Domain-Targeted, High Precision Knowledge Extraction

Bhavana Dalvi Mishra **Niket Tandon** **Peter Clark**

Allen Institute for Artificial Intelligence
2157 N Northlake Way Suite 110, Seattle, WA 98103
{bhavanad,nikett,peterc}@allenai.org

Abstract

Our goal is to construct a *domain-targeted, high precision* knowledge base (KB), containing general *(subject,predicate,object)* statements about the world, in support of a downstream question-answering (QA) application. Despite recent advances in information extraction (IE) techniques, no suitable resource for our task already exists; existing resources are either too noisy, too named-entity centric, or too incomplete, and typically have not been constructed with a clear scope or purpose. To address these, we have created a *domain-targeted, high precision knowledge extraction pipeline*, leveraging Open IE, crowdsourcing, and a novel canonical schema learning algorithm (called CASI), that produces high precision knowledge targeted to a particular domain - in our case, elementary science. To measure the KB's coverage of the target domain's knowledge (its "comprehensiveness" with respect to science) we measure recall with respect to an independent corpus of domain text, and show that our pipeline produces output with over 80% precision and 23% recall with respect to that target, a substantially higher coverage of tuple-expressible science knowledge than other comparable resources. We have made the KB publicly available[1].

1 Introduction

While there have been substantial advances in knowledge extraction techniques, the availability of high precision, general knowledge about the world, remains elusive. Specifically, our goal is a large, high precision body of *(subject,predicate,object)* statements relevant to elementary science, to support a downstream QA application task. Although there are several impressive, existing resources that can contribute to our endeavor, e.g., NELL (Carlson et al., 2010), ConceptNet (Speer and Havasi, 2013), WordNet (Fellbaum, 1998), WebChild (Tandon et al., 2014), Yago (Suchanek et al., 2007), FreeBase (Bollacker et al., 2008), and ReVerb-15M (Fader et al., 2011), their applicability is limited by both

- limited coverage of general knowledge (e.g., FreeBase and NELL primarily contain knowledge about Named Entities; WordNet uses only a few (< 10) semantic relations)
- low precision (e.g., many ConceptNet assertions express idiosyncratic rather than general knowledge)

Our goal in this work is to create a *domain-targeted knowledge extraction pipeline* that can overcome these limitations and output a high precision KB of triples relevant to our end task. Our approach leverages existing techniques of open information extraction (Open IE) and crowdsourcing, along with a novel schema learning algorithm.

There are three main contributions of this work. First, we present a high precision extraction pipeline able to extract *(subject,predicate,object)* tuples relevant to a domain with precision in excess of 80%. The input to the pipeline is a corpus, a sense-disambiguated domain vocabulary, and a small set of entity types. The pipeline uses a combination of text filtering, Open IE, Turker annotation on samples, and precision prediction to generate its output.

[1] This KB named as "Aristo Tuple KB" is available for download at http://data.allenai.org/tuple-kb

Transactions of the Association for Computational Linguistics, vol. 5, pp. 233–246, 2017. Action Editor: Patrick Pantel.
Submission batch: 11/2016; Revision batch: 2/2017; Published 7/2017.

Second, we present a novel canonical schema induction method (called CASI) that identifies clusters of similar-meaning predicates, and maps them to the most appropriate general predicate that captures that canonical meaning. Open IE, used in the early part of our pipeline, generates triples containing a large number of predicates (expressed as verbs or verb phrases), but equivalences and generalizations among them are not captured. Synonym dictionaries, paraphrase databases, and verb taxonomies can help identify these relationships, but only partially so because the meaning of a verb often shifts as its subject and object vary, something that these resources do not explicitly model. To address this challenge, we have developed a corpus-driven method that takes into account the subject and object of the verb, and thus can learn argument-specific mapping rules, e.g., the rule "(x:Animal,found in,y:Location) $\rightarrow$ (x:Animal,live in,y:Location)" states that if some animal is found in a location then it also means the animal lives in the location. Note that 'found in' can have very different meaning in the schema "(x:Substance,found in,y:Material). The result is a KB whose general predicates are more richly populated, still with high precision.

Finally, we contribute the science KB itself as a resource publicly available[2] to the research community. To measure how "complete" the KB is with respect to the target domain (elementary science), we use an (independent) corpus of domain text to characterize the target science knowledge, and measure the KB's recall at high ($>$80%) precision over that corpus (its "comprehensiveness" with respect to science). This measure is similar to recall at the point P=80% on the PR curve, except measured against a domain-specific sample of data that reflects the distribution of the target domain knowledge. Comprehensiveness thus gives us an approximate notion of the completeness of the KB for (tuple-expressible) facts in our target domain, something that has been lacking in earlier KB construction research. We show that our KB has comprehensiveness (recall of domain facts at $>$80% precision) of 23% with respect to science, a substantially higher coverage of tuple-expressible science knowledge than other comparable resources. We are making the KB publicly available.

[2]Aristo Tuple KB is available for download at `http://allenai.org/data/aristo-tuple-kb`

Outline

We discuss the related work in Section 2. In Section 3, we describe the domain-targeted pipeline, including how the domain is characterized to the algorithm and the sequence of filters and predictors used. In Section 4, we describe how the relationships between predicates in the domain are identified and the more general predicates further populated. Finally in Section 5, we evaluate our approach, including evaluating its comprehensiveness (high-precision coverage of science knowledge).

2 Related Work

There has been substantial, recent progress in knowledge bases that (primarily) encode knowledge about Named Entities, including Freebase (Bollacker et al., 2008), Knowledge Vault (Dong et al., 2014), DBPedia (Auer et al., 2007), and others that hierarchically organize nouns and named entities, e.g., Yago (Suchanek et al., 2007). While these KBs are rich in facts about named entities, they are sparse in general knowledge about common nouns (e.g., that bears have fur). KBs covering general knowledge have received less attention, although there are some notable exceptions constructed using manual methods, e.g., WordNet (Fellbaum, 1998), crowdsourcing, e.g., ConceptNet (Speer and Havasi, 2013), and, more recently, using automated methods, e.g., WebChild (Tandon et al., 2014). While useful, these resources have been constructed to target only a small set of relations, providing only limited coverage for a domain of interest.

