An automatic approach for ontology-based feature extraction from heterogeneous textual resources

Carlos Vicent*, David Sánchez, Antonio Moreno

Intelligent Technologies for Advanced Knowledge Acquisition (ITAKA), Departament d’Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili. Av. Països Catalans, 26, 43007 Tarragona, Catalonia, Spain

ARTICLE INFO

Article history:
Received 24 February 2012
Received in revised form 24 July 2012
Accepted 10 August 2012
Available online 13 September 2012

Keywords:
Information extraction
Feature extraction
Ontologies
Wikipedia

ABSTRACT

Data mining algorithms such as data classification or clustering methods exploit features of entities to characterise, group or classify them according to their resemblance. In the past, many feature extraction methods focused on the analysis of numerical or categorical properties. In recent years, motivated by the success of the Information Society and the WWW, which has made available enormous amounts of textual electronic resources, researchers have proposed semantic data classification and clustering methods that exploit textual data at a conceptual level. To do so, these methods rely on pre-annotated inputs in which text has been mapped to their formal semantics according to one or several knowledge structures (e.g. ontologies, taxonomies). Hence, they are hampered by the bottleneck introduced by the manual semantic mapping process. To tackle this problem, this paper presents a domain-independent, automatic and unsupervised method to detect relevant features from heterogeneous textual resources, associating them to concepts modelled in a background ontology. The method has been applied to raw text resources and also to semi-structured ones (Wikipedia articles). It has been tested in the Tourism domain, showing promising results.

© 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Since the creation of the World Wide Web, its structure and architecture have been in constant growth and development. Nowadays the Web has turned into what we know as the Social Web, where all users are able to add and modify its content. This fact has resulted in an exponential growth of the available information. However, its lack of structure has brought some problems: the access and retrieval of information is difficult and inaccurate, and the unstructured and mainly textual information can hardly be interpreted by IT applications (Fensel et al., 2002).

In order to solve these problems, the Semantic Web (Berners-Lee et al., 2001) has been proposed.

The Semantic Web is an evolving extension of the World Wide Web in which the semantics of information and services are explicitly defined, making it possible for the Web to understand and satisfy the requests of users and computers. One of the basic pillars of the Semantic Web concept is the idea of having explicit semantic information on the Web pages that can be used by intelligent agents in order to solve complex problems such as Information Retrieval and Question Answering (Brill, 2003). The final objective of the Semantic Web is to be able to semantically analyse and catalog Web contents. These tasks require a set of structures to model knowledge, and a linkage between the knowledge and Web contents. The Semantic Web relies on two basic components: ontologies and semantic annotations.

On one hand, ontologies are formal, explicit specifications of a shared conceptualization (Guarino, 1998). Ontologies are useful to model knowledge in a formal abstract way which can be interpreted and exploited by computers. On the other hand, annotations are linkages between textual contents of a document and their formal semantics, modelled in a knowledge structure such an ontology, enabling the development of knowledge-based systems (Setchi et al., 2011). The availability of such amount of electronic resources has brought a growing interest in the research community in the exploitation of Web contents. Data mining algorithms, and particularly data classification and clustering methods, aim to analyse, characterise, classify or group data. Since classical data analysis methods were designed to deal with numerical data, in recent years researchers have proposed novel semantically-grounded methods that, by exploiting structured knowledge bases like ontologies, taxonomies or folksonomies, are able to semantically interpret textual data such as Web contents (Hotho et al., 2002). In He et al. (2008) authors propose a data classification method that generalises set-valued textual data (e.g. query logs, clinical records, etc.) relying on domain taxonomies. In Batet et al. (2011, 2010) authors adapt well-known clustering methods so that they can semantically interpret (i.e. compare, group and aggregate) multi-featured textual
data (e.g. polls, electronic health-care records, etc.), exploiting general purpose and domain-specific ontologies. A similar process is carried out in Martínez et al. (2012), but focusing on semantic micro-aggregation of textual values for privacy-preserving purposes. Even though these methods deal with textual features, they suppose structured inputs rather than raw textual documents, in which words describing an entity have been extracted and associated to concepts whose semantics are modelled in a knowledge base (Hotho et al., 2002).

Unfortunately, semantically-tagged features, which can be useful for semantic data analysis, are scarce nowadays. Due to the semantic annotation bottleneck that characterises manual approaches, most of the existing textual resources are plain text documents or ad-hoc and partially annotated ones (such as Wikipedia articles). Hence, the applicability of semantic-based data mining methods is hampered by the availability of adequately processed input information.

To tackle this problem, this paper proposes a method that, given an input document describing an entity (e.g. a Web resource, a Wikipedia article or a plain text document) and an ontology stating which features should be extracted (e.g. tourism points of interest), it is able to detect, extract and semantically annotate relevant textual features describing that entity. The method has been designed in a general way so that it can be applied to a range of textual documents going from raw plain text to semi-structured resources (e.g. tagged Wikipedia articles). In this last case, the method is able to exploit the pre-processed input to complement its own learning algorithms. The key point of the work is to leverage the syntactical parsing and several natural language processing techniques with the knowledge contained in the input ontology, and the use of the Web as a corpus to assist the learning process to be able to (1) identify relevant features describing a particular entity in the input document, and (2) associate, if applicable, the extracted features to concepts contained in the input ontology. In this manner, the output of the system consists on tagged features which can be directly exploited by semantically grounded data analysis methods such as clustering (Batet et al., 2011).

