Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence

Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm

Andrea Moro and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
Via Salaria, 113, 00198 Roma, Italy
{*moro,navigli*}*@di.uniroma1.it*

Abstract

In this paper we present an approach aimed at enriching the Open Information Extraction paradigm with semantic relation ontologization by integrating syntactic and semantic features into its workflow. To achieve this goal, we combine deep syntactic analysis and distributional semantics using a shortest path kernel method and soft clustering. The output of our system is a set of automatically discovered and ontologized semantic relations.

1 Introduction

One of the long standing problems of Artificial Intelligence is Machine Reading (MR) [Mitchell, 2005; Etzioni *et al.*, 2006; Poon and et al., 2010], i.e. the problem of automatic, unsupervised understanding of text. Over time the Natural Language Processing (NLP) community has developed many tools for this problem. One major approach to the MR problem is via the Open Information Extraction (OIE) paradigm [Etzioni *et al.*, 2008; Wu and Weld, 2010], that is the extraction of a large number of relations from huge corpora such as the Web. However, these techniques are mainly concerned with the surface, i.e. textual, realization of relations, without performing any semantic or deep syntactic analysis.

Taking semantics explicitly into account is the main goal of knowledge acquisition techniques, which represent another important set of tools for tackling the MR problem. Such approaches are typically based on a small number of ontologized semantic relation types. One of the most well-known representative approaches is the Never Ending Language Learner (NELL) [Carlson *et al.*, 2010], which focuses on the continuous learning/extraction of a small, fixed number of semantic relations from the Web. Other approaches extract taxonomic structure information from raw text [Velardi *et al.*, 2013], or exploit semi-structured resources [Hovy *et al.*, 2013] such as WordNet [Miller *et al.*, 1990], Wikipedia[1] and Freebase[2]. The main outputs of these approaches are taxonomies [Ponzetto and Strube, 2011], ontologies [Poon and Domingos, 2010; Hoffart *et al.*, 2013], semantic networks [Navigli and Ponzetto, 2012a; Nastase and Strube, 2013], relation extraction rules [Krause *et al.*, 2012] and enriched computational lexicons [Pennacchiotti and Pantel, 2006; Navigli, 2005]. These techniques have already been shown to enhance the performance of automatic approaches for a wealth of tasks related to the MR problem, such as Word Sense Disambiguation [Navigli and Ponzetto, 2012b; Miller *et al.*, 2012], Information Extraction [Hoffmann *et al.*, 2011] and many others [Schierle and Trabold, 2008; Fernández *et al.*, 2011]. Nonetheless, one major drawback of these approaches is that, either they are not "open", i.e. they are limited in the number of learned relation types [Carlson *et al.*, 2010; Hoffart *et al.*, 2013; Nastase and Strube, 2013], or they do not scale with the dimension of the input corpus [Poon and Domingos, 2010].

Starting with Soderland and Mandhani [2007], middle ground approaches have been proposed which combine the "open" nature of OIE with the semantic awareness of knowledge acquisition techniques. The last year, especially, has seen an increasing interest in semantically-enhanced OIE [Nakashole *et al.*, 2012; Min *et al.*, 2012; Moro and Navigli, 2012; Sun and Grishman, 2012]. At the core of this new paradigm lie the language phenomena of synonymy and polysemy, both of which require deep language understanding. However, none of the existing approaches fully addresses these issues. In this paper we aim at filling this gap by providing the following novel contributions:

- we leverage syntactic analysis to improve the quality of the extracted surface realizations of relations;
- we integrate distributional semantics into syntactic analysis and define a new kernel-based similarity measure which we use for merging synonymous surface realizations into full-fledged semantic relations;
- we exploit category-based distributional semantics to provide semantic type signatures for the acquired semantic relations.

2 Related Work

Since the introduction of the OIE paradigm, the NLP community has begun to adopt a new point of view for the study of relation extraction: extending the textual-based ideas of the OIE paradigm towards a deeper and more complete understanding of text based on semantic analysis. Over the last year several approaches have been presented to the problem

[1] http://en.wikipedia.org/
[2] http://www.freebase.com/

of ontologizing semantic relations, i.e. automatically identifying synonymous relational phrases together with a semantic description of their typical arguments. A major difficulty of the ontologization problem regards two well-known features of language: synonymy and polysemy. One way to deal with synonymy is to exploit the distributional semantics of the relational phrases [Sun and Grishman, 2012; Moro and Navigli, 2012]. First the context terms occurring in a window surrounding the relational phrases are collected. Then, cosine similarity is utilized for identifying relational phrases with similar contexts. However, this approach suffers from the problem of data sparsity, in that it needs the exact relational phrases to occur several times within a given text. A second approach exploits the arguments of the relationship, in order to determine the similarity of relational phrases [Nakashole *et al.*, 2012]. However, the same arguments can be related by several semantic relations (e.g. *married to, is a friend of, started a company with*).