To overcome relation sparseness, the paradigm of Open IE (Banko et al., 2007; Soderland et al., 2013) extracts knowledge from text using an open set of relationships, and has been used to successfully build large-scale *(arg1,relation,arg2)* resources such as ReVerb-15M (containing 15 million general triples) (Fader et al., 2011). Although broad coverage, however, Open IE techniques typically produce noisy output. Our extraction pipeline can be viewed as an extension of the Open IE paradigm: we start with targeted Open IE output, and then apply a sequence of filters to substantially improve the

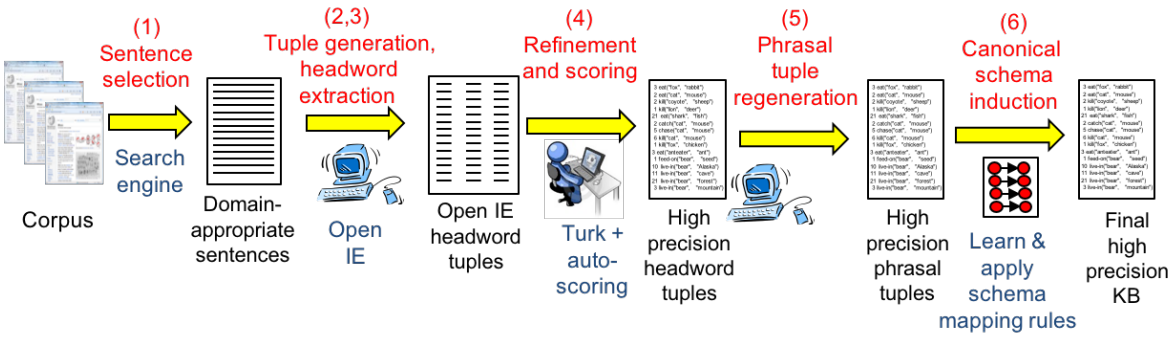

Figure 1: The extraction pipeline. A vocabulary-guided sequence of open information extraction, crowdsourcing, and learning predicate relationships are used to produce high precision tuples relevant to the domain of interest.

output's precision, and learn and apply relationships between predicates.

The task of finding and exploiting relationships between different predicates requires identifying both equivalence between relations (e.g., clustering to find paraphrases), and implication (hierarchical organization of relations). One class of approach is to use existing resources, e.g., verb taxonomies, as a source of verbal relationships, e.g., (Grycner and Weikum, 2014), (Grycner et al., 2015). However, the hierarchical relationship between verbs, out of context, is often unclear, and some verbs, e.g., "have", are ambiguous. To address this, we characterize semantic relationships not only by a verb but also by the types of its arguments. A second class of approach is to induce semantic equivalence from data, e.g., using algorithms such as DIRT (Lin and Pantel, 2001), RESOLVER (Yates and Etzioni, 2009), WiseNet (Moro and Navigli, 2012), and AMIE (Galárraga et al., 2013). These allow relational equivalences to be inferred, but are also noisy. In our pipeline, we combine these two approaches together, by clustering relations using a similarity measure computed from both existing resources and data.

A novel feature of our approach is that we not only cluster the (typed) relations, but also identify a canonical relation that all the other relations in a cluster can be mapped to, without recourse to human annotated training data or a target relational vocabulary (e.g., from Freebase). This makes our problem setting different from that of universal schema (Riedel et al., 2013) where the clusters of relations are not explicitly represented and mapping to canonical relations can be achieved given an existing KB like Freebase. Although no existing methods can be directly applied in our problem setting, the AMIE-based schema clustering method of (Galárraga et al., 2014) can be modified to do this also. We have implemented this modification (called AMIE*, described in Section 5.3), and we use it as a baseline to compare our schema clustering method (CASI) against.

Finally, interactive methods have been used to create common sense knowledge bases, for example ConceptNet (Speer and Havasi, 2013; Liu and Singh, 2004) includes a substantial amount of knowledge manually contributed by people through a Web-based interface, and used in numerous applications (Faaborg and Lieberman, 2006; Dinakar et al., 2012). More recently there has been work on interactive methods (Dalvi et al., 2016; Wolfe et al., 2015; Soderland et al., 2013), which can be seen as a "machine teaching" approach to KB construction. These approaches focus on human-in-the-loop methods to create domain specific knowledge bases. Such approaches are proven to be effective on domains where expert human input is available. In contrast, our goal is to create extraction techniques that need little human supervision, and result in comprehensive coverage of the target domain.

3 The Extraction Pipeline

We first describe the overall extraction pipeline. The pipeline is a chain of filters and transformations, outputting *(subject,predicate,object)* triples at the end. It uses a novel combination of familiar technologies, plus a novel schema learning module, described in

more detail in Section 4.

3.1 Inputs and Outputs

Unlike many prior efforts, our goal is a *domain-focused* KB. To specify the KB's extent and focus, we use two inputs:

1. A **domain vocabulary** listing the nouns and verbs relevant to the domain. In our particular application, the domain is Elementary science, and the domain vocabulary is the typical vocabulary of a Fourth Grader ($\sim$10 year old child), augmented with additional science terms from 4th Grade Science texts, comprising of about 6000 nouns, 2000 verbs, 2000 adjectives, and 600 adverbs.
2. A small **set of types** for the nouns, listing the primary types of entity relevant to the domain. In our domain, we use a manually constructed inventory of 45 types (animal, artifact, body part, measuring instrument, etc.).

In addition, the pipeline also uses:

3. a large, searchable **text corpus** to provide sentences for knowledge extraction. In our case, we use the Web via a search engine (Bing), followed by filters to extract clean sentences from search results.

3.2 Word Senses

Although, in general, nouns are ambiguous, in a targeted domain there is typically a clear, primary sense that can be identified. For example, while in general the word "pig" can refer to an animal, a person, a mold, or a block of metal, in 4th Grade Science it universally refers to an animal[3]. We leverage this for our task by assuming one sense per noun in the domain vocabulary, and notate these senses by manually assigning each noun to one of the entity types in the type inventory.

