Find your Advisor: Robust Knowledge Gathering from the Web

Ndapandula Nakashole, Martin Theobald, Gerhard Weikum
Max-Planck-Institute für Informatik
Saarbrücken, Germany
{nnakasho,mtb,weikum}@mpi-inf.mpg.de

ABSTRACT

We present a robust method for gathering relational facts from the Web, based on matching *generalized patterns* which are automatically learned from seed facts for relations of interest. Our approach combines these generalized patterns for *high recall* information extraction with a rule-based, declarative reasoning approach to also ensure *high precision*. Newly extracted candidate facts are assigned statistical weights which reflect the strengths of the patterns used to extract them. For checking the plausibility of candidate facts with respect to existing knowledge and competing hypotheses, we use an efficient algorithm for weighted Max-Sat over propositional-logic clauses. In contrast to prior work on reasoning-based information extraction, we employ richer statistics and smart pruning to bound the number of grounded rules passed on to the Max-Sat solver.

1. INTRODUCTION

1.1 Motivation

Knowledge-sharing communities like Wikipedia and recent advances on scalable information extraction have opened up opportunities for large-scale knowledge harvesting: automatically constructing comprehensive knowledge bases of entity-relationship-structured facts. Notable endeavors along these lines of a "Semantic Wikipedia" and a "machine-readable Web" include academic projects such as DBpedia [3], YAGO [21], Text2Onto [9], sindice/sig.ma [23], KnowItAll/TextRunner [13], IntelligenceInWikipedia [25], ReadTheWeb [6], Omnivore [5], StatSnowball [28], or DBLife [11], and also commercial endeavors like *freebase.com*, *trueknowledge.com*, *wolframalpha.com*, *www.google.com/squared*, or *entitycube.research.microsoft.com*. Such knowledge bases consist of relational facts about millions of named entities, their semantic types, and their relationships. They provide great assets for semantic search on the Web and in enterprises, entity reconciliation, knowledge-based reasoning, and other applications.

Knowledge harvesting typically works by pattern matching, statistical learning, or logical reasoning [12, 18] for entities and relationships embedded in semistructured Web pages such as Wikipedia infoboxes or natural-language texts such as Wikipedia articles, online news, biographies or homepages of researchers, and so on. As with many learning tasks, hand-labeled training data for supervised learning of information extraction is typically a key bottleneck. Therefore, a family of almost-unsupervised methods has become the prevalent approach: fact harvesting starts with a small set of *seed facts* for one or more relations of interest, then automatically finds markup, textual, or linguistic *patterns* in the underlying sources as indicators of facts, and finally uses these patterns to identify new *fact candidates* as further hypotheses to populate the relations in the knowledge base.

For example, to collect facts about the Alma Mater and doctoral advisors of researchers, one could start with seeds such as *graduatedFrom(Jim Gray, UC Berkeley)*, *graduatedFrom(Hector Garcia-Molina, Stanford)*, *hasAcademicAdvisor(Jim Gray, Mike Harrison)*, and *hasAcademicAdvisor(Hector Garcia-Molina, Gio Wiederhold)*. We could then find text patterns such as "x graduated at u", "x and his advisor y", and "professor y and his student x" (with placeholders x, u, y for the involved named entities), which in turn could lead to discovering the new facts such as *graduatedFrom(Susan Davidson, Princeton)* and *hasAcademicAdvisor(Susan Davidson, Hector Garcia-Molina)*.

This fact-pattern duality can be harnessed in an iterative manner, with statistical assessment of the indicative strengths of patterns and the confidence in fact candidates. Moreover, additional plausability tests can be employed based on logical consistency constraints, to reduce the false-positive rate. For example, if researcher x gradudated from university u under the supervision of y, then y would have to be a professor or lecturer with a position at or other connections to u.

