J Intell Inf Syst (2014) 43:231–246
DOI 10.1007/s10844-014-0320-9

# Learning from syntax generalizations for automatic semantic annotation

**Guido Boella · Luigi Di Caro · Alice Ruggeri · Livio Robaldo**

Received: 1 October 2013 / Revised: 26 March 2014 / Accepted: 30 March 2014 /
Published online: 27 May 2014


**Abstract** Nowadays, there is a huge amount of textual data coming from on-line social communities like Twitter or encyclopedic data provided by Wikipedia and similar platforms. This *Big Data Era* created novel challenges to be faced in order to make sense of large data storages as well as to efficiently find specific information within them. In a more domain-specific scenario like the management of legal documents, the extraction of semantic knowledge can support domain engineers to find relevant information in more rapid ways, and to provide assistance within the process of constructing application-based legal ontologies. In this work, we face the problem of automatically extracting structured knowledge to improve semantic search and ontology creation on textual databases. To achieve this goal, we propose an approach that first relies on well-known Natural Language Processing techniques like Part-Of-Speech tagging and Syntactic Parsing. Then, we transform these information into generalized features that aim at capturing the surrounding linguistic variability of the target semantic units. These new featured data are finally fed into a Support Vector Machine classifier that computes a model to automate the semantic annotation. We first tested our technique on the problem of automatically extracting semantic entities and involved objects within legal texts. Then, we focus on the identification of hypernym relations and definitional sentences, demonstrating the validity of the approach on different tasks and domains.

**Keywords** Ontology learning · Automatic annotation · Information extraction

## 1 Introduction

These days, the problem of managing and accessing textual data is more important than ever. The Web 2.0 induced people to create their own contents in on-line social micro-blogging

The work has been funded by the project ITxLaw with Compagnia di San Paolo.

G. Boella · L. Di Caro (✉) · A. Ruggeri · L. Robaldo
University of Turin, Torino, Italy
e-mail: dicaro@di.unito.it

communities like Twitter, Blogger, MySpace, Wordpress and several others. Twitter, for instance, has over 550 million registered users, generating about 60 million tweets daily and handling over 2 billion search queries per day.[1] Users can post tweets of maximum 140 characthers regarding their activities, moods, opinions, and so forth.[2]

In a completely different scenario, the social-community lever has put the basis for projects like Wikipedia,[3] to achieve a free encyclopedic information storage with around 4 millions English concepts, people, organizations, locations, and so on. YAGO (Hoffart et al. 2012) is a huge semantic knowledge base derived from Wikipedia and other resources like WordNet (Miller 1995), containing more than 10 million entities like persons and organizations, and with more than 120 million facts about such entities. BabelNet (Navigli and Ponzetto 2010) represents a significant effort to combine WordNet information with Wikipedia.

From another perspective, in the legal domain, million of multilingual documents of public administrations are now publicly available. They represent an important basis for specific applications like the semi-supervised construction of legal ontologies as well as smart searches within legislation.

Even if such data sources represent different domains with possibly different specific applications, the need of extracting semantic-aware knowledge bases is satisfied by converging technologies that face similar tasks: Information Extraction, Sentiment Analysis, Question Answering, Text Classification, Clustering, and Semantic Search are the most representative ones. Then, it is often important to have more structured data in the form of ontologies, in order to allow semantics-based retrieval and reasoning. Ontology Learning is a task that permits to automatically (or semi-automatically) extract structured knowledge from text. Manual construction of ontologies usually requires strong efforts from domain experts. Thus, some automatization strategies are needed.

In this paper, we present a novel technique for the identification of semantic units that can be used to extract structured knowledge as well as to efficiently compute semantic searches in texts belonging to different domains. Most of the existing work in this field uses automatic or semi-automatic generation of sequential patterns that induce semantic information. Although this approach can achieve good results, it is limited in the sense that it exclusively relies on the sequentiality of the expressions. Natural language offers potentially infinite ways for expressing concepts, without necessary imposing any limit on the length and complexity of the sentences. Our assumption is that syntax is less dependent than learned patterns on the length and the complexity of textual expressions. In some way, patterns grasp syntactic relationships, but without any linguistic knowledge. We thus investigated the plausibility of using two best performing methods for two separated tasks. On the one hand, the classification phase makes use of a Support Vector Machine classifier that automatically decides the features and the way they help for the discrimination of the training instances. This means that the classifier is used as a discoverer of semantic units that are concealed under syntactic surfaces. On the other hand, we fully exploit all the linguistic knowledge contained in a syntactic parser to create well-formed syntax-based features to be used by the forementioned classifier. It is important to note that such syntactic features do not necessarily reflect a complete and precise parse tree. Thus, our technique is not strictly

[1] http://www.statisticbrain.com/twitter-statistics/

[2] http://www.telegraph.co.uk/technology/twitter/9945505/Twitter-in-numbers.html

[3] http://www.wikipedia.org/

subjected to the errors given by the parser. Finally, we propose a method to generalize the features using the Part-of-Speech tags, with the goal of creating a feature space that is able to understand language variability as well as meaningful syntactic clusters.

For the evaluation, we apply our approach on the legal domain with the task of extracting semantic entities like roles and involved objects within legal prescriptions. Then, we focus on encyclopedic data to identify hyponyms and hypernyms using the same approach, showing how we can identify textual definitions and automatically construct ontologies.

Our overall vision is to make texts more meaningful and clear, and we need to use intelligent technologies as much as possible, from NLP to semantic search, as explained in Boella et al. (2012).