As for the problem of polysemy, to date soft clustering techniques have been applied. Soft clustering allows relational phrases to belong to one or more clusters. However, the current approaches address this problem, either when the set of semantic relations is small and known in advance [Min *et al.*, 2012], or by considering static methods, i.e. fixing a threshold and clustering together those relational phrases above the threshold [Moro and Navigli, 2012]. In contrast, in this paper we present an approach to clustering relational phrases into an unknown number of semantic relations by exploiting a soft version of the K-medoids algorithm [Han and Kamber, 2006].

The state-of-the-art systems most closely related to our approach are PATTY [Nakashole *et al.*, 2012] and WiSeNet [Moro and Navigli, 2012], neither of which, however, exploits deep syntactic analysis or advanced soft clustering techniques. In this paper we take the problem of the automatic ontologization of semantic relations to the next level, by integrating deep syntactic analysis together with distributional semantics into a new shortest path kernel method for clustering synonymous relational phrases. The use of a soft version of the K-medoids algorithm enables us to effectively take into account the problems of synonymy and polysemy associated with relational phrases.

3 Integrating Syntax and Semantics into OIE

Our approach consists of three steps: relation extraction, relation ontologization and relation disambiguation. During the first step we extract relational phrases from Wikipedia by exploiting deep syntactic analysis, e.g. we extract the relational phrases *is a member of, is a part of, is a territory of*. In the second step we define a shortest path kernel similarity measure that integrates semantic and syntactic features to automatically build relation synsets, i.e. clusters of synonymous relational phrases with semantic type signatures for their domain and range. For instance, we cluster together the relational phrases *is a member of* and *is a part of*, while we group together, in a separate cluster, *is a part of* and *is a territory of*. Finally, we disambiguate the relational phrases extracted from Wikipedia using these relation synsets, obtaining a large set of automatically ontologized semantic relations, e.g. we recognize that the relational phrase *is a part of* is a synonym of *is a territory of* when we consider the sentence *Nunavut is a part of Canada*, while it is a synonym of *is a member of* for the sentence *Taproot Theatre Company is a part of Theatre Communications Group*.

3.1 Step 1: Relation Extraction

In this section we describe the approach that we used to extract a large number of relational phrases and relation instances from Wikipedia.

Definition 1 *A **relational phrase** π is a sequence of words that comprises at least one verb and that can be used to express a semantic relation between a subject and an object.*

For instance, a good relational phrase is: *is located in*; a bad relational phrase is: *and located in*.

Definition 2 *A **relation instance** is a triple (p_1, π, p_2) where p_1, p_2 are concepts (i.e., Wikipedia pages) and π is a relational phrase.*

For instance, *(Nunavut, is a part of, Canada)* is a valid relation instance. We build upon the heuristic presented by Moro and Navigli [2012], summarized in the first part of Algorithm 1 (lines 4–13). During this first part, the algorithm analyzes each sentence of each Wikipedia page and extracts a huge number of relational phrases between hyperlink pairs using shallow syntactic analysis (line 10 in Algorithm 1). In contrast to [Etzioni *et al.*, 2008], this heuristic exploits the manual annotations in Wikipedia, by extracting unambiguously hyperlinked arguments. Even so, we still extract overspecific and noisy information (e.g. *is the name Gulliver gives his nurse in Book II of* and *but then lost to*). To overcome this problem, first, we filter out all the relational phrases that do not extract more than a minimum number η of relation instances (lines 14–17 in Algorithm 1, see Section 4 for details on tuning). Second, and more importantly, we apply a novel constraint to the surviving relational phrases based on their syntactic structure. As we have already extracted relation instances, we propose a simplified and computationally efficient[3] test to check if each relational phrase relates its arguments as subject and object. We build an artificial phrase for each relational phrase π by concatenating the character "*x*", the relational phrase π and the character "*y*", i.e. we obtain "$x\ \pi\ y$". Then we apply a dependency parser and check whether "*x*" and "*y*" are marked as subject of a word in π and object of a, not necessarily the same, word in π (see lines 18–22 of Algorithm 1). For example, the relational phrase *is located in* satisfies the constraint (see Figure 1a), while the relational phrase *and located in* does not (see Figure 1b).

3.2 Step 2: Relation Ontologization

As a result of the first step we obtain a set I of relation instances and a set P of relational phrases (line 23 of Algorithm 1). In this second step we ontologize the relational phrases P.

[3]The whole Wikipedia corpus consists of 88 million sentences with a mean length of 30 words, while using our approach we syntactically parsed just half a million sentences with a mean length of 10.

