES is a system designed to answer complex queries over large, noisy knowledge bases. As a motivating example, we consider the question "What vegetables help prevent osteoporosis?" As of this writing, Google has no pages explicitly stating 'kale helps prevent osteoporosis', making it challenging to return "kale" as an answer. However, there are numerous web pages stating that "kale is high in calcium" and others declaring that "calcium helps prevent osteoporosis". If we could combine those facts we could easily infer that "kale" is an answer to the question "What vegetables help prevent osteoporosis?" HOLMES was designed to make such inferences while accounting for uncertainty in the process.

Given a query, expressed as a conjunctive Datalog rule, HOLMES generates a probabilistic model using knowledge-based model construction (KBMC) (Wellman et al., 1992). Specifically, HOLMES utilizes fast, logical inference to find the subset of ground assertions and inference rules that may influence the answers to the query — enabling the construction of a small and focused Markov network. Since this graphical model is much smaller than one incorporating *all* ground assertions, probabilistic inference will be much faster than if naive compilation were used.

Figure 1 summarizes the operation of HOLMES. As with many theorem provers or KBMC systems, HOLMES takes three inputs:

1. A set of knowledge bases — databases of ground relational assertions, each with an estimate of its probability, which can be generated by TextRunner (Banko et al., 2007) or Kylin (Wu and Weld, 2007). In our example, we would extract the assertions `IsHighIn(kale, calcium)` and `Prevents(calcium, osteoporosis)` from those sentences.
2. A domain theory – A set of probabilistic inference rules written as Markov logic Horn clauses, which can be used to derive new assertions. The weight associated with each clause specifies its reliability.

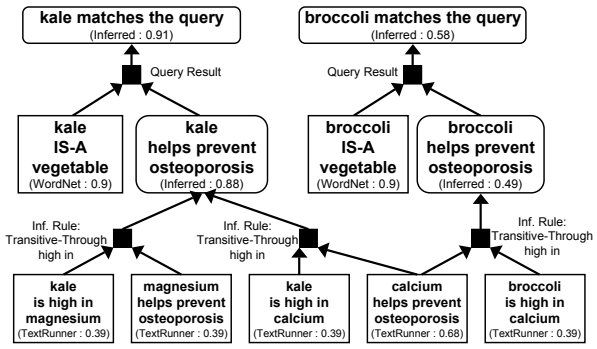

Figure 2: Partial proof 'tree' (DAG) for the query "What vegetables help prevent osteoporosis?" Rectangles depict ground assertions from a knowledge base, rounded boxes are inferred assertions, and shaded squared represent the application of inference rules. HOLMES converts this DAG into a Markov network in order to estimate the probability of each node.

In Section 2.3 we identify several domain-independent rules, but a user may (optionally) specify additional, domain-specific rules if desired. In our example, we assume we are given the domain-specific rule: `Prevents(X,Z) :- IsHighIn(X,Y) ∧ Prevents(Y,Z)`

3. A conjunctive query is specified as a Datalog rule. For example, the question "What vegetables help prevent osteoporosis?" could be written as: `query(X) :- IS-A(X,Vegetable) ∧ Prevents(X,osteoporosis)`

and returns a set of answers to the query, each with an associated probability.

2.1 Basic Operation

To find these answers and their associated probabilities, HOLMES first finds all ground assertions in the knowledge bases that are potentially relevant to the query. This is efficiently done using the inference rules to chain backwards from the query. Note that the generated candidate answers, themselves, are less important than the associated proof trees. Furthermore, since HOLMES uses these 'trees' (actually, DAGs) to generate a probabilistic graphical model, HOLMES seeks to find as many proof trees as possible for each query result — each may influence the final belief in that result. Figure 2 shows a partial proof tree for our example query.

To handle uncertainty, HOLMES now constructs a ground Markov network from the proof trees and the Markov-logic-encoded inference rules. Markov networks (Pearl, 1988) model the joint distribution of a set of variables by creating an undirected graph with one node for each random variable, and representing dependencies between variables with cliques in the graph. Each clique has a corresponding potential function ϕ_k, which returns a non-negative value based on the state of variables in the clique. The probability of a state, x, is given by

$$P(x) = \frac{1}{Z} \prod \phi_k(x_{\{k\}})$$

where the partition function Z is a normalizing term, and $x_{\{k\}}$ denotes the state of all the variables in clique k.

