SPred: Large-scale Harvesting of Semantic Predicates

Tiziano Flati and **Roberto Navigli**
Dipartimento di Informatica
Sapienza Università di Roma
{flati,navigli}@di.uniroma1.it

Abstract

We present SPred, a novel method for the creation of large repositories of semantic predicates. We start from existing collocations to form lexical predicates (e.g., *break* ∗) and learn the semantic classes that best fit the ∗ argument. To do this, we extract all the occurrences in Wikipedia which match the predicate and abstract its arguments to general semantic classes (e.g., *break* BODY PART, *break* AGREEMENT, etc.). Our experiments show that we are able to create a large collection of semantic predicates from the Oxford Advanced Learner's Dictionary with high precision and recall, and perform well against the most similar approach.

1 Introduction

Acquiring semantic knowledge from text automatically is a long-standing issue in Computational Linguistics and Artificial Intelligence. Over the last decade or so the enormous abundance of information and data that has become available has made it possible to extract huge amounts of patterns and named entities (Etzioni et al., 2005), semantic lexicons for categories of interest (Thelen and Riloff, 2002; Igo and Riloff, 2009), large domain glossaries (De Benedictis et al., 2013) and lists of concepts (Katz et al., 2003). Recently, the availability of Wikipedia and other collaborative resources has considerably boosted research on several aspects of knowledge acquisition (Hovy et al., 2013), leading to the creation of several large-scale knowledge resources, such as DBPedia (Bizer et al., 2009), BabelNet (Navigli and Ponzetto, 2012), YAGO (Hoffart et al., 2013), MENTA (de Melo and Weikum, 2010), to name but a few. This wealth of acquired knowledge is known to have a positive impact on important fields such as Information Retrieval (Chu-Carroll and Prager, 2007), Information Extraction (Krause et al., 2012), Question Answering (Ferrucci et al., 2010) and Textual Entailment (Berant et al., 2012; Stern and Dagan, 2012).

Not only are these knowledge resources obtained by acquiring concepts and named entities, but they also provide semantic relations between them. These relations are extracted from unstructured or semi-structured text using ontology learning from scratch (Velardi et al., 2013) and Open Information Extraction techniques (Etzioni et al., 2005; Yates et al., 2007; Wu and Weld, 2010; Fader et al., 2011; Moro and Navigli, 2013) which mainly stem from seminal work on *is-a* relation acquisition (Hearst, 1992) and subsequent developments (Girju et al., 2003; Pasca, 2004; Snow et al., 2004, among others).

However, these knowledge resources still lack semantic information about language units such as phrases and collocations. For instance, which semantic classes are expected as a direct object of the verb *break*? What kinds of noun does the adjective *amazing* collocate with? Recognition of the need for systems that are aware of the selectional restrictions of verbs and, more in general, of textual expressions, dates back to several decades (Wilks, 1975), but today it is more relevant than ever, as is testified by the current interest in semantic class learning (Kozareva et al., 2008) and supertype acquisition (Kozareva and Hovy, 2010). These approaches leverage lexico-syntactic patterns and input seeds to recursively learn the semantic classes of relation arguments. However, they require the manual selection of one or more seeds for each pattern of interest, and this selection influences the amount and kind of semantic classes to be learned. Furthermore, the learned classes are not directly linked to existing resources such as WordNet (Fellbaum, 1998) or Wikipedia.

The goal of our research is to create a large-scale repository of semantic predicates whose lexical arguments are replaced by their semantic classes. For example, given the textual expression *break a toe* we want to create the correspond-

ing semantic predicate *break a* BODY PART, where BODY PART is a class comprising several lexical realizations, such as *leg, arm, foot*, etc.

This paper provides three main contributions:

- We propose SPred, a novel approach which harvests predicates from Wikipedia and generalizes them by leveraging core concepts from WordNet.
- We create a large-scale resource made up of semantic predicates.
- We demonstrate the high quality of our semantic predicates, as well as the generality of our approach, also in comparison with our closest competitor.

2 Preliminaries

We introduce two preliminary definitions which we use in our approach.

Definition 1 (lexical predicate). A lexical predicate $w_1 \ w_2 \ \ldots \ w_i \ * \ w_{i+1} \ \ldots \ w_n$ is a regular expression, where w_j are tokens ($j = 1, \ldots, n$), $*$ matches any sequence of one or more tokens, and $i \in \{0, \ldots, n\}$. We call the token sequence which matches $*$ the filling argument of the predicate.

For example, *a * of milk* matches occurrences such as *a full bottle of milk, a glass of milk, a carton of milk*, etc. While in principle * could match any sequence of words, since we aim at generalizing nouns, in what follows we allow $*$ to match only noun phrases (e.g., *glass, hot cup, very big bottle*, etc.).

Definition 2 (semantic predicate). A semantic predicate is a sequence $w_1 \ w_2 \ \ldots \ w_i \ c \ w_{i+1} \ \ldots \ w_n$, where w_j are tokens ($j = 1, \ldots, n$), $c \in C$ is a semantic class selected from a fixed set C of classes, and $i \in \{0, \ldots, n\}$.

As an example, consider the semantic predicate *cup of* BEVERAGE,[1] where BEVERAGE is a semantic class representing beverages. This predicate matches phrases like *cup of coffee, cup of tea*, etc., but not *cup of sky*. Other examples include: MUSICAL INSTRUMENT *is played by, a* CONTAINER *of milk, break* AGREEMENT, etc.

Semantic predicates mix the lexical information of a given lexical predicate with the explicit semantic modeling of its argument. Importantly, the same lexical predicate can have different classes as its argument, like *cup of* FOOD vs. *cup of* BEVERAGE. Note, however, that different classes might convey different semantics for the same lexical

predicate, such as *cup of* COUNTRY, referring to cup as a prize instead of cup as a container.