1.2 Problem

The outlined methods for information extraction and seed-based knowledge harvesting face a three-way *trade-off* regarding the dimensions of *precision*, *recall*, and *efficiency*. Simple pattern-based extraction can achieve high recall (i.e., high fraction of potentially collectible facts) but typically has low or mediocre precision (i.e., low fraction of correct facts). This is because of false positives resulting from noisy or ambiguous patterns such as "x and his co-author y" which can easily be picked up from seed facts but dilutes the subsequent candidate gathering. Conservative pruning based on statistical confidence measures, on the other hand, would strengthen precision only at the cost of substantially lowering recall. To increase recall, one could consider patterns with ellipses (wild-cards for included substrings), but this can lead to misleading patterns by included appositions or relative clauses - an example is "y, a great mentor and advisor of many students, and her co-author x". Identifying this situation requires deep parsing of natural-language sentences, using a lexical dependency parser. This in turn is computationally expensive and would prohibit high-throughput harve-

WebDB '10 Indianapolis, IN USA

sting.

Recently, a family of constraints-based methods have been developed, for complementing the statistical evidence from patterns with reasoning on the consistency of fact candidates. These methods include relational learning with Markov Logic Networks [17], log-linear learning models with constraints [8], and our own work on the SOFIE [22] tool which casts reasoning about extraction hypotheses into a weighted Max-Sat problem. These methods have the potential to improve precision without neglecting recall, but all of them still tend to favor precision over recall and all of them have high computational costs.

The problem thus is to devise new methods that reconcile high precision, good recall, and affordable computational costs, with the final goal of scalable and robust knowledge harvesting. The approach explored in this paper is to intelligently combine and extend several building blocks from the methodological repertoire outlined above.

1.3 Contribution

In this paper, we develop a robust approach to knowledge harvesting that combines extensions of high-recall, pattern-driven extractors with high-precision consistency reasoning in an innovative way. Our approach extends prior work in the SOFIE [22] information extraction framework into two important directions: 1) generalized patterns and 2) statistical weights assigned to the patterns used for fact extraction. Thus, for *high recall*, we propose a new kind of patterns that consist of sets of variable-length n-grams, with an optional lifting to part-of-speech tags. For *high precision*, we compute extended statistics for the confidence in these patterns, and construct a set of weighted clauses for Max-Sat-based logical reasoning about the mutual consistency of fact candidates. Our experimental studies demonstrate major improvements on our previous work on the SOFIE tool.

2. PATTERN GATHERING AND ANALYSIS

In the first phase of our three-phase architecture, we gather all the patterns that we can find in text. The rationale of this phase is recall-oriented: strive to retain almost all useful patterns.

We are given a set of binary relations $R_1, \ldots, R_m$ of interest, each with a type signature, and we assume that we have an existing knowledge base with typed entities. We assume that this knowledge base more or less includes all individual entities as well as information about their types (e.g., *scientist* or more generally *person*). For many other domains, explicit dictionaries are available, for example, *imdb.com* for movies, *librarything.com* for books, MeSH (Medical Subject Headings or the YAGO [21] knowledge base when dealing with Wikipedia entities. This provides us with a fairly complete dictionary of the possible meanings of a surface string. These possible meanings are captured by a special *means* relation, which explicitly maps the different surface strings onto individual entities. As in the original SOFIE approach [22], disambiguation of surface strings onto entity names can be done at reasoning time.

We then consider a textual corpus, e.g., a set of Wikipedia articles, Web pages, or news articles, as input for pattern analysis. The corpus is pre-processed to produce meaningful units, e.g., sentences or passages which are represented as sequences of tokens. Finding new patterns is then based on first detecting entities or entity candidates within these different parts of speech of the sentence. Every sentence with two or more entities (each occurring in a different noun phrase) serves as source for a potential pattern. Initially, any subsequence between two such entities is considered as a basic pattern candidate.

2.1 Seed Patterns

The basic patterns from the pattern gathering phase are fed into a frequent *n-gram-itemset* mining algorithm for identifying strong patterns. To motivate this approach, consider the following sentences with two different formulations for the *alma mater* relation:
1: `Jennifer Widom` *received her* Bachelors degree from the Indiana University School of Music and her Computer Science *Ph.D. from* `Cornell University`.
2: `Jeff Ullman` *received his Ph.D. degree* in Electrical Engineering *from* `Princeton`.

The sentences show the relevant entities in typewriter font and interesting candidates for patterns in italics. Intuitively, the two sentences have very similar structure, so that it seems feasible to identify a common pattern. However, the common pattern is not easily recognizable by a computer. The basic patterns - substrings between the two entities of the sentences - differ widely.