Verbs are more challenging, because even within a domain they are often polysemous out of context (e.g., "have"). To handle this, we refer to verbs *along with their argument types*, the combination expressed as a verbal schema, e.g., (Animal,"have",BodyPart). This allows us to distinguish different contextual uses of a verb without introducing a proliferation of verb sense symbols. Others have taken a similar approach of using type restrictions to express verb semantics (Pantel et al., 2007; Del Corro et al., 2014).

[3] There are exceptions, e.g., in 4th Grade Science "bat" can refer to either the animal or the sporting implement, but these cases are rare.

3.3 The Pipeline

The pipeline is sketched in Figure 1 and exemplified in Table 1, and consists of six steps:

3.3.1 Sentence Selection

The first step is to construct a collection of (loosely) domain-appropriate sentences from the larger corpus. There are multiple ways this could be done, but in our case we found the most effective way was as follows:

a. List the core topics in the domain of interest (science), here producing 81 topics derived from syllabus guides.
b. For each topic, author 1-3 query templates, parameterized using one or more of the 45 domain types. For example, for the topic "animal adapation", a template was "[*Animal*] adaptation environment", parameterized by the type *Animal*. The purpose of query templates is to steer the search engine to domain-relevant text.
c. For each template, automatically instantiate its type(s) in all possible ways using the domain vocabulary members of those types.
d. Use each instantiation as a search query over the corpus, and collect sentences in the top (here, 10) documents retrieved.

In our case, this resulted in a generally domain-relevant corpus of 7M sentences.

3.3.2 Tuple Generation

Second, we run an open information extraction system over the sentences to generate an initial set of (np, vp, np) tuples. In our case, we use OpenIE 4.2 (Soderland et al., 2013; Mausam et al., 2012).

3.3.3 Headword Extraction and Filtering

Third, the np arguments are replaced with their headwords, by applying a simple headword filtering utility. We discard tuples with infrequent vps or verbal schemas (here vp frequency < 10, schema frequency < 5).

Pipeline Example Outputs:
Inputs: corpus + vocabulary + types
1. Sentence selection:
"In addition, green leaves have chlorophyll.")
2. Tuple Generation:
("green leaves" "have" "chlorophyll")
3. Headword Extraction:
("leaf" "have" "chlorophyll")
4. Refinement and Scoring:
("leaf" "have" "chlorophyll") @0.89 (score)
5. Phrasal tuple generation:
("leaf" "have" "chlorophyll") @0.89 (score)
("green leaf" "have" "chlorophyll") @0.89 (score)
6. Relation Canonicalization:
("leaf" "have" "chlorophyll") @0.89 (score)
("green leaf" "have" "chlorophyll") @0.89 (score)
("leaf" "contain" "chlorophyll") @0.89 (score)
("green leaf" "contain" "chlorophyll") @0.89 (score)

Table 1: Illustrative outputs of each step of the pipeline for the term "leaf".

3.3.4 Refinement and Scoring

Fourth, to improve precision, Turkers are asked to manually score a proportion (in our case, 15%) of the tuples, then a model is constructed from this data to score the remainder. For the Turk task, Turkers were asked to label each tuple as true or false/nonsense. Each tuple is labeled 3 times, and a majority vote is applied to yield the overall label. The semantics we apply to tuples (and which we explain to Turkers) is one of plausibility: if the fact is true for some of the arg1's, then score it as true. For example, if it is true that some birds lay eggs, then the tuple (bird, lay, egg) should be marked true. The degree of manual vs. automated can be selected here depending on the precision/cost constraints of the end application.

We then build a model using this data to predict scores on other tuples. For this model, we use logistic regression applied to a set of tuple features. These tuple features include normalized count features, schema and type level features, PMI statistics and semantic features. Normalized count features are based on the number of occurrences of tuples, and the number of unique sentences the tuple is extracted from. Schema and type level features are derived from the subject and object type, and frequency of schema in the corpus. Semantic features are based on whether subject and object are abstract vs. concrete (using Turney et al's abstractness database (Turney et al., 2011)), and whether there are any modal verbs (e.g. may, should etc.) in the original sentence. PMI features are derived from the count statistics of subject, predicate, object and entire triple in the Google n-gram corpus (Brants and Franz, 2006).

3.3.5 Phrasal Tuple Generation

Fifth, for each headword tuple (n, vp, n), retrieve the original phrasal triples (np, vp, np) it was derived from, and add sub-phrase versions of these phrasal tuples to the KB. For example, if a headword tuple (cat, chase, mouse) was derived from (A black furry cat, chased, a grey mouse) then the algorithm considers adding

(black cat, chase, mouse)
(black furry cat, chase, mouse)
(black cat, chase, grey mouse)
(black furry cat, chase, grey mouse)

Valid noun phrases are those following a pattern "<Adj>* <Noun>+". The system only retains constructed phrasal tuples for which both subject and object phrases satisfy PMI and count thresholds[4], computed using the Google N-gram corpus (Brants and Franz, 2006). In general, if the headword tuple is scored as correct and the PMI and count thresholds are met, then the phrasal originals and variants are also correct. (We evaluate this in Section 5.2).

3.3.6 Canonical Schema Induction

Finally, we induce a set of *schema mapping rules* over the tuples that identify clusters of equivalent and similar relations, and map them to a canonical, generalized relation. These canonical, generalized relations are referred to as canonical schemas, and the induction algorithm is called CASI (Canonical Schema Induction). The rules are then applied to the tuples, resulting in additional general tuples being added to the KB. The importance of this step is that generalizations among seemingly disparate tuples are made explicit. While we could then discard

[4] e.g., "black bear" is a usable phrase provided it occurs $> k_1$ times in the N-gram corpus and $log[p(\text{"black bear"})/p(\text{"black"}).p(\text{"bear"})] > k_2$ in the N-gram corpus, where constants k_1 and k_2 were chosen to optimize performance on a small test set.

tuples that are mapped to a generalized form, we instead retain them in case a query is made to the KB that requires the original fine-grained distinctions. In the next section, we describe how these schema mapping rules are learned.