3 Large-Scale Harvesting of Semantic Predicates

The goal of this paper is to provide a fully automatic approach for the creation of a large repository of semantic predicates in three phases. For each lexical predicate of interest (e.g., *break* *):

1. We extract all its possible filling arguments from Wikipedia, e.g., *lease, contract, leg, arm*, etc. (Section 3.1).
2. We disambiguate as many filling arguments as possible using Wikipedia, obtaining a set of corresponding Wikipedia pages, e.g., *Lease, Contract*, etc. (Section 3.2).
3. We create the semantic predicates by generalizing the Wikipedia pages to their most suitable semantic classes, e.g., *break* AGREEMENT, *break* LIMB, etc. (Section 3.3).

We can then exploit the learned semantic predicates to assign the most suitable semantic class to new filling arguments for the given lexical predicate (Section 3.4).

3.1 Extraction of Filling Arguments

Let π be an input lexical predicate (e.g., *break* *). We search the English Wikipedia for all the token sequences which match π, resulting in a list of noun phrases filling the $*$ argument. We show an excerpt of the output obtained when searching Wikipedia for the arguments of the lexical predicate *a * of milk* in Table 1. As can be seen, a wide range of noun phrases are extracted, from quantities such as *glass* and *cup* to other aspects, such as *brand* and *constituent*.

The output of this first step is a set L_π of triples (a, s, l) of filling arguments a matching the lexical predicate π in a sentence s of the Wikipedia corpus, with a potentially linked to a page l (e.g., see the top 3 rows in Table 1; $l = \epsilon$ if no link is provided, see bottom rows of the Table).[2] Note that Wikipedia is the only possible corpus that can be used here for at least two reasons: first, in order to extract relevant arguments, we need a large corpus of a definitional nature; second, we need wide-coverage semantic annotations of filling arguments.

3.2 Disambiguation of Filling Arguments

The objective of the second step is to disambiguate as many arguments in L_π as possible for the lex-

---

[2] We will also refer to l as the *sense* of a in sentence s.

a	full [[bottle]]	of milk
a	nice hot [[cup]]	of milk
a	cold [[glass]]	of milk
a	very big bottle	of milk
a	brand	of milk
a	constituent	of milk

Table 1: An excerpt of the token sequences which match the lexical predicate *a * of milk* in Wikipedia (filling argument shown in the second column; following the Wikipedia convention we provide links in double square brackets).

ical predicate π. We denote $D_\pi = \{(a, s, l) : l \neq \epsilon\} \subseteq L_\pi$ as the set of those arguments originally linked to the corresponding Wikipedia page (like the top three linked arguments in Table 1). Therefore, in the rest of this section we will focus only on the remaining triples $(a, s, \epsilon) \in U_\pi$, where $U_\pi = L_\pi \setminus D_\pi$, i.e., those triples whose arguments are not semantically annotated. Our goal is to replace ϵ with an appropriate sense, i.e., page, for a. For each such triple $(a, s, \epsilon) \in U_\pi$, we apply the following disambiguation heuristics:

- **One sense per page:** if another occurrence of a in the same Wikipedia page (independent of the lexical predicate) is linked to a page l, then remove (a, s, ϵ) from U_π and add (a, s, l) to D_π. In other words, we propagate an existing annotation of a in the same Wikipedia page and apply it to our ambiguous item. For instance, *cup of coffee* appears in the Wikipedia page *Energy drink* in the sentence "[...] energy drinks contain more caffeine than a strong *cup of coffee*", but this occurrence of *coffee* is not linked. However the second paragraph contains the sentence "[[Coffee]], tea and other naturally caffeinated beverages are usually not considered energy drinks", where *coffee* is linked to the *Coffee* page. This heuristic naturally reflects the broadly known assumption about lexical ambiguity presented in (Yarowsky, 1995), namely the one-sense-per-discourse heuristic.

- **One sense per lexical predicate:** if $\exists (a, s', l) \in D_\pi$, then remove (a, s, ϵ) from U_π and add (a, s, l) to D_π. If multiple senses of a are available, choose the most frequent one in D_π. For example, in the page *Singaporean cuisine* the occurrence of *coffee* in the sentence "[...] combined with a *cup of coffee* and a half-boiled egg" is not linked, but we have collected many other occurrences, all linked to the *Coffee* page, so this link

gets propagated to our ambiguous item as well. This heuristic mimes the one-sense-per-collocation heuristic presented in (Yarowsky, 1995).

- **Trust the inventory:** if Wikipedia provides only one sense for a, i.e., only one page title whose lemma is a, link a to that page. Consider the instance "At that point, Smith threw down a *cup of Gatorade*" in page *Jimmy Clausen*; there is only one sense for *Gatorade* in Wikipedia, so we link the unannotated occurrence to it.

As a result, the initial set of disambiguated arguments in D_π is augmented with all those triples for which any of the above three heuristics apply. Note that D_π might contain the same argument several times, occurring in different sentences and linked many times to the same page or to different pages. Notably, the discovery of new links is made through one scan of Wikipedia per heuristic. The three disambiguation strategies, applied in the same order as presented above, contribute to promoting the most relevant sense for a given word.

Finally, let A be the set of arguments in D_π, i.e., $A := \{a : \exists (a, s, l) \in D_\pi\}$. For each argument $a \in A$ we select the majority sense $sense(a)$ of a and collect the corresponding set of sentences $sent(a)$ marked with that sense. Formally, $sense(a) := \arg\max_l |\{(x, y, z) \in D_\pi : x = a \wedge z = l\}|$ and $sent(a) := \{s : (a, s, sense(a)) \in D_\pi\}$.