However, the basic patterns exhibit common sub-patterns such as *"received ... Ph.D. ... degree ... from"*. Our approach is to discover these sub-patterns. But as each of those is merely a short sequence of 2 or 3 words, it would not generalize well towards newly seen sentences which may or may not contain proper facts for the *graduatedFrom* relation. For example, we would be confused by a sentence such as "Jeff Ullman received the best paper award for the work he did with his Ph.D. students at Stanford". To overcome this problem, we use frequent *n-gram itemsets* as patterns, that is a set of co-occurring n-grams, for example *{received his, Ph.D., from}*. This is a powerful means for reducing false positives while retaining high recall.

There is still natural variety regarding pronouns or injected adjectives that could prevent us from identifying good patterns. For example, because of the variations *"received his"* and *"received her"*, we may dismiss good n-grams as too infrequent. We address this issue also by POS tagging, thus assigning word categories – nouns, verbs, prepositions, etc. – to the words in a sentence. We consider specific categories like pronouns, prepositions, and to adjectives as generalizations of the actual occurring words. By replacing the words with their POS tags, we obtain a more general form of n-gram pattern that we refer to as *lifted pattern*. These POS-lifted n-grams are considered in addition to the words-only n-grams, and our notion of n-gram-itemset patterns can freely combine both kinds, as determined by occurrence frequencies. To generate the n-gram-itemset patterns, we apply the technique of frequent itemset mining [2] which has been widely used to discover interesting relations among items in databases. While frequent itemset mining has been used for generalizing patterns in various domains, our approach is the first to augment the generated patterns with a logical reasoning component for ensuring high precision.

DEFINITION 1 (SEED EXAMPLES, COUNTEREXAMPLES). *A seed example* $(e1, e2) \in R_i$ *is an instance of the relation* R_i *with* $e1, e2$ *denoting uniquely identified entities. It is asserted that a seed example is indeed a valid fact. A* counterexample $(e3, e4) \notin R_i$ *is an entity pair for which it is asserted that the pair is not an instance of* R_i*. For relation* R_i*, we denote the set of seed examples and counterexamples by* $SX(R_i)$ *and* $CX(R_i)$*, respectively.*

DEFINITION 2 (LIFTED & N-GRAM-ITEMSET PATTERNS). *Given a set* $SX(R_i)$ *of seed examples for a relation* R_i *and an input set* $S \subseteq \Sigma^*$ *of sequences over tokens from the alphabet* Σ*, a* basic pattern $p \in \Sigma^*$ *is a sequence such that* $e1\ p\ e2$ *occurs in* S *for at least one pair* $(e1, e2) \in SX(R_i)$*. A* lifted pattern *is a pattern* p *where some tokens in* p *(with POS tags in a specified set* T *of tags) are replaced by those tags. An n-gram itemset pattern*

Pattern	Relation	Computed confidence
{PhD dissertation at}	graduatedFrom	1.0
{doctorate at the, in, with}	graduatedFrom	0.57
{doctorate at the, in, with}	hasAcademicAdvisor	0.43
{attended the}	graduatedFrom	0.96
{dissertation supervised by}	hasAcademicAdvisor	1.0
{academic career at the }	facultyAt	1.0
{was awarded the, along with}	hasCollaborator	0.44
{and associate}	hasCollaborator	1.0
{is a fellow of}	hasProfessionalAffiliation	1.0
{is a member of the }	hasProfessionalAffiliation	0.33

Table 1: Example seed patterns with computed confidence values

is a set $Q \subseteq (\Sigma \cup T)^*$ for which there is at least one sequence $s \in S$ that can be written as $s = h\ e1\ p\ e2\ t$ with a seed example $(e1, e2) \in SX(R_i)$ and $h, p, t \in \Sigma^*$, such that for all $q \in Q$ the length of q is at most n tokens and q is a subsequence of p.

Note that the occurrence of a pattern with a seed example for R_i does not yet mean that the corresponding sentence actually states anything about the relation R_i; the co-occurrence could be mere coincidence. To assess the goodness of a pattern, in particular, the significance of n-gram-itemset patterns, we gather frequency statistics about co-occurrence with seed examples and counterexamples.

