Exploiting Hyponymy in Extracting Relations and Enhancing Ontologies

Bhushan Mandhani Stephen Soderland
Dept. of Computer Science
University of Washington
Seattle USA 98195

Abstract

Relation extraction systems typically rely on local lexical and syntactic features as evidence. Recent work suggests that for a given relation, there might exist certain patterns which, if present in the graph of relationships between objects, provide additional evidence for that relation. While these relational patterns can be very useful, obtaining them on a per-relation basis can be difficult. We propose template relational patterns based on the hyponymy relation. These patterns are applicable for all relations. Existing resources like WordNet are a rich and reliable source of hyponymy relationships between entities. We present techniques for making use of these template patterns for extracting relationships and incorporating them into the ontology. Our experiments show performance improvements in both tasks.

1 Introduction

The problem of extracting relationships between entities from a corpus has recently received a lot of attention. It is an important information extraction task, and has many uses. It is a key component in building ontologies and structured knowledge bases from the web [6]. It also has uses in more traditional applications like question answering and information retrieval. A variety of techniques have been proposed for relation extraction. They primarily use lexical and syntactic information as evidence. A range of classifiers and probabilistic models have been used for deciding whether or not the relation holds, given the evidence.

However, there has been little work on employing semantic evidence in relation extraction. The most common such feature seems to be the class of the entities involved in the relationship [8, 3]. This feature is available only when the entity recognizer also assigns class labels to the entities. [2] propose a new source of semantic evidence, which they call "relational patterns". The idea here is that for a given relation, there might exist a pattern of relationships involving other relations, such that if this pattern holds between two entities, then the original relation is also likely to hold. We illustrate this with an example. Suppose our set of relations includes "Mother", "Father" and "Wife". A relational pattern predictive of entity X having mother Z would be that X has father Y and Y has wife Z. So, if these last two relationships are known, we have additional evidence for the "Mother" relation between X and Z. Note that this evidence is not only semantic, but also global. It is based on relationships extracted from elsewhere in the corpus.

It is instructive to express the semantics of relational patterns more formally. They represent Horn clauses in first-order logic. The above pattern can be written as follows:

$$\forall x, y, z \ \mathrm{Father}(x, y) \wedge \mathrm{Wife}(y, z) \Rightarrow \mathrm{Mother}(x, z)$$

Unlike first-order logic, these have a probabilistic interpretation. When the clause body fires, the head of the clause can't be "inferred". Instead, we have more evidence that it is true. We formalize this later in the paper.

Relational patterns can help improve precision and recall in relation extraction. [2] obtained improvements in F-Measure when they incorporated them into their extraction model. However, obtaining them on a per-relation basis is a challenge, especially when the number of relations is very large. Mining the set of extracted relationships seems like a promising approach. However, for a web-scale system with millions of extracted relationships, learning such rules may not be computationally feasible.

We propose Taxonomic Relational Patterns (TRPs), which are template patterns that can be applied for each distinct relation. Entities in extracted relationships are often mapped to objects in an existing ontology or semantic taxonomy like WordNet. This provides us with an accurate and largely complete set of hyponymy relationships between these entities. TRPs look to exploit these relationships. We present three TRPs in this paper.

Formula 1 presents our first TRP. "Relation" stands for an arbitrary relation, and our notational convention is that variables not explicitly quantified are universally quantified.

$$\mathrm{Hyponym}(x', x) \wedge \mathrm{Relation}(x, y) \Rightarrow \mathrm{Relation}(x', y) \quad (1)$$

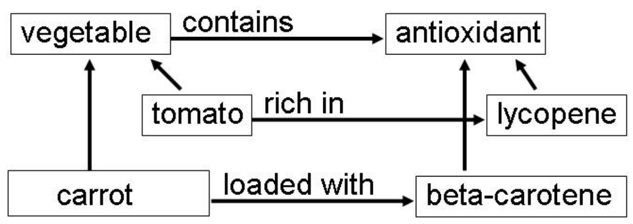

Figure 1. Relationships between the entities in the example discussed in text

This TRP is based on simple inheritance reasoning. For example, suppose we set "Relation" to the "Contains" relation. We respectively set the variables x', x, y to "tomato", "vegetable" and "antioxidant". In Figure 1, the upward arrows indicate hyponymy while the horizontal arrows indicate extractions. Suppose we have extracted *Contains(vegetable,antioxidant)*, and are now considering local evidence for *Contains(tomato,antioxidant)*. Formula 1 says that the former provides additional evidence for the latter.