HOLMES converts the proof trees into a Markov network in a manner pioneered by the Markov Logic framework of Richardson and Domingos (2006). A Boolean variable is created to represent the truth of each assertion in the proof forest. Next, HOLMES adds edges to the Markov network to create a clique corresponding to each application of an inference rule in the proof forest.

Following the Markov Logic framework, the potential function of a clique has form $\phi(x) = e^w$ if all member nodes are true (w denotes the weight of the inference rule), and $\phi(x) = 1$ otherwise. The probabilities of leaf nodes are derived from the underlying knowledge base,[2] and inferred nodes are biased with an exponential prior.

Finally, HOLMES computes the approximate probability of each answer by running a variant of loopy belief propagation (Pearl, 1988) over the Markov network. In our experience this method performs well on networks derived from our Horn clause proof forest, but one could use Monte Carlo techniques or even exact methods if desired.

Note that this architecture allows HOLMES to combine information from *multiple* web pages to infer assertions not explicitly seen in the textual corpus. Because this inference is done using a Markov network, it correctly handles uncertain extractions and probabilistic dependencies. By using KBMC to create a custom, focused network for each query, the

[2] In our experiments, ground assertions from WordNet get a uniformly high probability of correctness (0.9), but those extracted from the Web are assigned probabilities derived from redundancy statistics, following the intuition that frequently extracted facts are more likely to be true (Etzioni et al., 2005).

amount of probabilistic inference is reduced to manageable proportions.

2.2 Anytime, Incremental Expansion

Because exact probabilistic inference is #P-complete, HOLMES uses approximate methods, but even these techniques have problems if the Markov network gets too large. As a result, HOLMES creates the network incrementally. After the first proof trees are generated, HOLMES creates the model and performs approximate probabilistic inference. If more time is available then HOLMES searches for additional proof trees and updates the network (Figure 1). This incremental process allows HOLMES to return initial results (with preliminary probability estimates) as soon as they are discovered.

For efficiency, HOLMES exploits standard Datalog optimizations (*e.g.*, it only expands proofs of recently added nodes and it uses an approximation to magic sets (Ullman, 1989), rather than simple backwards chaining). For tractability, we also allow the user to limit the number of transitive inference steps for any inference rule.

HOLMES also includes a few enhancements for dealing with information extracted from natural language. For example, HOLMES's inference rules support substring/regex matching of ground assertions, to accommodate simple variations in text. HOLMES also can be restricted to only operate over proper nouns, which is useful for queries involving named entities.

2.3 Markov Logic Inference Rules

HOLMES is given the following set of six domain-independent rules, which are similar to the upward monotone rules introduced by (MacCartney and Manning, 2007).

1. Observed relations are likely to be true:
   `R(X,Y) :- ObservedInCorpus(X, R, Y)`
2. Synonym substitution preserves meaning:
   R_{TR}`(X',Y) :-` R_{TR}`(X,Y)` $\wedge$ `Synonym(X, X')`
3. R_{TR}`(X,Y') :-` R_{TR}`(X,Y)` $\wedge$ `Synonym(Y, Y')`
4. Generalizations preserve meaning:
   R_{TR}`(X',Y) :-` R_{TR}`(X,Y)` $\wedge$ `IS-A(X, X')`
5. R_{TR}`(X,Y') :-` R_{TR}`(X,Y)` $\wedge$ `IS-A(Y, Y')`
6. Transitivity of Part Meronyms:
   R_{TR}`(X,Y') :-` R_{TR}`(X,Y)` $\wedge$ `Part-Of(Y, Y')`
   where R_{TR} matches '* in' (*e.g.*, 'born in').

For example, if `Q(X):-BornIn(X,'France')`, and we know from WordNet that Paris is in France, then by inference rule 6, we know that `BornIn(X,'Paris')` will yield valid results for `Q(X)`. Although all of these rules contain at most two relations in the body, HOLMES allows an arbitrary number of relations in the query and rule bodies. However, we have found that even simple rules can dramatically improve some queries.

We set the rule weights to capture the intuition that deeper inferences decrease the likelihood (as there are more chances to make mistakes), whereas additional, independent proof trees increase the likelihood (as there is more supporting evidence). Specifically, in our experiments we set the prior on inferred facts to -0.75, the weight on rule 1 to 1.5, and the weights on all other rules to 0.6.

At present, we define these weights manually, but we expect to learn the parameter values in the future.