4 Canonical Schema Induction (CASI)

4.1 Task: Induce schema mapping rules

The role of the schema mapping rules is to make generalizations among seemingly disparate tuples explicit in the KB. To do this, the system identifies clusters of relations with similar meaning, and maps them to a canonical, generalized relation. The mappings are expressed using a set of *schema mapping rules*, and the rules can be applied to infer additional, general triples in the KB. Informally, mapping rules should combine evidence from both external resources (e.g., verb taxonomies) and data (tuples in the KB). This observation allows us to formally define an objective function to guide the search for mapping rules. We define:

- a **schema** is a structure
  (type1,verb phrase,type2)
  here the types are from the input type inventory.
- a **schema mapping rule** is a rule of the form
  $schema_i \rightarrow schema_j$
  stating that a triple using $schema_i$ can be re-expressed using $schema_j$.
- a **canonical schema** is a schema that does not occur on the left-hand side of any mapping rule, i.e., it does not point to any other schema.

To learn a set of schema mapping rules, we select from the space of possible mapping rules so as to:

- maximize the quality of the selected mapping rules, i.e., maximize the evidence that the selected rules express valid paraphrases or generalization. That is we are looking for synonymous and type-of edges between schemas. This evidence is drawn from both existing resources (e.g., WordNet) and from statistical evidence (among the tuples themselves).
- satisfy the constraint that every schema points to a canonical schema, or is itself a canonical schema.

We can view this task as a subgraph selection problem in which the nodes are schemas, and directed edges are possible mapping rules between schemas. The learning task is to select subgraphs such that all nodes in a subgraph are similar, and point to a single, canonical node (Figure 2). We refer to the blue nodes in Figure 2 as induced canonical schemas.

To solve this selection problem, we formulate it as as a linear optimization task and solve it using integer linear programming (ILP), as we now describe.

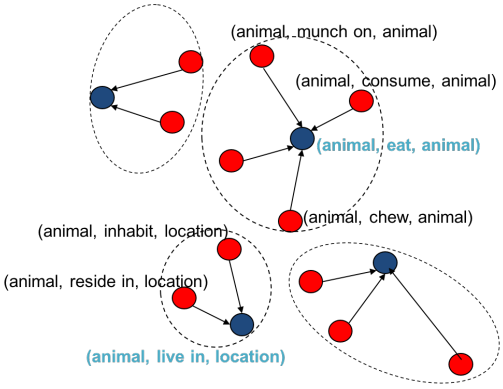

Figure 2: Learning schema mapping rules can be viewed as a subgraph selection problem, whose result (illustrated) is a set of clusters of similar schemas, all pointing to a single, canonical form.

4.2 Features for learning schema mapping rules

To assess the quality of candidate mapping rules, we combine features from the following sources: Moby, WordNet, association rules and statistical features from our corpus. These features indicate synonymy or type-of links between schemas. For each schema S_i e.g. (Animal, live in, Location) we define the relation r_i as being the verb phrase (e.g. "live in"), and v_i as the root verb of r_i (e.g. "live").

- **Moby:** We also use verb phrase similarity scores derived from the Moby thesaurus. Moby score M_{ij} for a schema pair is computed by a lookup in this dataset for relation pair r_i, r_j or root verb pair v_i, v_j. This is also a directed feature, i.e. $M_{ij} \neq M_{ji}$.
- **WordNet:** If there exists a troponym link path from schema r_i to r_j, then we define the WordNet score W_{ij} for this schema pair as the inverse of the number of edges that need to be

	Type		Use which parts of schema?			What kind of relations do they encode?		
Feature source	semantic	distributional	subject	predicate	object	synonym	type-of	temporal implication
Moby	✓			✓		✓	✓	
WordNet	✓			✓			✓	
AMIE-typed		✓	✓		✓	✓	✓	✓
AMIE-untyped		✓	✓		✓	✓	✓	✓

Table 2: The different features used in relation canonicalization capture different aspects of similarity.

$$
\begin{aligned}
\underset{\{X_{ij}\}}{\text{maximize}} \sum_{\text{i,j}} \text{X}_{\text{ij}} \big(& \lambda_1 * \text{M}_{\text{ij}} + \lambda_2 * \text{Wij} + \lambda_3 * \text{AT}_{\text{ij}} \\
& + \lambda_4 * \text{AU}_{\text{ij}} + \lambda_5 * \text{S}_{\text{ij}} \big) - \delta * \|\text{X}\|_1 \\
\text{subject to,} \quad & X_{ij} \in \{0,1\}, \quad \forall \langle i,j \rangle && X_{ij} \text{ are boolean.} \\
& X_{ij} + X_{ji} \leq 1, \quad \forall i,j && \text{schema mapping relation is asymmetric.} \\
& \sum_j X_{ij} \leq 1, \quad \forall i && \text{select at most one parent per schema.} \\
& X_{ij} + X_{jk} - X_{ik} \leq 1, \quad \forall \langle i,j,k \rangle && \text{schema mapping relation is transitive.}
\end{aligned} \tag{1}
$$

Figure 3: The ILP used for canonical schema induction

traveled to reach r_j from r_i. If such a path does not exist, then we look for a path from v_i to v_j. Since we do not know the exact WordNet synset applicable for each schema, we consider all possible synset choices and pick the best score as W_{ij}. This is a directed feature i.e., $W_{ij} \neq W_{ji}$. Note that even though WordNet is a high quality resource, it is not completely sufficient for our purposes. Out of 955 unique relations (verb phrases) in our KB, only 455 (47%) are present in WordNet. We can deal with these out of WordNet verb phrases by relying on other sets of features described next.

- **AMIE:** AMIE is an association rule mining system that can produce association rules of the form: "?a eat ?b $\rightarrow$?a consume ?b". We have two sets of AMIE features: typed and untyped. Untyped features are of the form $r_i \rightarrow r_j$, e.g., $eat \rightarrow consume$, whereas typed features are of the form $S_i \rightarrow S_j$, e.g., $(Animal, eat, Food) \rightarrow (Animal, consume, Food)$. AMIE produces real valued scores[5] between 0 to 1 for each rule. We define AU_{ij} and AT_{ij} as untyped and typed AMIE rule scores respectively.