The exact logical formulation specializes Formula 1 and is discussed later. In particular, it specifies that the relationship in the head of the clause must have already been extracted i.e., we have evidence for it somewhere in the corpus. We thus use TRPs as a source of additional evidence for extracted relationships. The proposed TRPs can be combined with any Horn clauses that we learn from the extractions. This makes them more general and widely applicable than Formula 1 suggests.

We use TRPs in two different but related tasks. The first is that of relation extraction. Our extraction model is a classifier. Adding features derived from the TRPs gave us significant improvements in the precision–recall curve. The second task is that of semantic taxonomy induction. A semantic taxonomy T is defined as a set of pairwise relations over some set of objects. Given a relation Rel, an object pair (i, j) may or may not be related according to Rel. [9] formulates the problem as that of assigning truth values to all extracted relationships, given the probabilities assigned by the extraction model, constraints that the resulting taxonomy must satisfy, and a criterion to optimize. Our formulation is similar but more general. We allow constraints that are "soft" and have attached weights. We later describe our framework formally, and it represents one of our main contributions. In experiments, upon adding the TRPs as soft constraints, the taxonomy obtained had a higher F-Measure.

The rest of this paper is organized as follows. We complete our discussion of TRPs in Section 2. Sections 3 and 4 respectively cover relation extraction and taxonomy induction. Section 5 discusses related work. We present experimental results in Section 6, and conclude in Section 7.

2 Taxonomic Relational Patterns

2.1 Logical Framework

We first introduce the framework in first-order logic in which we will develop the TRPs. Entities in relationships map to objects in a semantic taxonomy, which in turn are objects in our logical domain. While it would be natural to map relations to logical predicates, we want to be able to specify properties of relationships, such as whether they have been extracted or not. This won't be possible if relations are predicates. Relations too are thus made objects in our logical domain, with the exception of hyponymy and siblinghood which are made predicates for expositional clarity. We list our predicates below:

- Extracted(x, rel, y) is true if the relationship rel between entities x and y has been extracted from text.
- True(x, rel, y) is true if the relationship rel actually holds between entities x and y. We want to determine the truth values of those ground atoms which correspond to extractions.
- Hyponym(x, y) is true if x is a hyponym of y in the base ontology.
- Siblings(x, y) is true if x is a sibling of y in the base ontology i.e., they have the same direct hypernym.
- Entails(rel, rel') is true if relation rel entails rel'.

2.2 Entailments Between Relations

Learning arbitrary rules from extracted relationships is a separate problem that we don't address. However, we actually do learn rules having a specific form. We show an example below:

$$\forall x, y \text{ True}(x, \textit{rich-in}, y) \Rightarrow \text{True}(x, \textit{contains}, y)$$

These rules basically represent *entailments* between relations. Entails$(\textit{rich-in}, \textit{contains})$ is true if and only if the formula above is true. We employed simple heuristics for learning entailments. We will skip the details as the focus here is on how we can make use of these rather than how we can learn them.

Suppose we have extracted the relationships (*tomato,rich-in,lycopene*) and (*tomato,contains,lycopene*). If we have also learned that Entails$(\textit{rich-in}, \textit{contains})$ holds, the former relationship implies the latter. If it is true, it represents additional evidence for the latter. The Horn clause below captures this reasoning. The first literal has been included to ensure that we only consider relationships that have been extracted.

$$\text{Extracted}(x, rel, y) \wedge \text{True}(x, rel', y) \wedge \text{Entails}(rel', rel) \Rightarrow \text{True}(x, rel, y) \quad (2)$$

2.3 The Hyponymy TRP

We first present an example motivating the hyponymy TRP. We once again use the entities shown in Figure 1. Suppose we have extracted the relationships (*vegetable,rich-in,antioxidant*) and (*tomato,rich-in,lycopene*), which we denote by R_1 and R_2 respectively. R_1 says that each member of the "vegetable" class is rich in some antioxidant. R_2 says that a particular vegetable is rich in a particular antioxidant. It seems reasonable to consider R_1, if true, to be additional evidence that R_2 is true. The Horn clause below captures this reasoning. When grounding this clause to our example, (x, rel, y) would bind to R_2, and (x', rel', y') to R_1.

$$\text{Extracted}(x, rel, y) \wedge \text{True}(x', rel', y') \wedge \text{Hyponym}(x, x') \wedge \text{Hyponym}(y, y') \wedge rel = rel' \Rightarrow \text{True}(x, rel, y) \quad (3)$$

In fact, we can combine this kind of reasoning with the entailments between relations that we have learned, to make it more general. We can replace the requirement of equality between relations rel and rel' by entailment in either direction. Note that we adopt a probabilistic semantics for these Horn clauses which is discussed later in the paper. The system learns from labeled training data the degree to which any given rule is actually true. Thus, rules do not need to be exact like they would in a purely logical system.