The rest of the paper is organised as follows: Section 2 discusses related works proposing annotation systems for textual data. Section 3 provides and discusses the theoretical background, techniques and working environment for the proposed method. Section 4 details and formalises the proposed method, detailing also how it can be applied to different kinds of inputs (unstructured and semi-structured text). Section 5 evaluates the proposed method when dealing with touristic destinations, testing the behaviour and influence of learning parameters and input types and, finally, it is shown the global performance of the method when analysing different domains. The final section discusses and summarises the main contributions of the work.

2. Related work

In the context of the Semantic Web, several methods have been proposed to detect and annotate relevant entities from electronic resources. First, manual approaches have been developed to assist the user in the annotation process such as Annotea (Koivunen, 2005), CREAM (Handschuh et al., 2003), NOMOS (Niekrasz and Gruenstein, 2006) or Vannotea (Schroeter et al., 2003). Those systems rely on the skills and will of a community of users to detect and tag entities within Web content. Considering that there are 1 trillion pages on the Web, it is easy to envisage the unfeasibility of manual annotation of Web resources.

On the contrary, some authors have tried to automate some of the stages of the annotation process (proposing supervised systems) in order to overcome the bottleneck of manual solutions. Melita (Cravegna et al., 2002) is based on user-defined rules and pre-defined annotations, which are employed to propose new ones. Supervised approaches, however, are difficult to apply, due to the effort required to compile a large and representative training set. Cerno is another system for the semi-automated annotation of Web resources (Kiyavitskaya et al., 2005,2009). The authors propose the use of patterns/annotation schemas (addressed to extract objects such as email addresses, phone numbers, dates and prices) to obtain the candidates to annotate, and then this set is annotated by means of a domain conceptual model. That model represents the information of a particular domain through concepts, relationships and attributes (in an entity-relation based syntax). Supervised systems also use extraction rules obtained from a set of pre-tagged data (Califf and Mooney, 2003; Roberts et al., 2007). WebKB (Cafarella et al., 2005) and Armadillo (Alfonseca and Manandhar, 2002) use supervised techniques to extract information from computer science websites. Likewise, S-CREAM (Cunningham et al., 2002) uses Machine Learning techniques to annotate a particular document with respect to an ontology, given a set of annotated examples.

Other supervised systems present a more ad-hoc design and focus on a specific domain of knowledge. In Maedche et al. (2003) a system that works on the Tourism domain is proposed. They combine lexical knowledge, extraction rules and ontologies in order to extract information in the form of instantiated concepts and attributes that are stored in an ontology-like fashion (hotel names, number of rooms, prices, etc.). The most interesting characteristic is the fact that the pre-defined knowledge structures are extended as a result of the Information Extraction process, improving and completing them. They use several Ontology Learning techniques already developed for the OntoEdit system (Staab and Maedche, 2000). A human expert has to validate each extension before continuing. Another system focused on the Tourism domain is presented in Fellmayr et al. (2009). A domain-dependent crawler collects Web pages corresponding to accommodation websites. This corpus is passed to an extraction module based on the GATE framework (Cunningham et al., 2002) which provides a number of text engineering components. It performs an annotation of the Web pages in the corpus, supported by a domain-dependent ontology and rule set. The extracted tokens are ranked as a function of their frequency and relevance for the domain. Semantic-ART (Sapkota et al., 2011) is another system that aims to annotate texts associated to regulatory information to convert them into semantic models. The system assumes that a regulation typically comprises at least one statement (a sentence in the regulation), which must have a subject, an obligation and an action. The presence of an obligation (i.e., sentences containing obligatory verbs) in a paragraph helps to distinguish the regulation from the ordinary text. The subjects and objects of these sentences are directly searched in an input ontology. WordNet is used to look for synonyms of these words if a direct matching is not found.

Another domain-dependent system is SOBA (Buitelaar et al., 2008), a sub-component of the SmartWeb (a multi-modal dialog system that derives answers from unstructured resources such as the Web), which automatically populates a knowledge base with information extracted from soccer match reports found on the Web. The extracted information is defined with respect to an underlying ontology. The SOBA system consists of a Web crawler, linguistic annotation components and a module for the transformation of linguistic annotation into an ontology-based representation. The first component enables the automatic creation of a soccer corpus, which is kept up-to-date on a daily basis. Linguistic annotation is based in finite-state techniques and unification-based algorithms. It implements basic grammars for the annotation of people, locations, numerals and date and time expressions. On the top, rules for extracting soccer-specific entities, such as soccer players, teams and tournaments are implemented. Finally, data are transformed...
into ontological facts, by means of tabular processing (applying wrapper-like techniques) and text matching (by means of F-logic structures specified in a declarative form).