DEFINITION 3 (SUPPORT, CONFIDENCE). *For sets $SX(R_i)$ and $CX(R_i)$ of seed examples and counterexamples and an input set $S \subseteq \Sigma^*$, a basic (or lifted) pattern q has*

$$support(q) = \frac{|\{s \in S | \exists (e1, e2) \in SX(R_i) : q, e1, e2 \text{ occur in } s\}|}{|S|}$$

and

$$confidence(q) = \frac{|\{s \in S | \exists (e1, e2) \in SX(R_i) : q, e1, e2 \text{ occur in } s\}|}{|\{s \in S | \exists (e1, e2) \in SX(R_i) \cup CX(R_i) : q, e1, e2 \text{ occur in } s\}|}$$

Pattern analysis computes n-gram-itemset patterns with support and confidence values.

DEFINITION 4 (SEED PATTERN, SEED-PATTERN WEIGHT). *An n-gram-itemset pattern q, for given $SX(R_i)$, $CX(R_i)$, and input set $S \subseteq \Sigma^*$, is called a seed pattern if both $support(q)$ and $confidence(q)$ are above the specified thresholds. Pattern q is associated with a seed-pattern weight, set to:*

$$weight(q) = support(q) \times confidence(q).$$

The weight of a pattern q can be interpreted as the probability that we encounter q in a newly seen sentence (support factor) *and* that it is indeed good evidence for a fact in R_i (confidence factor). Table 1 depicts some example patterns we have learned along with their computed confidence values, for a few selected relation types.

2.2 New Patterns and Fact Candidates

Seed patterns are used to discover fact candidates as well as new patterns (which in turn can be applied to gather additional evidence for fact candidates and/or discover even more patterns). We consider all sequences $s \in S$ that contain two entities (x, y) of appropriate types for R_i (e.g., person names for the *hasAcademicAdvisor* relation) and whose subsequence p in $s = h\ x\ p\ y\ t$ in between the two entities x, y approximately matches one of the seed patterns. We do not insist on exact matching of a seed pattern q, as this would require presence of all n-grams of q in p. With the limitation of bootstrapping the entire extraction process by a few seed examples, such exact matching would be overly restrictive. This way, we can also discover new n-grams of interest, if they co-occur in a new pattern together with n-grams known from seed patterns.

The approximate matching of p against all seed patterns q is efficiently implemented by lookups in an n-gram index constructed from the seed patterns. This gives us also an efficient way of computing a matching score by the similarity between p and q.

DEFINITION 5 (PATTERN-MATCHING SIMILARITY). *A new pattern p in input sequence $s = h\ x\ p\ y\ t$ has similarity $sim(p, q)$ with seed pattern q, based on their Jaccard coefficient:*

$$\begin{aligned} sim(p,q) &= Jaccard(p,q) \\ &= \frac{|\{\text{n-grams} \in p\} \cap \{\text{n-grams} \in q\}|}{|\{\text{n-grams} \in p\} \cup \{\text{n-grams} \in q\}|} \end{aligned}$$

We process all input sequences $s = h\ x\ p\ y\ t$ this way, and again perform frequent-itemset mining to concentrate on the set of new patterns to those with support above a specified threshold (which does not need to be the same threshold as in the previous mining step for seed patterns only). Note that we cannot use confidence for thresholding here, because these patterns co-occur with fact candidates whose validity we do not know yet. The output of this step is a multi-set of weighted triples $(x, y, p)[w]$ where (x, y) is a fact candidate, p is an n-gram-itemset pattern, and w is the highest pattern-matching similarity of p with any seed pattern q. (Note that it is a multi-set rather than a set because the very same candidate could be seen in different sources.)

DEFINITION 6 (CANDIDATE MULTI-SET). *For given input set S and seed-pattern set Q, the fact-pattern candidate multi-set $C(S, P)$ is:*

$$C(S,P) = \{(x,y,p)[w] \quad | \quad \exists s \in S : s \text{ contains } x, y, p \ \wedge \ w = max\{sim(p,q) \times weight(q) | q \in Q\}\}$$

This candidate multi-set C is now grouped in two different ways:

1. by fact candidates (x, y), with an aggregated weight
$$weight(x,y) = \sum \{w | (x,y,p)[w] \in C\}$$
2. by new n-gram-itemset patterns p, with an aggregated weight
$$weight(p) = \sum \{w | (x,y,p)[w] \in C\}$$

We can interpret these weights as the aggregated evidence that (x, y) is a valid fact and p is a good pattern for further extraction steps for R_i. It is important to note that the two weights are different, as the aggregations are computed over different sets. Further note that the summations include also newly found patterns, not just seed patterns. Summations are over multi-sets, so that multiple occurrences of the same candidates in different sources are rewarded. The resulting weights are not normalized; they may be viewed as the expected number of good facts from a given pattern and expected number of good pattern occurrences for a given fact, respectively.