3.3 Generalization to Semantic Classes

Our final objective is to generalize the annotated arguments to semantic classes picked out from a fixed set C of classes. As explained below, we assume the set C to be made up of representative synsets from WordNet. We perform this in two substeps: we first link all our disambiguated arguments to WordNet (Section 3.3.1) and then leverage the WordNet taxonomy to populate the semantic classes in C (Section 3.3.2).

3.3.1 Linking to WordNet

So far the arguments in D_π have been semantically annotated with the Wikipedia pages they refer to. However, using Wikipedia as our sense inventory is not desirable; in fact, contrarily to other commonly used lexical-semantic networks such as WordNet, Wikipedia is not formally organized in a structured, taxonomic hierarchy. While it is true that attached to each Wikipedia page there are one or more categories, these categories just provide shallow information about the class the page

belongs to. Indeed, categories are not ideal for representing the semantic classes of a Wikipedia page for at least three reasons: i) many categories do not express taxonomic information (e.g., the English page *Albert Einstein* provides categories such as DEATHS FROM ABDOMINAL AORTIC ANEURYSM and INSTITUTE FOR ADVANCED STUDY FACULTY); ii) categories are mostly structured in a directed acyclic graph with multiple parents per category (even worse, cycles are possible in principle); iii) there is no clear way of identifying core semantic classes from the large set of available categories. Although efforts towards the taxonomization of Wikipedia categories do exist in the literature (Ponzetto and Strube, 2011; Nastase and Strube, 2013), the results are of a lower quality than a hand-built lexical resource. Therefore, as was done in previous work (Mihalcea and Moldovan, 2001; Ciaramita and Altun, 2006; Izquierdo et al., 2009; Erk and McCarthy, 2009; Huang and Riloff, 2010), we pick out our semantic classes C from WordNet and leverage its manually-curated taxonomy to associate our arguments with the most suitable class. This way we avoid building a new taxonomy and shift the problem to that of projecting the Wikipedia pages – associated with annotated filling arguments – to synsets in WordNet. We address this problem in two steps:

Wikipedia-WordNet mapping. We exploit an existing mapping implemented in BabelNet (Navigli and Ponzetto, 2012), a wide-coverage multilingual semantic network that integrates Wikipedia and WordNet.[3] Based on a disambiguation algorithm, BabelNet establishes a mapping $\mu : Wikipages \rightarrow Synsets$ which links about 50,000 pages to their most suitable WordNet senses.[4]

Mapping extension. Nevertheless, BabelNet is able to solve the problem only partially, because it still leaves the vast majority of the 4 million English Wikipedia pages unmapped. This is mainly due to the encyclopedic nature of most pages, which do not have a counterpart in the WordNet dictionary. To address this issue, for each unmapped Wikipedia page p we obtain its textual definition as the first sentence of the page.[5] Next,

we extract the hypernym from the textual definition of p by applying Word-Class Lattices (Navigli and Velardi, 2010, WCL[6]), a domain-independent hypernym extraction system successfully applied to taxonomy learning from scratch (Velardi et al., 2013) and freely available online (Faralli and Navigli, 2013). If a hypernym h is successfully extracted and h is linked to a Wikipedia page p' for which $\mu(p')$ is defined, then we extend the mapping by setting $\mu(p) := \mu(p')$. For instance, the mapping provided by BabelNet does not provide any link for the page *Peter Spence*; thanks to WCL, though, we are able to set the page *Journalist* as its hypernym, and link it to the WordNet synset *journalist*$_n^1$.

This way our mapping extension now covers 539,954 pages, i.e., more than an order of magnitude greater than the number of pages originally covered by the BabelNet mapping.

3.3.2 Populating the Semantic Classes

We now proceed to populating the semantic classes in C with the annotated arguments obtained for the lexical predicate π.

Definition 3 (semantic class of a synset). The semantic class for a WordNet synset S is the class c among those in C which is the most specific hypernym of S according to the WordNet taxonomy.

For instance, given the synset *tap water*$_n^1$, its semantic class is *water*$_n^1$ (while the other more general subsumers in C are not considered, e.g., *compound*$_n^2$, *chemical*$_n^1$, *liquid*$_n^3$, etc).

For each argument $a \in A$ for which a Wikipedia-to-WordNet mapping $\mu(sense(a))$ could be established as a result of the linking procedure described above, we associate a with the semantic class of $\mu(sense(a))$. For example, consider the case in which a is equal to *tap water* and $sense(a)$ is equal to the Wikipedia page *Tap water*, in turn mapped to *tap water*$_n^1$ via μ; we thus associate *tap water* with its semantic class *water*$_n^1$. If more than one class can be found we add a to each of them.[7]

Ultimately, for each class $c \in C$, we obtain a set *support*(c) made up of all the arguments $a \in A$ associated with c. For instance, *support*(*beverage*$_n^1$) = { *chinese tea*, *3.2% beer*, *hot cocoa*, *cider*, . . . , *orange juice* }. Note that, thanks to our extended mapping (cf. Section 3.3.1), the support of a class can also contain arguments not covered in WordNet (e.g., *hot cocoa* and *tejuino*).

$P_{class}(c\|\pi)$	c	$support(c)$
0.1896	$wine_n^1$	wine, sack, white wine, red wine, wine in china, madeira wine, claret, kosher wine
0.1805	$coffee_n^1$	turkish coffee, drip coffee, espresso, coffee, cappucino, caffè latte, decaffeinated coffee, latte
0.1143	$herb_n^2$	green tea, indian tea, black tea, orange pekoe tea, tea
0.1104	$water_n^1$	water, seawater
0.0532	$beverage_n^1$	chinese tea, 3.2% beer, orange soda, boiled water, hot chocolate, hot cocoa, tejuino, cider, beverage, cocoa, coffee milk, lemonade, orange juice
0.0403	$milk_n^1$	skim milk, milk, cultured buttermilk, whole milk
0.0351	$beer_n^1$	3.2% beer, beer
0.0273	$alcohol_n^1$	mead, umeshu, kava, rice wine, jägermeister, kvass, sake, gin, rum
0.0182	$poison_n^1$	poison

Table 2: Highest-probability semantic classes for the lexical predicate $\pi = $ *cup of* *, according to our set C of semantic classes.