Other systems like KnowItAll (Etzioni et al., 2005) rely on the redundancy of the Web to perform a bootstrapped information extraction process iteratively. The confirmation of the correctness of the obtained information is requested to the user to re-execute the process. KnowItAll is self-supervised; instead of using hand-tagged training data, the system selects and labels its own training examples and iteratively bootstrap its learning process. For a given relation, the set of generic patterns is used to automatically instantiate relation-specific extraction rules, which are then used to learn domain-specific extraction rules. The rules are applied to Web pages identified via search-engine queries, and the resulting extractions are assigned a probability using information-theoretic measures derived from search engine hit counts. Next, KnowItAll uses frequency statistics computed by querying search engines to identify which instantiations are most likely to be members of the class. KnowItAll is relation-specific in the sense that it requires a laborious bootstrapping process for each relation of interest, and the set of relations has to be named by the human user in advance. This is a significant obstacle to open-ended extraction because unanticipated concepts and relations are often encountered while processing text.

Completely automatic and unsupervised annotation systems are scarce. SemTag (Dill et al., 2003) performs automated semantic tagging from large corpora based on the Seeker platform for text analysis and tags a large number of pages with the terms included in a domain ontology named TAP. This ontology contains lexical and taxonomic information about music, movies, sports, health, and other issues, and SemTag detects the occurrence of these entities in Web pages. It disambiguates the sense of the words using neighbour tokens and corpus statistics, picking the best label for a token. KIM (Kiryakov et al., 2004) is another example of unsupervised domain-independent system. It scans documents looking for entities corresponding to instances in its input ontology.

The system proposed in Zavitsanos et al. (2010) follows a sequence of annotation steps. First, all the ontology concepts (and their stems) are directly searched on the text. In a second stage, WordNet is used to obtain synonyms of the ontology concepts, and the system looks for them in the analysed text. Later, the authors propose the use of a semantic similarity measure (path length in WordNet) which is computed between all the words in the text and all the ontology concepts. Finally, another similarity measure based on Wikipedia is also considered. Since only words contained in the ontology or those corresponding to their synonyms or semantically similar ones are annotated, this work is limited to the coverage offered by the input knowledge repositories. Named entities or recently minted terms which cannot be found in Wikipedia, for example, are unlikely to result in any annotation.

Another interesting annotation application is presented in Michelson and Knoblock (2007). In this case, authors use a reference set of elements (e.g., on-line collections containing structured data about cars, comics or general facts) to annotate ungrammatical sources like texts contained in posts. First of all, the elements of those posts are evaluated using the TF-IDF metric. Then, the most promising tokens are matched with the reference set. In both cases, limitations may be introduced by the availability and coverage of the background knowledge (i.e., ontology or reference set).

From the applicability point-of-view, Pankow (Cimiano et al., 2004) is the most promising system. It uses a range of well-studied syntactic patterns to mark-up candidate phrases in Web pages without having to manually produce an initial set of marked-up Web pages, and without depending on previous knowledge. Its context-driven version, C-Pankow (Cimiano et al., 2005), improves the first by reducing the number of queries to the search engine. However, the final association between text entities and a domain ontology is not addressed.

3. Background

This section discusses the basic characteristics of the working environment and presents the main existing techniques (lexical patterns and Web-scale statistics) in which the proposed method relies.

3.1. The Web as a tool to assist the learning process

When aiming to develop an unsupervised system dealing with textual data, one cannot rely on experts opinions to validate the results (e.g. to evaluate which extracted terms should be mapped to the input ontology or which is the best mapping for a given term). However, one can profit from predefined knowledge to assess the adequacy of the extracted conclusions. To be domain-independent, however, no domain-specific knowledge structures or corpora can be used. Hence, to support our proposed method we rely on the Web as a domain-independent corpus which can aid the process of learning entity features.

The Web has many interesting characteristics in comparison with other information sources. As stated in Brill (2003), many classical techniques for information extraction and knowledge acquisition present performance limitations because they typically rely on a limited corpus. According to Brill (2003), the Web is the biggest repository of information available. In fact, the amount and heterogeneity of information in the Web is so high that it can be assumed to approximate the real distribution of information in the world (Cilibrasi and Vitányi, 2006). Moreover, the high redundancy of information, derived from the huge amount of available resources, can represent a measure of its relevance, i.e., the more times a piece of information appears on the Web, the greater its relevance (Brill, 2003; Ciravegna et al., 2003; Etzioni et al., 2004; Rosso et al., 2005).

In other words, we can give more confidence to a fact that is enounced by a considerable amount of independent sources. Thanks to the aforementioned characteristics, the Web has demonstrated its validity as a corpus to assist learning processes (Etzioni et al., 2004; Sánchez et al., 2009; Velásquez et al., 2011).

In our work the Web is used as a general tool in which the feature extraction process relies to assess the relevancy of potential extractions found in an input document (Section 4.3), and to improve the recall of the feature annotation towards concepts included in the input ontology (Section 4.4). To this end, Web-scale information distribution is exploited to assist the learning process and to extract robust conclusions.

3.2. Learning techniques

In the following, the main techniques in which our work relies to extract potential features and assess their relevancy are presented. They are focused on linguistic patterns and Web-scale statistics.

3.2.1. Linguistic patterns

The problem of semantically annotating/classifying/matching features extracted from text to their formal semantics can be simplified to the problem of discovering the ontological concept of which the discovered feature is an instance (e.g., Barcelona is a city). In essence, instance–concept relationships define a taxonomical link which should be discovered.