This work is an extended version of Boella and Di Caro (2013) and Boella et al. (2013), presenting a generalization of the approach with experiments on different domains.

## 2 Related work

In this section we present an overview of the existing techniques concerning the extraction of semantic knowledge from texts. More in detail, we take into consideration the state of the art related to the extraction of semantic entities in the legal domain and the identification of hypernyms, hyponyms, and textual definitions within encyclopedic data.

### 2.1 Ontology learning in the legal domain

To the best of our knowledge, there is still a small literature concerning ontology learning and semantic search on the legal domain, while most of the efforts has been dedicated to standard classification tasks. de Maat et al. (2010), for instance, used a set of rules to find patterns suggestive of a particular semantic class. However, their classification task was quite different from ours since their classes were types of norms like delegations and penalizations, while we categorize pieces of text as related to specific semantic labels. Biagioli et al. (2005) achieved an accuracy of 92 % in the task of classifying 582 paragraphs from Italian laws into ten different semantic categories such as 'Prohibition Action', 'Obligation Addressee', 'Substitution', and so on. Lesmo et al. (2013) proposed a method to detect modificatory provisions, i.e., fragments of text that make a change to one or more sentences in the text or in the normative arguments. Our aim is instead to use classification techniques for finding and extracting information that can allow semantic search and smart navigation of the data.

### 2.2 Hypernyms and hyponyms extraction

According to Biemann (2005) and Buitelaar et al. (2005), the problem of extracting ontologies from text can be faced at different levels of granuarity. According to the former, our approach belongs to the extraction of *terminological ontologies* based on IS-A relations, while for the latter we refer to the *concept hierarchies* of their *Ontology Learning layer cake*.

As for the task of definition extraction, most of the existing approaches use symbolic methods that are based on lexico-syntactic patterns, which are manually crafted or deduced automatically. The seminal work of Hearst (1992) represents the main approach based on fixed patterns like "$NP_x$ *is a/an* $NP_y$" and "$NP_x$ *such as* $NP_y$", that usually imply $< x \text{IS-A } y >$.The main drawback of such technique is that it does not face the high vari-

ability of how a relation can be expressed in natural language. Still, it generally extracts single-word terms rather than well-formed and compound concepts. The work of Navigli and Velardi (2010) and Velardi et al. (2012) is based on graph structures that generalize over the POS-tagged patterns between $x$ and $y$. Berland and Charniak (1999) proposed similar lexico-syntactic patterns to extract *part-whole* relationships.

Del Gaudio and Branco (2007) proposed a rule-based approach to the extraction of hypernyms that, however, leads to very low accuracy values in terms of Precision.

Ponzetto and Strube (2007) proposed a technique to extract hypernym relations from Wikipedia by means of methods based on the connectivity of the network and classical lexico-syntactic patterns. Yamada et al. (2009) extended their work by combining extracted Wikipedia entries with new terms contained in additional web documents, using a distributional similarity-based approach.

Moschitti and Bejan (2004) proposed a technique that uses parse subtrees kernels to classify predicate-argument attachments, demonstrating the efficacy of using syntactic information rather than patterns. However, our method represents a computationally lighter approach since the feature space remains limited and manageable with ease.

Finally, pure statistical approaches present techniques for the extraction of hierarchies of terms based on words frequency as well as co-occurrence values, relying on clustering procedures Candan et al. (2008), Fortuna et al. (2006), and Yang and Callan (2008). The central hypothesis is that similar words tend to occur together in similar contexts (Harris 1954). Despite this, they are defined by Biemann (2005) as *prototype-based ontologies* rather than formal terminological ontologies, and they usually suffer from the problem of data sparsity in case of small corpora.

### 2.3 Identification of textual definitions

Considering the initial formal representation proposed by Storrer and Wellinghoff (2006), a *definitional sentence* is composed by different information fields:

- a *definiendum* (DF), i.e., the word being defined with its modifiers,
- a *definitor* (VF), i.e., the verb phrase to introduce the definition,
- a *definiens* (GF), i.e., the genus phrase that usually contains the hypernym,
- and the *rest* of the sentence (REST), that can contain additional clauses.

An example of annotated definition is represented by the following sentence:

> [In computer science, a $[pixel]_{DF}$ $[is]_{VF}$ [a **dot**]$_{GF}$ [that is part of a computer image]$_{REST}$.

In this paper, we will use the term *definitional sentence* referring to the more general meaning given by Navigli and Velardi (2010): *A sentence that provides a formal explanation for the term of interest*, and more specifically as a sentence containing at least one hypernym relation.

So far, most of the proposed techniques rely on lexico-syntactic patterns, either manually or semi-automatically produced (Hovy et al. 2003; Zhang and Jiang 2009; Westerhout 2009). Such patterns are sequences of words like "*is a*" or "*refers to*", rather than more complex sequences including part-of-speech tags.

In the work of Westerhout (2009), after a manual identification of types of definitions and related patterns contained in a corpus, the author successively applied Machine Learning techniques on syntactic and location features to improve the results.

A fully-automatic approach has been proposed by Borg et al. (2009), where the authors applied genetic algorithms to the extraction of English definitions containing the keyword "*is*". In detail, they assign weights to a set of features for the classification of definitional sentences, reaching a precision of 62 % and a recall of 52 %.

Then, Cui et al. (2007) proposed an approach based on *soft patterns*, i.e., probabilistic lexico-semantic patterns that are able to generalize over rigid patterns enabling partial matching by calculating a generative degree-of-match probability between a test instance and the set of training instances.