- **Specificity:** We define specificity of each relation as its IDF score in terms of the number of argument pairs it occurs with, compared to total number of argument type pairs in the corpus. The specificity score of a schema mapping rule favors more general predicates on the parent side of the rules.

$$
\begin{aligned}
& specificity(r) = IDF(r) \\
& SP(r) = \frac{specificity(r)}{\max_{r'} specificity(r')} \\
& S_{ij} = SP(r_i) - SP(r_j)
\end{aligned}
$$

  Further, we have a small set of very generic relations like "have" and "be" that are considered as relation stopwords by setting their $SP(r)$ scores to 1.

These features encode different aspects of similarity between schemas as described in Table 2. In this work we combine semantic high-quality features from WordNet, Moby thesaurus with weak distributional similarity features from AMIE to generate schema mapping rules. We have observed that thesaurus features are very effective for predicates which are less ambiguous e.g. eat, consume, live in. Association rule features on the other hand have evidence for predicates which are very ambiguous e.g. have, be. Thus these features are complementary.

Further, these features indicate different kinds of

[5] We use PCA confidence scores produced by AMIE.

relations between two schemas: synonymy, type-of and temporal implication (refer Table 2). In this work, we want to learn the schema mapping rules that capture synonymy and type-of relations and discard the temporal implications. This makes our problem setting different from that of knowledge base completion methods e.g., (Socher et al., 2013). Our proposed method CASI uses an ensemble of semantic and statistical features enabling us to promote the synonymy and type-of edges, and to select the most general schema as canonical schema per cluster.

4.3 ILP model used in CASI

The features described in Section 4.2 provide partial support for possible schema mapping rules in our dataset. The final set of rules we select needs to comply with asymmetry, transitive closure and at most one parent per schema constraints. We use an integer linear program to find the optimal set of schema mapping rules that satisfy these constraints, shown formally in Figure 3.

We decompose the schema mapping problem into multiple independent sub-problems by considering schemas related to a pair of argument types, e.g, all schemas that have domain or range types Animal, Location would be considered as a separate sub-problem. This way we can scale our method to large sets of schemas. The ILP for each sub-problem is presented in Equation 1.

In Equation 1, each X_{ij} is a boolean variable representing whether we pick the schema mapping rule $S_i \rightarrow S_j$. As described in Section 4.2, $M_{ij}, W_{ij}, AT_{ij}, AU_{ij}, S_{ij}$ represent the scores produced by Moby, WordNet, AMIE-typed, AMIE-untyped and Specificity features respectively for the schema mapping rule $S_i \rightarrow S_j$. The objective function maximizes the weighted combination of these scores. Further, the solution picked by this ILP satisfies constraints such as asymmetry, transitive closure and at most one parent per schema. We also apply an L_1 sparsity penalty on X, retaining only those schema mapping edges for which the model is reasonably confident.

For n schemas, there are $O(n^3)$ transitivity constraints which make the ILP very inefficient. Berant et al. (2011) proposed two approximations to handle a large number of transitivity rules by decomposing the ILP or solving it in an incremental way. Instead we re-write the ILP rules in such a way that we can efficiently solve our mapping problem without introducing any approximations. The last two constraints of this ILP can be rewritten as follows:

$$\begin{aligned}&\big(\textstyle\sum_j X_{ij} \leq 1, \forall i \\ &\text{AND } X_{ij} + X_{jk} - X_{ik} \leq 1, \quad \forall \langle i,j,k \rangle\big) \\ &\implies \text{If}(X_{ij} = 1) \text{ then } X_{jk} = 0 \; \forall k\end{aligned}$$

This results in $O(n^2)$ constraints and makes the ILP efficient. Impact of this technique in terms of runtime is described in Section 5.3.

We then use an off-the-shelf ILP optimization engine called SCPSolver (Planatscher and Schober, 2015) to solve the ILP problems. The output of our ILP model is the schema mapping rules. We then apply these rules onto KB tuples to generate additional, general tuples. Some examples of the learned rules are:

(Organism, have, Phenomenon)
→ (Organism, undergo, Phenomenon)
(Animal, have, Event)
→ (Animal, experience, Event)
(Bird, occupy, Location)
→ (Bird, inhabit, Location)

5 Evaluation

5.1 KB Comprehensiveness

Our overall goal is a high-precision KB that has reasonably "comprehensive" coverage of facts in the target domain, on the grounds that these are the facts that a domain application is likely to query about. This notion of KB comprehensiveness is an important but under-discussed aspect of knowledge bases. For example, in the automatic KB construction literature, while a KB's size is often reported, this does not reveal whether the KB is near-complete or merely a drop in the ocean of that required (Razniewski et al., 2016; Stanovsky and Dagan, 2016). More formally, we define comprehensiveness as: *recall at high ($> 80\%$) precision of domain-relevant facts*. This measure is similar to recall at the point P=80% on the PR curve, except recall is measured with respect to a different distribution of facts (namely facts about elementary science) rather than a held-out sample of data used to build the KB. The particular target precision value is not critical; what

KB	Precision	Coverage of Tuple-Expressible Science Knowledge (Recall on science KB)	KB comprehensiveness w.r.t. Science domain (Science recall @80% precision)
WebChild	89%	3.4%	3.4%
NELL	85%	0.1%	0.1%
ConceptNet	40%	8.4%	n/a (p<80%)
ReVerb-15M	55%	11.5%	n/a (p<80%)
Our KB	81%	23.2%	**23.2%**

Table 3: Precision and coverage of tuple-expressible elementary science knowledge by existing resources vs. our KB. Precision estimates are within +/-3% with 95% confidence interval.

is important is that the same precision point is used when comparing results. We choose 80% as subjectively reasonable; at least 4 out of 5 queries to the KB should be answered correctly.