Since not all classes are equally relevant to the lexical predicate π, we estimate the conditional probability of each class $c \in C$ given π on the basis of the number of sentences which contain an argument in that class. Formally:

$$P_{class}(c|\pi) = \frac{\sum_{a \in support(c)} |sent(a)|}{Z}, \quad (1)$$

where Z is a normalization factor calculated as $Z = \sum_{c' \in C} \sum_{a \in support(c')} |sent(a)|$. As an example, in Table 2 we show the highest-probability classes for the lexical predicate *cup of* *.

As a result of the probabilistic association of each semantic class c with a target lexical predicate $w_1\ w_2\ \ldots\ w_i\ *\ w_{i+1}\ \ldots\ w_n$, we obtain a semantic predicate $w_1\ w_2\ \ldots\ w_i\ c\ w_{i+1}\ \ldots\ w_n$.

3.4 Classification of new arguments

Once the semantic predicates for the input lexical predicate π have been learned, we can classify a new filling argument a of π. However, the class probabilities calculated with Formula 1 might not provide reliable scores for several classes, including unseen ones whose probability would be 0.

To enable wide coverage we estimate a second conditional probability based on the distributional semantic profile of each class. To do this, we perform three steps:

1. For each WordNet synset S we create a distributional vector $\vec{S}$ summing the noun occurrences within all the Wikipedia pages p such that $\mu(p) = S$. Next, we create a distributional vector for each class $c \in C$ as follows:

$$\vec{c} = \sum_{S \in desc(c)} \vec{S},$$

where $desc(c)$ is the set of all synsets which are descendants of the semantic class c in WordNet. As a result we obtain a predicate-independent distributional description for each semantic class in C.

2. Now, given an argument a of a lexical predicate π, we create a distributional vector $\vec{a}$ by summing the noun occurrences of all the sentences s such that $(a, s, l) \in L_\pi$ (cf. Section 3.1).

3. Let C_a be the set of candidate semantic classes for argument a, i.e., C_a contains the semantic classes for the WordNet synsets of a as well as the semantic classes associated with $\mu(p)$ for all Wikipedia pages p whose lemma is a. For each candidate class $c \in C_a$, we determine the cosine similarity between the distributional vectors $\vec{c}$ and $\vec{a}$ as follows:

$$sim(\vec{c}, \vec{a}) = \frac{\vec{c} \cdot \vec{a}}{||\vec{c}||\ ||\vec{a}||}.$$

Then, we determine the most suitable semantic class $c \in C_a$ of argument a as the class with the highest distributional probability, estimated as:

$$P_{distr}(c|\pi, a) = \frac{sim(\vec{c}, \vec{a})}{\sum_{c' \in C_a} sim(\vec{c}', \vec{a})}. \quad (2)$$

We can now choose the most suitable class $c \in C_a$ for argument a which maximizes the probability mixture of the distributional probability in Formula 2 and the class probability in Formula 1:

$$P(c|\pi, a) = \alpha P_{distr}(c|\pi, a) + (1 - \alpha) P_{class}(c|\pi), \quad (3)$$

where $\alpha \in [0, 1]$ is an interpolation factor.

We now illustrate the entire process of our algorithm on a real example. Given a textual expression such as *virus replicate*, we: (i) extract all the filling arguments of the lexical predicate * *replicate*; (ii) link and disambiguate the extracted filling arguments; (iii) query our system for the available *virus* semantic classes (i.e., $\{virus_n^1, virus_n^3\}$); (iv) build the distributional vectors for

the candidate semantic classes and the given input argument; (v) calculate the probability mixture. As a result we obtain the following ranking, $virus_n^1$:0.250, $virus_n^3$:0.000894, so that the first sense of *virus* in WordNet 3.0 is preferred, being an "ultramicroscopic infectious agent that replicates itself only within cells of living hosts".

4 Experiment 1: Oxford Lexical Predicates

We evaluate on the two forms of output produced by SPred: (i) the top-ranking semantic classes of a lexical predicate, as obtained with Formula 1, and (ii) the classification of a lexical predicate's argument with the most suitable semantic class, as produced using Formula 3. For both evaluations, we use a lexical predicate dataset built from the Oxford Advanced Learner's Dictionary (Crowther, 1998).

4.1 Set of Semantic Classes

The selection of which semantic classes to include in the set C is of great importance. In fact, having too many classes will end up in an overly fine-grained inventory of meanings, whereas an excessively small number of classes will provide little discriminatory power. As our set C of semantic classes we selected the standard set of 3,299 core nominal synsets available in WordNet.[8] However, our approach is flexible and can be used with classes of an arbitrary level of granularity.

4.2 Datasets

The Oxford Advanced Learner's Dictionary provides usage notes that contain typical predicates in various semantic domains in English, e.g., Traveling.[9] Each predicate is made up of a fixed part (e.g., a verb) and a generalizable part which contains one or more nouns.