Algorithm 1 Extracting relation instances and relational phrases from Wikipedia.

input: W, the set of Wikipedia pages.
output: I, the set of relation instances;
P, the set of relational phrases.
function EXTRACTRELATIONINSTANCESANDPHRASES(W)
  $I := \emptyset$; $P := \emptyset$;
  for each $page \in W$ **do**
    for each $sent \in page$ **do**
      $H(sent) :=$ all the pairs of hyperlinks in $sent$
      for each $(h_1, h_2) \in H(sent)$ **do**
        $\pi :=$ *text between* h_1 and h_2
        if π contains a verb **then**
          $I := I \cup \{(h_1, \pi, h_2)\}$
          $count[\pi] + +$
          $P := P \cup \{\pi\}$
  for each $\pi \in P$ **do**
    if $count[\pi] < \eta$ **then**
      $I := I \setminus \{(p, \pi, q) \in I : \exists p, q \in W\}$
      $P := P \setminus \{\pi\}$
  for each $\pi \in P$ **do**
    $dG := depParser(\text{“}x\ \text{”} + \pi + \text{“}\ y\text{”})$
    if *according to dG, x is not a subject of a word in* π ***or*** *y is not an object of a word in* π **then**
      $I := I \setminus \{(p, \pi, q) \in I : \exists p, q \in W\}$
      $P := P \setminus \{\pi\}$
  return I, P

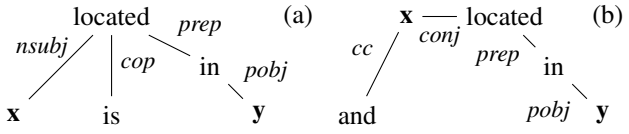

Figure 1: The dependency trees for the sentences *x is located in y* and *x and located in y*.

Like any other language component, relational phrases are affected by the well-known phenomena of synonymy (e.g. *is situated in* and *is located in*) and polysemy[4] (e.g. *is part of* can be replaced with *is a territory of* and *is a member of* depending on the context). To take into account these issues we exploit soft clustering techniques to synergistically cluster synonymous relational phrases, while at the same time letting polysemous relational phrases belong to more than one cluster. Finally, we create two semantic type signatures of the subjects and objects considered by the relational phrases in each cluster.

In order to perform this step, we consider three different aspects of each relational phrase: its dependency structure, the distributional semantics of its words and the semantics of its arguments.

Distributional Semantics and Shortest Path Dependency Kernel. Given a relational phrase π and the dependency tree of the artificial phrase “$x\ \pi\ y$”, we define π^{sp} as the sequence of words on the shortest path between x and y and $\pi^{sp}[i]$ as the i-th word on the path. For instance, given $\pi =$ *is located in* we have $\pi^{sp} = (\textit{located, in})$ and $\pi^{sp}[2] = \textit{in}$ (see Figure 1a). Given two relational phrases π_1 and π_2, we assume that π_1 and π_2 can be synonyms only if they share the same sequence of dependency relations and semantically close words within their shortest paths. We formulate this assumption as follows:

$$sim(\pi_1, \pi_2) = \begin{cases} 0, & |\pi_1^{sp}| \neq |\pi_2^{sp}| \text{ or } \exists i, \text{s.t.} \\ & type(\pi_1^{sp}[i], \pi_1^{sp}[i+1]) \neq \\ & type(\pi_2^{sp}[i], \pi_2^{sp}[i+1]) \\ & \text{or } dsim(\pi_1^{sp}[i], \pi_2^{sp}[i]) < \theta_1 \\ \frac{g(\pi_1^{sp}, \pi_2^{sp})}{Z} & \text{otherwise} \end{cases} \quad (1)$$

where $type(\pi_j^{sp}[i], \pi_j^{sp}[i+1])$ is the syntactic dependency between the i-th and $(i+1)$-th words of the shortest path π_j^{sp}, $dsim(w_1, w_2)$ is a measure of semantic similarity between words and $g(\pi_1^{sp}, \pi_2^{sp})$ is our kernel, described hereafter, and Z is a normalization factor.

Since state-of-the-art kernel methods are either used to find synonymous phrases without considering the special role of the shortest paths between the arguments of the phrases [Croce *et al.*, 2011], or they are exploited to learn a classifier for a specific set of relations [Bunescu and Mooney, 2005], we define a new kernel similarity measure based on our shortest path assumption.