Fahmi and Bouma (2006) used three different Machine Learning algorithms to distinguish actual definitions from other sentences, relying on syntactic features and reaching high accuracy levels.

The work of Klavans and Muresan (2001) relies on a rule-based system that makes use of "cue phrases" and structural indicators that frequently introduce definitions, reaching 87 % of precision and 75 % of recall on a small and domain-specific corpus.

Finally, Navigli and Velardi (2010) proposed a system based on Word-Class Lattices (WCL), i.e., graph structures that try to generalize over the POS-tagged definition patterns found in the training set. Nevertheless, these mechanisms are not properly able to handle linguistic exceptions and linguistic ambiguity.

## 3 Approach

In this section we present our approach to learn the linguistic variability of specific semantic information contained in text corpora in order to build automatic annotation systems to support the users in the construction of ontologies rather than in semantic search scenarios.

Our methodology consists in seeing the problem in the following way: given a set of semantic annotations $rel(x, L)$ between a piece of text $x$ and a semantic label $L$, the task is to build a set of features that aim at representing the syntactic context of $x$ such that a classifier would able to autonomously associate it with the label $L$. The only assumption is that all the words that are associated with some semantic label must be common nouns (or syntactic chunks involving a main common noun). Then, given a sentence $S$, all common nouns are extracted by means of a Part-Of-Speech tagger and considered as possible candidates. In the next sections we present the details of the whole process.

### 3.1 Local syntactic information

One way to study the relationship between a term and a semantic label is to focus on the syntactic context in which the relationship takes place. The idea is that a semantic label may be characterized by limited sets of syntactic contexts. According to this assumption, the task can be seen as a classification problem where each common noun $t$ in a sentence has to be associated with a specific semantic label by analyzing the syntactic structure of the text around it (Table 1).

In our work, text is syntactically analyzed via dependency parsers. For the English language we used the Stanford Toolkit,[4] while for the Italian language we used the dependency parser TULE (Lesmo 2009). The extracted dependencies are transformed into generalized textual representations in the form of triples. In particular, for each syntactic dependency

[4] http://nlp.stanford.edu/software/index.shtml

**Table 1** The instance created for the noun $word_2$ is composed by three items (one for each syntactic dependency related to $word_2$)

| Dependence | Instance item |
|---|---|
| $dep\text{-}type_1(word_2, word_1)$ | $dep\text{-}type_1$-**target**-$word_1$ |
| $dep\text{-}type_3(word_2, word_3)$ | $dep\text{-}type_3$-**target-noun** |
| $dep\text{-}type_4(word_5, word_2)$ | $dep\text{-}type_4$-**noun-target** |

Note that the considered noun $word_2$ is replaced by the generic term "**target**", while the other nouns are replaced with "**noun**" (in the example, this happens for terms $word_3$ and $word_5$)

$dep(a, b)$ (or $dep(b, a)$) of a considered noun $a$, we create a generalized token $dep\text{-}target\text{-}\hat{b}$ (or $dep\text{-}\hat{b}\text{-}target$), where $\hat{b}$ becomes the generic string "*noun*" in case it is another noun; otherwise it is equal to $b$. Thus, common nouns are transformed into coarse-grained context abstractions, creating a level of generalization of the feature set that collapses the variability of the nouns involved in the syntactic dependencies. The string "*target*" is useful to determine the exact position of the considered noun in a syntactic dependency (as a left argument, or as a right argument). For instance, consider a sentence formed by 5-words:

$$word_1\ [word_2]_L\ word_3\ word_4\ word_5.$$

and assume the term $word_2$ is labeled with the semantic label $L$. The result of the Part-Of-Speech tagging procedure will produce the following output:

$$word_1|pos_1\ [word_2]_L|pos_2\ word_3|pos_3\ word_4|pos_4\ word_5|pos_5.$$

where $pos_k$ identifies a specific Part-of-Speech tag. Then, the syntactic parsing will produce a sequence of dependencies like in the following example:

$$\begin{aligned} &dep - type_1(word_2, word_1)\\ &dep - type_2(word_1, word_4)\\ &dep - type_3(word_2, word_3)\\ &dep - type_2(word_4, word_3)\\ &dep - type_2(word_1, word_3)\\ &dep - type_4(word_5, word_2) \end{aligned}$$

where each dependency $dep\text{-}type_k$ indicates a specific kind ok syntactic connection (e.g., determiners, subjects and objects of the verb, and so forth).

At this point, the system creates one instance for each term labeled as "noun" by the POS-tagger. For example, let us assume the term $word_2$ is a noun, the instance will be represented by three abstract terms, as shown in Table 3. In the instance, the noun under evaluation is replaced by the generic term **target**, while all the other nouns are replaced with **noun** (in the example, this happens for terms $word_3$ and $word_5$).

Once the instance for the noun $word_2$ is created, it is passed to the classification process that will decide if it can be considered as part of a candidate term to be associated with the semantic label $L$. This is done for each noun in a sentence.

### 3.2 Learning phase

Once all nouns in the labeled data are transformed into syntax-based generalizations, we create labeled numeric vectors in order to be able to use standard Machine Learning approaches for the automatic classification step. More in detail, given a sentence $S$ containing terms associated with a semantic label $L$, the system produces as many input instances as the number of common nouns contained in $S$. Only those that are associated with $L$ will be positive instances for the classifier, while the other nouns will be negative examples. More specifically, for each noun $n$ in $S$, we create an instance $S^n$ labeled as *positive* if $rel(n, L)$ exists; otherwise, it is labeled as *negative*.