Examples include *fix an election/the vote*, *bacteria/microbes/viruses spread*, *spend money/savings/a fortune*. In the case that more than one noun was provided, we split the textual expression into as many items as the number of nouns. For instance, from *spend money/savings/a fortune* we created three items in our dataset, i.e., *spend money*, *spend savings*, *spend a fortune*. The splitting procedure generated 6,220 instantiated lexical predicate items overall.

k	Prec@k	Correct	Total
1	0.94	46	49
2	0.87	85	98
3	0.86	124	145
4	0.83	160	192
5	0.82	194	237
6	0.81	228	282
7	0.80	261	326
8	0.78	288	370
9	0.77	318	414
10	0.76	349	458
11	0.75	379	502
12	0.75	411	546
13	0.75	445	590
14	0.76	479	634
15	0.75	510	678
16	0.75	544	721
17	0.76	577	763
18	0.76	612	806
19	0.76	643	849
20	0.75	671	892

Table 3: Precision@k for ranking the semantic classes of lexical predicates.

4.3 Evaluating the Semantic Class Ranking

Dataset. Given the above dataset, we generalized each item by pairing its fixed verb part with * (i.e., we keep "verb predicates" only, since they are more informative). For instance, the three items *bacteria/microbes/viruses spread* were generalized into the lexical predicate * *spread*. The total number of different lexical predicates obtained was 1,446, totaling 1,429 distinct verbs (note that the dataset might contain the lexical predicate * *spread* as well as *spread* *).[10]

Methodology. For each lexical predicate we calculated the conditional probability of each semantic class using Formula 1, resulting in a ranking of semantic classes. To evaluate the top ranking classes, we calculated precision@k, with k ranging from 1 to 20, by counting all applicable classes as correct, e.g., $location_n^1$ is a valid semantic class for *travel to* * while $emotion_n^1$ is not.

Results. We show in Table 3 the precision@k calculated over a random sample of 50 lexical predicates.[11] As can be seen, while the classes quality is pretty high with low values of k, performance gradually degrades as we let k increase. This is mostly due to the highly polysemous nature of the predicates selected (e.g., *occupy* *, *leave* *, *help* *, *attain* *, *live* *, etc.). We note that high performance, attaining above 80%, can be achieved

---

by focusing up to the first 7 classes output by our system, with a 94% precision@1.

4.4 Evaluating Classification Performance

Dataset. Starting from the lexical predicate items obtained as described in Section 4.2, we selected those items belonging to a random sample of 20 usage notes among those provided by the Oxford dictionary, totaling 3,245 items. We then manually tagged each item's argument (e.g., *virus* in *viruses spread*) with the most suitable semantic class (e.g., $virus_n^1$), obtaining a gold standard dataset for the evaluation of our argument classification algorithm (cf. Section 3.4).

Methodology. In this second evaluation we measure the accuracy of our method at assigning the most suitable semantic class to the argument of a lexical predicate item in our gold standard. We use three customary measures to determine the quality of the acquired semantic classes, i.e., precision, recall and F1. Precision is the number of items which are assigned the correct class (as evaluated by a human) over the number of items which are assigned a class by the system. Recall is the number of items which are assigned the correct class over the number of items to be classified. F1 is the harmonic mean of precision and recall.

Tuning. The only parameter to be tuned is the factor α that we use to mix the two probabilities in Formula 3 (cf. Section 3.4). For tuning α we used a held-out set of 8 verbs, randomly sampled from the lexical predicates not used in the dataset. We created a tuning set using the annotated arguments in Wikipedia for these verbs: we trained the model on 80% of the annotated lexical predicate arguments (i.e., the class probability estimates in Formula 1) and then applied the probability mixture (i.e., Formula 3) for classifying the remaining 20% of arguments. Finally, we calculated the performance in terms of precision, recall and F1 with 11 different values of $\alpha \in \{0, 0.1, \ldots, 1.0\}$, achieving optimal performance with $\alpha = 0.2$.

Results. Table 4 shows the results on the semantic class assignments. Our system shows very high precision, above 85%, while at the same time attaining an adequate 68% recall. We also compared against a random baseline that randomly selects one out of all the candidate semantic classes for each item, achieving only moderate results. A subsequent error analysis revealed the common types of error produced by our system: terms for which we could not provide (1) any WordNet concept

Method	Precision	Recall	F1
SPred	85.61	68.01	75.80
Random	40.96	40.96	40.96

Table 4: Performance on semantic class assignment.

(e.g., *political corruption*) or (2) any candidate semantic class (e.g., *immune system*).

4.5 Disambiguation heuristics impact

As a follow-up analysis, for each dataset we considered the impact of each disambiguation heuristic described in Section 3.2 according to how many times it was triggered. Starting from the entire set of 1,446 lexical predicates from the Oxford dictionary (see Section 4.3), we counted the number of argument triples (a, s, l) already disambiguated in Wikipedia (i.e., $l \neq \epsilon$) and those disambiguated thanks to our disambiguation strategies. Table 5 shows the statistics. We note that, while the amount of originally linked arguments is very low (about 2.5% of total), our strategies are able to considerably increase the size of the initial set of linked instances. The most effective strategies appear to be the *One sense per page* and the *Trust the inventory*, which contribute 26.16% and 31.33% of the total links, respectively.

Even though most of the triples (i.e., 68 out of almost 74 million) remain unlinked, the ratio of distinct arguments which we linked to WordNet is considerably higher, calculated as 3,723,979 linked arguments over 12,431,564 distinct arguments, i.e., about 30%.

5 Experiment 2: Comparison with Kozareva & Hovy (2010)

Due to the novelty of the task carried out by SPred, the resulting output may be compared with only a limited number of existing approaches. The most similar approach is that of Kozareva and Hovy (2010, K&H) who assign supertypes to the arguments of arbitrary relations, a task which resembles our semantic predicate ranking. We therefore performed a comparison on the quality of the most highly-ranked supertypes (i.e., semantic classes) using their dataset of 24 relation patterns (i.e., lexical predicates).