Our kernel computes the similarity score between two relational phrases sharing their shortest path as follows:

$$g(\pi_1^{sp}, \pi_2^{sp}) = \sum_{i=1}^{|\pi_1^{sp}|} dsim(\pi_1^{sp}[i], \pi_2^{sp}[i]) + f(\pi_1^{sp}[i], \pi_2^{sp}[i])$$

$$f(n_1, n_2) = \sum_{\substack{w_1 \in Children(n_1), w_1 \notin \pi_1^{sp}, \\ w_2 \in Children(n_2), w_2 \notin \pi_2^{sp}, \\ type(n_1, w_1) = type(n_2, w_2)}} dsim(w_1, w_2)$$

Our kernel sums the distributional similarity of the words along the considered shortest paths and that of their children (i.e. the function $f(n_1, n_2)$ in the above equation). Consider $\pi_1 =$ *is a territory of* and $\pi_2 =$ *is a part of*. We start by comparing the first word of each shortest path, i.e. *territory* and *part* (see Figure 2), and, since they are not the same, we consider their distributional similarity score which is equal to 0.8. Then we consider the corresponding children (i.e. *is* and *a*), that are not on the shortest paths, of these nodes. In this case, they are equal in number, type of edges and words, and so we add 1 for the word *is* and another 1 for the word *a* to our score. Finally, we do the same for the last node on the shortest path *of*, obtaining a score of 3.8. We then normalize this score by the maximum number of considered words, i.e. 4, obtaining a similarity score of 0.95. Instead, the similarity score between *is a territory of* and *is a member of* is 0.75.

To calculate the aforementioned distributional similarity measure $dsim(w_1, w_2)$ between the words w_1 and w_2, we exploit the distributional hypothesis. Given a word w, following Mitchell and Lapata [2010], we define a distributional vector $distrVect_l(w)$ for the words occurring in the left window of

[4]Min *et al.* [2012] estimated that roughly 20% of relational phrases can represent at least two different interpretations as different semantic relations.

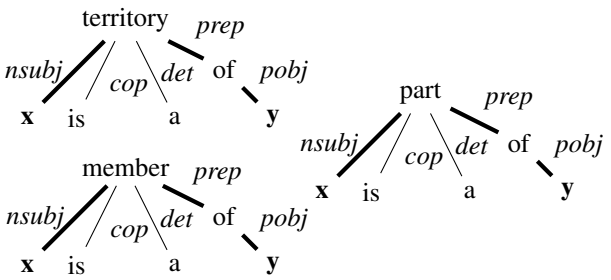

Figure 2: The shortest paths (in bold) between x and y of the following phrases: *x is a territory of y, x is a part of y, x is a member of y*.

w in which each component is computed as the probability of a given context word cw to occur on the left of w divided by its prior probability, $\frac{P_{left}(cw|w)}{P_{left}(cw)}$. Similarly, we define the distributional vector $distrVect_r(w)$ for the context words that appear to the right of the word w. The probabilities are estimated from a big corpus (see Section 4 for details). Then, we compute the cosine similarity of the left and right vectors:

$$\begin{aligned} ds_{left}(w_1, w_2) &= cos(distrVect_l(w_1), distrVect_l(w_2)) \\ ds_{right}(w_1, w_2) &= cos(distrVect_r(w_1), distrVect_r(w_2)), \end{aligned}$$

and finally, by applying the harmonic mean between the two scores above, we obtain a single value of distributional similarity for each pair of words:

$$dsim(w_1, w_2) = HaMean(ds_{left}(w_1, w_2), ds_{right}(w_1, w_2)).$$

Distributed Soft Kernel K-medoids Algorithm. We will now describe the clustering algorithm that we use to build the set Σ of relation synsets.

Definition 3 *A **relation synset** σ is a set of synonymous relational phrases.*

For instance, $\sigma = \{$*is a territory of, is a part of*$\}$. To obtain the set Σ of relation synsets we exploit a soft version of the K-medoids algorithm, whose pseudocode is shown in Algorithm 2. As our initial centers $\mathcal{C}$ we select all the relational phrases in P (line 4), thereby avoiding tuning the number of clusters K. We use the K-medoids algorithm instead of K-means because we cannot explicitly compute the centroids, but instead have to select the best approximation by finding the relational phrase that maximizes the similarity scores against the other relational phrases in the cluster [Zhang and Rudnicky, 2002].

Next, we start the distributed soft kernel K-medoids algorithm. We define a membership matrix M which describes the assignment of relational phrases to clusters. Because we selected each relational phrase as an initial center of a cluster, M is a $|P| \times |P|$ square matrix. The i-th row contains the membership scores of the i-th relational phrase with respect to the various clusters, while the j-th column contains the membership scores of the relational phrases against the j-th cluster.

Algorithm 2 Distributed Soft Kernel K-medoids

1: **input:** P, the set of relational phrases.
2: **output:** Σ, the set of relation synsets.
3: **function** SOFTCLUSTERS(P)
4: $\quad\mathcal{C} := P$
5: $\quad$**repeat**
6: $\quad\quad$Distributed update of M using equation 2;
7: $\quad\quad\mathcal{C}' := \mathcal{C}$;
8: $\quad\quad\mathcal{C} :=$ Distributed update of $\mathcal{C}$ using equation 3;
9: $\quad$**until** $\mathcal{C}' \neq \mathcal{C}$
10: $\quad$**return** $\Sigma :=$ Extract the relation synsets from M;