3. REASONING FRAMEWORK

The logical reasoning phase serves to ensure mutual consistency of facts that are ultimately accepted as more likely to be true among the fact candidates. We apply SOFIE's basic reasoning model, which makes use of propositional first-order logic formulas, referred to as *rules*. These rules represent semantic knowledge that needs to be upheld to ensure the correctness of extracted facts.

SOFIE introduced a number of general rules that can be applied across relations and domains. One of the rules that relate text patterns to fact candidates states that, for a target relation R_i, if a pattern p occurs with an entity pair (x, y), then x and y stand in the R_i relation. The rule can be stated as follows:

$$\text{R1:} \quad occurs(p, x, y) \quad \wedge \quad expresses(p, R_i) \quad \Rightarrow \quad R_i(x, y)$$

For example, if the relation *hasAcademicAdvisor* has a seed pattern "studied under" and we encounter a sequence "Barbara Liskov studied under John McCarthy" then a new candidate fact *hasAcademicAdvisor(Barbara Liskov, John McCarthy)* is generated.

A similar rule states that, if it is known that an entity pair (x, y) stands in the relation R_i and we encounter a pattern p with (x, y), then p expresses the relation and can be written as follows:

$$\text{R2:} \quad R_i(x, y) \quad \wedge \quad occurs(p, x, y) \quad \Rightarrow \quad expresses(p, R_i)$$

First-order rules are grounded by replacing the placeholders with entity names, which results in weighted clauses in conjunctive normal form (CNF). The task of the logical reasoner is thus to assign truth values to fact candidates with the objective of maximizing the sum of weights among rules that are satisfied. Thus the entire problem can be cast as a weighted maximum satisfiability problem (Max-Sat) (see [22] for a complete list of rules used in SOFIE, including *entity disambiguation* and *functional properties* of relations).

3.1 Statistically Weighted Clauses

Our approach extends the original SOFIE reasoning framework in an important direction: introducing a more informed *weighting scheme* for the clauses which serve as input to the weighted Max-Sat solver.

SOFIE already statically assigned weights to clauses generated from rules. However, not all candidate facts are equally important. Candidate facts may have been extracted from different patterns which do not have the same strength, and thus their weights should also include occurrence statistics.

The desired impact of weights can be seen from two main perspectives in this setting. First, in the case of functional relations where only one of a pair of candidate facts can be true, the one that has better evidence will have higher weight, thus guiding the Max-Sat solver to the correct solution. Second, and more generally, the weighted Max-Sat problem is NP-hard, and it is typically solved using approximation algorithms. The solutions delivered by approximation algorithms are not necessarily optimal and may include some randomization. Thus high weights serve an important role in guiding the solver into the correct direction. Consequently, an approach relying on fixed weights lacks important information represented by weights derived from evidence in the data.

We compute the weights of clauses generated by grounding the rules on the candidate multi-set $C = \{(x, y, p)[w]\}$, using the aggregated weights $w(x, y)$ and $w(p)$ constructed in the pattern-analysis phase. Thus, for rules $R1$ and $R2$, we generate clauses with their weights set as follows:

$$\text{Rule } R1: \quad \sum\{w|(\alpha, \beta, \pi)[w] \in C \quad \alpha = x, \beta = y, \pi = p\} \times w(p).$$

$$\text{Rule } R2: \quad \sum\{w|(\alpha, \beta, \pi)[w] \in C \quad \alpha = x, \beta = y, \pi = p\} \times w(x, y).$$

4. EXPERIMENTAL RESULTS

We carried out experiments to extract academia-related information. The corpus was generated by crawling the homepages of the most prolific authors from DBLP, then augmenting these with articles of scientists from Wikipedia. Additionally, names of scientists were used to query Google for further documents. The resulting corpus consists of 87,470 documents. The knowledge base used in the experiments is the YAGO [21] ontology.