Dataset. The dataset contained 14 lexical predicates (e.g., *work for* * or * *fly to*), 10 of which were expanded in order to semantify their left- and right-side arguments (e.g., * *work for* and *work for* *); for the remaining 4 predicates just a single

Total triples	Linked in Wikipedia	One sense per page	One sense per lexical predicate	Trust the inventory	Not linked
73,843,415	1,795,608	1,433,634	533,946	1,716,813	68,363,414

Table 5: Statistics on argument triple linking for all the lexical predicates in the Oxford dataset.

k	Prec@k	Correct	Total
1	0.88	21	24
2	0.90	43	48
3	0.88	63	72
4	0.89	85	96
5	0.91	109	120
6	0.91	131	144
7	0.92	154	168
8	0.91	175	192
9	0.92	198	216
10	0.92	221	240
11	0.92	242	264
12	0.92	264	288
13	0.91	284	312
14	0.90	304	336
15	0.91	327	360
16	0.91	348	384
17	0.90	367	408
18	0.89	386	432
19	0.89	407	456
20	0.89	429	480

Table 6: Precision@k for the semantic classes of the relations of Kozareva and Hovy (2010).

Predicate	Number of args	Number of classes
cause *	181,401	1,339
live in *	143,628	600
go to *	134,712	867
* cause	92,160	1,244
work in *	79,444	770
* go to	71,794	746
* live in	61,074	541
work on *	58,760	840
work for *	58,332	681
work at *	31,904	511
* work in	24,933	528
* celebrate	23,333	408

Table 7: Number of arguments and associated classes for the 12 most frequent lexical predicates of Kozareva and Hovy (2010) extracted by SPred from Wikipedia.

side was generalized (e.g., *dress*). While most of the relations apply to persons as a supertype, our method could find arguments for each of them.

Methodology. We carried out the same evaluation as in Section 4.3. We calculated precision@k of the semantic classes obtained for each relation in the dataset of K&H. Because the set of applicable classes was potentially unbounded, we were not able to report recall directly.

Results. K&H reported an overall accuracy of the top-20 supertypes of 92%. As can be seen in Table 6 we exhibit very good performance with increasing values of k. A comparison of Table 3 with Table 6 shows considerable differences in performance between the two datasets. We attribute this difference to the higher average WordNet polysemy of the verbal component of the Oxford predicates (on average 2.64 senses for K&H against 6.52 for the Oxford dataset).

Although we cannot report recall, we list the number of Wikipedia arguments and associated classes in Table 7, which provides an estimate of the extraction capability of SPred. The large number of classes found for the arguments demonstrates the ability of our method to generalize to a variety of semantic classes.

6 Related work

The availability of Web-scale corpora has led to the production of large resources of relations (Etzioni et al., 2005; Yates et al., 2007; Wu and Weld, 2010; Carlson et al., 2010; Fader et al., 2011). However, these resources often operate purely at the lexical level, providing no information on the semantics of their arguments or relations. Several studies have examined adding semantics through grouping relations into sets (Yates and Etzioni, 2009), ontologizing the arguments (Chklovski and Pantel, 2004), or ontologizing the relations themselves (Moro and Navigli, 2013). However, analysis has largely been either limited to ontologizing a small number of relation types with a fixed inventory, which potentially limits coverage, or has used implicit definitions of semantic categories (e.g., clusters of arguments), which limits interpretability. For example, Mohamed et al. (2011) use the semantic categories of the NELL system (Carlson et al., 2010) to learn roughly 400 valid ontologized relations from over 200M web pages, whereas WiSeNet (Moro and Navigli, 2012) leverages Wikipedia to acquire relation synsets for an open set of relations. Despite these efforts, no large-scale resource has existed to date that contains ontologized lexical predicates. In contrast, the present work provides a high-coverage method for learning argument supertypes from a broad-coverage ontology (WordNet), which can potentially be leveraged in relation extraction to ontolo-

gize relation arguments.

Our method for identifying the different semantic classes of predicate arguments is closely related to the task of identifying selectional preferences. The most similar approaches to it are taxonomy-based ones, which leverage the semantic types of the relations arguments (Resnik, 1996; Li and Abe, 1998; Clark and Weir, 2002; Pennacchiotti and Pantel, 2006). Nevertheless, despite their high quality sense-tagged data, these methods have often suffered from lack of coverage. As a result, alternative approaches have been proposed that eschew taxonomies in favor of rating the quality of potential relation arguments (Erk, 2007; Chambers and Jurafsky, 2010) or generating probability distributions over the arguments (Rooth et al., 1999; Pantel et al., 2007; Bergsma et al., 2008; Ritter et al., 2010; Séaghdha, 2010; Bouma, 2010; Jang and Mostow, 2012) in order to obtain higher coverage of preferences.

In contrast, we overcome the data sparsity of class-based models by leveraging the large quantity of collaboratively-annotated Wikipedia text in order to connect predicate arguments with their semantic class in WordNet using BabelNet (Navigli and Ponzetto, 2012); because we map directly to WordNet synsets, we provide a more readily-interpretable collocation preference model than most similarity-based or probabilistic models.

Verb frame extraction (Green et al., 2004) and predicate-argument structure analysis (Surdeanu et al., 2003; Yakushiji et al., 2006) are two areas that are also related to our work. But their generality goes beyond our intentions, as we focus on semantic predicates, which is much simpler and free from syntactic parsing.

Another closely related work is that of Hanks (2013) concerning the Theory of Norms and Exploitations, where norms (exploitations) represent expected (unexpected) classes for a given lexical predicate. Although our semantified predicates do, indeed, provide explicit evidence of norms obtained from collective intelligence and would provide support for this theory, exploitations present a more difficult task, different from the one addressed here, due to its focus on identifying property transfer between the semantic class and the exploited instance.