There exist many approaches for detecting this kind of relations. As stated in Cimiano et al. (2004), three different learning paradigms can be exploited. First, some approaches rely on the document-based notion of term subsumption (Sanderson and
Table 1

<table>
<thead>
<tr>
<th>Pattern</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Such NP as (NP,)* (and/or) NP</td>
<td>Such countries as Poland</td>
</tr>
<tr>
<td>NP ( ) such as (NP,)* (and/or) NP</td>
<td>Cities such as Barcelona</td>
</tr>
<tr>
<td>NP ( ) including (NP,)* (and/or) NP</td>
<td>Capital cities including London</td>
</tr>
<tr>
<td>NP ( ) specially (NP,)* (and/or) NP</td>
<td>Science fiction films, specially Matrix</td>
</tr>
<tr>
<td>NP ( ) (and/or) other NP</td>
<td>The Sagrada Familia and other churches</td>
</tr>
</tbody>
</table>

Croft, 1999). Secondly, some researchers claim that words or terms are semantically similar to the extent to which they share similar syntactic contexts (Bisson et al., 2000; Caraballo, 1999). Finally, several researchers have attempted to find taxonomic relations expressed in texts by matching certain patterns associated to the language in which documents are presented (Ahmad et al., 2003; Berland and Charniak, 1999).

Pattern-based approaches are heuristic methods using regular expressions that have been successfully applied in information extraction (Sánchez, 2010; Sánchez and Isern, 2011). The text is scanned for instances of distinguished lexical-syntactic patterns that indicate a relation of interest. This is especially useful for detecting specialisations of concepts that can represent is-a (taxonomic) relations (Hearst, 1992; Sánchez and Moreno, 2008b) or individual facts (Etzioni et al., 2005). Table 1 shows patterns proposed by M. Hearst to detect taxonomical relationships (NP stands for a noun phrase).

However, the quality of pattern-based extractions can be compromised by the problems of de-contextualisations and ellipsis. For example, de-contextualisations can easily be found in sentences like “There are several newspapers sited in big cities such as El Pais and El Mundo”; without a more exhaustive linguistic analysis we might erroneously extract El Pais and El Mundo as instances of “city” instead of “newspapers”. Hence, a way to validate the correctness of the detected relationships is needed in order to avoid depending solely on individual observations (which can be affected by natural language ambiguity).

Another limitation of pattern-based approaches is the fact that they usually present a relatively high precision but typically suffer from low recall due to the fact that the patterns are rare in reduced corpora (Cimiano et al., 2004). Fortunately, as it was stated in Section 4.5.2, this data sparseness problem can be tackled by exploiting the Web (Buitelaar et al., 2004; Velardi et al., 2005) as a corpus.

3.2.2. Web-scale statistics

As stated above, pattern-based analysis is hampered by the fact that conclusions are extracted from individual observations. To base the conclusions on a wider evidence, the use of statistical measures about information distribution (term occurrence and co-occurrence, in our case) in a large enough corpus have aided in the past (Lin, 1998). To avoid data sparseness problems that may be derived from statistics computed from reduced corpora, some authors like Brill (2003) have demonstrated the convenience of using a corpus as big as the Web to improve the quality of classical statistical methods.

However, the analysis of such an enormous repository for computing statistics is, in most cases, impracticable. To tackle this problem, some researchers (Cilibari and Vitányi, 2006; Cimiano et al., 2004; Etzioni et al., 2005) have used Web search engines to obtain good quality and relevant statistics.

To assess the degree of relatedness between a pair of terms (in our case, for example, between an extracted feature and its potential conceptualization), term collocation measures such as the well-known Pointwise Mutual Information (PMI, Eq. (1)) (Church et al., 1991) can be used. PMI statistically assesses the relation between two terms \((a, b)\) as the conditional probability of \(a\) and \(b\) co-occurring within the text

\[
PMI(a, b) = \log_2 \frac{p(ab)}{p(a)p(b)} \tag{1}
\]

here \(p(ab)\) is the probability that \(a\) and \(b\) co-occur. If \(a\) and \(b\) are statistically independent, then the probability that they co-occur is given by the product \(p(a)p(b).\) If they are not independent, and they have a tendency to co-occur, then \(p(ab)\) will be greater than \(p(a)p(b).\) Therefore the ratio between \(p(ab)\) and \(p(a)p(b)\) is a measure of the degree of statistical dependence between \(a\) and \(b.\)

To compute robust PMI values from the Web, term probabilities can be approximated to the chance of retrieving each term from the Web, when asking a search engine. Concretely, Eq. (2) approximates the PMI score by assessing term occurrence and co-occurrence probabilities as the hit count provided by a search engine (Turney, 2001)

\[
PMI_{hit}(a, b) = \log_2 \frac{hits(ab)}{hits(a)hits(b)} \tag{2}
\]

4. A novel ontology-based feature extraction method

This section provides a thorough description and formalization of the proposed methodology, whose aim is to discover those features modelled in an input ontology that can be found in a textual document describing an entity. The general algorithm is introduced in Section 4.1, and the following three sections detail its basic components. Sections 4.5 and 4.6 explain its adaptation to the analysis of plain text documents and semi-structured resources (Wikipedia articles). Section 4.7 analyses the temporal cost of the method according to the type of input.