During the first phase of the iterations, we update the membership matrix with the new membership scores with respect to the current centers (line 6 in Algorithm 2). We distribute this computation over all the relational phrases, as this phase requires only the old membership scores of the considered relational phrase and the similarity scores against the current centers. We use the following equation to update the membership score of a relational phrase π against the i-th cluster represented by its center $c_i \in \mathcal{C}$:

$$newW_t(\pi, i) = \frac{(e^t - 1)M_{t-1}(\pi, i) + sim(\pi, c_i)}{e^t}$$

$$M_t(\pi, i) = \begin{cases} newW_t(\pi, i), & \text{if } newW_t(\pi, i) > \theta_2. \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

where t is the number of the current iteration of the algorithm. We use an exponential update of the membership scores to ensure a smoother convergence of the algorithm. Moreover, we discard all the membership scores less than a threshold θ_2 so as to limit the number of clusters each relational phrase belongs to (see Section 4 for tuning details). During the second phase of the clustering algorithm we compute the new centers $\mathcal{C}$ for each cluster $C_i = \{\pi_j : M(j, i) > 0\}$ in the following manner:

$$c_i = \underset{\pi^\star \in C_i}{\operatorname{argmax}} \sum_{\pi \in C_i} sim(\pi, \pi^\star). \quad (3)$$

In order to select each center c_i, Formula 3 considers only the relational phrases in the i-th cluster C_i, so we can distribute the computation over the number of clusters (line 8 in Algorithm 2). Notice that, as we select the best centroid approximation, some of the clusters end up being merged during the iterations by selecting the same relational phrase as center.

We run the algorithm until the centers remain stable (the convergence is assured by the exponential update) and then we extract, from the membership matrix, the description of the automatically created relation synsets Σ, i.e. we create a set of synonymous relational phrases for each column in the membership matrix by selecting those relational phrases with a positive membership score.

Semantic Type Signatures of the Relation Synsets. We now describe the construction of the semantic description of the relation synsets that will be used to identify which kind of arguments a relation synset considers and that will be used in the next step to disambiguate ambiguous relational phrases.

Definition 4 *The **semantic type signatures** for the domain (i.e. the concepts occurring to the left) and range (i.e. the*

concepts occurring to the right) of a relation synset σ are two distributional vectors $d(\sigma)$ and $r(\sigma)$ of Wikipedia categories.

Since the arguments of our relation instances, extracted during the first step (see Section 3.1), are Wikipedia pages, we can exploit the categories associated with each Wikipedia page to build semantic type signatures for the domain and range of relational phrases.

We define a category vector $catVect_d(\pi)$ for the concepts in the domain of π. Each component of the vector is computed as the probability that a given category will appear within the domain of π divided by its prior probability: $\frac{P_{domain}(category|\pi)}{P_{domain}(category)}$. Similarly, we define the category vector $catVect_r(\pi)$ for the range of π.

In order to avoid overspecific Wikipedia categories and to keep the number of dimensions low, we consider only the categories which are at distance ≤ 2 from the root[5] of the Wikipedia category hierarchy. In this way we obtain a set C of 657 top-level categories. Given a relational phrase π we have at least η Wikipedia pages in the domain and range of π (see lines 14–17 of Algorithm 1). We count the number of times we reach each top-level category in C starting from the categories of the Wikipedia pages and going up the Wikipedia category hierarchy. Using these counts we estimate the aforementioned probabilities for the domain and range of π.

To extend these category vectors from relational phrases to relation synsets we merge (using a weighted arithmetic mean on the number of extracted relation instances) each category vector of the relational phrases contained in the same relation synset. As a result, we obtain two category vectors $d(\sigma)$ and $r(\sigma)$ for each relation synset σ, i.e. our semantic type signature of σ. In Table 1 we show some of the obtained relation synsets.

3.3 Step 3: Relation Disambiguation

After the first step we obtained a set I of relation instances and a set P of relational phrases. Then, in the second step we ontologized the relational phrases in P obtaining a large set of relation synsets Σ. Now, in the third step, we disambiguate each textual relation instance in I with the semantically closest relation synset in Σ.

To disambiguate a relation instance $(p_1, \pi, p_2) \in I$ we consider all the relation synsets $\Sigma_\pi = \{\sigma \in \Sigma : \pi \in \sigma\}$ that contain π and we select the semantically closest relation synset:

$$(p_1, \operatorname*{argmax}_{\sigma \in \Sigma_\pi} semsim(p_1, \sigma, p_2), p_2).$$

Given a relation synset σ with its semantic type signatures $d(\sigma)$ and $r(\sigma)$ and given the distributional vectors $catVect(p_1)$ and $catVect(p_2)$ computed similarly to $catVect_l(\pi)$ and $catVect_r(\pi)$, we calculate the cosine similarity between them:

$$ss_d(p_1, \sigma) = cos(catVect(p_1), d(\sigma))$$
$$ss_r(p_2, \sigma) = cos(catVect(p_2), r(\sigma)).$$