The closest technical approach to ours is that of Kozareva and Hovy (2010), who use recursive patterns to induce semantic classes for the arguments of relational patterns. Whereas their approach requires both a relation pattern and one or more seeds, which bias the types of semantic classes that are learned, our proposed method re-quires only the pattern itself, and as a result is capable of learning an unbounded number of different semantic classes.

7 Conclusions

In this paper we present SPred, a novel approach to large-scale harvesting of semantic predicates. In order to semantify lexical predicates we exploit the wide coverage of Wikipedia to extract and disambiguate lexical predicate occurrences, and leverage WordNet to populate the semantic classes with suitable predicate arguments. As a result, we are able to ontologize lexical predicate instances like those available in existing dictionaries (e.g., *break a toe*) into semantic predicates (such as *break* a BODY PART).

For each lexical predicate (such as *break* $*$), our method produces a probability distribution over the set of semantic classes (thus covering the different expected meanings for the filling arguments) and is able to classify new instances with the most suitable class. Our experiments show generally high performance, also in comparison with previous work on argument supertyping.

We hope that our semantic predicates will enable progress in different Natural Language Processing tasks such as Word Sense Disambiguation (Navigli, 2009), Semantic Role Labeling (Fürstenau and Lapata, 2012) or even Textual Entailment (Stern and Dagan, 2012) – each of which is in urgent need of reliable semantics. While we focused on semantifying lexical predicates, as future work we will apply our method to the ontologization of large amounts of sequences of words, such as phrases or textual relations (e.g., considering Google n-grams appearing in Wikipedia). Notably, our method should, in principle, generalize to any semantically-annotated corpus (e.g., Wikipedias in other languages), provided lexical predicates can be extracted with associated semantic classes.

In order to support future efforts we are releasing our semantic predicates as a freely available resource.[12]

---

[12]http://lcl.uniroma1.it/spred

References

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2012. Learning entailment relations by global graph structure optimization. *Computational Linguistics*, 38(1):73–111.

Shane Bergsma, Dekang Lin, and Randy Goebel. 2008. Discriminative learning of selectional preference from unlabeled text. In *Proc. of EMNLP*, pages 59–68, Stroudsburg, PA, USA.

Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. 2009. DBpedia - a crystallization point for the Web of Data. *Web Semantics*, 7(3):154–165.

Gerlof Bouma. 2010. Collocation Extraction beyond the Independence Assumption. In *Proc. of ACL, Short Papers*, pages 109–114, Uppsala, Sweden.

Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and Tom M. Mitchell. 2010. Toward an architecture for never-ending language learning. In *Proc. of AAAI*, pages 1306–1313, Atlanta, Georgia.

Nathanael Chambers and Dan Jurafsky. 2010. Improving the use of pseudo-words for evaluating selectional preferences. In *Proc. of ACL*, pages 445–453, Stroudsburg, PA, USA.

Tim Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for fine-grained semantic verb relations. In *Proc. of EMNLP*, pages 33–40, Barcelona, Spain.

Jennifer Chu-Carroll and John Prager. 2007. An experimental study of the impact of information extraction accuracy on semantic search performance. In *Proc. of CIKM*, pages 505–514, Lisbon, Portugal.

Massimiliano Ciaramita and Yasemin Altun. 2006. Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger. In *Proc. of EMNLP*, pages 594–602, Sydney, Australia.

Stephen Clark and David Weir. 2002. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

Jonathan Crowther, editor. 1998. *Oxford Advanced Learner's Dictionary*. Cornelsen & Oxford, 5th edition.

Flavio De Benedictis, Stefano Faralli, and Roberto Navigli. 2013. GlossBoot: Bootstrapping multilingual domain glossaries from the Web. In *Proc. of ACL*, Sofia, Bulgaria.

Gerard de Melo and Gerhard Weikum. 2010. MENTA: Inducing Multilingual Taxonomies from Wikipedia. In *Proc. of CIKM*, pages 1099–1108, New York, NY, USA.

Katrin Erk and Diana McCarthy. 2009. Graded word sense assignment. In *Proc. of EMNLP*, pages 440–449, Stroudsburg, PA, USA.

Katrin Erk. 2007. A Simple, Similarity-based Model for Selectional Preferences. In *Proc. of ACL*, pages 216–223, Prague, Czech Republic.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: an experimental study. *Artificial Intelligence*, 165(1):91–134.

Anthony Fader, Stephen Soderland, and Oren Etzioni. 2011. Identifying Relations for Open Information Extraction. In *Proc. of EMNLP*, pages 1535–1545, Edinburgh, UK.

Stefano Faralli and Roberto Navigli. 2013. A Java framework for multilingual definition and hypernym extraction. In *Proc. of ACL, Comp. Volume*, Sofia, Bulgaria.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Database*. MIT Press, Cambridge, MA.

David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. 2010. Building Watson: an overview of the DeepQA project. *AI Magazine*, 31(3):59–79.

Hagen Fürstenau and Mirella Lapata. 2012. Semi-supervised semantic role labeling via structural alignment. *Computational Linguistics*, 38(1):135–171.

Roxana Girju, Adriana Badulescu, and Dan Moldovan. 2003. Learning semantic constraints for the automatic discovery of part-whole relations. In *Proc. of HLT-NAACL*, pages 1–8, Edmonton, Canada.

Rebecca Green, Bonnie J. Dorr, and Philip Resnik. 2004. Inducing Frame Semantic Verb Classes from WordNet and LDOCE. In *Proc. of ACL*, pages 375–382, Barcelona, Spain.

Patrick Hanks. 2013. *Lexical Analysis: Norms and Exploitations*. University Press Group Limited.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proc. of COLING*, pages 539–545, Nantes, France.

Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. 2013. Yago2: A spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*, 194:28–61.

Eduard H. Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27.

Ruihong Huang and Ellen Riloff. 2010. Inducing Domain-Specific Semantic Class Taggers from (Almost) Nothing. In *Proc. of ACL*, pages 275–285, Uppsala, Sweden.