2.4 The Siblinghood TRP

Our final TRP is based on siblinghood, which we defined in Section 2.1. For many relations Rel, there exist certain class pairs (C_1, C_2) such that many or all true extractions of Rel are of the form (X, Rel, Y) where X and Y are particular members of C_1 and C_2 respectively. For example, we could have many true extractions of the form (*Person X,Born-In,City Y*). For the relationship (X, Rel, Y), we want to consider other true relationships (X', Rel, Y') where (X, X') and (Y, Y') are siblings, as additional evidence. This leads us to the following:

$$\text{Extracted}(x, rel, y) \wedge \text{True}(x', rel', y') \wedge \text{Siblings}(x, x') \wedge \text{Siblings}(y, y') \wedge rel = rel' \Rightarrow \text{True}(x, rel, y) \quad (4)$$

Like for the hyponymy TRP, we can generalize this reasoning to include entailments between rel and rel'.

3 Using TRPs in Relation Extraction

This section describes how evidence from TRPs can help improve precision and recall in relation extraction. Our extraction system [1] relies on local evidence, including syntactic features coming from POS (part-of-speech) tags. The exact mechanics of its working is not relevant here. It gives us a set S of extractions of the form (X, Rel, Y), which forms our input. The source sentence and POS tags are also available for each extraction. We address the problem of identifying true extractions in S. In particular, we want to assign probabilities rather than true/false labels, so that it is possible to tradeoff precision with recall.

We will use a classifier for this task. Let (X, Rel, Y) be an example extraction, which we denote R. The baseline feature set F_0 consists of POS tags of Rel, bigrams of these tags, the number of words in Rel, and the number of times R was extracted from the corpus.

We now enhance F_0 with features derived from the different types of additional evidence we discussed in Section 2. Truth values of atoms of the "True" predicate are unknown, and we will approximate them with corresponding atoms of "Extracted". We first consider evidence from entailments. In Formula 2, we bind (x, rel, y) to R. Suppose R_0 is some extraction (X, Rel_0, Y) such that binding rel' to Rel_0 satisfies the clause body. R_0 thus supports R through entailment. The more such R_0, the more the evidence for R. We add to F_0 as a feature the number of such extractions R_0 giving us the feature set F_1.

To be able to use evidence from TRPs, we first need to map all entities X, Y in the extractions to a taxonomy like WordNet. We incorporate evidence from TRPs similarly to how we did for entailments. In Formula 3, which represents the Hyponymy TRP, we bind (x, rel, y) to R. Suppose R' is some extraction (X', Rel', Y') such that binding (x', rel', y') to R' satisfies the clause body. R' thus supports R through the Hyponymy TRP. We add to F_0 as a feature the number of such extractions R'. We add similar features corresponding to the other two TRPs. We denote the resulting feature set F_2. In Section 6, we compare the precision–recall curves obtained for each of F_0, F_1 and F_2.

4 Semantic Taxonomy Induction

4.1 Problem Framework

Our formulation of the taxonomy induction problem is along the lines of [9]. Our input is still the set S of extracted relationships, with a probability assigned to each. The extracted entities represent objects in an existing semantic taxonomy. *The goal is to choose a subset of S to add to this taxonomy such that the resulting taxonomy T maximizes a criterion which includes a measure of how well T satisfies a* *set C of constraints expressed in first-order logic.* Choosing a subset of S is equivalent to assigning true/false labels to its members. The key difference from the relation extraction task is that these labels can no longer be assigned using independent classifiers.

While [9] have hard constraints, we allow soft constraints. We adopt a probabilistic semantics for them that is the same as in Markov Logic Networks (MLNs) [7]. This gives us a natural generalization of the criterion proposed in [9]. In MLNs, formulae in first-order logic have attached weights. While our discussion is self-contained, the reader is encouraged to see [7] to learn more about MLNs and better understand how our formulation arises out of them.