At the end of this process, a training set is built for the target semantic label $L$, namely the $L$-set. All the instances of the dataset are transformed into numeric vectors according to the Vector Space Model (Salton et al. 1975), and fed into a Support Vector Machine classifier (Cortes and Vapnik 1995). In particular, we used the Sequential Minimal Optimization implementation of the Weka framework (Hall et al. 2009). We refer to the resulting model as the $L$-model. This model is a binary classifier that, given the local syntactic information of a noun, tells us if the noun can/cannot be associated with the semantic label $L$. An example for the sentence illustrated in the previous section is shown in Table 2.

The whole set of instances $L$-set is fed into a Support Vector Machine classifier. At this point, it is possible to classify each term as possible candidates for the semantic label $L$.

Notice that our approach is susceptible from the errors given by the POS-tagger and the syntactic parser. In spite of this, our approach demonstrates how syntax can be more robust for identifying semantic relations. Our approach does not make use of the full parse tree, thus we are not dependent on a complete and correct result of the parser.

## 4 Semantic entities in the legal domain

In this section, we present how we applied our approach in the identification of relationships between legal texts and semantic labels. Let us start considering the following text about a legal prescription:

**Table 2** The instances created for the sentence of the example (one for each noun)

| Noun | Instance | Label L |
|---|---|---|
| $word_1$ | $dep\text{-}type_1$-target-$word_1$ | *negative* |
| | $dep\text{-}type_2$-target-$word_4$ | |
| | $dep\text{-}type_2$-$word_1$-noun | |
| $word_2$ | $dep\text{-}type_1$-target-$word_1$ | *positive* |
| | $dep\text{-}type_3$-target-noun | |
| | $dep\text{-}type_4$-noun-target | |
| $word_3$ | $dep\text{-}type_3$-noun-target | |
| | $dep\text{-}type_2$-$word_4$-target | |
| | $dep\text{-}type_2$-$word_1$-target | |
| $word_4$ | $dep\text{-}type_2$-$word_1$-target | *negative* |
| | $dep\text{-}type_2$-target-noun | |
| $word_5$ | $dep\text{-}type_4$-target-noun | *negative* |

*A pena di una ammenda da 2500 a 6400 euro o dell'arresto da tre a sei mesi, il datore di lavoro deve mantenere in efficienza i dispositivi di protezione individuale e assicurare le condizioni d'igiene per i dipendenti, mediante la manutenzione, le riparazioni e le sostituzioni necessarie e secondo le eventuali indicazioni fornite dal fabbricante.*

*[Under penalty of 2500 to 6400 euros or a three to six months detention, the work supervisor must maintain the personal protective equipment and ensure the hygiene conditions for the employees through maintenance, repairs and replacements necessary and in accordance with any instructions provided by the manufacturer.]*

This legal prescription contains the following semantic annotations:

rel(*datore*, ACTIVE − ROLE))
rel(*dipendenti*, PASSIVE − ROLE)
rel(*condizioni*, INVOLVED − OBJECT)
rel(*dispositivi*, INVOLVED − OBJECT)

This means that, in order to automatically identify the three semantic labels, we had to learn three different models, one for each label. Considering this example, the result of the parsing procedure will be the following:

ARG(pena − 2, a − 1)
RMOD(dovere − 24, pena − 2)
ARG(ammenda − 5, di − 3)
ARG(ammenda − 5, un − 4)
RMOD(pena − 2, ammenda − 5)
ARG(2500 − 7, da − 6)
RMOD(ammenda − 5, 2500 − 7)
ARG(euro − 10, a − 8)
ARG(euro − 10, 6400 − 9)
RMOD(dovere − 24, euro − 10)
COORD(arresto − 13, o − 11)
ARG(arresto − 13, di − 12)
RMOD(pena − 2, arresto − 13)
ARG(tre − 15, da − 14)
...

where SUBJ stands for subject relations, OBJs are themes, ARGs are mandatory arguments, COORDs are cordinations, and RMODs are modifiers.

Then, the following terms are identified as nouns by the POS-tagger: *pena, ammenda, euro, arresto, mesi, datore, lavoro, efficienza, dispositivi, protezione condizioni, igiene, manutenzione, riparazioni, sostituzioni, indicazioni, fabbricante*.

At this point, the system creates one instance for each identified noun. For example, for the noun phrase "*datore di lavoro*" (work supervisor), the instance will be represented

by three abstract terms, as shown in Table 3. In the instance, the noun under evaluation is replaced by the generic term *target*, while all the other nouns are replaced with *noun*. It is important to note that only the term "datore" (i.e., "supervisor") is taken into account, since "di lavoro" (i.e., "of work") is one of its modifiers.

The dataset used for evaluating our approach contains 560 legal texts annotated with various semantic information, with a total of 6939 nouns. In particular, the data include an extended structure for prescriptions, which has been described in Boella et al. (2012) as individual legal obligations derived from legislation. For our experiments we used three types of semantic labels:

Active role The active role indicates an active agent involved within the situation described in the text. Examples of common entites related to active roles are directors of banks, doctors, security managers.

Passive role The passive role indicates an agent that is the beneficiary of the described norm. Examples of agents associated with passive roles are workers and work supervisors.

Involved Object An involved object represents an entity that is central for the situation being described. Examples are types of risk for a worker, the location of a specific work, and so on.