3 Scaling Inference to the Web

If TI is applied to a corpus containing hundreds of millions or even billions of pages, its run time has to be at most linear in the size of the corpus. This section shows that under some reasonable assumptions inference *does* scale linearly.

We start our analysis with two simplifications. First, we assume that the number of distinct, ground assertions in the KBs, $|A|$, grows at most linearly with the size of the textual corpus. This is certainly true for assertions extracted by TextRunner and Kylin, and follows from our exclusion of texts with complex quantified sentences. Our analysis now proceeds to consider scaling with respect to $|A|$ for a fixed query and set of inference rules.

Our second assumption is that the size of every proof tree is bounded by some constant, m. This is a strong assumption and one that depends on the precise set of inference rules and pattern of ground assertions. However, it holds in our experience, and if necessary could be enforced by terminating the search for proof trees at a certain depth, *e.g.*, $\log(m)$.

HOLMES's knowledge-based model construction has two parts: construction of the proof forest and conversion of the forest into a Markov network. Since the Markov network is essentially isomorphic to the proof forest, the conversion will be $\mathcal{O}(|A|)$ if the forest is linear in size, which is ensured if the time to construct the proof trees is $\mathcal{O}(|A|)$. We show

this in the remainder of this section.

Recall that HOLMES requires inference rules to be function-free Horn clauses. While this limits expressivity to some degree, it provides a huge speed benefit — logical inference over Horn clauses can be done in polynomial time, whereas general propositional inference (*i.e.*, from grounded first-order rules) is NP-complete.

Alas, even low-order polynomial blowup is unacceptable when the textual corpus reaches Web scale; we seek linear growth. Intuitively, there are two places where polynomial expansion could cause trouble. First, the number of different *types* of proofs (*i.e.*, first order proofs) could grow too quickly, and secondly, a given type of proof tree might apply to too many ground assertions ("tuples" in database lingo). We treat these issues in turn.

Under our assumptions, each proof tree can be represented as an expression in relational algebra with at most m equijoins (Ullman, 1989),[3] each stemming from the application of an inference rule. Since the number of rules is fixed, as is m, there are a constant number of possible first-order proof trees.

The bigger concern is that any one of these first-order trees might result in a polynomial number of ground trees; if so, the size of the ground forest (and corresponding Markov network) could grow too quickly. In fact, polynomial growth is a common phenomena in database query evaluation. Luckily, most relations in the Web corpus behave more favorably. We introduce a property of relations that ensures m-way joins, and therefore all proof trees up to size m, can be computed in $\mathcal{O}(|A|)$ time.

The intuition is that most relations derived from large corpora have a 'heavy-tailed' distribution, wherein a few objects appear many times in a relation, but most appear only once or twice, thus joins involving rare objects lead to a small number of results, and so the main limitation on scalability is common objects. We now prove that if these common objects account for a small enough fraction of the relation, then joins will still scale linearly. We focus on binary relations, but these results can easily be extended to relations of larger arity.

[3]Note that an inference rule of the form `H(X) :- R1(X,Y),R2(Y,Z)` is equivalent to the algebraic expression $\pi_X(R_1 \bowtie R_2)$. First a join is performed between R_1 and R_2 testing for equality between values of Y; then a projection eliminates all columns besides X.

Definition 1 *A relation, $R = \{(x_i, y_i)\} \subseteq \mathcal{X} \times \mathcal{Y}$, is* pseudo-functional *(PF)* in x with degree k, *if* $\forall x \in \mathcal{X} : |\{y|(x,y) \in R\}| \leq k$. *When the precise variable and degree is irrelevant to discussion, we simply say "R is PF."*

An m-way equijoin over relations that are PF in the join variables will have at most $k^m * |R|$ results. Since k^m is constant for a given join and $|R|$ scales linearly in the size of the textual corpus, proof tree construction over PF relations also scales linearly.

However, due to their heavy-tailed distributions, most relations extracted from the Web fit the pseudo-functional definition in most, but not all values of $\mathcal{X}$. Fortunately, it turns out that in most cases these "bad" values of $\mathcal{X}$ are rare and hence don't influence the join size significantly. We formalize this intuition by defining a class of approximately pseudo-functional (APF) relations and proving that joining two APF relations produces at most a linear number of results.