4.1. General algorithm

Algorithm 1 provides a high-level pseudo-code description of the proposed algorithm. Table 2 contains the explanation of the

Table 2

<table>
<thead>
<tr>
<th>Element</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ae (analysed entity)</td>
<td>Name of the real entity which is being analysed (e.g. Barcelona)</td>
</tr>
<tr>
<td>wd (web document)</td>
<td>Web document which contains the information about the ae.</td>
</tr>
<tr>
<td>do (domain ontology)</td>
<td>Domain ontology used in order to specify which kind of concepts will be taken into account during the feature extraction process</td>
</tr>
<tr>
<td>pd (parsed document)</td>
<td>Web document parsed in a readable format for the system</td>
</tr>
<tr>
<td>PNE (Potential Named Entities)</td>
<td>List of potential Named Entities (entities extracted from the text that have not been checked and filtered yet)</td>
</tr>
<tr>
<td>NE (Named Entities)</td>
<td>List of selected PNEs, which are those with a relevancy score higher than the NE_THRESHOLD</td>
</tr>
<tr>
<td>SC (Subsumer Concepts)</td>
<td>Abstractions of collections of real entities which share common characteristics. Subsumer concepts and their abstractions are taxonomically related. For example, the subsumer concept of the real entities The Sagrada Familia and St. Peter’s Basilica could be basilica.</td>
</tr>
<tr>
<td>OC (Ontological Classes)</td>
<td>List of all the classes (i.e., concepts) modelled in the input ontology</td>
</tr>
<tr>
<td>SOC (Subsumer Ontological Classes)</td>
<td>Subset of the ontological concepts that correspond to the subsumer concepts of a NE</td>
</tr>
<tr>
<td>ac (annotated class)</td>
<td>Subsumer ontological concept which represents the final annotation of the analysed NE</td>
</tr>
</tbody>
</table>
basic elements of the algorithm. Several of the proposed functions are abstract and can be overwritten in order to adapt the analysis to different kinds of inputs (plain text documents or semi-structured ones). Moreover, since the feature extraction process is guided by the input ontology, by using different domain ontologies (e.g. in Medicine (Spackman, 2004) or Chemical engineering (Morbach et al., 2007)), the system is able to adapt the feature extraction process to the domain of interest. As inputs, the system receives the document to be analysed (e.g. a text describing a city) and the ontology detailing the features to be extracted (e.g. tourism activities or points of interest).

Algorithm 1. Ontology-based feature extraction method.

1 OntologyBasedExtraction(WebDocument wd, String ae, DomainOntology do){
2 /* Document Parsing */
3 pd:=parse_document(wd)
4
5 /* Extraction and selection of Named Entities from Document */
6 PNE:=extract_potential_NEs(pd)
7 ∀ pne ∈ PNE { 8 if NE_Score(pne, ae) > NE_THRESHOLD{ 9 NE:=NE ∪ pne, } 10 } 11 }
12 /* Retrieval of potential subsumer concepts for each NE*/
13 ∀ nei ∈ NE{ 14 SC:=extract_subsumer_concepts(nei ) 15 nei:=add_subsumer_concepts_list(SC ) 16 }
17 }
18 /* Annotation of NEs with ontological classes */
19 OC:=extract_ontological_classes(do)
20 ∀ ne ∈ NE { 21 /* Retrieval of Subsumer Ontological Classes (i.e. potential annotations) for each Subsumer Concept of each NE*/
22 SC:=get_subsumer_concepts_list(nei )
23 /* Application of direct matching */
24 SO:=extract_direct_matching(OC, SC)
25 /* If direct matching fails, semantic matching is applied */
26 if |SO| = 0 { 27 SO:=extract_semantic_matching(OC, SC, ne, ae)
28 } 29 /* if a similar ontological class is found, the best annotation is chosen and the annotation is performed */
30 if |SO| > 0 { 31 SO:=SOC_Score(SOC, ne, ae) 32 ac:=select_SOC_max_score(SOC, AC_THRESHOLD)
33 ne := add_annotation(ac)
34 }
35 }
36 return NE
37 }

In order to discover the relevant features of an entity, we focus on the extraction and selection of Named Entities (NEs) found in the text. A NE is a Noun Phrase which refers to a real world entity and can be considered as an instance of a conceptual abstraction (e.g. Barcelona is a NE and an instance of the conceptual abstraction city). Due to their concrete nature, it is assumed that NEs describe, in a less ambiguous way than general words, the relevant features of a particular entity (Abril et al., 2011).

To select which of the detected NEs are the most related to the analysed entity, a relevance-based analysis relying on Web co-occurrence statistics is performed. Afterwards, the selected NEs are matched to the ontological concepts of which they can be considered instances. In this manner the extracted features are presented in an annotated fashion, easing the posterior application of semantically-grounded data analyses. In the following subsections, the main algorithm steps are described in detail.

4.2. Document parsing

The first step is to parse the input Web document (line 3) which is supposed to describe a particular real world entity, from now on ae. The Parse_document function depends on the kind of document that is being analysed. If it is a HTML document, then it is necessary to extract raw text from it by means of HTML parsers which are able to remove headers, templates, HTML tags, etc. Otherwise, if the document comes from a semi-structured source such as Wikipedia, then ad-hoc tools are used to filter and extract the main text.