To quantify how the various aspects of our approach, which we refer to as PROSPERA, affect performance, we evaluated the *hasAvademicAdvisor* relation, Table 2 shows the results.

PROSPERA has the highest recall at high precision. SOFIE produced many extractions but with low precision. The *hasAcademicAdvisor* relation is not straightforward to extract because it can be expressed by patterns that may be misleading, for example the pattern, "*x* worked with *y*" may or may not indicate that *y* was the doctoral advisor of *x*. These misled SOFIE but the robustness of PROSPERA withstood them as it identifies these cases through pattern occurrence statistics. The two systems extracted more or less the same facts, however each system also extracted some tupples the other did not. For example both systems extracted the pair *(Jeffrey Shallit, Manuel Blum)*, but only PROSPERA extracted the pair *(Serge Lang, Emil Artin)*, whereas only SOFIE extracted the pair *(Ravi Sethi, Jeffrey Ullman)*.

Consistency checking plays a significant role in ensuring high precision as reflected in the results of the PROSPERA-NoReasoner method. The reasoner thus acts as a well-placed filter, performing type checking as well as ensuring that the logical rules are upheld. The PROSPERA-NoCounterExamples method shows the impact of counter-examples. Without the counter-examples, even weak patterns may lead to extractions, as can be seen, this degrades precision slightly. The PROSPERA-Unweighted method shows the impact of the weights. Disregarding pattern weights all together results in slightly reduced recall, this is attributed to the fact that the weights guide the reasoner to the correct answer, and without them there may be misleading cases, causing the reasoner to reject facts that might be true.

The number of fact candidates reflect the number of candidates passed on to the reasoner. It can be seen in Table 2 that all variations of PROSPERA provide considerable pruning of fact candidates and

Method	# extractions	Precision	# fact candidates	Runtime (min)
SOFIE	1,845	22%	105,016	122
PROSPERA	372	83%	22,340	35
PROSPERA-NoReasoner	22,340	1.9%	n/a	22
PROSPERA-NoCounterExamples	404	79%	24,328	35
PROSPERA-Unweighted	338	83%	24,328	35

Table 2: Performance for the *hasAcademicAdvisor* relation

this results in shorter execution times compared to SOFIE. Comparisons were also carried out using various other relations, the results are shown in Table 3.

	Relation	# extractions	Precision
PROS-PERA	hasAcademicAdvisor	372	83%
	hasCollaborator	122	91%
	facultyAt	1,274	94%
	graduatedFrom	1,310	89%
	hasProfessionalAffiliation	107	90%
	hasWonPrize	1,309	99%
SOFIE	hasAcademicAdvisor	1,845	22%
	hasCollaborator	8	100%
	facultyAt	3,147	49%
	graduatedFrom	5,088	56%
	hasProfessionalAffiliation	7	100%
	hasWonPrize	1,553	99%

Table 3: Performance for all relations

PROSPERA has high precision across all the relations whereas SOFIE's precision varies widely across relations. Furthermore, for the *hasCollaborator* and *hasProfessionalAffiliation* relations, PROSPERA has much higher recall, this is because these two relations had the fewest number of seeds and the generilization capability of patterns in PROSPERA enabled further instances to be discovered without requiring exact matches between patterns. For the *graduatedFrom* and *facultyAt* relations, SOFIE's recall suffers because these two relations have overlapping instances, since a person can be a faculty remember at the institution where they graduated from. Here again PROSPERA is robust to this scenario. PROSPERA has the same precision as SOFIE for the *hasWonPrize* relation. This is because this relation is typically expressed with the same patterns and thus the exact pattern matching in SOFIE works well. PROSPERA has a slightly lower recall than SOFIE for this relation, primarily because certain weak seed patterns that do not meet the requirements in PROSPERA mean that instances expressed with similar patterns are not discovered.

To quantify the impact of the number of seeds used, we evaluated performance of the *hasAcademicAdvisor* for varying numbers of seeds. Table 4 shows the results.

	# seeds	# extractions	Precision
PROSPERA	15	48	89%
	50	115	81%
	212	372	83%
SOFIE	15	4	100%
	50	131	31%
	212	1,845	22%

Table 4: Precision and recall for the *hasAcademicAdvisor* for varying numbers of seeds

PROSPERA has high precision and reasonable recall even when only a small number of seeds are used. SOFIE on the other hand has high precision when only a few seeds are used, however, in this case SOFIE's recall is extremely low. When many seeds are used, there is a high chance of noisy patterns occurring with a few instances, for a relation like the *hasAcademicAdvisor* relation, for this reason SOFIE's precision degrades, PROSPERA on the other hand still achieves high precision further demonstrating robustness.