Then we combine these values into their harmonic mean:

$$semsim(p_1, \sigma, p_2) = HaMean(ss_d(p_1, \sigma), ss_r(p_2, \sigma)).$$

[5] en.wikipedia.org/wiki/Category:Main_topic_classifications

Domain	Relation Synset	Range
Arts	{is located in the small village of, . . ., is located in the small rural town of}	Places
Corporate groups	{is a member of an, . . ., were the members of the}	Corporate groups
Geography	{is a valley of, is a zone of, . . ., is a territory of}	Geography by place

Table 1: Examples of relation synsets together with their top Wikipedia category for their domain and range.

4 Experimental Setup

Step 1. To run our relation extraction step we used the English Wikipedia dump of December 1st, 2012. To fix the value of our parameter $\eta = 2$ (see line 15 of Algorithm 1) we manually evaluated the accuracy of a random set of 250 relational phrases for $\eta = \{1, \ldots, 5\}$. To syntactically parse the phrases we used the Stanford parser [Marneffe *et al.*, 2006].

Step 2. To obtain the distributional description of the words within the relational phrases we used Gigaword [Parker *et al.*, 2011]. The probabilities of the distributional vectors of domain and range were also estimated from Gigaword, using the optimal parameters of Mitchell and Lapata [2010]. To set up the two thresholds θ_1 and θ_2 (see Formula 1 and 2) needed by our system we built a tuning set composed of 200 manually clustered relational phrases. Then we selected the values that maximized the pair-counting F-Measure (i.e. the number of relational phrase pairs clustered in the same way as the tuning set). We found the following optimal values: $\theta_1 = 0.6, \theta_2 = 0.86$.

Evaluations. We used Amazon Mechanical Turk for our evaluations. To ensure high-quality annotations by competent Turkers we built a gold standard for each evaluation task which will be described in the following section. To evaluate the agreement between judges we used the free-margin multirater kappa [Warrens, 2010] obtaining an agreement greater than 85% for each setup.

Statistics. During the first step we extracted $2,271,807$ relation instances with $278,945$ distinct relational phrases. As a result of step 2, after 80 iterations, we obtained $29,440$ relation synsets with two or more relational phrases and $155,207$ relation synsets with a single relational phrase. The number of ambiguous relational phrases, i.e. occurring in multiple synsets, was $18,457$ with a mean ambiguity of almost 3.

5 Experiments

Step 1: Ambiguous relation instances. We created a random sample of $2,000$ relation instances extracted during the first step. For each relation instance (p_1, π, p_2), we presented three judges with the title of p_1 and p_2 and the relational phrase π together with the first paragraph of the Wikipedia page of p_1 and p_2. Then we asked if the relation instance was correct. The results are shown in the first row of Table 2 together with the accuracy of our closest competitor,

Task	Our approach	WiSeNet
relation instances	91.8%	82.8%
relational phrases	94.5%	79.8%
relation synsets	85.0%	82.1%
disambiguated relation instances	88.6%	76.7%

Table 2: Accuracy of the outputs of our approach versus WiSeNet.

i.e. WiSeNet. An error analysis identified the wrong hyperlinks annotation in Wikipedia as the main class of error, e.g. *(Rajura, lies in the heart of the, Coal)* is extracted from the sentence: *Rajura lies in the heart of the coal- and cement-producing areas of Maharashtra*, where *coal* is a hyperlink and *Maharashtra* is not linked.

Step 1: Relational phrases. We created a second random sample of $2,000$ relational phrases. We presented the judges with each relational phrase and we asked if they could think of a subject and an object that would fit the phrase. The results are shown in the second row of Table 2. An error analysis identified the wrong syntactic parsing of relational phrases as the main class of error, e.g. *x site and was designated a y* in which *site* is considered as a verb instead of a noun compound modifier.

Step 2: Relation synsets. The third sample consisted of $2,000$ randomly chosen relation synsets which contained at least two relational phrases. We asked the judges if two relational phrases randomly chosen from the considered relation synset could be exchanged with each other to express the same semantic relation. The results are shown in the third row of Table 2. Antonymy was identified as the main class of error, i.e. we cluster together *was a predecessor of* and *was the successor of*.

Step 3: Disambiguated relation instances. In this evaluation we considered only the relation instances (p_1, π, p_2) which contained relational phrases associated with more than one relation synset, i.e. ambiguous relational phrases which we disambiguated with our relation synsets in step 3. We created a random sample containing $2,000$ of these disambiguated relation instances. We presented three judges with the title of p_1 and p_2 and a randomly chosen relational phrase from the disambiguated relation synset together with the first paragraph of the Wikipedia page of p_1 and p_2. We then asked if the relation instance was correct, as in the first evaluation. The results are shown in the fourth row of Table 2.

