Refining Non-Taxonomic Relation Labels with External Structured Data to Support Ontology Learning

Abstract

This paper presents a method to integrate external knowledge sources such as DBpedia and OpenCyc into an ontology learning system that automatically suggests labels for unknown relations in domain ontologies based on large corpora of unstructured text. The method extracts and aggregates verb vectors from semantic relations identified in the corpus. It composes a knowledge base which consists of (i) verb centroids for known relations between domain concepts, (ii) mappings between concept pairs and the types of known relations, and (iii) ontological knowledge retrieved from external sources. Applying semantic inference and validation to this knowledge base yields a refined relation label suggestion. A formal evaluation compares the accuracy and average ranking precision of this hybrid method with the performance of methods that solely rely on corpus data and those that are only based on reasoning and external data sources.

Key words: Ontologies, Semantic Web, Ontology Learning, Relation Labeling, Machine Learning

1. Introduction

As formal conceptualizations of an application domain [1], ontologies provide the means for a common understanding of domain concepts and relations among different stakeholder groups. When domains evolve, there is a constant need to update and refine domain-specific ontologies to ensure their usefulness. The bottleneck and cost-driver in ontology learning tends to be the availability of expertise and qualified human resources. Automated approaches address this problem by supporting ontology engineers, improving their productivity, and reducing the human input required.

1.1. Related Work

Classifying previous approaches to (semi-)automatically learning ontologies [2**?** , 3, 4], this paper distinguishes between (i) *corpus analysis*, extracting information from corpus resources; (ii) *corpus enrichment*, integrating external resources such as Wikipedia (www.wikipedia.org), Google (www.google.com) and WordNet [5] into the extraction process; and (iii) *semantic inference and validation*, incorporating data from Semantic Web sources and investigating relations by reasoning upon this data.

Identifying and labeling non-taxonomic relations are among the ontology learning subtasks that are considered most challenging [6]. Natural Language Processing (NLP) competitions such as the one held in conjunction with the International Workshop on Semantic Evaluations (Task 04 at SemEval 2007 [7], previously known as SensEval, nlp.cs.swarthmore.edu/semeval) underscore the growing importance of identifying semantic relations. Building upon the categories available in this competition (corpus analysis and enrichment methods applying WordNet, Google, etc.), the following presentation of the state of the art distinguishes between techniques that rely exclusively on text corpora, and those that incorporate external data sources as well.

Corpus analysis applies linguistic patterns [8, 9, 10], association rules [11], kernel-based approaches [12] and other techniques from the fields of artificial intelligence, statistics, and mathematics to the problem of relation discovery. For instance, methods combining syntactic, semantic and lexical features and multiple models such as decision trees, decision rules, logistic regression and lazy classifiers (e.g., k-nearest-neighbor) tend to perform well in evaluations [13]. Snow et al. [14] optimize taxonomies by maximizing the probability of a certain taxonomy given the available evidence. They use taxonomic constraints such as the transitivity of hypernyms to determine all relations implied by a new hypernym and apply the Bayes algorithm to verify whether the proposed change would increase or decrease the probability of the taxonomy given the available evidence.

Ruiz-Casado et al. [15] extend WordNet with relations learned from the simple English version of Wikipedia (simple.wikipedia.org). They use known relations from WordNet to identify textual patterns indicating hyponym, hyperonym, holonym and meronym relations and automatically generalize these patterns. Afterwards, they apply the learned patterns to extract new relations from Wikipedia.

Zouaq and Nkambou [16] extract concept maps from textual resources by computing the text's keywords, analyzing the syntactic structure of key sentences, and a pattern-based semantic analysis. They identify important concepts based on the concept's number of relationships in the concept map and apply textual patterns to discover instances, taxonomic relations and attributes.

Navigli and Velardi [17] introduce OntoLearn, a tool which extracts domain terminology from Web sites. The paper's main focus is the disambiguation of the domain terminology to obtain a tree of domain concepts. The authors identify relations by training an inductive machine learning program with a set of manually tagged relations using the concepts participating in the relations and all their WordNet hyponyms as features. Finally, they apply the trained classifier to determine the labels of unlabeled relations.

Navigli and Velardi [18] describe a pattern-based method to enrich an ontology with definitions from a glossary. They apply manually defined regular expressions which consider (i) lexical similarity, (ii) part-of-speech tags, and (iii) syntactic and semantic constraints to capture relevant gloss fragments.

Many relation learning approaches are limited to in-corpus analysis as well. Snow et al. [19] apply a bootstrapping approach for learning hypernyms. They identify large numbers of lexico-syntactic patterns indicating hypernym relations by extending a set of example relations using a supervised learning algorithm. They improve their classifier by considering coordinate relations between nouns with a common hypernym in the relation detection process.

Kavalec and Svátek [6] identify verbs that express relations between concepts by applying a heuristic statistical measure called *above expectation*. This measure determines the most significant (concept$_1$, verb, concept$_2$) triples. Since relation names are more loosely linked to lexical items than names of concepts, the authors apply a manual step for mapping the identified verbs to domain ontology relation labels.

Ciaramita et al. [20] introduce an unsupervised, domain-independent method for learning arbitrary relations between named entities by using the dependency structure generated by a constituent syntactic parser [21] to extract candidate relations. The authors apply the chi-square test to select the *relations* most strongly associated with ordered pairs of named entities from the candidate list and then evaluate these relations manually. Reinberger et al. [22] extract subject-verb-object triples from a corpus composed of Medline abstracts and from a legal corpus. They apply a shallow parser to identify subject-verb-object structures and clustering to build classes of terms sharing a certain relation.

Rinaldi et al. [23] use deep linguistic parsing and manually created patterns to extract relations from the GENIA corpus, which consists of 2000 manually annotated Medline abstracts (www-tsujii.is.s.u-tokyo.ac.jp/GENIA). They apply (i) *syntactic patterns*; (ii) *semantic rules* (which combine multiple syntactic patterns into rules covering syntactic variants - e.g., active, passive, etc.; and (iii) *ontological constraints* to obtain domain-specific relations.

Corpus enrichment integrates external resources to increase the accuracy of the relation labeling. Sanchez and Moreno [24] present an approach using verbs from sentences containing domain concept identifiers and search engine queries for relation labeling. Giuliano et al. [25] use WordNet synsets and hypernym relations to refine kernel methods for extracting semantic relations. Chagnoux et al. [26] use corpus enrichment methods to identify relations using Hearst-style patterns [27]. They apply the Watson [28] semantic search engine to suggest unlabeled relations, which are then used to learn additional patterns from the domain corpus. Etzioni et al. [29] introduce a system for Web-scale information extraction from Web pages. They bootstrap their approach using a set of generic extraction patterns and use statistics computed by querying search engines to assess the correctness of the extracted relations. Yang and Callan [30] present an approach combining the high accuracy of pattern-based methods with the advantages of clustering-based techniques, which can identify implicit relations - even those that do not occur in the text. The clustering is based on an ontology metric which considers (i) in-corpus evidences such as co-occurrence, minipar syntactic distance, and lexical-syntactic patterns, as well as (ii) unstructured data retrieved from search engines and Google term definitions to identify hypernyms [27, 19], siblings (using conjunctions) and meronyms [31, 9].

Many knowledge mining approaches yield flat lists of unlinked lexical semantic knowledge [32] which needs to be grounded to unfold its full potential. Pennacchiotti and Pantel [32, 33] present an approach for disambiguating concepts participating in relations and linking them to their WordNet sense.

Semantic Inference and Validation integrates structured data from semantic Web resources, a method that has become quite popular in recent years. Using publicly available structured data from external sources to learn domain ontologies is a natural step in the evolution of ontology learning methods. Scarlet (scarlet.open.ac.uk) and Watson (watson.kmi.open.ac.uk) leverage ontological knowledge from single or multiple sources to determine the relation between concept pairs.

The DBpedia project (www.DBpedia.org) extracts information from Wikipedia and makes this information accessible via SPARQL endpoints and Web services. Its database currently contains 4.7 billion RDF triples with more than 2.6 million things, 3.1 million links to external Web pages, and more than 4.9 million links to external RDF datasets [34]. Lehmann et al. [35] query structured data from DBpedia to identify relations between concepts by identifying paths between these concepts. While their method has been applied successfully, it still suffers from a number of shortcomings: (i) the interpretation of the terms is based on DBpedia and therefore not domain-specific, and (ii) it is not trivial to derive *one* relation label from the paths determined by this method.

Despite the potential of the approaches presented above, their usefulness is limited by the so-called *knowledge acquisition bottleneck* [36], a term that refers to the difficulty of creating and maintaining extensive knowledge bases. This problem is also reflected in the current structure of the Semantic Web, which only comprises a relatively small number of extensive domain ontologies - as compared to a large number of easy-to-maintain, lightweight ontologies [28]. To overcome the restrictions imposed by the knowledge acquisition bottleneck, the approach presented in this paper combines machine learning methods with reasoning based on structured data from external sources.