Definition 2 *A relation, R, is* approximately pseudo-functional *(APF)* in x with degree k, *if $\mathcal{X}$ can be partitioned into two sets $\mathcal{X}_G$ and $\mathcal{X}_B$ such that for all $x \in \mathcal{X}_G$ R is PF with degree k and*

$$\sum_{x \in \mathcal{X}_B} |\{y|(x,y) \in R\}| \leq k * log(|R|)$$

Theorem 1. *If relation R_1 is APF in y with degree k_1 and R_2 is APF in y with degree k_2 then the relation $Q = R_1 \bowtie R_2$ has size at most $\mathcal{O}(max(|R_1|, |R_2|))$.*

Proof. Since R_1 and R_2 are APF, we know that $\mathcal{Y}$ can be partitioned into four groups: $\mathcal{Y}_{BB} = \mathcal{Y}_{B1} \bigcap \mathcal{Y}_{B2}$, $\mathcal{Y}_{BG} = \mathcal{Y}_{B1} \bigcap \mathcal{Y}_{G2}$, $\mathcal{Y}_{GB} = \mathcal{Y}_{G1} \bigcap \mathcal{Y}_{B2}$, $\mathcal{Y}_{GG} = \mathcal{Y}_{G1} \bigcap \mathcal{Y}_{G2}$.[4] We can show that each group leads to at most $\mathcal{O}(|A|)$ entries in Q. For $y \in \mathcal{Y}_{BB}$ there are at most $k_1 * k_2 * log(|R_1|) * log(|R_2|)$ entries in Q. The $y \in \mathcal{Y}_{GB}$ and $y \in \mathcal{Y}_{BG}$ lead to at most $k_1 * k_2 * log(|R_2|)$ and $k_1 * k_2 * log(|R_1|)$ entries, respectively. For $y \in \mathcal{Y}_{GG}$ there are at most $k_1 * k_2 * max(|R_1|, |R_2|)$. Summing the results from the four partitions, we see that $|Q|$ is $\mathcal{O}(max(|R_1|, |R_2|))$, thus it is $\mathcal{O}(|A|)$. □

This theorem and proof can easily be extended to

[4]$\mathcal{Y}_{BB}$ are the "doubly bad" values of y that violate the PF definition for both relations, $\mathcal{Y}_{GG}$ are the values that do not violate the PF definition for either relation, and $\mathcal{Y}_{BG}$ and $\mathcal{Y}_{GB}$ are the values that violate it in only R_1 or R_2, resp.

an m-way equijoin, as long as each relation is APF in all arguments that are being joined.

Theorem 2. *If Q is the relation obtained by an equijoin over m relations $R_{1..m}$, each having size at most $\mathcal{O}(|A|)$, and if all $R_{1..m}$ are APF in all arguments that they are joined in with degree at most k_{max}, and if $\prod_{1 \leq i \leq m} log(|R_i|) \leq |A|$, then $|Q|$ is $\mathcal{O}(|A|)$.*

The inequality in Theorem 2 relates the sizes of the relations ($|R|$), the join (m) and the number of ground assertions ($|A|$). However, in many cases we are interested in much smaller values of m than the inequality enables. We can relax the APF definition to allow a broader, but still scalable, class of m-way-APF relations.

Corollary 3. *If Q is the relation obtained by an m-way join, and if each participating relation is APF in their joined variables with a bound of $k_i * \sqrt[m]{|R_i|}$ instead of $k_i * log(|R_i|)$, then the join is $\mathcal{O}(|A|)$.*

The final step in our scaling argument concerns probabilistic inference, which is #P-Complete if performed exactly. This is addressed in two ways. First, HOLMES uses approximate methods, *e.g.*, loopy belief propagation, which avoids the cost of exact inference — at the cost of reduced precision. Secondly, at a practical level, HOLMES's incremental construction of the graphical model (Figure 1) allows it to bound the size of the network by terminating the search for additional proofs.

4 Experimental Results

This section reports on measurements that confirm that linear scaling with $|A|$ occurs in practice, and that HOLMES's inference is not only scalable but also improves precision/recall on sample queries in a diverse set of domains. After describing the experimental domains and queries, Section 4.2 reports on the boost to the area under the precision/recall curve for a set of example queries in three domains: geography, business, and nutrition. Section 4.3 then shows that APF relations are very common in the Web corpus, and finally Section 4.4 demonstrates empirically that HOLMES's inference time scales linearly with the number of pages in the corpus.

4.1 Experimental Setup

HOLMES utilized two knowledge bases in these experiments: TEXTRUNNER and WordNet. TEXTRUNNER contains approximately 183 million distinct ground assertions extracted from over 117 million web pages, and WordNet contains 159 thousand manually created IS-A, Part-Of, and Synonym assertions.