We will define a probability distribution over all possible taxonomies. *The criterion we will maximize is $P(T)$, the probability of T.* Let $S = \{R_1, \ldots, R_n\}$, with $W(R_i)$ denoting the probability assigned to relationship R_i by the extraction model. Let $C = \{C_1, \ldots, C_k\}$, with $W(C_i)$ denoting the weight for constraint C_i. Setting each variable in C_i to some object in our logical domain gives us a particular grounded formula (or grounding). Let $G = \{G_1, \ldots, G_m\}$ be the set of all possible groundings of members of C. If some G_j was obtained from some C_i, we set its weight $W(G_j)$ to be $W(C_i)$. We will use the notation $T \models G_j$ to indicate that T satisfies formula G_j. The probability for taxonomy T is shown below (Z is a normalization constant):

$$P(T) = \frac{1}{Z} \prod_{R_i \in T} W(R_i) \cdot \prod_{R_i \notin T} (1 - W(R_i)) \cdot \prod_{T \models G_j} W(G_j) \cdot \prod_{T \not\models G_j} (1 - W(G_j)) \tag{5}$$

We now explain the connection with MLNs. The above formula comes from defining a MLN with two kinds of formulae. The first comes from the set S. For each R_i representing some extraction (X, Rel, Y), we have the formula True(X, Rel, Y) with weight $W(R_i)$. The second kind are just the specified constraints C.

4.2 Using TRPs in Taxonomy Induction

We set the weight $W(R_i)$ for an extraction R_i to the probability assigned to it by the extraction model. This allows us to plug in the local evidence for R_i into our model. In our experiments, we obtained this probability by using a classifier with the feature set F_0.

We will incorporate the different types of additional evidence discussed in Section 2 by adding them as constraints. For the baseline taxonomy T_0, the set C of constraints is empty. For the taxonomy T_1, we include evidence from entailments. C now consists of Formula 2, with the corresponding weight learned from labeled training data. For the taxonomy T_2, we incorporate evidence from TRPs. C now consists of Formulae 3, 4, and 1 (actually, a rewrite of 1). The weights are again learned from training data. Note that a single constraint formula will result in a large number of grounded constraints. Now that we have a more powerful model, we no longer need to approximate atoms of the unknown "True" predicate with those of "Extracted".

Taking the logarithm on both sides of (5), we can see that maximizing $P(T)$ reduces to a variant of the weighted satisfiability problem. The clauses here are of the form R_i and G_j. We want to try and satisfy those with weight greater than 0.5, and keep the rest unsatisfied. We use MaxWalkSat [4] for finding the T that maximizes $P(T)$.

5 Related Work

Relation extraction techniques mostly rely on lexical and syntactic information as evidence. Syntactic features are obtained from parsers and POS taggers. Extraction models used include a range of classifiers, kernel methods, maximum entropy models [3], conditional random fields [2], etc.

There are two key differences between our formulation of the taxonomy induction problem and that presented in [9]. Firstly, they have hard constraints. Effectively, they have $W(G_j) = 1$ for each grounding G_j, so that the contribution of the constraints term to $P(T)$ in Formula 5 is exactly 0 or 1. They thus don't explicitly include this term. Secondly, while we maximize the posterior probability $P(T|E)$ where E indicates all the local evidence for extracted relationships, they maximize the likelihood $P(E|T)$.

6 Experiments

6.1 Experimental Setup

Our input set S was a subset of a set of relationships extracted from a corpus of over a million web pages on health and nutrition. We picked those extractions where one of the entities was a WordNet hyponym of "edible fruit" or "vegetable" in some sense, and the other was present in WordNet. After merging duplicates, this gave us 272,960 distinct relationships spanning 6850 distinct relations.

Mapping to WordNet: We used WordNet as the semantic taxonomy being extended with extracted relationships. Mapping entities to WordNet synsets is basically the word sense disambiguation (WSD) problem. For each extraction (X, Rel, Y), we know the sentences from which this relationship was extracted. Our WSD algorithm uses the context words for X, and distributional similarity, to choose the best sense for X. We skip further details. A WSD accuracy of 73% was obtained on a sample of 100 correct extractions.