1.2. Problem Statement

This paper introduces a method to suggest labels for unlabeled relations in domain ontologies. The underlying research aimed to ensure the general applicability of the method and addresses the shortcomings of the ontology learning framework introduced in Liu et al. [8]. This framework extracts domain terminology and the following relations from domain corpora: (i) *hypernym/hyponym relations*, (ii) *synonyms*, and (iii) *unlabeled relations*, for which no relation label could be determined.

The goal was to extend the relation labeling process to support the labeling of arbitrary relations based on a set of labels specified by domain experts. This set of candidate labels is determined by the conceptualization of the application domain by these experts and therefore reflects the domain's shared understanding.

Our approach uses an improved version of the relation labeling method developed by Weichselbraun et al. [37] to suggest relation labels to these *unlabeled relations* (Section 2.1) and integrates external structured sources such as DBpedia.org and OpenCyc (www.cyc.com/cyc/opencyc) to refine its suggestions (Section 2.2-2.4) by applying domain, range, and property restrictions based on information derived from these sources.

Snow et al. [14] note that many approaches to relation detection focus on particular relation types such as hypernyms [27, 19], synonyms [38], meronyms [31, 9], verb relations [39] and general purpose analogy relations [40]. The other extreme are domain-independent approaches which extract arbitrary relations from text corpora such as work done by Reinberger et al. [22], Etzioni et al. [29], and Banko and Etzioni [41]. These techniques do not consider the mapping of such relations to "valid" labels corresponding to the domain model.

In contrast to the first group of methods, our approach is applicable to arbitrary domain labels, but considers a number of predefined, already axiomatized predicates and integrates their domain, range and property restrictions into the relation label learning algorithm. Kavalec and Svátek [6] detect arbitrary relation types based on a semi-automatically established mapping between these types and verbs significant for this particular relation. Ciaramita et al. [20] and Rinaldi et al. [23] also support the detection of arbitrary relations but do not apply any external data sources to refine their results.

Navigli and Paola's [18] approach for enriching ontologies with term definitions from ontologies is related in regard to their use of regular expressions, together with syntactic and semantic constraints such as domain and range restrictions, to identify domain concepts in glosses.

Inductive Logic Programing (ILP), a technique for learning logical theories from data is another method related to this work. Nevertheless, Buitelaar and Cimiano [3] note that, it is more relevant for ontology refinement than for ontology learning, because ILP theories differ significantly from ontologies, which reflect a shared understanding of a domain of interest. Nevertheless, they emphasize the importance of analyzing how inductively derived models can supplement and support ontology learning, which they define as an *interactive* and *cooperative* process between engineers and ontology learning systems [42].

1.3. Paper Structure

The remainder of this paper is structured as follows: Section 2 presents the relation labeling component and elaborates on the integration of external structured resources into the identification process. Section 3 evaluates the link type labeling suggestion architecture using different experimental setups.

Section 3.4 discusses the results of the evaluations. The paper closes with a summary of the achieved results as well as an outlook on possible future research avenues in Section 4.

2. Method

The method presented in this section suggests labels for unlabeled relations in domain ontologies. It is independent from any particular ontology learning system, but has been developed as a component of the framework introduced by Liu et al. [8], which utilizes text corpora to extract domain concepts and instances ($\mathcal{C}$), as well as relations ($\mathcal{R}$), both taxonomic and unlabeled. Figure 1 illustrates the labeling process. The relation labeling component utilizes three types of data to identify relations in a particular domain including optional specifications of their domain, range, and property restrictions: (i) domain documents, (ii) the XML/RDF serialized domain ontology containing unlabeled relations ($\mathcal{R}_{m^*n^*}$) from the ontology learning framework [8], and (iii) a reusable "relation description" meta-ontology (Section 2.2) which contains the set of relation labels to be used.

The system extracts verbs from sentences with terms that represent two domain concepts or instances ($\mathcal{C}_m$, $\mathcal{C}_n$) participating in the relation $\mathcal{R}_{mn}$, and then stores this data in its knowledge base. The similarity between verb vectors of an unlabeled relation $\mathcal{R}_{m^*n^*}$ and the data in the knowledge base (Section 2.1) helps determine the label of unknown links, in conjunction with domain knowledge retrieved from sources such as DBpedia and OpenCyc.

Section 2.2-2.4 and Figure 3 describe the integration of *external knowledge* into this process. Querying the DBpedia page found for the respective concept label via SPARQL yields a set of links to external OWL-based ontologies. Many DBpedia pages contain links to both the DBpedia ontology as well as external data sources such as OpenCyc. The referenced sources contain structured information that describes the concepts participating in a particular relation. Matching this information with the suggested relation label's constraints allows to remove invalid relation labels or to decrease their similarity score. Concept grounding is a prerequisite and therefore a crucial step for applying ontological constraints to refine the similarity scores as provided by the vector space model.

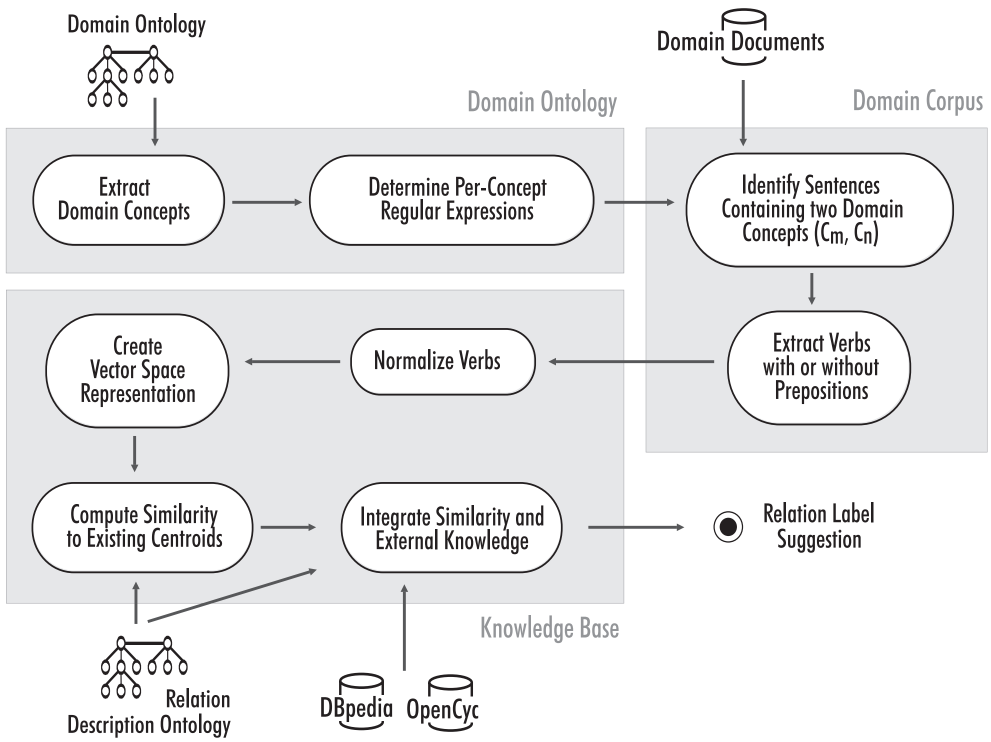

Figure 1: Architecture of the relation labeling component

2.1. Composing and Comparing Verb Vectors

The approach presented in this paper is based on corpus analysis algorithms developed by Weichselbraun et al. [37].

The regular expressions are generated automatically by determining all WordNet synonyms for a given concept[1] and generating a pattern which matches their singular and plural forms. Applying this procedure to the first WordNet sense of the concept "solar cell", for example, yields the list of synonyms "solar cell, photovoltaic cell" and the regular expression "(solar cells?)|(photovoltaic cells?)".

Equation 1 defines the list of verbs L_{mn} that characterize the semantic

[1]We obtain the term's sense from the input ontology. In cases where no senses are provided by the ontology, a word sense disambiguation algorithm such as the one introduced by Pantel and Pennacchiotti [33] can be used.

relation between the entities $\mathcal{C}_m$ and $\mathcal{C}_n$.

$$L_{mn} = \{verbs(s_i) \quad | \quad match(\mathcal{C}^r_m, s_i) \wedge match(\mathcal{C}^r_n, s_i) \\ \wedge \quad idx(\mathcal{C}^r_m, s_i) < idx(\mathcal{C}^r_n, s_i) \quad \} \tag{1}$$

L_{mn} is composed of the verbs occurring in sentences s_i together with terms matching the regular expressions for the concepts or instances $\mathcal{C}^r_m$ and $\mathcal{C}^r_n$. The *match* operators returns true if sentence s_i matches at least one of the regular expressions in the list $\mathcal{C}^r$. Multiple matches of a regular expression pair $\mathcal{C}^r_m$ and $\mathcal{C}^r_n$ at independent positions in a sentence are allowed, but the current implementation is limited to capture two matches.