In the corpus there are 509 annotated active roles, 142 passive roles, and 615 involved objects out of a total of 6939 nouns.

The result of this evaluation is threefold: first, we evaluate the ability of the proposed approach to identify and annotate active roles; then we focus on the passive roles; finally, we face the more challenging recognition of involved objects, given their high level of semantic abstraction. Table 4 shows the accuracy levels reached by the approach using the 10-folds cross validation scheme.

As can be noticed, the approach works almost perfectly with the *active role* semantic tag. This means that the syntactic context of the active roles are well circumscribed, thus it is easy for the classifier to build the model. Regarding the *passive role* tag, even if the approach is sufficiently good when identifying the right semantic label (68.7 % of Precision), it returns many false negative (32.4 % of Recall). In a semi-supervised context of an ontology learning process, this can be anyway a good support, since all of what has been automatically identified is likely to be correct. Finally, the *involved object* semantic tag gave quite low results in terms of Precision and Recall. On average, only six to ten nouns classified as involved objects were actually annotated with the correct semantic label. This is due to the very wide semantic coverage of this specific tag, and its consequently broad syntactic context.

**Table 3** The instance created for the noun "*datore*" is composed by three items (one for each syntactic dependency related to "*datore*")

| Dependency | Instance item |
|---|---|
| ARG(datore, il) | ARG-**target**-il |
| SUBJ(dovere, datore) | SUBJ-dovere-**target** |
| RMOD(datore, lavoro) | RMOD-**target**-noun |

Note that the considered noun "*datore*" is replaced by the generic term "**target**", while the other nouns are replaced with "*noun*"

**Table 4** Precision, Recall and F-Measure values for the identification of active roles, passive roles, and involved objects, using 10-folds cross validation

| | *Precision* | *Recall* | *F-Measure* |
|---|---|---|---|
| Active role | | | |
| *yes* | 91.0 % | 89.6 % | 90.3 % |
| *no* | 99.2 % | 99.3 % | 99.2 % |
| Passive role | | | |
| *yes* | 68.7 % | 32.4 % | 44.0 % |
| *no* | 98.6 % | 99.7 % | 99.1 % |
| Involved object | | | |
| *yes* | 60.1 % | 25.7 % | 36.0 % |
| *no* | 93.2 % | 98.3 % | 95.7 % |

## 5 Hypernym relations in wikipedia entries

In this section we present the results of our approach for the extraction of hypernyms and hyponyms from text. In fact, these semantic information can be considered as semantic labels and on which semantic search strategies can work with. In this section we present the evaluation of our approach to extract hyponyms and hypernyms, individually. We used an annotated dataset of definitional sentences (Navigli et al. 2010) containing 4,619 sentences extracted from Wikipedia.

Table 5 shows the results, in terms of Precision, Recall, and F-Measure. As can be noticed, the approach is able to identify correct $x$ and $y$ with high accuracy. Interestingly, hyponyms seem to have more stable syntactic contexts rather than hypernyms. Moreover, while Recall seems to be quite similar between the two, Precision is much higher (+11.6 %) for the extraction of hyponyms.

While these results demonstrate the potential of the approach, it is interesting to analyze which syntactic information frequently reveal hyponyms and hypernyms. Table 6 shows the top 10 most important features for both the $x$ and the $y$ in a $hyp(x, y)$ relation, computing the value of the chi-squared statistics with respect to the class ($x$ and $y$, respectively). A part from dataset-specific features like *amod-target-geologic* (marked in *italics*), many interesting considerations can be done by looking at Table 6.

For example, the syntactic dependency $nsubj$ results to be important for the identification of both hyponyms and hypernyms. The formers, in fact, are often syntactic subjects of a clause, and vice versa for the latters. Interestingly, *nsubj-noun-target* (marked in **bold** in Table 6) is important to both identify a correct hyponym and to reveal that a noun *is not* a

**Table 5** Accuracy levels for the classification of single hyponyms ($x$) and hypernyms ($y$) using their local syntactic context, in terms of Precision ($P$), Recall ($R$), and F-Measure ($F$), using 10-folds cross validation

| Target | $P$ | $R$ | $F$ |
|---|---|---|---|
| $x$ | 93.85 % | 79.04 % | 85.81 % |
| $y$ | 82.26 % | 76.77 % | 79.42 % |

**Table 6** The top 10 most relevant features for the classification of single hyponyms and hypernyms from a sentence, computing the value of the chi-squared statistic with respect to the class ($x$ and $y$, respectively)

| Top features for $x$ | Top features for $y$ |
|---|---|
| **nsubj-noun-target** | cop-target-be |
| det-target-a | nsubj-target-noun |
| nsubj-refer-target | det-target-a |
| cop-target-be | prepin-target-noun |
| nsubj-target-noun | **nsubj-noun-target** |
| prepof-noun-target | partmod-target-use |
| prepof-target-noun | prepto-refer-target |
| nn-noun-target | prepof-target-noun |
| det-noun-a | det-target-any |
| nsubjpass-define-target | *amod-target-geologic* |

The feature "nsubj-noun-target" (marked in **bold**) is important to identify a correct hyponym and to estimate that a noun *is not* a hypernym, while this seems not true for "nsubj-target-noun". Clear dataset-specific features are marked in *italic*

hypernym (*nsubj-noun-target* is present in both the two columns $x$ and $y$), while this seems not true for *nsubj-target-noun* (it is only important to say if a noun can be a hypernym, and not to say if such noun *is not* a hyponym).