We also compared PROSPERA to the SNOWBALL[1] system, using one of their experiments (for which a non-copyright-protected part of the data was available[22]). In this test run, with the goal of extracting the headquarters of companies, PROSPERA reached 85% for a recall of 42 newly extracted, correct facts, SOFIE also extracted 42 correct facts with 91% precision, whereas the original SNOWBALL reached 57% precision with 37 correct facts.

In general, not many information-extraction systems are publicly available for comparative experiments. Moreover, many results in the literature cannot be reproduced in a full experiment because the papers do not disclose sufficient details about their experiments (e.g., the datasets and chosen seeds). Therefore, we cannot present a broader set of comparisons with other systems.

5. RELATED WORK

Using various forms of pattern matching for fact extraction from natural-language documents has a long history in the NLP, AI and DB communities, dating back to the work by Hearst [15]. Hearst patterns [15] were the first part-of-speech-enriched regular expressions (so-called *lexico-syntactic* patterns) which aim to identify instances of predefined relationship types from free text. Hearst patterns are hand-crafted; for arbitrary target relations (such as *hasCollaborator* or *hasAcademicAdvisor*) it would be difficult to come up with an exhaustive set of expressive yet accurate patterns.

Seminal work by Brin [4] was centered around the *duality of facts and patterns* which refers to the iterative mutual enrichment between patterns and facts, whereby seeds can be used to find new patterns and new patterns can be used to find more seeds. The iterative process outlined above is powerful but it is susceptible to drifting away from its target. While high recall is easily possible, the precision would often be unacceptable. In our approach, reasoning serves to ensure high precision.

KnowItAll [14], and Text2Onto [9] improved the statistical assessment of fact candidates and patterns in a variety of ways, regarding robustness (lower false-positive rate while still retaining high recall), expressiveness (e.g., by adding part-of-speech tagging and other NLP-based features), and efficiency (lower-cost estimates the statistical measures). The LEILA approach [20] uses dependency-parsing-based features for boosted precision, and also extends Brin's bootstrapping technique by incorporating both positive and negative seeds. TextRunner [27] extended the pattern-fact bootstrapping paradigm to *Open IE*, where the harvesting is not focused on a particular relation but considers all relationships expressed in verbal phrases. This relaxation however makes it very difficult to apply any form of consistency reasoning on the extracted data.

More recent work on rule-based fact gathering is based on DB-style *declarative IE*. [10, 16, 19] have shown how to combine query processing for extraction and consistency-centered inferencing into a unified framework. A very nice showcase is the (at least largely) automated construction and maintenance of the *DBLife* community portal (*dblife.cs.wisc.edu*), which is based on the *Cimple* tool suite [10]. Rule-based fact extraction has also been customized to Wikipedia as a knowledge source, primarily to exploit the great asset provided by infoboxes. DBpedia [3] has pioneered the massive extraction of infobox facts. It uses simple, recall-oriented techniques and essentially places all attribute-value pairs into its knowledge base as they are. YAGO [21], on the other hand, uses a suite of carefully designed rules for frequently used infobox attributes to extract and normalize the corresponding values.

Learning and reasoning-based methods include the StatSnowball [28], a powerful machinery for fact harvesting that makes intensive use of Markov Logic Networks (MLNs) and Conditional Random Fields (CRFs). Moreover, the Kylin/KOG framework [24, 26] is an interesting application of MLNs which also aims to infer "missing infobox values" in Wikipedia. SOFIE [22] combines pattern learning with consistency reasoning cast as a weighted max-sat problem to disambiguate entities and extract facts from from text. In the ReadTheWeb project [7], semi-supervised learning ensembles have been combined with constraints for extracting entities and facts from a large Web corpus.

6. CONCLUSIONS

Previous work in the SOFIE system introduced coupling pattern matching with logical reasoning for information extaction. The pattern matching used there is based on exact matches between patterns and does not evaluate the strength of patterns. This has negative implications for recall and precision. We introduced an approach for associating patterns with quality measures. Our approach allowed us to prune and bound the clauses that are passed on to the reasoner. Experiments demonstrated performance improvements on precision, recall and runtimes.