In all queries, HOLMES utilizes the domain-independent inference rules described in Section 2.3. HOLMES additionally makes use of two domain-specific inference rules in the Nutrition domain, to demonstrate the benefits of including domain-specific information. Estimating the precision and relative recall of HOLMES requires extensive and careful manual tagging of HOLMES output. To make this feasible, we restricted ourselves to a set of twenty queries in three domains, but made the domains diverse to illustrate the broad scope of the system.

We now describe each domain briefly.

Geography: the query issued is: "Who was born in one of the following countries?" More formally, `Q(X) :- BornIn(X,{country})` where {country} is bound to each of the following nine countries in turn {France, Germany, China, Thailand, Kenya, Morocco, Peru, Columbia, Guatemala}, yielding a total of nine queries.

Because Web text often refers to a person's birth city rather than birth country, this query illustrates how combining an ground assertion (*e.g.*, `BornIn(Alberto Fujimori, Lima)`) with background knowledge (*e.g.*, `LocatedIn(Lima, Peru)`) enables the system to draw new conclusions (*e.g.*, `BornIn(Alberto Fujimori, Peru)`).

Business: we issued the following two queries.

1) Which companies are acquiring software companies? Formally, `Q(X) :- Acquired(X, Y) ∧ Develops(Y, 'software')` This query tests HOLMES's ability to scalably join a large number of assertions from multiple pages.

2) Which companies are headquartered in the USA? `Q(X) :- HeadquarteredIn(X, 'USA') ∧ IS-A(X, 'company')`

Answering this query comprehensively requires HOLMES to combine a join (over the relations HeadquarteredIn and IS-A) with transitive inference on PartOf (*e.g.*, Seattle is PartOf Washington which is PartOf the USA) and on IS-A (*e.g.*, Microsoft IS-A software company which IS-A company). The IS-A assertions came from both TEXTRUNNER (using patterns from (Hearst, 1992)) and WordNet.

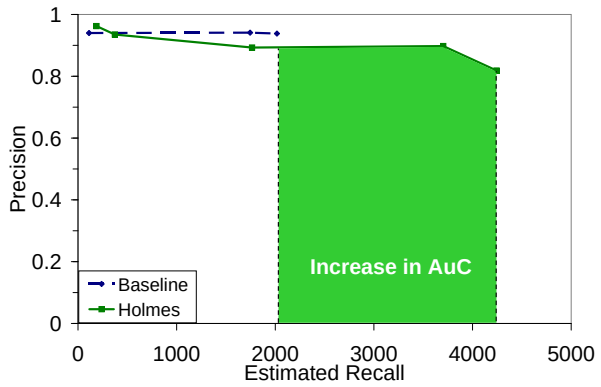

Figure 3: PR Curve for BornIn(X, {country}). Inference boosts the Area under the PR Curve (AuC) by 102 %.

Domain	Increase in AuC	Total Inference Time
Geography	+102%	55 s
Business	+2,643%	145 s
Nutrition	+5,595%	64 s

Table 1: Improvement in the AuC of HOLMES over the BASELINE and total inference time taken by HOLMES. Results are summed over all queries in the geography, business, and nutrition domains. Inference time measured on unoptimized prototype.

Nutrition: the nine queries issued are instances of "What foods prevent disease?" Where a food is a member of one of the classes: fruit, vegetable, or grain, and a disease is one of: anemia, scurvy, or osteoporosis. More formally, `Q(X, {disease}) :- Prevents(X, {disease}) ∧ IS-A(X, {food})`

Our experiments in the nutrition domain utilized two domain-specific inference rules in addition to the ones presented in Section 2.3:

`Prevents(X,Y):-HighIn(X,Z) ∧ Prevents(Z,Y)`

`Prevents(X,Y):-Contains(X,Z) ∧ Prevents(Z,Y)`

4.2 Effect of Inference on Recall

To measure the cost and benefit of HOLMES's inference we need to define a baseline for comparison. Answering the conjunctive queries in the business and nutrition domains requires computing joins, which TEXTRUNNER does not do. Thus, we defined a baseline system, BASELINE, which has access to the underlying Knowledge Bases (KBs) (TEXTRUNNER and WordNet), and the ability to compute joins using information explicitly stated in either KB, but does *not* have the ability to infer new assertions.