The *verbs(s_i)* operator typically includes a verb lemmatization step and returns the infinitive form of all verbs present in sentence s_i. The verbs are detected upon the part-of-speech annotations of the sentence, optionally WordNet is used to identify wrongly classified words. The order of the concepts is important for the evaluation process. We define that $\mathcal{R}_{mn}(\mathcal{C}_m, \mathcal{C}_n) := \mathcal{R}_{nm}(\mathcal{C}_n, \mathcal{C}_m)^{-1}$, which effectively reverses the direction of a relation. The *idx* operator in the second term of the definition ensures that the regular expression for the first entity ($\mathcal{C}^r_m$) occurs before the second entity ($\mathcal{C}^r_n$) in a sentence s_i.

Equation 2 computes the centroid $\vec{V}_{mn}$, which represents the verb vector for the relation $\mathcal{R}_{mn}$ between the two concepts or instances $\mathcal{C}_m$, $\mathcal{C}_n$. The operator vsm_{20} yields the vector space representation of the 20 most relevant verbs in the verb list, in regard to the tf-idf [43] measure.

$$\vec{V}_{mn} = \frac{vsm_{20}(L_{mn})}{\left|vsm_{20}(L_{mn})\right|} \tag{2}$$

The method is trained by computing the centroids for concepts and instances $(\mathcal{C}_m, \mathcal{C}_n)$ with known relation types (training relations) and saving the corresponding mapping in $M_{mn \to j}$ which translates concept pairs to relation labels (j). The relation label j^* of an unlabeled relation $(\mathcal{R}_{m^*n^*})$ between the entries $\mathcal{C}_{m^*}$, $\mathcal{C}_{n^*}$ is computed by

1. applying Equation 1 and 2 to sentences containing terms referring to these concepts, a process that yields the centroid $\vec{V}_{m^*n^*}$ for the unlabeled relation;
2. comparing this centroid to all centroids with known relation label using a similarity function $sim := sim(\vec{V}_{m^*n^*}, \vec{V}_{mn})$ such as the cosine measure; and

3. assigning the relation label (j) of the best match (highest similarity) to the unlabeled relation ($\mathcal{R}_{m^*n^*}$) by applying the mapping function $M_{mn\to j}$. Section 2.4 will introduce more advanced label selection strategies, and Section 3.3 will evaluate their impact on labeling performance.

Table 1 outlines this process. We determine the label (j^*) for the relation ($\mathcal{R}_{m^*n^*}$) between *scientist* and *greenhouse effect* based on the similarity of its centroid ($\vec{V}_{m^*n^*}$) to the centroids of known relations ($\vec{V}_{mn}$). Those similarity computations rely on centroids built from the list of verbs L_{mn} as defined in Equation 2. In this example, the relation between *oil* and *fossil fuel* is most similar to the unlabeled relation ($sim(\vec{V}_{m^*n^*}, \vec{V}_{mn}) = 0.33$). Therefore, its relation label ($j =$*subClassOf*) would be assigned to the unknown relation.

$\mathcal{C}_m$, $\mathcal{C}_n$ $\xrightarrow{M_{mn\to j}}$	j	L_{mn}	sim
politician, carbon tax	takeActionBy	{raise, pay, ...}	0.12
oil, fossil fuel	subClassOf	{be, have, ... }	0.33
NOAA, climate change	study	{say, describe, ... }	0.30
scientist, green energy	study	{develop, use, ... }	0.29

Table 1: Determining the best matching relation label ($\mathcal{R}_{m^*n^*}$) for the relation between *scientist* ($\mathcal{C}_{m^*}$) and *greenhouse effect* ($\mathcal{C}_{n^*}$)

2.2. Integration of External Knowledge

Experts define the set of relation labels in a meta ontology to describe the relations used in the domain ontology. This “relation description ontology” contains the concepts necessary to specify all valid relation labels and optional domain, range and property restrictions to clarify the use of these labels. It is important to note that classes and relation descriptions from the meta ontology can be reused between different domain ontologies.

The component to integrate external knowledge into the label suggestions uses these restrictions to refine the similarities obtained from the comparison of verb vectors as described in Section 2.1. Figure 2 visualizes the classes and properties for the relation description meta ontology used for the following examples. The figure also demonstrates the specification of domain and range restrictions based on the “study” relation.

Figure 3 outlines the refinement process. The *structured data integration component* obtains the entries $\mathcal{C}_{m^*}$ and $\mathcal{C}_{n^*}$ participating in an unlabeled relation as well as a ranked list of relation labels, suggested by the vector space

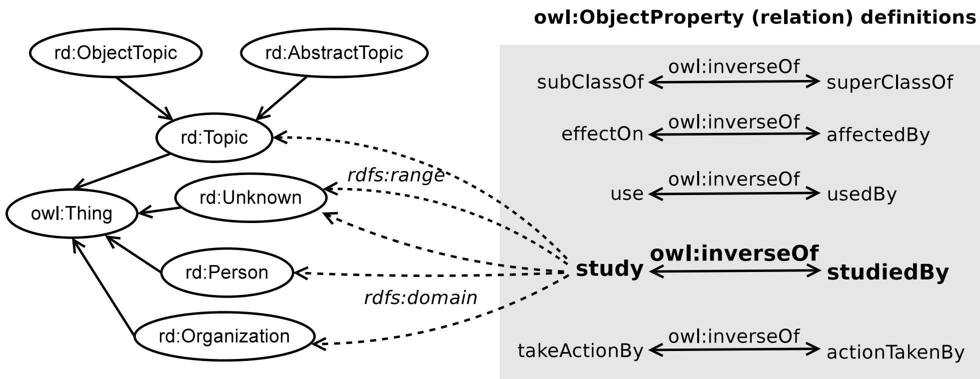

Figure 2: Relation description ontology

approach described above. It uses structured data from external sources such as DBpedia and OpenCyc to ground $\mathcal{C}_{m^*}$, $\mathcal{C}_{n^*}$ to the relation description concepts that specify the restrictions on the label's use. The system then verifies the constraints for each suggested relation label, adjusts its weights accordingly, and computes a refined ranking of labels for the relation between the entries $\mathcal{C}_{m^*}$ and $\mathcal{C}_{n^*}$.

Figure 4 illustrates the verification process for the relation between the concepts *NOAA* and *climate change*. The concepts are of type *Organization* and *Topic*. Combining this information with the domain and range restrictions specified for the relation *study* and *studiedBy* suggests that it is unlikely that *studiedBy* is the correct label, because it violates the relation type's domain restrictions. In contrast, the link label *study* satisfies domain and range restrictions, which increases the likelihood that it is the correct label.

The code snippet below illustrates an ontology fragment for the relation type *study*, based on our conceptualization of the climate change domain (the relation label *study* may apply to the range of *Person* and *Organization* in other domains such as medicine and psychology). The concept type *Unknown* is added to all domain/range restrictions implicitly, in order to cover situations where no grounding was possible.

```
<owl:ObjectProperty rdf:ID="study">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="rd:Person"/>
          <owl:Class rdf:about="rd:Organization"/>
          <owl:Class rdf:about="rd:Unknown"/>
        </owl:unionOf>
```

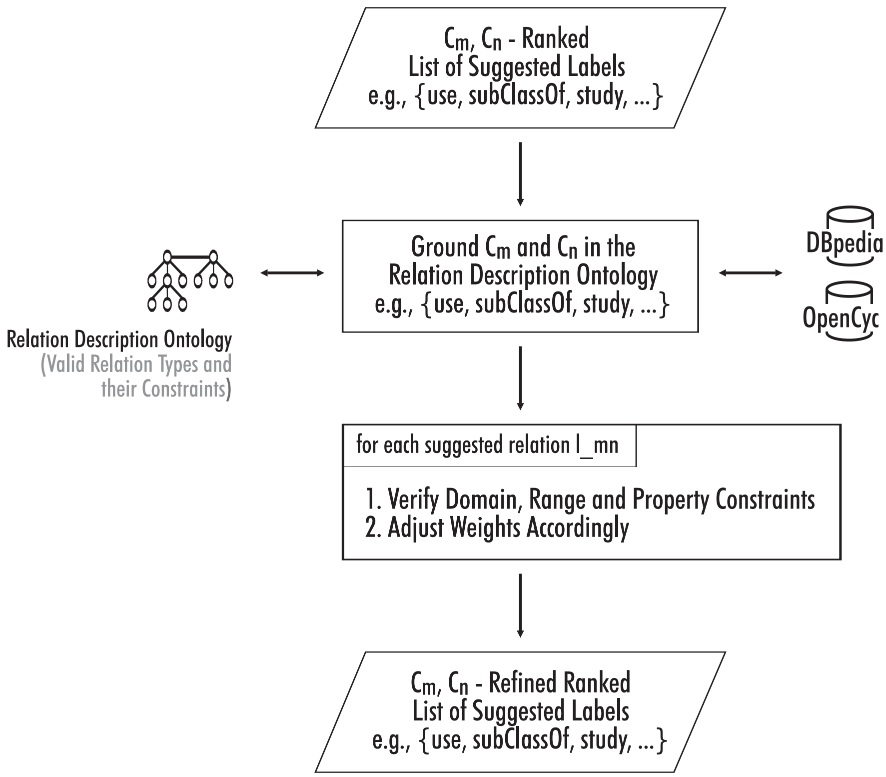

Figure 3: Integration of external knowledge

```
      </owl:Class>
    </rdfs:domain>
    <rdfs:range>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <owl:Class rdf:about="rd:Topic"/>
          <owl:Class rdf:about="rd:Unknown"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:range>
</owl:ObjectProperty>
```