We label as *definitional* all the sentences that contain at least one hypernym and one noun hyponym in the same sentence. Thus, given an input sentence:

1. we extract all the nouns (POS-tagging),
2. we extract all the syntactic dependencies of the nouns (dependency parsing),
3. we classify each noun (i.e., its instance) with the $x$-model and to the $y$ model,
4. we check if there exist at least one noun classified as $x$ and one noun classified as $y$: in this case, we classify the sentences as *definitional*.

As in the previous task, we used the dataset of definitional sentences presented in Navigli et al. (2010). Table 7 shows the accuracy of the approach for this task. As can be seen, our proposed approach has a high Precision, with a high Recall. Although Precision is lower than the pattern matching approach proposed by Navigli and Velardi (2010), our Recall is higher, leading to an higher F-Measure.

**Table 7** Evaluation results for the classification of definitional sentences, in terms of Precision ($P$), Recall ($R$), F-Measure ($F$), and Accuracy ($Acc$), using 10-folds cross validation

| Algorithm | $P$ | $R$ | $F$ | $Acc$ |
|---|---|---|---|---|
| WCL-1 (Nav. Vel. 2010) | 99.88 % | 42.09 % | 59.22 % | 76.06 % |
| WCL-3 (Nav. Vel. 2010) | 98.81 % | 60.74 % | 75.23 % | 83.48 % |
| Star patterns (Nav. Vel. 2010) | 86.74 % | 66.14 % | 75.05 % | 81.84 % |
| Bigrams (Cui et al. 2007) | 66.70 % | 82.70 % | 73.84 % | 75.80 % |
| Our approach | 88.09 % | 76.01 % | 81.61 % | 89.67 % |

**Table 8** Evaluation results for the hypernym relation extraction, in terms of Precision ($P$), Recall ($R$), and F-Measure ($F$)

| Algorithm | $P$ | $R$ | $F$ |
|---|---|---|---|
| WCL-1 (Nav. Vel. 2010) | 77.00 % | 42.09 % | 54.42 % |
| WCL-3 (Nav. Vel. 2010) | 78.58 % | 60.74 % | 68.56 % |
| Baseline | 57.66 % | 21.09 % | 30.76 % |
| Our approach | 83.05 % | 68.64 % | 75.16 % |

These results are obtained using 10-folds cross validation

Our method for extracting hypernym relations makes use of two models: one fo the hypernyms extraction and one for the hyponyms, as for the the task of classifying definitional sentences. If exactly one $x$ and one $y$ are identified in the same sentence, they are directly connected and the relation is extracted. The only constraint is that $x$ and $y$ must be connected within the same parse tree. In case the sentence contains more than one noun that is classified as hypernym (or hyponym), there are two possible scenarios:

1. there are actually more than one hypernym (or hyponym), or
2. the classifiers returned some false positive.

Up to now, we decided to keep all the possible combinations, without further filtering operations.[5] Finally, in case the system finds multiple hypernyms and multiple hyponyms at the same time, the problem becomes to select which hypernym is linked to which hyponym. To do this, we simply calculate the distance between these terms in the parse tree (the closer the terms, the better the connection between the two). Nevertheless, in the used corpus, only around 1.4 % of the sentences are classified with multiple hypernyms and hyponyms.

The results of our approach in this task is shown in Table 8. We still used the dataset of definitional sentences of Navigli et al. (2010).

Table 8 shows the results of the extraction of the whole hypernym relations. We also added the performance of a system named "Baseline", which implements our strategy but only using the POS tags of the nouns' neighbor words instead of their syntactic dependencies. Its low effectiveness demonstrates the importance of the syntactic information, independently from the learning phase. Finally, note that our approach reached high levels of accuracy. In particular, our system outperforms the pattern matching algorithm proposed by Navigli and Velardi (2010) in terms of both Precision and Recall.

### 5.1 Further considerations

The data provided by Navigli et al. (2010) also contain a dataset of over 300,000 sentences retrieved from the UkWac Corpus (Ferraresi et al. 2008). Unfortunately, Precision was only manually validated, therefore we could not be able to make any fair comparison. Nevertheless, they made available a subset of 99 definitional sentences. On such data, our technique obtained a Recall of 59.6 % (59 out of 99), while their approaches reached 39.4 %, 56.6 %, and 63.6 % respectively for WCL-1, WCL-3, and Star Patterns.

In the dataset, the syntactic parser found hundreds of cases of coordinated hyponyms, while the annotation provides only one hyponym for each sentence. For this reason, we

[5] We only used the constraint that the hypernym has to be different from the hyponym.

were not able to evaluate our method on the extraction of all possible relations with all coordinated hyponyms.

## 6 Further experiments

In this section we evaluate the approach on different types of data. More in detail, in addition to legal texts and Wikipedia entries, we experimented our approach also on social network data. In particular, we used a dataset of 1-million Twitter posts (called tweets)[6] from which we automatically extracted 100 well-formed sentences (i.e., no anomalies were detected in the use of punctuation, all the used words were checked with the WordNet dictionary, and there was no presence of hashtags) with a number of characters close to the maximum allowed (140). Since tweets contain texts that usually do not contain taxonomical information, we only considered tweets having trigger keywords like 'to be' and 'kind of'. In the first 100,000 tweets, we found only 124 texts following these constraints. We randomly selected 100 texts from them, manually evaluating the results of our approach. Of course, given the nature of these data, it has been difficult to find definitions and hypernyms. In spite of this, for instance, the tweet "*An alarm clock is a device for waking up people who do not have small children...*" contains the relation between alarm clock and device, even if the text represents an ironic expression rather than a definition. During the manual annotation, only 4 tweets resulted to be definitions with hypernym relations, and the system was able to extract them. To the contrary, 2 non-definitional tweets have been tagged as definitional. Therefore, in this domain, the approach got a precision of 66.67 % and a recall of 100 % for the definitional tweets, and a precision of 100 % and a recall of 97.91 % for the non-definitional ones.