We compared HOLMES with BASELINE in all three domains. Figure 3 depicts the combined precision/relative recall curves for the nine Geography queries. HOLMES yields substantially higher recall (the shaded region) at modestly lower precision, doubling the area under the precision/recall curve (AuC). The other precision/recall curves also showed a slight drop in precision for substantial gains in recall. Table 1 summarizes the results, along with the total runtime needed for inference. Because relations in the business domain are much larger than in the other domains (*i.e.*, 100x ground assertions), inference is slower in this domain.

We note that inference is particularly helpful with rarely mentioned instances. However, inference can lead to errors when the proof tree contains joins on generic terms (*e.g.*, "company") or common extraction errors (*e.g.*, "LLC" as a company name). This is a key area for future work.

4.3 Prevalence of APF Relations

To determine the prevalence of APF relations in Web text, we examined a sample of 500 binary relations selected randomly from TEXTRUNNER's ground assertions. The surface forms of the relations and arguments may misrepresent the true properties of the underlying concepts, so to better estimate the true properties we merged synonymous values as given by Resolver (Yates and Etzioni, 2007) or the most frequent sense of the word in WordNet. For example, we would consider `BornIn(baby, hospital)` and `BornAt(infant, infirmary)` to represent the same concept, and so would merge them into one instance of the 'Born In' relation. The largest two relations had over 1.25 million unique instances each, and 52% of the relations had more than 10,000 instances.

For each relation R, we first found all instances of R extracted by TEXTRUNNER and merged all synonymous instances as described above. Then, for each argument of R we computed the smallest value, K_{min}, such that R is APF with degree K_{min}. Since many interesting assertions can be inferred by simply joining two relations, we also considered the special case of 2-way joins using Corollary 3. We computed the smallest value, $K_{2\bowtie}$, such that the relation is two-way-APF with degree $K_{2\bowtie}$.

Figure 4 shows the fraction of relations with K_{min} and $K_{2\bowtie}$ of at most K as a function of varying

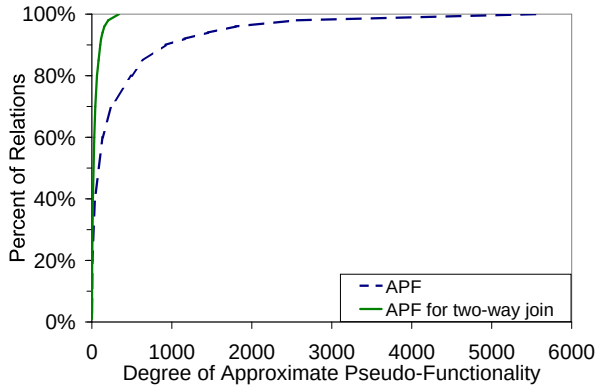

Figure 4: Prevalence of APF relations in Web text. The x-axis depicts the *degree* of pseudo-functionality, *e.g.*, K_{min} and $K_{2\bowtie}$, (see definition 2); the y-axis lists the percent of relations that are APF with that degree. Results are averaged over both arguments.

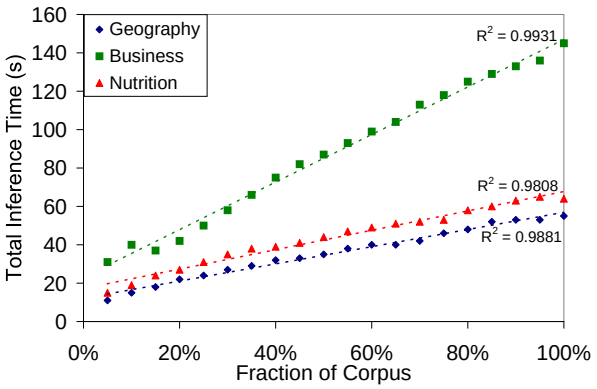

Figure 5: The effects of corpus size on total inference time. We see approximately linear growth in all domains, and display the best fit lines and coefficient of determination (R^2) of each.

values of K. The results are averaged over both arguments of each binary relation. For arbitrary joins in this KB, 80% of the relations are APF with degree less than 496; for 2-way joins (like the ones in our inference rules and test queries), 80% of the relations are APF with degree less than 65. These results indicate that the majority of relations TEXTRUNNER extracted from text are APF, and so we can expect HOLMES's techniques will allow efficient inference over most relations.