Sean P. Igo and Ellen Riloff. 2009. Corpus-based semantic lexicon induction with Web-based corroboration. In *Proc. of UMSLLS*, pages 18–26, Stroudsburg, PA, USA.

Rubén Izquierdo, Armando Suárez, and German Rigau. 2009. An Empirical Study on Class-Based Word Sense Disambiguation. In *Proc. of EACL*, pages 389–397, Athens, Greece.

Hyeju Jang and Jack Mostow. 2012. Inferring selectional preferences from part-of-speech n-grams. In *Proc. of EACL*, pages 377–386, Stroudsburg, PA, USA.

Boris Katz, Jimmy J. Lin, Daniel Loreto, Wesley Hildebrandt, Matthew W. Bilotti, Sue Felshin, Aaron Fernandes, Gregory Marton, and Federico Mora. 2003. Integrating Web-based and Corpus-based Techniques for Question Answering. In *Proc. of TREC*, pages 426–435, Gaithersburg, Maryland.

Zornitsa Kozareva and Eduard Hovy. 2010. Learning Arguments and Supertypes of Semantic Relations Using Recursive Patterns. In *Proc. of ACL*, pages 1482–1491, Uppsala, Sweden.

Zornitsa Kozareva, Ellen Riloff, and Eduard H. Hovy. 2008. Semantic Class Learning from the Web with Hyponym Pattern Linkage Graphs. In *Proc. ACL/HLT*, pages 1048–1056, Columbus, Ohio.

Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. 2012. Large-scale learning of relation-extraction rules with distant supervision from the web. In *Proc. of ISWC 2012, Part I*, pages 263–278, Boston, MA.

Hang Li and Naoki Abe. 1998. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244.

Rada Mihalcea and Dan Moldovan. 2001. eXtended WordNet: Progress report. In *Proceedings of the NAACL-01 Workshop on WordNet and Other Lexical Resources*, pages 95–100, Pittsburgh, Penn.

Thahir Mohamed, Estevam Hruschka, and Tom Mitchell. 2011. Discovering Relations between Noun Categories. In *Proc. of EMNLP*, pages 1447–1455, Edinburgh, Scotland, UK.

Andrea Moro and Roberto Navigli. 2012. WiSeNet: Building a Wikipedia-based semantic network with ontologized relations. In *Proc. of CIKM*, pages 1672–1676, Maui, HI, USA.

Andrea Moro and Roberto Navigli. 2013. Integrating Syntactic and Semantic Analysis into the Open Information Extraction Paradigm. In *Proc. of IJCAI*, Beijing, China.

Vivi Nastase and Michael Strube. 2013. Transforming wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, 194:62–85.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for Definition and Hypernym Extraction. In *Proc. of ACL*, pages 1318–1327, Uppsala, Sweden.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM Computing Surveys*, 41(2):1–69.

Patrick Pantel, Rahul Bhagat, Timothy Chklovski, and Eduard Hovy. 2007. ISP: learning inferential selectional preferences. In *Proc. of NAACL*, pages 564–571, Rochester, NY.

Marius Pasca. 2004. Acquisition of categorized named entities for web search. In *Proc. of CIKM*, pages 137–145, New York, NY, USA.

Marco Pennacchiotti and Patrick Pantel. 2006. Ontologizing semantic relations. In *Proc. of COLING*, pages 793–800, Sydney, Australia.

Simone Paolo Ponzetto and Michael Strube. 2011. Taxonomy induction based on a collaboratively built knowledge repository. *Artificial Intelligence*, 175(9-10):1737–1756.

Philip Resnik. 1996. Selectional constraints: An information-theoretic model and its computational realization. *Cognition*, 61(1):127–159.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proc. of ACL*, pages 424–434, Uppsala, Sweden. ACL.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via EM-based clustering. In *Proc. of ACL*, pages 104–111, Stroudsburg, PA, USA.

Diarmuid O Séaghdha. 2010. Latent variable models of selectional preference. In *Proc. of ACL*, pages 435–444, Uppsala, Sweden.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2004. Learning Syntactic Patterns for Automatic Hypernym Discovery. In *NIPS*, pages 1297–1304, Cambridge, Mass.

Asher Stern and Ido Dagan. 2012. Biutee: A modular open-source system for recognizing textual entailment. In *Proc. of ACL 2012, System Demonstrations*, pages 73–78, Jeju Island, Korea.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proc. ACL*, pages 8–15, Stroudsburg, PA, USA.

M. Thelen and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. In *Proc. of EMNLP*, pages 214–221, Salt Lake City, UT, USA.

Paola Velardi, Stefano Faralli, and Roberto Navigli. 2013. OntoLearn Reloaded: A graph-based algorithm for taxonomy induction. *Computational Linguistics*, 39(3).

Yorick Wilks. 1975. A preferential, pattern-seeking, semantics for natural language inference. *Artificial Intelligence*, 6(1):53–74.

Fei Wu and Daniel S. Weld. 2010. Open Information Extraction Using Wikipedia. In *Proc. of ACL*, pages 118–127, Uppsala, Sweden.

Akane Yakushiji, Yusuke Miyao, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2006. Automatic construction of predicate-argument structure patterns for biomedical information extraction. In *Proc. of EMNLP*, pages 284–292, Stroudsburg, PA, USA.

David Yarowsky. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. In *Proc. of ACL*, pages 189–196, Cambridge, MA, USA.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34(1):255.

Alexander Yates, Michael Cafarella, Michele Banko, Oren Etzioni, Matthew Broadhead, and Stephen Soderland. 2007. TextRunner: open information extraction on the web. In *Proc. of NAACL-Demonstrations*, pages 25–26, Stroudsburg, PA, USA.