There are several ways this target distribution of required facts can be modeled. To fully realize the ambition of this metric, we would directly identify a sample of required end-task facts, e.g., by manual analysis of questions posed to the end-task system, or from logs of the interaction between the end-task system and the KB. However, given the practical challenges of doing this at scale, we take a simpler approach and approximate this end-task distribution using facts extracted from an (independent) *domain-specific* text corpus (we call this a *reference corpus*). Note that these facts are only a sample of domain-relevant facts, not the entirety. Otherwise, we could simply run our extractor over the reference corpus and have all we need. Now we are in a strong position, because the reference corpus gives us a fixed point of reference to measure comprehensiveness: we can sample facts from it and measure what fraction the KB "knows", i.e., can answer as true (Figure 4).

For our specific task of elementary science QA, we have assembled a reference corpus[6] of ∼1.2M sentences comprising of multiple elementary science textbooks, multiple dictionary definitions of all fourth grade vocabulary words, and simple Wikipedia pages for all fourth grade vocabulary words (where such pages exist). To measure our KB's comprehensiveness (of facts within the expressive power of our KB), we randomly sampled 4147 facts, expressed as headword tuples, from

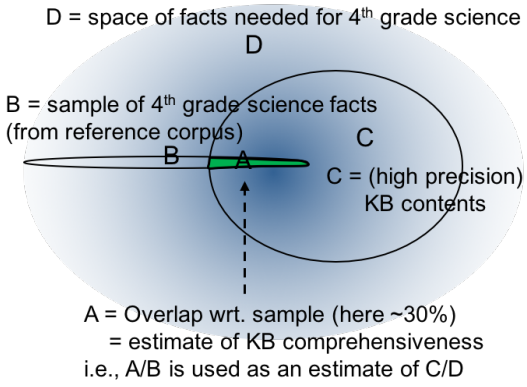

Figure 4: Comprehensiveness (frequency-weighted coverage C of the required facts D) can be estimated using coverage A of a reference KB B as a surrogate sampling of the target distribution.

the reference corpus. These were generated semi-automatically using parts of our pipeline, namely information extraction then Turker scoring to obtain true facts[7]. We call these facts the Reference KB[8]. To the extent our tuple KB contains facts in this Reference KB (and under the simplifying assumption that these facts are representative of the science knowledge our QA application needs), we say our tuple KB is comprehensive. Doing this yields a value of 23% comprehensiveness for our KB (Table 3).

We also measured the precision and science coverage of other, existing fact KBs. For precision, we took a random sample of 1000 facts in each KB, and followed the same methodology as earlier so that the

[6]This corpus named as "Aristo MINI Corpus" is available for download at `http://allenai.org/data/aristo-tuple-kb`

[7]This method will of course miss many facts in the reference corpus, e.g., when extraction fails or when the fact is in a non-sentential form, e.g., a table. However, we only assume that the *distribution* of extracted facts is representative of the domain.

[8]These 4147 test facts are published with the dataset at `http://allenai.org/data/aristo-tuple-kb`

comparison is valid: Turkers label each fact as true or false/nonsense, each fact is labeled 3 times, and the majority label is the overall label. The precisions are shown in Table 3. For ConceptNet, we used only the subset of facts with frequency > 1, as frequency=1 facts are particularly noisy (thus the precision of the full ConceptNet would be lower).

We also computed the science coverage (= comprehensiveness, if p>80%) using our reference KB. Note that these other KBs were not designed with elementary science in mind and so, not surprisingly, they do not cover many of the relations in our domain. To make the comparison as fair as possible, given these other KBs use different relational vocabularies, we first constructed a list of 20 very general relations (similar to the ConceptNet relations, e.g., causes, uses, part-of, requires), and then mapped relations used in both our reference facts, and in the other KBs, to these 20 relations. To compare if a reference fact is in one of these other KBs, only the general relations need to match, and only the subject and object headwords need to match. This allows substantial linguistic variation to be permitted during evaluation (e.g., "contain",. "comprise", "part of" etc. would all be considered matching). In other words, this is a generous notion of "a KB containing a fact", in order to be as fair as possible.

As Table 4 illustrates, these other KBs cover very little of the target science knowledge. In the case of WebChild and NELL, the primary reason for low recall is low overlap between their target and ours. NELL has almost no predicate overlap with our Reference KB, reflecting it's Named Entity centric content. WebChild is rich in part-of and location information, and covers 60% of part-of and location facts in our Reference KB. However, these are only 4.5% of all the facts in the Reference KB, resulting in an overall recall (and comprehensiveness) of 3%. In contrast, ConceptNet and ReVerb-15M have substantially more relational overlap with our Reference KB, hence their recall numbers are higher. However, both have lower precision, limiting their utility.

This evaluation demonstrates the limited science coverage of existing resources, and the degree to which we have overcome this limitation. The extraction methods used to build these resources are not directly comparable since they are starting with different input/output settings and involve significantly different degrees of supervision. Rather, the results suggest that general-purpose KBs (e.g., NELL) may have limited coverage for specific domains, and that our domain-targeted extraction pipeline can significantly alleviate this in terms of precision and coverage when that domain is known.

Extraction stage output	#schemas	#tuples	% Avg. precision
2. Tuple generation	-	7.5M	54.2
3. Headword tuples	29.3K	462K	68.0
4. Tuple scoring	15.8K	156K	87.2
5. Phrasal tuples	15.8K	286K	86.5
6. Canonical schemas	15.8K	340K	80.6

Table 4: Evaluation of KB at different stages of extraction. Precision estimates are within +/-3% with 95% confidence interval.

5.2 Performance of the Extraction Pipeline

In addition, we measured the average precision of facts present in the KB after every stage of the pipeline (Table 4). We can see that the pipeline take as input 7.5M OpenIE tuples with precision of 54% and produces a good quality science KB of over 340K facts with 80.6% precision organized into 15K schemas. The Table also shows that precision is largely preserved as we introduce phrasal triples and general tuples.

5.3 Evaluation of Canonical Schema Induction

In this section we will focus on usefulness and correctness of our canonical schema induction method.