4.3. Named entity detection

This step consists in extracting relevant named entities from the analysed document. The function extract_potential_NEs (line 6) returns a set of Named Entities (PNE). The implementation of this function depends on the type of input, as will be described in Sections 4.5 and 4.6. However, as stated above, only a subset of the elements of PNE really describe the main features of ae; the rest of the elements of PNE could introduce noise because they may be unrelated to the analysed entity (they just happen to appear in the Web page describing the entity but are not part of its basic distinguishing characteristics). Thus, it is necessary to have a way of separating the relevant NEs from the irrelevant ones (NE filtering, line 8). To do that, a Web-based co-occurrence measure that tries to assess the degree of relationship between ae and each NE is used. Concretely, a version of the Pointwise Mutual Information (PMI, introduced in 3.2.2) relatedness measure adapted to the Web is computed (Church et al., 1991)

\[ NE_{SCORE}(pne, ae) = \frac{\text{hits}(pne, ae)}{\text{hits}(pne)} \] (3)

In the NE_{SCORE} (Eq. (3)), concept probabilities are approximated by Web hit counts provided by a Web search engine as proposed by Turney (2001). Since the intention of the algorithm is to rank a set of choices –PNEs- for a given domain (i.e. the ae) the hits(ae) term can be dropped because it has the same value for all choices. NEs that have a score exceeding an empirically determined threshold (NE_THRESHOLD, line 8) are considered as relevant, whereas the rest are removed. The value of the threshold determines a compromise between the precision and the recall of the system, as will be shown in the evaluation results presented in Section 5.1.

4.4. Semantic annotation

In this work, the aim of the semantic annotation step is to match the extractions found in a text describing an entity with the appropriate classes contained in the input ontology, which models the features of interest.

Some approaches have been proposed in this area. One way to assess the relationship between two terms (which, in our case, would be a NE and an ontology class) is to use a general thesaurus like WordNet to compute a similarity measure based on the number of semantic links among them. However, those measures are
hampered by WordNet’s limited coverage of NEs and, in consequence, it is usually not possible to compute the similarity between a NE and an ontological class in this way. There are approaches that try to discover automatically taxonomic relationships (Bisson et al., 2000; Sanderson and Croft, 1999), but they require a considerable amount of background documents and linguistic parsing. Finally, another possibility is to compute the co-occurrence between each NE and each ontological class using Web-scale statistics (Turney, 2001), but this solution is not scalable because of the huge amount of required queries (Cimiano et al., 2005).

The method proposed in this paper employs this last technique, but introducing a previous step that reduces the number of queries to be performed. To do so, the semantic annotation step is divided in two parts: the discovery of potential subsumer concepts (line 14) and their matching with the ontology classes (lines 19–38).

### 4.4.1. Discovering potential subsumer concepts

This first stage is proposed in order to minimise the number of queries (NE, ontology class) to be performed by the final statistical assessor. It may be noticed that the problem of semantic annotation is to find a bridge between the instance level (i.e., a NE) and the conceptual level (i.e., an ontology concept for which the NE is an instance). As stated in Section 3.2.1 NEs and concepts are semantically related by means of taxonomic relationships. Thus, the way to go from the instance level to the conceptual level is by discovering taxonomic abstractions, which are represented by subsumer concepts. So, in this stage, the aim is to automatically discover possible subsumer concepts for each NE.

This is done by means of the function `extract_subsumer_concepts`, which depends on the kind of input document (see Sections 4.5 and 4.6 for specific details). As a result, we obtain a set of subsumer concepts for each NE (e.g., the subsumer concepts of Porsche could be car, automobile, auto, motorcar and machine). Notice that those concepts are abstractions of the NE and they do not depend on any ontology. This means that subsumer concepts do not necessarily match with ontological classes. However, at this point the algorithm is working at a conceptual level (rather than at the instance level) and thus it is possible to match NEs with ontological classes more easily and efficiently in the following step, as detailed in the next section.

### 4.4.2. Matching subsumers to ontological classes

This stage tries to discover matches between the subsumer concepts of a NE and the ontological classes. To do so, two different cases are considered: Direct Matching and Semantic Matching.

#### 4.4.2.1. Direct matching

First, the system tries to find a direct match between the subsumers of a NE and the ontological classes. This phase begins with the extraction of all the classes contained in the domain ontology (line 19). Then, for each Named Entity `ne`, all its potential subsumer concepts (`sc ∈ SC`) are compared against each ontology class in order to discover lexically similar ontological classes (`soc ∈ SOC`, lines 23–25), i.e., classes whose name matches the subsumer itself or a subset of it (e.g., if one of the potential subsumers is Gothic cathedral, it would match an ontology class called Cathedral).

A stemming algorithm (Porter, 1997) is applied to both the `sc` and the ontology classes in order to discover terms that have the same root (e.g., city and cities). If one (or several) ontology classes match with the potential subsumers, they are included in `SOC` as candidates for the final annotation of `ne`. This direct matching step is quite easy and computationally efficient; however, its main problem is that, in many cases, subsumers do not appear as ontological classes with exactly the same name. As a result, potentially good candidates for annotation are not discovered.

#### 4.4.2.2. Semantic matching

This step faces the non-trivial matching between semantically similar subsumers and ontological classes that, because they were represented with different textual labels, could not be discovered in the previous stage. Details on how this task is performed are given in Algorithm 2, which specifies what it is done in line 28 of Algorithm 1.