## 7 Ontology learning from text: a further look

An important aspect to take into consideration when facing a semantic extraction process for ontology learning from textual data is how the meaning is encapsulated at sentence and discourse level, instead of at word-level only. For instance, in the case of linguistic modifiers, it is important to understand whether they are necessary or if their absence would change the meaning of the whole linguistic construction. In fact, the composition into single lexical units (syntagms) creates unique and indivisible concepts. On the other hand, when a modifier is not necessary, the semantics expressed by the text remains the same (even if less specific or lightly different). A major layer of specialization is certainly useful for the reader, but the underlying ontological concepts remain the same.

Another interesting fact to further investigate is when a linguistic modifier is a noun and not an adjective, because it usually reflects the presence of a single syntagm. An examples is "circuit board": the single words "circuit" and "board" refer to distinct concepts compared to the one of their composition. But it is not always the case. For instance, the noun modifier of the construct "round table" suggests only something about its functionalities (for instance it is a type of table that is particularly safe for kids because of the absence of edges), but it does not represent a completely different concept with respect to "table".

[6] http://thinknook.com/wp-content/uploads/2012/09/Sentiment-Analysis-Dataset.zip

In the light of this, we are certainly talking about a higher level of semantics, which is obviously complex to treat even at the ontological level. In fact the correct understanding of a single word suggests us a mental representation. This means that there is a direct link with the descriptive meaning of the considered concept that we have in mind.

In this section, we only want to introduce the reader to the concept of linguistic affordances, that is the graded relationship between words and modifiers to construct meanings that somehow reflect some mental models. We may approach this problem by considering terms compositions in a dynamic way where the meaning is distributed among subjects, objects, functionalities, and mental representations. In general, the concept of "affordance" is linked to the meaning of an action that is dynamically created by the interaction of the involved agents. Dropping this principle into language, an action (for example suggested through the use of a verbal construct) will have a certain meaning that is given by the interaction between the agent and the receiver (subject / object), and more particularly by their properties. The idea is that different combinations of words with different properties are likely to lead to "different" meanings.

One of the main problem currently faced by computational linguists is to solve the ambiguity of natural language at word level. In future works we may consider to see words not as isolated entity, but as bricks in a context where the interaction plays a fundamental role in creating the actual meaning. The notion of "affordance" was first suggested by Gibson (1977) in his theory of perception and was later re-articulated by Norman in the field of interface design (Norman 1999).

## 8 Conclusions

In this work we proposed a general approach for the automatic extraction of semantic information to improve semantic search and ontology building from textual data. First, we rely on Natural Language Processing techniques to obtain rich lexical and syntactic information. Then, we transform these knowledge into generalized features that aim at capturing the surrounding linguistic variability of the target semantic labels. Finally, such extracted data are fed into a Support Vector Machine classifier which creates a model to automate the semantic annotation and to provide semantic-aware search queries. We tested our technique on different tasks both in the legal domain and in the Wikipedia knowledge base, reaching high accuracy levels. In future work, we aim at integrating our approach with existing methods (both unsupervised and supervised) for ontology learning.

## References

Berland, M., & Charniak, E. (1999). Finding parts in very large corpora. In *Annual meeting association for computational linguistics* (Vol. 37, pp. 57–64). Association for Computational Linguistics.

Biagioli, C., Francesconi, E., Passerini, A., Montemagni, S., Soria, C. (2005). Automatic semantics extraction in law documents. In *Proceedings of the 10th international conference on artificial intelligence and law: ICAIL* (pp. 133–140). ACM.

Biemann, C. (2005). Ontology learning from text: a survey of methods. In *LDV forum* (Vol. 20, pp. 75–93).

Boella, G., di Caro, L., Humphreys, L., Robaldo, L., van der Torre, L. (2012). Nlp challenges for eunomos, a tool to build and manage legal knowledge. In *Proceedings of the 8th international conference on language resources and evaluation (LREC)*.

Boella, G., & Di Caro, L. (2013). Supervised learning of syntactic contexts for uncovering definitions and extracting hypernym relations in text databases. In *Machine learning and knowledge discovery in databases* (pp. 64–79). Berlin Heidelberg: Springer.

Boella, G., Di Caro, L., Robaldo, L. (2013). Semantic relation extraction from legislative text using generalized syntactic dependencies and support vector machines. In *Theory, practice, and applications of rules on the web* (pp. 218–225). Berlin Heidelberg: Springer.

Boella, G., Martin, M., Rossi, P., van der Torre, L., Violato, A. (2012). Eunomos, a legal document and knowledge management system for regulatory compliance. In *Proceedings of information systems: a crossroads for organization, management, accounting and engineering (ITAIS) conference*. Berlin: Springer.

Borg, C., Rosner, M., Pace, G. (2009). Evolutionary algorithms for definition extraction. In *Proceedings of the 1st workshop on definition extraction* (pp. 26–32). Association for Computational Linguistics.

Buitelaar, P., Cimiano, P., Magnini, B. (2005). Ontology learning from text: an overview. *Ontology Learning from Text: Methods, Evaluation and Applications*, *123*, 3–12.