The parameters of the ILP model (see Equation 1) i.e., $\lambda_1 \ldots \lambda_5$ and δ are tuned based on sample accuracy of individual feature sources and using a small schema mapping problem with schemas applicable to vocabulary types Animal and Body-Part.

$$\lambda_1 = 0.7, \lambda_2 = 0.9, \lambda_3 = 0.3,$$
$$\lambda_4 = 0.1, \lambda_5 = 0.2, \delta = 0.7$$

Further, with $O(n^3)$ transitivity constraints we could not successfully solve a single ILP problem with 100 schemas within a time limit of 1 hour. Whereas, when we rewrite them with $O(n^2)$ constraints as explained in Section 4.3, we could solve 443 ILP sub-problems within 6 minutes with average runtime per ILP being 800 msec.

Canonical schema induction method	Comprehensiveness
None	20.0%
AMIE*	20.9%
CASI (our method)	**23.2%**

Table 5: Use of the CASI-induced schemas significantly (at the 99% confidence level) improves comprehensiveness of the KB.

As discussed in Section 2, we not only cluster the (typed) relations, but also identify a canonical relation that all the other relations in a cluster can be mapped to, without recourse to human annotated training data or a target relational vocabulary. Although no existing methods do this directly, the AMIE-based schema clustering method of (Galárraga et al., 2014) can be extended to do this by incorporating the association rules learned by AMIE (both typed and untyped) inside our ILP framework to output schema mapping rules. We call this extension AMIE*, and use it as a baseline to compare the performance of CASI against.

5.3.1 Canonical Schema Usefulness

The purpose of canonicalization is to allow equivalence between seemingly different schema to be recognized. For example, the KB query ("polar bear", "reside in", "tundra")?[9] can be answered by a KB triple ("polar bear", "inhabit", "tundra") if schema mapping rules map one or both to the same canonical form e.g., ("polar bear", "live in", "tundra") using the rules:

(Animal, inhabit, Location)
→ (Animal, live in, Location)
(Animal, reside in, Location)
→ (Animal, live in, Location)

One way to quantitatively evaluate this is to measure the impact of schema mapping on the comprehensiveness metric. Table 5 shows that, before applying any canonical schema induction method, the comprehensiveness score of our KB was 20%. The AMIE* method improves this score to 20.9%, whereas our method achieves a comprehensiveness of 23.2%. This latter improvement over the original KB is statistically significant at the 99% confidence level (sample size is the 4147 facts sampled from the reference corpus).

5.3.2 Canonical Schema Correctness

A second metric of interest is the correctness of the schema mapping rules (just because comprehensiveness improves does not imply every mapping rule is correct). We evaluate correctness of schema mapping rules using following metric:

Precision of schema mapping rules: We asked Turkers to directly assess whether particular schema mapping rules were correct, for a random sample of rules. To make the task clear, Turkers were shown the schema mapping rule (expressed in English) along with an example fact that was rewritten using that rule (to give a concrete example of its use), and they were asked to select one option "correct or incorrect or unsure" for each rewrite rule. We asked this question to three different Turkers and considered the majority vote as final evaluation[10].

The comparison results are shown in Table 6. Starting with 15.8K schemas, AMIE* canonicalized only 822 of those into 102 canonical schemas (using 822 schema mapping rules). In contrast, our method CASI canonicalized 4.2K schemas into 2.5K canonical schemas. We randomly sampled 500 schema mapping rules generated by each method and asked Turkers to evaluate their correctness, as described earlier. As shown in Table 6, the precision of rules produced was CASI is 68%, compared with AMIE* which achieved 59% on this metric. Thus CASI could canonicalize five times more schemas with 9% more precision.

5.4 Discussion and Future Work

Next, we identify some of the limitations of our approach and directions for future work.

1. Extracting Richer Representations of Knowledge: While triples can capture certain kinds of knowledge, there are other kinds of information, e.g. detailed descriptions of events or processes, that cannot be easily represented by a set of independent tuples. An extension of this work would be to extract event frames, capable of representing a richer set of

[9] e.g., posed by a QA system trying to answer the question "Which is the likely location in which a polar bear to reside in? (A) Tundra (B) Desert (C) Grassland"

[10] We discarded the unsure votes. For more than 95% of the rules, at least 2 out of 3 Turkers reached clear consensus on whether the rule is "correct vs. incorrect", indicating that the Turker task was clearly defined.

Canonical schema induction method	#input schemas	#schema mapping rules	#induced canonical schemas	Precision of schema mapping rules
AMIE*	15.8K	822	102	59%
CASI (our method)	15.8K	**4.2K**	2.5K	**68%**

Table 6: CASI canonicalizes five times more schemas than AMIE*, and also achieves a small (9%) increase in precision, demonstrating how additional knowledge resources can help the canonicalization process (Section 4.2). Precision estimates are within +/-4% with 95% confidence interval.

roles in a wider context compared to a triple fact. For example in the news domain, while representing an event "public shooting", one would like to store the shooter, victims, weapon used, date, time, location and so on. Building high-precision extraction techniques that can go beyond binary relations towards event frames is a potential direction of future research.

2. Richer KB Organization: Our approach organizes entities and relations into flat entity types and schema clusters. An immediate direction for extending this work could be a better KB organization with deep semantic hierarchies for predicates and arguments, allowing inheritance of knowledge among entities and triples.

3. Improving comprehensiveness beyond 23%: Our comprehensiveness score is currently at 23% indicating 77% of potentially useful science facts are still missing from our KB. There are multiple ways to improve this coverage including but not limited to 1) processing more science corpora through our extraction pipeline, 2) running standard KB completion methods on our KB to add the facts that are likely to be true given the existing facts, and 3) improving our canonical schema induction method further to avoid cases where the query fact is present in our KB but with a slight linguistic variation.

4. Quantification Sharpening: Similar to other KBs, our tuples have the semantics of plausibility: If the fact is generally true for some of the arg1s, then score it as true. Although frequency filtering typically removes facts that are rarely true for the arg1s, there is still variation in the quantifier strength of facts (i.e., does the fact hold for all, most, or some arg1s?) that can affect downstream inference. We are exploring methods for quantification sharpening, e.g., (Gordon and Schubert, 2010), to address this.