**Algorithm 2.** Semantic matching method (line 28, algorithm 1).

```plaintext
1 extract_semantic_matching(OC, SC, ne, ae) {
2 ∀ sc ∈ SC {
3 SYNSETS := get_wordnet_synsets(sc) {
4 if |SYNSETS| = 0 { return }
5 else if |SYNSETS| = 1 {
6 RELATED_TERMS ← get_related_terms(synset0)
7 }
8 else /* when the sc has more than 1 synset, disambiguation is needed */
9 CONTEXT := get_context(ne)
10 CONTEXT := CONTEXT + get_web_snippets(ne, ae)
11 ∀ contextı ∈ CONTEXT {
12 /* the similarity between the context and each synset is calculated */
13 ∀ synsetk ∈ SYNSET {
14 similarity := get_cosine_distance(contextı, synsetk)
15 update_average_synset(synsetı, similarity)
16 }
17 }
18 }
19 disambiguated_synset := get_synset_max_average(SYNSETS)
20 RELATED_TERMS := get_related_terms(disambiguated_synset)
21 }
22 }
23 return extract_direct_matching(OC, RELATED_TERMS)
24 }
```

The semantic matching step is performed when the direct matching has not produced any result. Its main goal is to increase the number of elements in SOC, so that the direct matching can be re-applied with a wider set of terms. The new potential subsumers are concepts semantically related to any of the initial subsumers (synonyms, hypernyms and hyponyms). As the algorithm works at a conceptual level, WordNet (Fellbaum, 1998) has been used to obtain these related terms and to increase the SOC set.

The main problem of semantic matching is that a word may be polysemous and, before extracting the related concepts from WordNet, it is necessary to discover to which sense it corresponds (i.e., a semantic disambiguation step must be performed to choose the correct synset). To do so, one possible solution is to use the context (i.e., the sentence from which `ne` was extracted, line 9 of Algorithm 2) but, usually, that is not enough to disambiguate the meaning. To minimise this problem, the Web is used to assist the disambiguation.

We propose a Web-based approach that combines the contexts of Named Entities, WordNet definitions and the cosine distance. The Web is used in this case to find new evidences of the relationship between the `ne` and the `ae` in order to increase the contexts (where those terms occurs together) that enable a better disambiguation of the meaning of polysemous words. First, the algorithm retrieves the contexts in which the NE appears (line 10 of Algorithm 2), framed in the scope of the original `ae` (by querying a search engine for common appearances of `ae` and `ne`). Then, the set of Web snippets returned by the search engine is analysed. Web snippets (see Fig. 1) are used instead of complete documents because they precisely
contain, in a concise manner, the context of the occurrence of the web query. Several snippets (e.g. 10) can be obtained with a single Web query, so their analysis is much more efficient than the one of complete Web resources.

Then, the system calculates the cosine distance between each snippet and every WordNet synset of the element of SC (lines 11 to 18 of Algorithm 2). The aim is to assess which of the senses of each subsumer concept is the most similar to the context in which the NE appears. The synset with the highest average value is finally selected as the appropriate conceptualization of the NE appearing. The synset with the highest average value is the most similar to the context in which the NE appears. The synset with the highest average value is the most similar to the context in which the NE appears. The synset with the highest average value is the most similar to the context in which the NE appears.

Let us consider an example to illustrate this method. Table 3 depicts the input data of the problem. The first three rows are the analysed entity, the named entity and a subsumer concept. The last three rows show the query performed in order to retrieve Web snippets and the two WordNet synsets of the subsumer concept.

Table 4 shows a subset of the retrieved snippets, which represent contexts of the NE, and the cosine distance between the context and each of the subsumer’s synsets. Synset 1 obtains the highest score, so it will be selected as the most appropriate conceptualization of the SC and, hence, its synonyms, hyponyms and hypernyms will be retrieved in the semantic matching step. In this example, minister, church and church building are used as terms to expand the direct matching between the SC cathedral and the ontology classes.

4.4.3. Class selection

After applying the matching process between subsumers and ontological classes, a final assessor decides if the selected matching is adequate enough (if only one matching was found) or which of the matchings is the most appropriate, if several ones were obtained (line 32 of Algorithm 1). In this case, a Web-based statistical measure is applied again in order to choose the most representative one (Class Selection, lines 33–34 of Algorithm 1). Notice that, at this stage, the number of matches between NEs and ontological classes is low compared with all the possible combinations between NEs and ontological classes (as discussed in Section 4.4.1). As a result, a much lower amount of queries is needed to select the final annotation. This shows the benefits of exploiting automatically discovered subsumer concepts as a bridge to enable more direct semantic annotations.

The selection is based on the degree of relatedness between the Named Entity and each element of the matched ontological classes in SOC, assessed again with the Web-based version of PMI introduced in Section 3.2.2. However, it must be noted that the elements of SOC can also be polysemous, and can be referring to different concepts depending on the context (line 33 of Algorithm 1). So, in Eq. (4), the analysed entity ae has been introduced to contextualise the relationship of each element of SOC with ae.

The score (Eq. (4)) computes the probability of the co-occurrence of the named entity ne and each ontology class proposed for annotation soc from the Web hit count provided by a search engine when querying these two terms (contextualised with ae). Finally, the annotation with the highest score which exceeds the AC_THRESHHOLD (lines 34–35 of Algorithm 1) is annotated. If no elements in SOC exceed the threshold, the NE remains unannotated. This will indicate that, even though the NE and its corresponding SC may be correct, there are not enough evidences (statistically assessed from the Web) to support the annotation in the context of the domain defined by the input ontology.