Using OWL Lite property restrictions on classes from the relation description ontology helps include even more information on the proper usage of a relation. Only an *Organization* typed concept, for example, can be a *subClassOf* another *Organization* typed concept. The following example specifies this restriction on the *Organization* concept in the relation descrip-

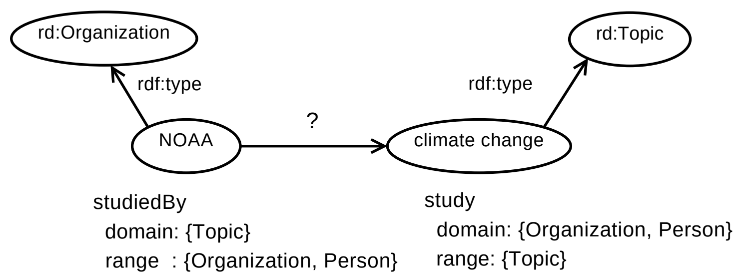

Figure 4: Considering external knowledge when selecting link types

tion ontology:

```
<owl:Class rdf:ID="rd:Organization">
    <rdfs:subClassOf>
        <owl:Restriction>
            <owl:onProperty rdf:resource="#subClassOf" />
            <owl:allValuesFrom rdf:resource="#Organization">
        </owl:Restriction>
    </rdfs:subClassOf>
</owl:Class>
```

Before the restrictions of the relation description ontology can be applied in the semantic inference and validation step, the system has to ground the concepts of the domain ontology using data from external sources such as DBpedia and OpenCyc. The goal of this step is to assign one of the concept types defined in the relation description ontology to every concept.

Concept labels from the domain ontology are mapped to DBpedia page names. In cases where no corresponding DBpedia page exists, or the result is a DBpedia disambiguation page, no grounding is possible. The issue of disambiguation will be addressed in future research. Figure 5 illustrates the mapping of the concept *NOAA* from the example above to its respective types (*Organization*). The system follows links from DBpedia to external ontologies such as OpenCyc, and applies an ontology reasoner to the external ontologies which tries to map the respective external classes to a relation description ontology concept.

The relation description ontology specifies its classes as the union of external class URIs, which are mapped onto the respective class in the grounding process. This process provides a set of ontology fragments to ground the domain concepts in the relation description ontology, as illustrated in the following example:

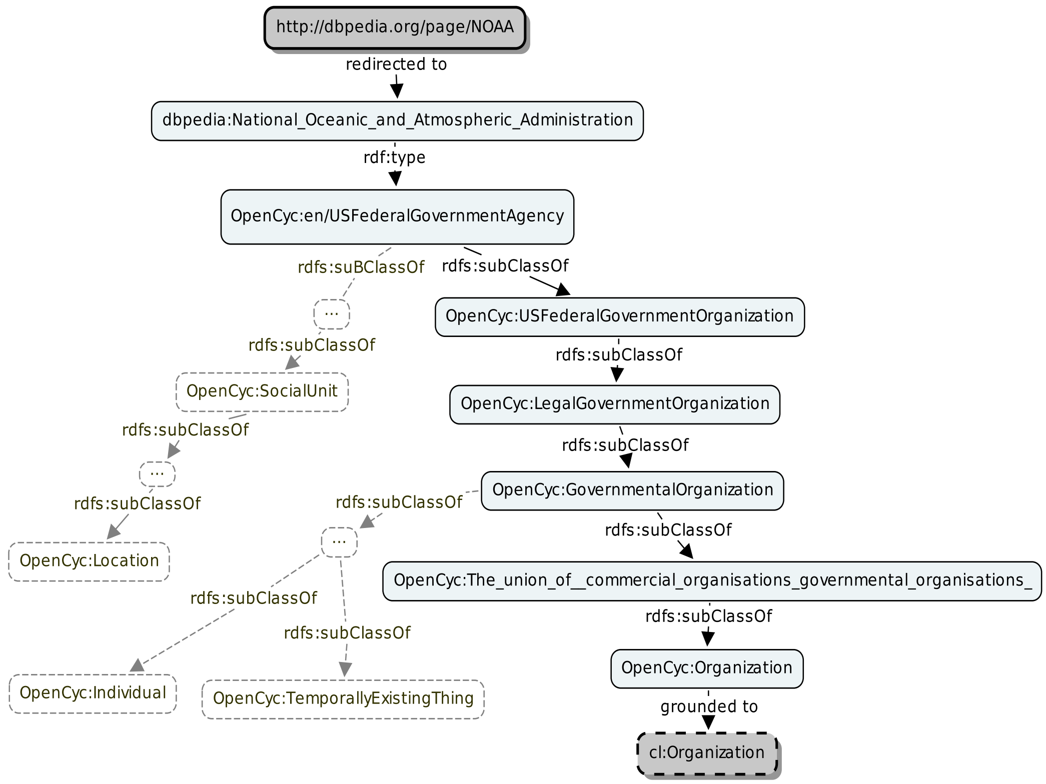

Figure 5: Grounding example for the concept "NOAA"

```
<!-- we define rd:Organization as the union of external classes which
     are mapped to rd:Organization-->
<owl:Class rdf:ID=rd:Organization>
  <owl:unionOf rdf:parseType="Collection">
    <owl:Class rdf:about="http://sw.opencyc.org/concept/Mx4r..." />
    <owl:Class rdf:about="http://dbpedia.org/ontology/Organisation" />
    . . .
  </owl:unionOf>
</owl:Class>

<!-- information derived from reasoning -->
<http://dbpedia.org/page/NOAA> <rdfs:subClassOf
   rdf:resource="http://sw.opencyc.org/concept/Mx4..."/>
```

2.3. Knowledge Base

The knowledge base of the framework consists of (i) a list of all centroids $\vec{V}_{m_i n_j}$ representing the relation $\mathcal{R}_{m_i n_j}$, (ii) the mapping $M_{mn \to j}$ assigning a label j to the relation $\mathcal{R}_{m_i n_j}$, and (iii) the ontology O which consists of the relation description ontology O_{rd} and the ontology fragments $\{O_1, O_2, ...O_n\}$ retrieved from external sources, containing formalized knowledge of the domain (Section 2.2).

$$KB = (\{\vec{V}_{m_1 n_1}, ...\vec{V}_{m_k n_l}\}, M_{mn \to j}, O_{rd}, O_1, ..., O_n) \quad (3)$$

2.4. Relation Labeling

The first step in determining the relation's label is computing the similarity (sim) between centroid $\vec{V}_{m^* n^*}$ representing the unlabeled relation (Equation 1 and 2) and and all known centroids in the knowledge base (see Section 2.1). Applying semantic inference and validation to the ontology fragments refines the similarity as outlined in Equation 4 by computing weights w_{o,m^*,n^*} to integrate domain knowledge (Equation 5) into the refined similarity s_{mn}.

$$s_{mn} = w_{o,m^* n^*}(\underbrace{M_{mn \to j}(\mathcal{C}_m, \mathcal{C}_n)}_{j}) \cdot sim(\vec{V}_{m^* n^*}, \vec{V}_{mn}) \quad (4)$$

The current architecture uses the cosine measure as similarity function. The factor w_{o,m^*,n^*} considers domain knowledge using the following heuristic.

$$w_{o,m^* n^*}(j) = \begin{cases} 1.0 & \text{if } O \models \mathcal{C}_{m^*} \in dom(j) \quad \wedge \\ & \quad O \models \mathcal{C}_{n^*} \in range(j) \wedge O(j(\mathcal{C}_{m^*}, \mathcal{C}_{n^*})) \\ 0.01 & \text{if } O \models \mathcal{C}_{m^*} \notin dom(j) \quad \vee \\ & \quad \mathcal{C}_{n^*} \notin range(j) \vee \neg O(j(\mathcal{C}_{m^*}, \mathcal{C}_{n^*})) \\ 0.8 & \text{if } O \models \mathcal{C}_{m^*} \in dom(j) \quad \vee \\ & \quad \mathcal{C}_{n^*} \in range(j) \\ 0.6 & \text{otherwise.} \end{cases} \quad (5)$$

Equation 5 determines the weights $w_{o,m^* n^*}$ based on whether the ontology implies ($\models$) the domain, range and property restrictions from the ontology or not. In cases where the component cannot verify any domain and range restrictions (relations with no domain and range constraints; concepts for which no type could be identified), a weight of 0.6 is assigned. If the suggested relation type only meets the domain *or* range restrictions – as the type of one of the two concepts could not be determined – the heuristic assigns a weight of 0.8. The weight of 0.01 is applied in cases where the restrictions on the suggested relations are in conflict with the concept types, the value was chosen in order to preserve the information from the original ranking. The weights (1.0, 0.8, 0.6 and 0.01) were chosen after conducting a series of experiments with different weight distributions and were selected independently of the evaluations.