Candan, K., Di Caro, L., Sapino, M. (2008). Creating tag hierarchies for effective navigation in social media. In *Proceedings of the 2008 ACM workshop on search in social media* (pp. 75–82). ACM.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, *20*(3), 273–297.

Cui, H., Kan, M.Y., Chua, T.S. (2007). Soft pattern matching models for definitional question answering. *ACM Transactions on Information Systems*, *25*(2). doi:10.1145/1229179.1229182.

Del Gaudio, R., & Branco, A. (2007). Automatic extraction of definitions in portuguese: a rule-based approach. *Progress in Artificial Intelligence*, 659–670.

Fahmi, I., & Bouma, G. (2006). Learning to identify definitions using syntactic features. In *Proceedings of the EACL 2006 workshop on learning structured information in natural language applications* (pp. 64–71).

Ferraresi, A., Zanchetta, E., Baroni, M., Bernardini, S. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th web as corpus workshop (WAC-4) can we beat Google* (pp. 47–54).

Fortuna, B., Mladenič, D., Grobelnik, M. (2006). Semi-automatic construction of topic ontologies. *Semantics, Web and Mining*, 121–131.

Gibson, J. (1977). The concept of affordances. In *Perceiving, acting, and knowing* (pp. 67–82).

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H. (2009). The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.

Harris, Z. (1954). Distributional structure. *Word*, *10*(23), 146–162.

Hearst, M. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on computational linguistics* (Vol. 2, pp. 539–545). Association for Computational Linguistics.

Hoffart, J., Suchanek, F.M., Berberich, K., Weikum, G. (2012). Yago2: a spatially and temporally enhanced knowledge base from wikipedia. *Artificial Intelligence*.

Hovy, E., Philpot, A., Klavans, J., Germann, U., Davis, P., Popper, S. (2003). Extending metadata definitions by automatically extracting and organizing glossary definitions. In *Proceedings of the 2003 annual national conference on digital government research* (pp. 1–6). Digital Government Society of North America.

Klavans, J., & Muresan, S. (2001). Evaluation of the definder system for fully automatic glossary construction. In *Proceedings of the AMIA symposium* (p. 324). American Medical Informatics Association.

Lesmo, L. (2009). The turin university parser at evalita 2009. *Proceedings of EVALITA*, 9.

Lesmo, L., Mazzei, A., Palmirani, M., Radicioni, D.P. (2013). Tulsi: an nlp system for extracting legal modificatory provisions. *Artificial Intelligence and Law*, 1–34.

de Maat, E., Krabben, K., Winkels, R. (2010). Machine learning versus knowledge based classification of legal texts. In *Proceedings of legal knowledge and information systems conference: JURIX 2010* (pp. 87–96). IOS Press. http://portal.acm.org/citation.cfm?id=1940559.1940573.

Miller, G.A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, *38*(11), 39–41.

Moschitti, A., & Bejan, C.A. (2004). A semantic kernel for predicate argument classification. In *CoNLL-2004*.

Navigli, R., & Ponzetto, S.P. (2010). Babelnet: building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 216–225). Association for Computational Linguistics.

Navigli, R., & Velardi, P. (2010). Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th annual meeting of the association for computational linguistics* (pp. 1318–1327). Uppsala: Association for Computational Linguistics. http://www.aclweb.org/anthology/P10-1134.

Navigli, R., Velardi, P., Ruiz-Martnez, J.M. (2010). An annotated dataset for extracting definitions and hypernyms from the web. In *Proceedings of the 7th international conference on language resources and evaluation (LREC'10)*. Valletta: European Language Resources Association (ELRA).

Norman, D.A. (1999). Affordance, conventions, and design. *Interactions*, *6*(3), 38–43.

Ponzetto, S., & Strube, M. (2007). Deriving a large scale taxonomy from wikipedia. In *Proceedings of the national conference on artificial intelligence* (Vol. 22, p. 1440). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press. 1999.

Salton, G., Wong, A., Yang, C.S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, *18*(11), 613–620. doi:10.1145/361219.361220.

Storrer, A., & Wellinghoff, S. (2006). Automated detection and annotation of term definitions in german text corpora. In *Proceedings of LREC* (Vol. 2006).

Velardi, P., Faralli, S., Navigli, R. (2012). Ontolearn reloaded: a graph-based algorithm for taxonomy induction.

Westerhout, E. (2009). Definition extraction using linguistic and structural features. In *Proceedings of the 1st workshop on definition extraction, WDE '09* (pp. 61–67). Stroudsburg: Association for Computational Linguistics. http://dl.acm.org/citation.cfm?id=1859765.1859775.

Yamada, I., Torisawa, K., Kazama, J., Kuroda, K., Murata, M., De Saeger, S., Bond, F., Sumida, A. (2009). Hypernym discovery based on distributional similarity and hierarchical structures. In *Proceedings of the 2009 conference on empirical methods in natural language processing* (Vol. 2, pp. 929–937). Association for Computational Linguistics.

Yang, H., & Callan, J. (2008). Ontology generation for large email collections. In *Proceedings of the 2008 international conference on digital government research* (pp. 254–261). Digital Government Society of North America.

Zhang, C., & Jiang, P. (2009). Automatic extraction of definitions. In: *2nd IEEE international conference on computer science and information technology, 2009. ICCSIT 2009* (pp. 364–368). doi:10.1109/ICCSIT.2009.5234687.