While Theorem 2 guarantees that joins over those relations will be $\mathcal{O}(|R|)$, that notation hides a potentially large constant factor of ${K_{min}}^m$. Fortunately the constant factor is significantly smaller in practice. To see why, we re-examine the proof: the large factor comes from assuming that *all* of R's first arguments which meet the PF definition are associated with exactly K_{min} distinct second arguments. However, in our corpus 83% of first arguments are associated with only *one* second argument. Clearly, our worst-case analysis substantially over-estimates inference time for most queries. Moreover, in additional experiments (omitted due to space limitations), measured join sizes grew linearly in the size of the corpus, but were on average two to three orders of magnitude smaller than the bounds given in the theory. This observation held across relations with different sizes and values of K_{min}.

While the results in Figure 4 may vary for other sets of relations, we believe the general trends hold. This is promising for Question Answering and Textual Inference systems, since if true it implies that combining information from multiple difference source is feasible, and can allow such systems to infer answers not explicitly seen in any source.

4.4 Scalability of Inference Speed

Since the previous subsection showed that most relations are APF in their arguments, our theory predicts HOLMES's inference will scale linearly. We tested this hypothesis empirically by running inference over the test queries in our three domains, while varying the number of pages in the textual corpus.

Figure 5 shows how the inference time HOLMES used to answer all queries in each domain scales with KB size. For these queries, and several others we tested (not shown here), inference time grows linearly with the size of the KB. Based on these results we believe that HOLMES can provide scalable inference over a wide variety of domains.

5 Related Work

Textual Entailment systems are given two textual fragments, text T and hypothesis H, and attempt to decide if the meaning of H can be inferred from the meaning of T (Dagan et al., 2005). While many approaches have addressed this problem, our work is most closely related to that of (Raina et al., 2005; MacCartney and Manning, 2007; Tatu and Moldovan, 2006; Braz et al., 2005), which convert the inputs into logical forms and then attempt to 'prove' H from T plus a set of axioms. For instance, (Braz et al., 2005) represents T, H, and a set of rewrite rules in a description logic framework, and determines entailment by solving an integer lin-

ear program derived from that representation.

These approaches and related ones (*e.g.*, (Van Durme and Schubert, 2008)) use highly expressive representations, enabling them to express negation, temporal information, and more. HOLMES's representation is much simpler—Markov Logic Horn Clauses for inference rules coupled with a massive database of ground assertions. However, this simplification allows HOLMES to tackle a "text" of enormously larger size: 117 million Web pages versus a single paragraph. A second, if smaller, difference stems from the fact that instead of determining whether a single hypothesis sentence, H, follows from the text, HOLMES tries to find all consequents that match a conjunctive query.

HOLMES is also related to open-domain question-answering systems such as Mulder (Kwok et al., 2001), AskMSR (Brill et al., 2002), and others (Harabagiu et al., 2000; Brill et al., 2001). However, these Q/A systems attempt to find individual documents or sentences containing the answer. They often perform deep analysis on promising texts, and back off to shallower, less reliable methods if those fail. In contrast, HOLMES utilizes TI and attempts to combine information from multiple different sentences in a scalable way.

While its ability to combine information from multiple sources is promising, HOLMES has several limitations these Q/A systems do not have. Since HOLMES relies on an information extraction system to convert sentences into ground predicates, any limitations of the IE system will be propagated to HOLMES. Additionally, the logical representation HOLMES uses limits the reasoning and types of questions it can answer. HOLMES is geared towards answering questions which are naturally expressed as properties and relations of entities, and is not well suited to answering more abstract or open ended questions. Although we have demonstrated that HOLMES is scalable, further work is needed to make it to run at interactive speeds.

Finally, research in statistical relational learning such as MLNs (Richardson and Domingos, 2006), RMNs (Taskar et al., 2002), and others (Getoor and Taskar, 2007) have studied techniques for combining logical and probabilistic inference. Our inference rules are more restrictive than those allowed in MLNs, but this trade-off allows us to efficiently scale inference to large, open domain corpora. By constructing only cliques for satisfied inference rules, HOLMES explicitly models the intuition behind LazySAT inference (Singla and Domingos, 2006) as used in MLNs. *I.e.*, most Horn clause inference rules will be trivially satisfied since their antecedents will be false, so we only need to worry about ones where the antecedent is true.

6 Conclusions

This paper makes three main contributions:

1. We introduce and evaluate the HOLMES system, which leverages KBMC methods in order to scale a class of TI methods to the Web.