From the computed similarity s_{mn} and the mapping $M_{mn \to j}$ we determine a list of triples, which contains the relation $\mathcal{R}_{mn}$, the matching relation label j, and its similarity to the unlabeled relation $\mathcal{R}_{m^*n^*}$.

The method determines the relation label j^* using one of the following strategies: (i) selecting the relation label j from the relation $\mathcal{R}_{mn}$ with the highest similarity s_{mn}, (ii) computing the average of the similarity measures s_{mn} for each relation label j (over all training relations associated with this relation label) and selecting the label with the highest average similarity, or (iii) determining the average of the highest 30% of the similarity measures for each relation label j, and selecting the label corresponding to the highest average. The evaluation in Section 3 compares the performance of these three approaches.

$\mathcal{C}_m$,$\mathcal{C}_n$ $\xrightarrow{M_{mn \to j}}$	j	*sim*	constraints domain	range	w_{o,m^*n^*}	s_{mn}
oil, fossil fuel	subClassOf	0.33	v	v	0.01	0.003
			c	-	0.8	0.264
			-	c	0.8	0.264
			-	-	0.6	0.198
NOAA, climate change	study	0.30	c	c	1.00	0.30
			c	-	0.8	0.24
			-	c	0.8	0.24
			-	-	0.6	0.18

Table 2: Label suggestion for the relation *scientist (Person)* → *greenhouse effect (Object-Topic)*. The letters "c" and "v" indicate information "corresponding to" or "violating" domain constraints; "-" shows that concept grounding was not successful

Table 2 exemplifies the weighting process with the unlabeled relation *scientist* → *greenhouse effect*. The system distinguishes four degrees of success in terms of concept grounding: (i) both concepts could be grounded correctly, (ii) only the subject was grounded, (iii) only the object was grounded, and (iv) no grounding was possible. Computing the vector space similarity between the known and unlabeled relations yields the similarity values ($sim := sim(\vec{V}_{m^*n^*}, \vec{V}_{mn})$). The mapping function $M_{mn\to j}$ determines the corresponding labels (j) for the concept pairs. External domain knowledge refines these similarities by computing weights based on Equation 5, which considers domain, range and property restrictions.

Table 2 outlines this process: the first step computes similarities between the verb vectors in the knowledge base and the verb vectors of the unlabeled relation between *scientist* and *greenhouse effect*. The mapping $M_{mn\to j}$ yields the relation label (j) which corresponds to the concept pairs from the knowledge base. The domain restrictions are then incorporated based on the grounding of the concepts from the unlabeled relation (*scientist* and *greenhouse effect*). The suggested relation type *subClassOf* receives a weight of only 0.01 because the concept types (*Person*, *ObjectTopic*) conflict with property restrictions for the concept types – i.e., if the subject of *subClassOf* is a *Person*, its object has to be of type *Person* as well. If only the subject or object concept type is known and there is no conflict, then the weighting value of 0.8 is used (Equation 5). The relation type *study* satisfies domain, range and property restrictions and therefore receives a weight of 1.0. In cases where both concept types were identified correctly, the method suggests the label *study*. If external domain knowledge could not be incorporated via grounding, the system would (wrongly) suggest the *subClassOf* type.

2.5. Integration of User Feedback

The knowledge base (KB) stores all training relations – i.e., known relations and their types from the domain ontology, as well as the (optional) relations specified by domain experts manually. The system presents suggestions for unlabeled relations to domain experts who either confirm or discard the suggested relations. User feedback on relation types is incorporated by adding the relation $\mathcal{R}_{m_i n_j}$ and the corresponding centroid $\vec{V}_{m_i n_j}$ and its mapping $M_{mn\to j}$, to the knowledge base.

Results violating the relation description ontology are reported back to the ontology engineer, who might either update the ontology or discard the automatically generated feedback. This semi-automated process refines the knowledge base and constantly improves the component's accuracy.

3. Evaluation

This section describes the experiments conducted to evaluate the performance of the relation labeling method introduced in Section 2. Two domain experts manually extended relation sets that were identified by the webLyzard ontology extension architecture [8] for the climate change domain. The resulting set of 313 high-quality relations ($\mathcal{R}$) was doubled in size by adding relations with the concepts in reverse order ($\mathcal{R}^{-1}$, e.g. car $\xrightarrow{subClassOf}$ vehicle; vehicle $\xrightarrow{superClassOf}$ car). These labels were verified by two other domain experts who double-checked all labels independently with an inter-expert agreement of 90.2%. Most of the conflicting classifications were caused by ambiguities between *takeActionBy/actionTakenBy* and *effectOn/affectedBy*. A smaller percentage of dissent related to *subClassOf/superClassOf* versus *effectOn/affectedBy*.

To improve validity, a pre-processing step eliminated relations that were identified in fewer than ten distinct sentences within the corpus. Based on the provided corpus, such relations cannot be considered domain-specific. Applying this cleanup process reduced the size of the evaluation set from 626 to 461 relations. Table 3 contrasts the number of relations per relation type defined by the domain experts with the number of relations available after the cleanup process.

Rel. label	before cleanup	after cleanup
subClassOf	55	41
superClassOf	56	40
use	58	48
usedBy	58	48
effectOn	67	48
affectedBy	67	48
takeActionBy	70	50
actionTakenBy	70	46
study	62	48
studiedBy	62	44

Table 3: Number of relations per relation type before and after the cleanup step

3.1. Domain-Specific Evaluation Corpus

To create the evaluation corpus, the webLyzard suite of Web mining tools (www.weblyzard.com) was used to crawl 156 Anglo-American news media sites selected from the Newslink.org, Kidon.com and ABYZNewsLinks.com directories. From the 200,000 documents gathered each week, a domain detection service based on regular expressions compiled an extensive domain-specific corpus consisting of 157,817 documents published between December 2008 and July 2009. Additional documents where gathered from environmental blogs and the Web sites of environmental organizations.

The evaluation component uses the corpus to create vector space representations of verbs appearing in (i) the same sentence as, or (ii) in a sliding window of five and seven words together with regular expression matches for concepts or instances ($\mathcal{C}_m^r$, $\mathcal{C}_n^r$). Table 4 lists the relations ($\mathcal{R}$) used for the labeling process and the number of sentences in the corpus satisfying Equation 1 (see Section 2) from which verb vectors for that particular relation type could be extracted. We used a total of 160,456 sentences from the corpus to evaluate the method, 126,163 of which were unique.

Rel. label	sentences	Rel. label^{-1}	sentences
subClassOf	8,877	superClassOf	8,717
use	11,780	usedBy	19,751
study	27,794	studiedBy	26,565
effectOn	6,605	affectedBy	4,229
takeActionBy	29,442	actionTakenBy	16,696

Table 4: Relation labels used in the evaluation and number of sentences per relation found in the corpus

Similar to Snow et al. [19], we trained each directed relation in the knowledge base with a set of about 50 pre-defined concept-relation patterns. Table 3 provides the exact numbers, while Table 5 contains examples of such patterns.

The number of verbs extracted from the corpus depends on the extraction mode (whole sentence versus sliding window). In *whole sentence* mode the average number of extracted verbs was 1,398.88 per training relation, with a maximum of 41,039. For a *seven word sliding window* the numbers are obviously lower, 313.58 verbs on average with a maximum of 8,734. As part of the evaluation, two different *tf-idf* significance thresholds (20 and 150) for building the verb vectors (Equation 2) were compared as well.

$\mathcal{C}_m^r$	$\mathcal{C}_n^r$	Rel. label
wind energ(y|ies)	energy sources?	subClassOf
solar cells?|photovoltaic cells?	solar energ(y|ies)|solar power	use
compan(y|ies)	energy savings?	takeActionBy

Table 5: Example training patterns

3.2. Integration of External Knowledge

A correct concept grounding is a precondition for verifying range, domain and property restrictions (see Section 2.2). Therefore, the relation description ontology is used to determine the type of concepts participating in unlabeled relations. If the concept grounding suggests multiple possible concept types, simple preference rules supplied by domain experts (e.g., *Person* is preferred over *ObjectTopic*) are used to resolve these conflicts. For example, a *politician* is of type *Person*, but also of the OpenCyc type *thing existing stably in time*. In contrast, an *ObjectTopic* such as *fossil fuel* should never be grounded to the *Person* class.