7. REFERENCES

[1] E. Agichtein, L. Gravano. Snowball: extracting relations from large plain-text collections. *ACM DL*, 2000.

[2] R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in large databases. *SIGMOD*, 1993.

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

[4] S. Brin. Extracting patterns and relations from the world wide web. *WebDB*, 1998.

[5] M. J. Cafarella. Extracting and querying a comprehensive web database. *CIDR*, 2009.

[6] A. Carlson, J. Betteridge, E. R. Hruschka, T. M. Mitchell. Coupling semi-supervised learning of categories and relations. *NAACL-HLT SemiSupLearn*, 2009.

[7] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka Jr., T. M. Mitchell. Coupled semi-supervised learning for information extraction. *WSDM*, 2010.

[8] M.-W. Chang, L.-A. Ratinov, N. Rizzolo, D. Roth. Learning and inference with constraints. *AAAI*, , 2008.

[9] P. Cimiano and J. Völker. Text2onto - a framework for ontology learning and data-driven change discovery. *NLDB*, 2005.

[10] P. DeRose, W. Shen, F. Chen, A. Doan, R. Ramakrishnan. Building structured web community portals: A top-down, compositional, and incremental approach. *VLDB*, 2007.

[11] P. DeRose, W. Shen, F. Chen, Y. Lee, D. Burdick, A. Doan, R. Ramakrishnan. DBlife: A community information management platform for the database research community. *CIDR*, 2007.

[12] A. Doan, L. Gravano, R. Ramakrishnan, S. Vaithyanathan. (Eds.). Special issue on information extraction. *SIGMOD Record*, 37(4), 2008.

[13] O. Etzioni, M. Banko, S. Soderland, D. S. Weld. Open information extraction from the web. *Commun. ACM*, 51(12), 2008.

[14] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, A. Yates. Web-scale information extraction in KnowItAll. *WWW*, 2004.

[15] M. A. Hearst. Automatic acquisition of hyponyms from large text corpora. *COLING*, 1992.

[16] F. Reiss, S. Raghavan, R. Krishnamurthy, H. Zhu, S. Vaithyanathan. An algebraic approach to rule-based information extraction. *ICDE*, 2008.

[17] M. Richardson, P. Domingos. Markov Logic Networks. *Machine Learning*, 2006.

[18] S. Sarawagi. Information extraction. *Foundations and Trends in Databases*, 1(3), 2008.

[19] W. Shen, A. Doan, J. F. Naughton, R. Ramakrishnan. Declarative information extraction using Datalog with embedded extraction predicates. *VLDB*, 2007.

[20] F. M. Suchanek, G. Ifrim, G. Weikum. LEILA: Learning to extract information by linguistic analysis. *2nd Workshop on Ontology Learning and Population*, 2006.

[21] F. M. Suchanek, G. Kasneci, G. Weikum. YAGO: A large ontology from Wikipedia and WordNet. *J. Web Sem.*, 6(3), 2008.

[22] F. M. Suchanek, M. Sozio, G. Weikum. SOFIE: a self-organizing framework for information extraction. *WWW*, 2009.

[23] G. Tummarello. SIG.MA: Live views on the web of data. *WWW*, 2010.

[24] D. S. Weld, R. Hoffmann, F. Wu. Using Wikipedia to bootstrap open information extraction. *SIGMOD Record*, 37(4), 2008.

[25] D. S. Weld, F. Wu, E. Adar, S. Amershi, J. Fogarty, R. Hoffmann, K. Patel, M. Skinner. Intelligence in Wikipedia. *AAAI*, 2008.

[26] F. Wu, D. S. Weld. Automatically refining the Wikipedia infobox ontology. *WWW*, 2008.

[27] A. Yates, M. Banko, M. Broadhead, M. J. Cafarella, O. Etzioni, S. Soderland. TextRunner: Open information extraction on the web. *HLT-NAACL*, 2007.

[28] J. Zhu, Z. Nie, X. Liu, B. Zhang, J.-R. Wen. StatSnowball: a statistical approach to extracting entity relationships. *WWW*, 2009.