Learning Entailments Between Relations: For a relation Rel, we define the ordered pair (X, Y) to be an instance

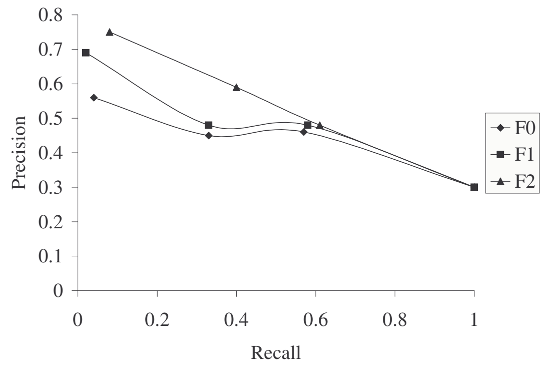

Figure 2. Precision–Recall Tradeoff for Different Feature Sets

	Precision	Recall	F-Measure
T_0	0.54	0.47	0.50
T_1	0.60	0.42	0.49
T_2	0.68	0.42	0.52

Table 1. F-Measure for Different Taxonomies

of Rel if $(X, Rel, Y) \in S$. Suppose Rel_1, Rel_2 have instance sets I_1, I_2 respectively. If Entails(Rel_1, Rel_2) holds we would have $I_1 \subseteq I_2$. $|I_1 \cap I_2|/|I_1|$ measures the extent to which this containment holds. After some experimentation, we decided to use $|I_1 \cap I_2|/|I_1| \log|I_2|$ as the score to assign to an entailment. We chose the top 20,000 entailments as our true entailments. An accuracy of 68% was obtained on a sample of 50 entailments.

6.2 Results

We first consider the relation extraction task. The goal here was to assign probabilities to the extractions in S independently. We used a logistic regression classifier [10] for this. Our training set consisted of 400 hand-tagged extractions. A relationship was tagged true if it represented something that a reader would accept as true rather than strictly following the semantics of universal quantification. For example, (*salad,is-topped-with,dressing*) was tagged true.

Figure 2 shows the precision–recall tradeoff for the feature sets F_0, F_1 and F_2. To determine each individual point on the curve, the top K extractions were chosen. Their precision was estimated by taking a random sample of 50, and hand-tagging them. We had a fixed recall set of 100 true extractions. We can see that F_2 gives the best performance. It seems like the TRP features, when active, provide reliable evidence that an extraction is true. The curve for F_2 thus starts off at a much higher precision.

We now consider the taxonomy induction problem. We used the MLN implementation provided by [5] for learning weights for constraints, as well as for MaxWalkSat. The training data used was the same as for the classifier. The training labels are now interpreted as truth values of atoms of the "True" predicate.

Each extraction is set to either true or false in a taxonomy. Computation of precision and recall is thus straightforward. We used the same recall set as used above. Table 1 compares the F-Measure for taxonomies T_0, T_1 and T_2. T_2 is the best once again showing the benefits of making use of the additional evidence provided by TRPs.

7 Conclusions

In this paper, we introduced the notion of TRPs, and described three concrete TRPs based on hyponymy. They represent a source of evidence that is weak but useful and easily obtained. We also generalized the semantic taxonomy induction problem to allow soft constraints with probabilistic semantics. We believe this provides a powerful framework for integrating different kinds of global evidence into relation extraction systems.

Acknowledgements

This research was supported in part by NSF grants IIS-0535284 and IIS-0312988, ONR grant N00014-08-1-0431 as well as gifts from Google, and carried out at the University of Washington's Turing Center.

References

[1] M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open Information Extraction from the Web. In *IJCAI*, 2007.
[2] A. Culotta, A. McCallum, and J. Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *HLT*, 2006.
[3] N. Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *ACL*, 2004.
[4] H. Kautz, B. Selman, and Y. Jiang. A general stochastic approach to solving problems with hard and soft constraints. In *The Satisfiability Problem: Theory and Applications*, 1996.
[5] S. Kok, P. Singla, M. Richardson, and P. Domingos. The alchemy sysem for statistical relational ai. Technical report, Dept of CSE, University of Washington, Seattle, 2006.
[6] T. Mitchell. Reading the Web: A Breakthrough Goal for AI. In *AI Magazine*, volume 26, pages 12–16, 2005.
[7] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2), 2006.
[8] D. Roth and W.-T. Yih. Probabilistic reasoning for entity & relation recognition. In *COLING*, pages 1–7, 2002.
[9] R. Snow, D. Jurafsky, and A. Y. Ng. Semantic taxonomy induction from heterogenous evidence. In *ACL*, 2006.
[10] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.