Discussion. In conclusion, our four evaluations showed the high quality of our outputs and greater accuracy over that reported by WiSeNet. More importantly, the last evaluation on the disambiguated relation instances shows that our intermediate steps are strong, as we maintain a small gap (roughly 3%) between the accuracy of the relation instances extracted during the first step and the relation instances disambiguated in the last step, while our closest competitor has a gap of more than 6%.

Gold Standard	Our approach	PATTY	YAGO2
163	129	126	31
	DBpedia	**Freebase**	**NELL**
	39	69	13

Table 3: Number of semantic relations covered by different resources.

5.1 Coverage of Semantic Relations

To assess the coverage of the extracted semantic relations we compared our relation synsets against a public dataset described in [Nakashole *et al.*, 2012]. This dataset is made up of 163 semantic relations manually identified in five Wikipedia pages about musicians. The dataset's authors calculated the number of these relations covered by well-known knowledge bases, i.e. YAGO2 [Hoffart *et al.*, 2013], DBpedia [Auer *et al.*, 2007], Freebase [Bollacker *et al.*, 2008], NELL [Carlson *et al.*, 2010] and PATTY [Nakashole *et al.*, 2012] (see Table 3). As was done in [Nakashole *et al.*, 2012], we manually associated each of these semantic relations with one of our relation synsets that best represented the considered semantic relation in terms of relational phrases and semantic type signatures, obtaining an increase in the number of recognized semantic relations over the best system, i.e. PATTY. As can be seen from Table 3, the number of semantic relations contained in the other resources is considerably lower, showing that those resources are still missing a lot of useful information. In contrast, our approach and PATTY demonstrate their ability to find and extract a wealth of semantic relations. Moreover, in contrast to PATTY, our approach takes into account the phenomenon of polysemous relational phrases.

6 Conclusions and Future Work

We presented an approach that provides a novel integration of automatic ontologization of semantic relations into the OIE paradigm by exploiting syntactic and semantic analysis. We demonstrated the quality of our approach by carrying out extensive manual evaluations and by comparing our approach against state-of-the-art resources, with competitive results.

The main contributions of this paper can be summarized as follows: i) we presented an efficient way to integrate syntax into the OIE workflow; ii) we introduced a new kernel similarity measure that combines syntax and distributional semantics; iii) we presented an algorithm for associating a semantic description with the typical arguments of a set of synonymous relational phrases, and which can also be exploited to disambiguate semantic relation instances.

We will release all the data to the research community (http://lcl.uniroma1.it/wisenet). As future work we plan to extend our current approach from binary to n-ary semantic relations, test the approach on other languages and, most importantly, build a hierarchical structure on the relation synsets along the lines of PATTY.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Starting Grant Multi-JEDI No. 259234.

References

[Auer *et al.*, 2007] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A Nucleus for a Web of Open Data. In *Proc. of ISWC/ASWC*, pages 722–735, 2007.

[Bollacker *et al.*, 2008] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proc. of SIGMOD*, pages 1247–1250, 2008.

[Bunescu and Mooney, 2005] R. C. Bunescu and R. J. Mooney. A Shortest Path Dependency Kernel for Relation Extraction. In *Proc. of HLT*, pages 724–731, 2005.

[Carlson *et al.*, 2010] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E.R. Hruschka Jr., and T.M. Mitchell. Toward an Architecture for Never-Ending Language Learning. In *Proc. of AAAI*, pages 1306–1313, 2010.

[Croce *et al.*, 2011] D. Croce, A. Moschitti, and R. Basili. Structured Lexical Similarity via Convolution Kernels on Dependency Trees. In *Proc. of EMNLP*, pages 1034–1046, 2011.

[Etzioni *et al.*, 2006] O. Etzioni, M. Banko, and M. J. Cafarella. Machine Reading. In *Proc. of AAAI*, pages 1517–1519, 2006.

[Etzioni *et al.*, 2008] O. Etzioni, M. Banko, S. Soderland, and D. S. Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008.

[Fernández *et al.*, 2011] M. Fernández, I. Cantador, V. Lopez, D. Vallet, P. Castells, and E. Motta. Semantically enhanced Information Retrieval: An ontology-based approach. *J. Web Sem.*, 9(4):434–452, 2011.

[Han and Kamber, 2006] J. Han and M. Kamber. *Data Mining. Concepts and Techniques*. Morgan Kaufmann, 2nd edition, 2006.

[Hoffart *et al.*, 2013] J. Hoffart, F. M. Suchanek, K. Berberich, and G. Weikum. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013.

[Hoffmann *et al.*, 2011] R. Hoffmann, C. Zhang, X. Ling, L. S. Zettlemoyer, and D. S. Weld. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *Proc. of ACL*, pages 541–550, 2011.