Table 6 shows a number of concept examples and their respective types. Concepts for which no type could be discovered are labeled as *unknown* and treated according to Equation 5.

concept or instance ($\mathcal{C}$)	type
expert, scientist	rd:Person
NOAA, IPCC, OPEC	rd:Organization
fossil fuel, ecosystem	rd:ObjectTopic
exploitation, peak oil	rd:AbstractTopic

Table 6: Concept and instance types in the relation description ontology

Using the described method for determining the concept types succeeded in 110 out of 168 cases. Table 7 presents the results of a manual evaluation of the concept grounding performed by domain experts.

In seven cases the grounding yielded incorrect concept types. Some of the links from DBpedia to OpenCyc are questionable, resulting in incorrectly grounded concepts. For example, the concept *bus* (*www.dbpedia.org-/resource/Bus*) has an *owl:sameAs* link to OpenCyc's *bus line* (*sw.OpenCyc.org/2008/06/10/concept/Mx4...*), which is an OpenCyc subclass of *transportation organization*, and therefore gets grounded to the concept *Organization*, although it should be an *ObjectTopic*. DBpedia's redirects (see Figure 4)

	type detection	number of concepts
grounded	correct	110 of 168 (65.4%)
	incorrect	7 of 168 (4.2%)
not grounded	no DBpedia entry found	10 of 168 (6.0%)
	no path to a matching concept	41 of 168(24.4%)

Table 7: Success of the concept grounding of all 168 concepts

can be another source of problems – e.g., *activist* is redirected to *activism* which we therefore map to *AbstractTopic* instead of the correct *Person*. Such cases have a negative impact on the results of the relation label suggestion system – it is better to have no type information at all than an incorrect one.

For ten concepts, no DBpedia page could be found. The concept labels simply do not exist in DBpedia, examples are: *combustion process*, *oil demand*, *environmental problem*. Possible ways to tackle this problem in the future are the acquisition of synonyms or term resolution techniques such as the ones used by Wong et al. [44].

For 41 concepts, a DBpedia page was found but did not provide sufficient information to detect the type. In those cases no links to OpenCyc or the DBpedia ontology were available, or those links did not yield appropriate grounding information. Occasionally, the DBpedia disambiguation page was returned, for example for *pipeline* or *creation*. Future research will apply disambiguation techniques to address such issues. For other concepts such as *photovoltaic effect*, DBpedia does not provide sufficient structured data to successfully apply concept grounding. The ongoing extension and refinement of DBpedia and other semantic resources will improve the precision and recall of our approach.

3.3. Results

In the experiments, the relation label suggestion component assigns an ordered list of the labels introduced in Table 4 to each unlabeled relation. The results are based on the average of seven evaluation runs with 461 relations from the test ontology, which were randomly split into training and testing sets of equal size for each run. Table 8 summarizes the configuration parameters used in the subsequent experiments.

Three different aggregation strategies were used to generate the ordered list of label suggestions (compare Section 2.4): (i) suggest the single best training relation with the highest similarity value, (ii) apply the average

No.	Target	Description
1	verbs	This setting affects the process of collecting verbs from the sentences matching the regular expressions representing a particular relation. The **verb extraction modes** (e.g., *whole sentence*, *sliding window size 7*, or *sliding window size 5*) determine the verbs to be collected from a sentence.
2	vector-building	The initial implementation used simple verb frequencies to build the verb vectors. We compute verb significance with **tf-idf** and evaluates the use of the **20** and **150** most significant verbs, denoted as *tf-idf 150* and *tf-idf 20*.
3	relations	The various **aggregation modes** of similarity scores, denoted as (i)-(iii) are discussed below. For rows marked with *direction: yes*, both the type of a relation *and* its direction are computed. Rows identified by the term *direction: no* only consider the correct relation type for the evaluation.
4	grounding	The evaluation distinguishes between suggestions derived from **corpus analysis** (vector space model), and suggestions refining this model by applying **semantic inference and validation** (SIV). Results solely obtained using the vector space model are marked with VSM. SIV denotes experiments which apply the vector space model and external knowledge.

Table 8: Configuration settings for the relation suggestion component

similarity value per relation label, and (iii) use the average of the best 30% of relations per predicate. The results for these three distinct strategies are indicated by the literals *(i)*, *(ii)* and *(iii)* in Table 9, which compares the strategies based on the average ranking precision (ARP). This measure specifies the average number of tries required to pick the correct relation label from an ordered list of suggestions (the table contrasts computations based on a sliding window size with those based on whole sentences). The ARP measure is highly relevant to capture the usefulness of a method to assist domain experts in assigning relation labels - it indicates how many choices the domain expert has to check on average to identify the correct label.

The ARP for randomly chosen relation labels is 3.0 for guessing the correct relation label and 5.5 for picking the right relation label and direction (those values are part of the evaluation tables denoted as *Baseline: rand*).

To demonstrate the potential of semantic inference and validation, we conducted a second evaluation restricted to a set of 437 (from 461) relations for which at least one concept *type* - according to the relation description ontology - could be extracted. The results of these additional computations are given in parenthesis.

	Method	direction	sl. window 7	sentence
(i)	SIV	no	1.655 (1.607)	1.533 (1.479)
	Baseline: VSM	no	2.044 (2.018)	1.851 (1.816)
	Baseline: rand	no	3.000 (3.000)	3.000 (3.000)
	SIV	yes	2.139 (2.028)	2.178 (2.058)
	Baseline: VSM	yes	3.091 (3.036)	3.196 (3.136)
	Baseline: rand	yes	5.500 (5.500)	5.500 (5.500)
(ii)	SIV	no	1.616 (1.550)	1.535 (1.463)
	Baseline: VSM	no	1.924 (1.877)	1.872 (1.820)
	Baseline: rand	no	3.000 (3.000)	3.000 (3.000)
	SIV	yes	2.024 (1.912)	2.116 (2.008)
	Baseline: VSM	yes	2.726 (2.656)	2.997 (2.940)
	Baseline: rand	yes	5.500 (5.500)	5.500 (5.500)
(iii)	SIV	no	1.591 (1.530)	1.520 (1.451)
	Baseline: VSM	no	1.915 (1.874)	1.847 (1.797)
	Baseline: rand	no	3.000 (3.000)	3.000 (3.000)
	SIV	yes	2.035 (1.930)	2.103 (1.993)
	Baseline: VSM	yes	2.778 (2.716)	3.003 (2.946)
	Baseline: rand	yes	5.500 (5.500)	5.500 (5.500)

Table 9: Average Ranking Precision (ARP) of the SIV method, compared to VSM and random baselines

Table 9 shows that the combined approach including semantic validation (*SIV*) clearly outperforms the VSM-only method. In the case of non-directed relations using aggregation strategy *(iii)* and *sentence mode*, for example, an ARP of 1.52 (1.45) could be reached with the integration of structured data, as compared to 1.85 (1.80) with VSM only. For directed relations, we observe an ARP of 2.10 (1.99) for SIV as compared to 3.00 (2.95) for VSM. The evaluations presented in Table 9 were computed with a *tf-idf 20* configuration, which uses the 20 most significant verbs per relation in the verb vectors (the evaluation with a *tf-idf 150* configuration yielded similar results and is therefore omitted for brevity).

Applying Scarlet (scarlet.open.ac.uk), a method solely based on querying Semantic Web resources, to the evaluation task only provided relation types for eight out of 461 evaluated relations. This is attributable to the knowledge acquisition bottleneck discussed in the introduction. Four out of eight relations were labeled correctly by Scarlet. We also encountered a case in which Scarlet inaccurately labeled relations due to an incorrect *subClassOf* relation in an underlying ontology (*oil subClassOf industry*) as described by d'Aquin et al. [28]. Currently, Scarlet does not significantly influence the evaluation results, so it is not included in Table 9 and 10. Nevertheless, the authors decided to incorporate Scarlet into the presented framework, since the growth of the Semantic Web will result in significant improvements in terms of Scarlet's precision recall.