2. We define the notion of *Approximately Pseudo-Functional* (APF) relations and prove that, for a APF relations, HOLMES's inference time increases *linearly* with the size of the input corpus. We show empirically that APF relations appear to be prevalent in our Web corpus (Figure 4), and that HOLMES's runtime does scale linearly with the size of its input (Figure 5), taking only a few CPU minutes when run over 183 million distinct ground assertions.

3. We present experiments demonstrating that, for a set of queries in the domains of geography, business, and nutrition, HOLMES substantially improves the quality of answers (measured by AuC) relative to a "no inference" baseline.

In the future, we plan more extensive tests to characterize when HOLMES's inference is helpful. We also hope to examine in what cases jointly performing extraction and inference (as opposed to performing them separately) is feasible at scale. Finally, we plan to examine methods for HOLMES to learn both rule weights and new inference rules.

Acknowledgements

We thank the following for helpful comments on previous drafts: Fei Wu, Michele Banko, Mausam, Doug Downey, and Alan Ritter. This research was supported in part by NSF grants IIS-0535284, IIS-0312988, and IIS-0307906, ONR grants N00014-08-1-0431 and N00014-06-1-0147, CALO grant 03-000225, the WRF / TJ Cable Professorship as well as gifts from Google. The work was performed at the University of Washington's Turing Center.

References

M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. 2007. Open information extraction from the Web. In *Procs. of IJCAI*.

R. Braz, R. Girju, V. Punyakanok, D. Roth, and M. Sammons. 2005. An inference model for semantic entailment in natural language. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 1678–1679.

E. Brill, J. Lin, M. Banko, S. T. Dumais, and A. Y. Ng. 2001. Data-intensive question answering. In *Procs. of Text REtrieval Conference (TREC-10)*, pages 393–400.

Eric Brill, Susan Dumais, and Michele Banko. 2002. An analysis of the AskMSR question-answering system. In *EMNLP '02: Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, pages 257–264, Morristown, NJ, USA. Association for Computational Linguistics.

I. Dagan, O. Glickman, and B. Magnini. 2005. The PASCAL Recognising Textual Entailment Challenge. *Proceedings of the PASCAL Challenges Workshop on Recognising Textual Entailment*, pages 1–8.

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

L. Getoor and B. Taskar. 2007. *Introduction to Statistical Relational Learning*. MIT Press.

S. Harabagiu, M. Pasca, and S. Maiorano. 2000. Experiments with open-domain textual question answering. In *Procs. of the COLING-2000.*

M. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Procs. of the 14th International Conference on Computational Linguistics*, pages 539–545, Nantes, France.

C.C.T. Kwok, O. Etzioni, and D.S. Weld. 2001. Scaling question answering to the Web. *Proceedings of the 10th international conference on World Wide Web*, pages 150–161.

B. MacCartney and C.D. Manning. 2007. Natural Logic for Textual Inference. In *Workshop on Textual Entailment and Paraphrasing*.

G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. 1990. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–312.

Judea Pearl. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA.

Rajat Raina, Andrew Y. Ng, and Christopher D. Manning. 2005. Robust textual inference via learning and abductive reasoning. In *Proceedings of AAAI 2005*. AAAI Press.

M. Richardson and P. Domingos. 2006. Markov Logic Networks. *Machine Learning*, 62(1-2):107–136.

Parag Singla and Pedro Domingos. 2006. Memory-efficient inference in relational domains. In *AAAI*.

B. Taskar, P. Abbeel, and D. Koller. 2002. Discriminative probabilistic models for relational data. *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)*.

Marta Tatu and Dan Moldovan. 2006. A logic-based semantic approach to recognizing textual entailment. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 819–826, Morristown, NJ, USA. Association for Computational Linguistics.

J. Ullman. 1989. *Database and knowledge-base systems*. Computer Science Press.

B. Van Durme and L.K. Schubert. 2008. Open knowledge extraction through compositional language processing. In *Symposium on Semantics in Systems for Text Processing*.

M. Wellman, J. Breese, and R. Goldman. 1992. From knowledge bases to decision models. *The Knowledge Engineering Review*, 7(1):35–53.

F. Wu and D. Weld. 2007. Autonomously semantifying Wikipedia. In *Proceedings of the ACM Sixteenth Conference on Information and Knowledge Management (CIKM-07)*, Lisbon, Porgugal.

A. Yates and O. Etzioni. 2007. Unsupervised resolution of objects and relations on the Web. In *Procs. of HLT*.