[Hovy *et al.*, 2013] E. H. Hovy, R. Navigli, and S. P. Ponzetto. Collaboratively built semi-structured content and Artificial Intelligence: The story so far. *Artificial Intelligence*, 194:2–27, 2013.

[Krause *et al.*, 2012] S. Krause, H. Li, H. Uszkoreit, and F. Xu. Large-scale learning of relation-extraction rules with distant supervision from the web. In *Proc. of ISWC*, pages 263–278, 2012.

[Marneffe *et al.*, 2006] M. Marneffe, B. MacCartney, and C. D. Manning. Generating Typed Dependency Parses from Phrase Structure Trees. In *Proc. of LREC*, 2006.

[Miller *et al.*, 1990] G. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. Miller. Introduction to WordNet: An On-Line Lexical Database. *Int. Journal of Lexicography*, 3(4):235–244, 1990.

[Miller *et al.*, 2012] T. Miller, C. Biemann, T. Zesch, and I. Gurevych. Using Distributional Similarity for Lexical Expansion in Knowledge-based Word Sense Disambiguation. In *Proc. of COLING*, 2012.

[Min *et al.*, 2012] B. Min, S. Shi, R. Grishman, and C. Lin. Ensemble Semantics for Large-scale Unsupervised Relation Extraction. In *Proc. of EMNLP-CoNLL*, pages 1027–1037, 2012.

[Mitchell and Lapata, 2010] J. Mitchell and M. Lapata. Composition in Distributional Models of Semantics. *Cognitive Science*, 34(8):1388–1429, 2010.

[Mitchell, 2005] T. M. Mitchell. Reading the Web: A Breakthrough Goal for AI. *AI Magazine*, 2005.

[Moro and Navigli, 2012] A. Moro and R. Navigli. WiSeNet: building a wikipedia-based semantic network with ontologized relations. In *Proc. of CIKM*, pages 1672–1676, 2012.

[Nakashole *et al.*, 2012] N. Nakashole, G. Weikum, and F. M. Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proc. of EMNLP-CoNLL*, pages 1135–1145, 2012.

[Nastase and Strube, 2013] V. Nastase and M. Strube. Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194:62–85, 2013.

[Navigli and Ponzetto, 2012a] R. Navigli and S. P. Ponzetto. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[Navigli and Ponzetto, 2012b] R. Navigli and S. P. Ponzetto. Joining forces pays off: Multilingual Joint Word Sense Disambiguation. In *Proc. of EMNLP*, pages 1399–1410, Jeju, Korea, 2012.

[Navigli, 2005] R. Navigli. Semi-automatic extension of large-scale linguistic knowledge bases. In *Proc. of FLAIRS*, pages 548–553, Clearwater Beach, Florida, USA, 2005.

[Parker *et al.*, 2011] R. Parker, D. Graff, J. Kong, K. Chen, and K. Maeda. English Gigaword Fifth Edition. LDC 2011.

[Pennacchiotti and Pantel, 2006] M. Pennacchiotti and P. Pantel. Ontologizing semantic relations. In *Proc. of COLING-ACL*, pages 793–800, 2006.

[Ponzetto and Strube, 2011] S. P. Ponzetto and M. Strube. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9–10):1737–1756, 2011.

[Poon and Domingos, 2010] H. Poon and P. Domingos. Unsupervised ontology induction from text. In *Proc. of ACL*, pages 296–305, 2010.

[Poon and et al., 2010] H. Poon and et al. Machine Reading at the University of Washington. In *Proc. of NAACL-HLT*, pages 87–95, 2010.

[Schierle and Trabold, 2008] M. Schierle and D. Trabold. Multilingual Knowledge-Based Concept Recognition in Textual Data. In *GfKl*, pages 327–336, 2008.

[Soderland and Mandhani, 2007] S. Soderland and B. Mandhani. Moving from Textual Relations to Ontologized Relations. In *Proc. of AAAI*, pages 85–90, 2007.

[Sun and Grishman, 2012] A. Sun and R. Grishman. Active learning for relation type extension with local and global data views. In *Proc. of CIKM*, pages 1105–1112, 2012.

[Velardi *et al.*, 2013] P. Velardi, S. Faralli, and R. Navigli. OntoLearn Reloaded: A Graph-based Algorithm for Taxonomy Induction. *Computational Linguistics*, 39(3), 2013.

[Warrens, 2010] M. J. Warrens. Inequalities between multi-rater kappas. *Adv. Data Analysis and Classification*, 4(4):271–286, 2010.

[Wu and Weld, 2010] F. Wu and D. S. Weld. Open Information Extraction Using Wikipedia. In *Proc. of ACL*, pages 118–127, 2010.

[Zhang and Rudnicky, 2002] R. Zhang and A. I. Rudnicky. A Large Scale Clustering Scheme for Kernel K-Means. In *Proc. of ICPR*, pages 289–292, 2002.