	Method	direction	sl. window 7	sentence
(i)	SIV	no	66.96 (68.12)	70.56 (71.73)
	Baseline: VSM	no	56.15 (56.68)	64.78 (65.61)
	Baseline: KS	no	32.41 (33.10)	27.41 (28.45)
	Baseline: rand	no	20.00 (20.00)	20.00 (20.00)
	SIV	yes	48.32 (49.96)	43.79 (45.10)
	Baseline: VSM	yes	33.23 (33.99)	26.65 (26.95)
	Baseline: KS	yes	19.24 (19.60)	13.61 (14.24)
	Baseline: rand	yes	10.00 (10.00)	10.00 (10.00)
(ii)	SIV	no	68.14 (69.64)	71.55 (73.17)
	Baseline: VSM	no	61.55 (62.66)	63.04 (64.16)
	Baseline: KS	no	31.24 (32.55)	29.68 (30.25)
	Baseline: rand	no	20.00 (20.00)	20.00 (20.00)
	SIV	yes	51.30 (52.80)	45.47 (46.48)
	Baseline: VSM	yes	39.13 (39.91)	27.45 (27.41)
	Baseline: KS	yes	18.36 (19.22)	16.83 (16.96)
	Baseline: rand	yes	10.00 (10.00)	10.00 (10.00)
(iii)	SIV	no	69.50 (70.75)	72.61 (74.03)
	Baseline: VSM	no	61.80 (62.59)	63.79 (64.69)
	Baseline: KS	no	33.57 (34.56)	29.86 (30.71)
	Baseline: rand	no	20.00 (20.00)	20.00 (20.00)
	SIV	yes	51.55 (52.99)	47.20 (48.24)
	Baseline: VSM	yes	37.95 (38.59)	28.26 (28.20)
	Baseline: KS	yes	20.41 (20.99)	16.51 (16.95)
	Baseline: rand	yes	10.00 (10.00)	10.00 (10.00)

Table 10: Percentage of relation labels which were correctly identified on the first guess (sliding window size of seven words)

The results for extracting verbs with sliding windows produced similar results as verbs from whole sentences, although the *sentence* configuration seems to perform better when only suggesting the relation type but not the direction (*direction: no*). Sliding windows, by contrast, extract a narrower context and therefore achieve better ARP scores when also detecting relation direction.

The average similarity over all relations grouped by relation label (ii) and average of the best 30% of relations per label (iii) strategies deliver the best results and outperform the single best vector (i) approach. We attribute this to the fact that strategies including average building are more robust against outliers, which may harm the performance of (i).

Table 10 summarizes the results as a percentage of correctly identified relation labels, more precisely the percentage of relations correctly identified by the first suggestion (as opposed to *2nd guess correct* in Table 11). The literals (i)-(iii) have the same meaning as in Table 9. This table also contains an additional baseline score *Baseline: KS*, which is based on an adopted version of the *above expectation* measure by Kavalec and Svátek [6]. To integrate this measure into the evaluation process, we had to replace the semi-automatic mapping with a custom heuristic which does not require the input of domain experts.

The evaluation results provided in Table 10 present a system performance of over 70% of correctly labeled suggestions for non-directional configurations, and around 50% for detecting relation labels including direction. The observation made for the ARP measure, namely that the *sentence* verb extraction mode is superior regarding the non-directional (*direction: no*) settings, and sliding windows are better in the directional (*direction: yes*) setting, holds for *first guess* and *second guess* (see below), too. The performance of the adopted *above expectation* measure was rather limited. We attribute this not to the heuristic itself, but to the difficulty of transforming the method appropriately to our automated label suggestion and evaluation schema.

Table 11 presents the results where either the first or second relation label are correct. For the sake of brevity we limit this table to strategy (iii).

It is obviously more challenging to guess the correct relation type and direction out of ten possibilities with a probability of 10% of randomly picking the correct label (compare Table 4), as compared to guessing only the relation type with a 20% chance of randomly guessing the correct label.

Conducting a Chi-squared test on the results presented in Table 11 showed that the observable increases in accuracy are significant at the 0.01 level as

	Method	direction	sl. window 7	sentence
(iii)	SIV	no	84.29 (85.47)	86.40 (87.84)
	Baseline: VSM	no	76.21 (76.92)	77.33 (78.23)
	Baseline: KS	no	52.15 (53.61)	45.99 (46.53)
	Baseline: rand	no	40.00 (40.00)	40.00 (40.00)
	SIV	yes	77.70 (79.42)	76.65 (78.44)
	Baseline: VSM	yes	61.86 (62.65)	56.52 (57.13)
	Baseline: KS	yes	34.90 (35.73)	28.54 (29.00)
	Baseline: rand	yes	20.00 (20.00)	20.00 (20.00)

Table 11: Percentage of correctly identified relation labels on second guess in the evaluation (sliding window size of seven words)

compared to the VSM, KS and rand baseline scores for directed relations, and at the 0.05 level in the case of non-directed relations. The accuracy of 72.61 % for determining the correct label at the first guess (86.40 % for second guess) in Table 11 is equivalent to an F-measure of 0.84 (0.92) when retrieving relation types only. For relations where at least one concept could be grounded, values of 74.03 % for first suggestion correct and 87.84% for first or second suggestion correct yield F-measures of 0.85 and 0.94, respectively. A comparison between the VSM method based on first guess data and other baseline scores also delivers Chi-squared significance values above 99.9%.

3.4. Interpretation

As expected, additional experiments show that the performance of the proposed methods depends on the specific relation type. The approach performs best for the relation type *study* with up to 90% of correct suggestions (for relations including direction) at the first guess, and an ARP around 1.15. *Study* is particularly well suited as it has a clearly defined subject domain (*Person*, *Organization*) and object range (*ObjectTopic*, *AbstractTopic*). The *use* relation performed quite well, too, with a first guess correct for the directional setting of approximately 65% (nodir: 84%). As already mentioned at the beginning of Section 3, even domain experts disagreed on the use of the *effectOn* versus the *takeActionBy* relation types, which is also reflected by the lower accuracy in the individual performances of the respective predicates. The *subClassOf/superClassOf* relations are characterized by varying accuracy depending on the configuration settings, but the average performance is the lowest among all relations, especially for directional settings.

Table 12 presents an overview of the *a posteriori* accuracy of relation detection approaches found in literature to provide the reader with an impression of the performance of current state-of-the-art solutions. Please note that these methods are *not directly comparable* due to heterogeneous corpora, evaluation methods and settings. Another important factor to consider is whether evaluations are conducted *a priori* or *a posteriori*. Schutz and Buitelaar [45] point out that the average precision of a system can easily be 10% higher if evaluated *a posteriori*. We provide *a priori* evaluation results because the correct labels of the evaluation relations were defined *before* our method was applied.

Authors	Domain	Evaluation Corpora	% correct
Ciaramita et al. [20]	Biomedicine	GENIA	76.5%
Reinberger et al. [22]	Biomedicine	Medline abstracts	42.0%
Rinaldi et al. [23]	Biomedicine	GENIA	68.2 - 84.8%

Table 12: Approaches towards relation detection (accuracy in the case of Rinaldi et al. varies by corpus and relation type)

Overall, the method presented in this paper is particularly promising in situations where the domain and range restrictions of the relations used in the domain ontology are rather tight, especially if additional property restrictions on classes are available.

4. Conclusions

This paper presents a method for computing relation labels for unlabeled relations based on corpus analysis as well as semantic inference and validation. It evaluates several strategies for integrating a machine learning technique based on the vector space model with structured data from external sources such as DBpedia and OpenCyc. The main contributions of this research are: (i) introducing a method that integrates external knowledge with a machine learning approach; (ii) conducting an extensive formal evaluation to assess the performance of this method; (iii) outlining the advantages of hybrid approaches and current problems with methods solely relying on structured data.

The evaluation presented in Section 3 shows that refining relation label suggestions using external data sources yields superior results. Currently, certain data quality issues such as incorrect mappings (e.g. *activist* → *activism* → *AbstractTopic*) and missing data (e.g. Scarlet, where only eight out of 159 relations could be found) remain to be solved for the method to unfold its full potential. Nevertheless, as methods leveraging Semantic Web resources become more popular and collaborative approaches such as DBpedia.org and GeoNames.org continue to attract volunteers, the quality and quantity of structured data sources is expected to increase significantly.

Future research should emphasize the integration of additional, heterogeneous data sources - including strategies for resolving conflicts between annotations from multiple sources. Disambiguation and mediation techniques are a cornerstone for addressing this challenge and providing a fine-grained and accurate assessment of relation types. Support for multiple relations between concepts, and thresholds to detect if no relation label at all is appropriate in a particular case, will extend the range of possible applications for the method.

References

[1] T. R. Gruber, Toward principles for the design of ontologies used for knowledge sharing?, International Journal of Human-Computer Studies 43 (5-6) (1995) 907–928.

[2] A. Maedche, S. Staab, Ontology learning for the semantic web, IEEE Intelligent Systems 16 (2) (2001) 72–79.

[3] P. Buitelaar, P. Cimiano, On the “ontology” in ontology learning, in: P. Buitelaar, P. Cimiano (Eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Vol. 167 of Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, 2008, pp. v–ix.

[4] T. Morita, N. Fukuta, N. Izumi, T. Yamaguchi, DODDLE-OWL: Interactive Domain Ontology Development with Open Source Software in Java, IEICE Transactions on Information and Systems 91-D (4) (2008) 945–958.

[5] C. Fellbaum, Wordnet - an electronic lexical database, Computational Linguistics 25 (2) (1998) 292–296.