5. Can the pipeline be easily adapted to a new domain? Our proposed extraction pipeline expects high-quality vocabulary and types information as input. In many domains, it is easy to import types from existing resources like WordNet or FreeBase. For other domains like medicine, legal it might require domain experts to encode this knowledge. However, we believe that manually encoding types is a much simpler task as compared to manually defining all the schemas relevant for an individual domain. Further, various design choices, e.g., precision vs. recall tradeoff of final KB, the amount of expert input available, etc. would depend on the domain and end task requirements.

6 Conclusion

Our goal is to construct, a domain-targeted, high precision knowledge base of *(subject,predicate,object)* triples to support an elementary science application. We have presented a scalable knowledge extraction pipeline that is able to extract a large number of facts targeted to a particular domain. The pipeline leveraging Open IE, crowdsourcing, and a novel schema learning algorithm, and has produced a KB of over 340,163 facts at 80.6% precision for elementary science QA.

We have also introduced a metric of *comprehensiveness* for measuring KB coverage with respect to a particular domain. Applying this metric to our KB, we have achieved a comprehensiveness of over 23% of science facts within the KB's expressive power, substantially higher than the science coverage of other comparable resources. Most importantly, the pipeline offers for the first time a viable way of extracting large amounts of high-quality knowledge targeted to a specific domain. We have made the KB publicly available at `http://data.allenai.org/tuple-kb`.

Acknowledgments

We are grateful to Paul Allen whose long-term vision continues to inspire our scientific endeavors. We would also like to thank Peter Turney and Isaac Cowhey for their important contributions to this project.

References

S. Auer, C. Bizer, J. Lehmann, G. Kobilarov, R. Cyganiak, and Z. Ives. 2007. DBpedia: A nucleus for a web of open data. In *In ISWC/ASWC*.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global learning of typed entailment rules. In *ACL*.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *SIGMOD*.

Thorsten Brants and Alex Franz. 2006. Web 1T 5-gram version 1 LDC2006T13. Philadelphia: Linguistic Data Consortium.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. 2010. Toward an architecture for never-ending language learning. In *AAAI*, volume 5, page 3.

Bhavana Dalvi, Sumithra Bhakthavatsalam, Chris Clark, Peter Clark, Oren Etzioni, Anthony Fader, and Dirk Groeneveld. 2016. IKE - An Interactive Tool for Knowledge Extraction. In *AKBC@NAACL-HLT*.

Luciano Del Corro, Rainer Gemulla, and Gerhard Weikum. 2014. Werdy: Recognition and disambiguation of verbs and verb phrases with syntactic and semantic pruning. In *2014 Conference on Empirical Methods in Natural Language Processing*, pages 374–385. ACL.

Karthik Dinakar, Birago Jones, Catherine Havasi, Henry Lieberman, and Rosalind Picard. 2012. Common sense reasoning for detection, prevention, and mitigation of cyberbullying. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 2(3):18.

Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*.

Alexander Faaborg and Henry Lieberman. 2006. A goal-oriented web browser. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 751–760. ACM.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics. ReVerb-15M available at `http://openie.cs.washington.edu`.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Luis Galárraga, Christina Teflioudi, Katja Hose, and Fabian M. Suchanek. 2013. AMIE: association rule mining under incomplete evidence in ontological knowledge bases. In *WWW*.

Luis Galárraga, Geremy Heitz, Kevin Murphy, and Fabian M. Suchanek. 2014. Canonicalizing open knowledge bases. In *CIKM*.

Jonathan Gordon and Lenhart K Schubert. 2010. Quantificational sharpening of commonsense knowledge. In *AAAI Fall Symposium: Commonsense Knowledge*.

Adam Grycner and Gerhard Weikum. 2014. Harpy: Hypernyms and alignment of relational paraphrases. In *COLING*.

Adam Grycner, Gerhard Weikum, Jay Pujara, James R. Foulds, and Lise Getoor. 2015. RELLY: Inferring hypernym relationships between relational phrases. In *EMNLP*.

Dekang Lin and Patrick Pantel. 2001. DIRT - discovery of inference rules from text. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 323–328. ACM.

Hugo Liu and Push Singh. 2004. ConceptNet: a practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.

Mausam, Michael Schmitz, Stephen Soderland, Robert Bart, and Oren Etzioni. 2012. Open language learning for information extraction. In *EMNLP*.

Andrea Moro and Roberto Navigli. 2012. WiSeNet: building a wikipedia-based semantic network with ontologized relations. In *CIKM*.

Patrick Pantel, Rahul Bhagat, Bonaventura Coppola, Timothy Chklovski, and Eduard H Hovy. 2007. ISP: Learning inferential selectional preferences. In *HLT-NAACL*, pages 564–571.

Hannes Planatscher and Michael Schober. 2015. SCP solver. `http://scpsolver.org`.

Simon Razniewski, Fabian M Suchanek, and Werner Nutt. 2016. But what do we actually know? In *Proc. AKBC'16*.

Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with

matrix factorization and universal schemas. In *HLT-NAACL*.

Richard Socher, Danqi Chen, Christopher D. Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *NIPS*.

Stephen Soderland, John Gilmer, Robert Bart, Oren Etzioni, and Daniel S. Weld. 2013. Open Information Extraction to KBP Relations in 3 Hours. In *TAC*.

Robert Speer and Catherine Havasi. 2013. Conceptnet 5: A large semantic network for relational knowledge. In *The Peoples Web Meets NLP*, pages 161–176. Springer.

Gabriel Stanovsky and Ido Dagan. 2016. Creating a large benchmark for open information extraction. In *EMNLP*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *WWW*.

Niket Tandon, Gerard de Melo, Fabian Suchanek, and Gerhard Weikum. 2014. WebChild: Harvesting and Organizing Commonsense Knowledge from the Web. In *WSDM*.

Peter D. Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and metaphorical sense identification through concrete and abstract context. In *EMNLP*.

Travis Wolfe, Mark Dredze, James Mayfield, Paul McNamee, Craig Harman, Timothy W. Finin, and Benjamin Van Durme. 2015. Interactive knowledge base population. *CoRR*, abs/1506.00301.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*.