[6] M. Kavalec, V. Svátek, A study on automated relation labelling in ontology learning, in: P. Buitelaar, P. Cimiano, B. Magnini (Eds.), Ontology Learning from Text: Methods, Evaluation and Applications, no. 123 in Frontiers in Artificial Intelligence and Applications, IOS Press, 2005, pp. 44–58.

[7] R. Girju, P. Nakov, V. Nastase, S. Szpakowicz, P. Turney, D. Yuret, Semeval-2007 task 04: Classification of semantic relations between nominals, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 13–18.

[8] W. Liu, A. Weichselbraun, A. Scharl, E. Chang, Semi-automatic ontology extension using spreading activation, Journal of Universal Knowledge Management 0 (1) (2005) 50–58.

[9] P. Cimiano, J. Wenderoth, Automatic acquisition of ranked qualia structures from the web, in: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, Vol. 45, The Association for Computer Linguistics, 2007, pp. 888–895.

[10] M. Poesio, A. Almuhareb, Identifying concept attributes using a classifier, in: Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition, Association for Computational Linguistics, Ann Arbor, Michigan, 2005, pp. 18–27.

[11] A. Maedche, V. Pekar, S. Staab, Ontology learning part one - on discovering taxonomic relations from the web, in: N. Zhong, J. Liu, Y. Yao (Eds.), Web Intelligence, Springer, 2002, pp. 301–322.

[12] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, The Journal of Machine Learning Research 3 (2003) 1083–1106.

[13] C. Nicolae, G. Nicolae, S. Harabagiu, UTD-HLT-CG: Semantic architecture for metonymy resolution and classification of nominal relations, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 454–459.

[14] R. Snow, D. Jurafsky, A. Y. Ng, Semantic taxonomy induction from heterogenous evidence, in: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (ACL), Association for Computational Linguistics, Morristown, NJ, USA, 2006, pp. 801–808.

[15] M. Ruiz-Casado, E. Alfonseca, P. Castells, Automatising the learning of lexical patterns: An application to the enrichment of wordnet by extracting semantic relationships from wikipedia, Data & Knowledge Engineering 61 (3) (2007) 484 – 499.

[16] A. Zouaq, R. Nkambou, Enhancing learning objects with an ontology-based memory, IEEE Transactions on Knowledge and Data Engineering 21 (2009) 881–893.

[17] R. Navigli, P. Velardi, Learning domain ontologies from document warehouses and dedicated web sites, Computational Linguistics 30 (2) (2004) 151–179.

[18] R. Navigli, P. Velardi, Enriching a formal ontology with a thesaurus: an application in the cultural heritage domain, in: Proceedings of the 2nd Workshop on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Association for Computational Linguistics, Sydney, Australia, 2006, pp. 1–9.

[19] R. Snow, D. Jurafsky, A. Y. Ng, Learning syntactic patterns for automatic hypernym discovery, in: L. K. Saul, Y. Weiss, L. Bottou (Eds.), Advances in Neural Information Processing Systems 17, MIT Press, Cambridge, MA, 2005, pp. 1297–1304.

[20] M. Ciaramita, A. Gangemi, E. Ratsch, J. Saric, I. Rojas, Unsupervised learning of semantic relations between concepts of a molecular biology ontology, in: P. Buitelaar, P. Cimiano (Eds.), Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Vol. 167 of Frontiers in Artificial Intelligence and Applications, IOS Press, Amsterdam, Netherlands, 2008, pp. 91–103.

[21] S. Buchholz, E. Marsi, CoNLL-X shared task on multilingual dependency parsing, in: Proceedings of the 10th Conference on Computational Natural Language Learning, 2006, pp. 189–210.

[22] M.-L. Reinberger, P. Spyns, A. J. Pretorius, W. Daelemans, Automatic initiation of an ontology, in: R. Meersman, Z. Tari (Eds.), On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, Vol. 3290 of Lecture Notes in Computer Science, Springer, Berlin / Heidelberg, 2004, pp. 600–617.

[23] F. Rinaldi, G. Schneider, K. Kaljurand, M. Hess, M. Romacker, An environment for relation mining over richly annotated corpora: the case of genia., BMC Bioinformatics 7 (S-3) (2006) S3.

[24] D. Sánchez, A. Moreno, Learning non-taxonomic relationships from web documents for domain ontology construction, Data & Knowledge Engineering 64 (3) (2008) 600–623.

[25] C. Giuliano, A. Lavelli, D. Pighin, L. Romano, FBK-IRST: Kernel methods for semantic relation extraction, in: Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007), Association for Computational Linguistics, Prague, Czech Republic, 2007, pp. 141–144.

[26] M. Chagnoux, N. Hernandez, N. Aussenac-Gilles, An interactive pattern based approach for extracting non-taxonomic relations from texts, in: P. Buitelaar, P. Cimiano, G. Paliouras, M. Spilliopoulou (Eds.), Workshop on Ontology Learning and Population (associated to ECAI 2008) (OLP), University of Patras, Patras, Greece, 2008, pp. 1–6.

[27] M. A. Hearst, Automatic acquisition of hyponyms from large text corpora, in: Proceedings of the Fourteenth International Conference on Computational Linguistics (COLING'92), Nantes, France, 1992, pp. 539–545.

[28] M. d'Aquin, E. Motta, M. Sabou, S. Angeletou, L. Gridinoc, V. Lopez, D. Guidi, Toward a new generation of semantic web applications, IEEE Intelligent Systems 23 (3) (2008) 20–28.

[29] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, A. Yates, Web-scale information extraction in knowitall:(preliminary results), in: Proceedings of the 13th international conference on World Wide Web, ACM, New York, NY, USA, 2004, pp. 100–110.

[30] H. Yang, J. Callan, A metric-based framework for automatic taxonomy induction, in: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL2009), Singapore, 2009, pp. 271–279.

[31] R. Girju, A. Badulescu, D. I. Moldovan, Learning semantic constraints for the automatic discovery of part-whole relations, in: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003), Association for Computational Linguistics, Morristown, NJ, USA, 2003, pp. 1–8.

[32] M. Pennacchiotti, P. Pantel, Ontologizing semantic relations, in: ACL, The Association for Computer Linguistics, 2006, pp. 793–800.

[33] P. Pantel, M. Pennacchiotti, Automatically harvesting and ontologizing semantic relations, in: P. Buitelaar, P. Cimiano (Eds.), Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge, Vol. 167 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2008, pp. 171–195.

[34] C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, S. Hellmann, DBpedia - a crystallization point for the web of data, Journal of Web Semantics: Science, Services and Agents on the World Wide Web 7 (3) (2009) 154–165.

[35] J. Lehmann, J. Schüppel, S. Auer, Discovering unknown connections - the dbpedia relationship finder, in: S. Auer, C. Bizer, C. Müller, A. V. Zhdanova (Eds.), The Social Semantic Web 2007, Proceedings of the 1st Conference on Social Semantic Web (CSSW), Vol. 113 of LNI, GI, 2007, pp. 99–110.

[36] J. Cullen, A. Bryman, The knowledge acquisition bottleneck: Time for reassessment?, Expert Systems 5 (3) (1988) 216–225.

[37] A. Weichselbraun, G. Wohlgenannt, A. Scharl, M. Granitzer, T. Neidhart, A. Juffinger, Discovery and evaluation of non-taxonomic relations in domain ontologies, International Journal of Metadata, Semantics and Ontologies 4 (3) (2009) 212–222.

[38] D. Lin, S. Zhao, L. Qin, M. Zhou, Identifying synonyms among distributionally similar words, in: Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-2003), 2003, pp. 1492–1493.

[39] T. Chklovski, P. Pantel, Verbocean: Mining the web for fine-grained semantic verb relations, in: D. Lin, D. Wu (Eds.), Proceedings of EMNLP 2004, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 33–40.

[40] P. D. Turney, M. L. Littman, Corpus-based learning of analogies and semantic relations, Machine Learning 60 (1-3) (2005) 251–278.

[41] M. Banko, O. Etzioni, The tradeoffs between open and traditional relation extraction, in: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-08: HLT), Association for Computational Linguistics, Columbus, Ohio, 2008, pp. 28–36.

[42] A. Mädche, S. Staab, Handbook on ontologies, in: S. Staab, R. Studer (Eds.), Handbook on Ontologies, International Handbooks on Information Systems, Springer, 2004, Ch. Ontology Learning, pp. 173–190.

[43] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information Processing & Management 24 (5) (1988) 513–523.

[44] W. Wong, W. Liu, M. Bennamoun, Acquiring semantic relations using the web for constructing lightweight ontologies, in: 13th Pacific-Asia Conference on Knowledge Discovery and Data Mining, Bangkok, Thailand, 2009, pp. 277–288.

[45] A. Schutz, P. Buitelaar, Relext: A tool for relation extraction from text in ontology extension, in: Proceedings of the 4th International Semantic Web Conference (ISWC 2005), Galway, Ireland, 2005, pp. 593–606.