4.5. Extraction of features from raw texts

So far, the generic feature extraction algorithm has been presented. This section discusses how the functions extract_potential_NEs (line 6 of Algorithm 1) and extract_subsumer_concepts (line 14 of Algorithm 1) have been implemented in order to apply it to raw text.

4.5.1. Named entities detection

The main problem related with the detection of NEs from raw text is the fact that they are unstructured and unlimited by nature (Sánchez et al., 2010). This implies that, in most cases, NEs are not contained in classical repositories like WordNet due to its potential size and its dynamic character. Different approaches have been proposed in the field of NE detection. Roughly, they can be divided into supervised and unsupervised methods.

Supervised approaches rely on a specific set of extraction rules learned from pre-tagged examples (Fleischman and Hovy, 2002;
Stevenson and Gaizauskas, 2000), or predefined knowledge bases such as lexicons and gazetteers (Mikheev and Finch, 1997). However, the effort required to assemble large tagged sets or lexicons binds the NE recognition to either a limited domain (e.g., medical imaging), or a small set of predefined, broad categories of interest (e.g., persons, countries, organisations, products), hampering the recall (Pasca, 2004).

In unsupervised approaches like (Lamparter et al., 2004), it has been proposed to use a thesaurus as background knowledge (i.e., if a word does not appear in a dictionary, it is considered as a NE). Despite the fact that this approach is not limited by the coverage of the thesaurus, misspelled words are wrongly considered as NEs whereas correct NEs composed by a set of common words are rejected, providing inaccurate results.

Other approaches take into consideration the way in which NEs are presented in a specific language. Concretely, languages such as English distinguish proper names from other nouns through capitalisation. The main problem is that basing the detection of NEs on individual observations may produce inaccurate results if no additional analyses are applied. For example, a noun phrase may be arbitrary capitalised to stress its importance or due to its placement within the text. However, this simple idea, combined with linguistic pattern analysis, as it has been applied by several authors (Cimiano et al., 2004; Downey et al., 2007; Pasca, 2004), provides good results without depending on manually annotated examples or specific categories.

We have defined the following unsupervised, domain-independent method to detect NEs when dealing with raw text (i.e., to implement extract_potential_NEs, line 6). First, a natural language processing parser is used. Concretely, the four modules of the OPENNLP1 parser (Sentence Detector, Tokenizer, Tagging and Chunking) are applied in order to analyse syntactically the input text of the Web document. Then, all the detected Noun Phrases (NP) which contain one or more words beginning with a capital letter are considered as a Potential Named Entities (PNE).

Table 5 shows an example of the extracted PNEs from the first fragment of text of the Wikipedia article about Tarragona.

<table>
<thead>
<tr>
<th>Detected sentences</th>
<th>Extracted PNE</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NP Tarragona/EX]</td>
<td>Tarragona</td>
<td>Yes</td>
</tr>
<tr>
<td>[NP Catalonia/NN]</td>
<td>Catalonia</td>
<td>Yes</td>
</tr>
<tr>
<td>[NP Spain/NNP]</td>
<td>Spain</td>
<td>Yes</td>
</tr>
<tr>
<td>[NP Sea/NNP]</td>
<td>Sea</td>
<td>No</td>
</tr>
<tr>
<td>[NP Tarragonès/VBZ]</td>
<td>Tarragonès</td>
<td>Yes</td>
</tr>
<tr>
<td>[NP the/VBZ/Vegueria/NNPS]</td>
<td>The Vegueria</td>
<td>No</td>
</tr>
</tbody>
</table>

4.5.2. Discovering potential subsumer concepts

To extract subsumer concepts of NEs from raw text, it is necessary to look for linguistic evidences stating their implicit taxonomical relationship. As discussed in Section 3.2.1, we use Hearst's taxonomic linguistic patterns, which have proved their effectiveness to retrieve hyponym/hypernym relationships (Hearst, 1992). The Web is used to assist this learning process, looking for pattern matches that provide the required linguistic evidences.

To do so, the system constructs a Web query for each NE and each pattern. Each query is sent to a Web search engine, which returns as a result a set of Web snippets. Finally, all these snippets are analysed in order to extract a list of potential subsumer concepts (i.e., expressions that denote concepts of which the NE may be considered an instance).

Table 6 summarises the linguistic patterns that have been used (CONCEPT represents the retrieved potential subsumer concept and NE the Named Entity that is being studied).

4.6. Extracting features from semi-structured documents

In this work, Wikipedia has been used to show the applicability of the proposed method when applied to semi-structured resources. Wikipedia has some particularities which can ease the information extraction when compared with raw text. This work focuses on the exploitation of internal links and category links. The first ones represent connections between terms that appear in a Wikipedia article and other articles that describe them. This is an indication of the fact that the linked term represents a distinguished entity by itself, easing the detection of NEs. On the other hand, category links group different articles (corresponding to entities) in areas that are related in some way and give articles a kind of categorisation. Wikipedia’s category system can be thought of as consisting of overlapping trees. Hence, Wikipedia categories can be considered as rough taxonomical abstractions of entities and can be exploited to aid the annotation process.

This section details how the functions extract_potential_NEs (line 6) and extract_subsumer_concepts (line 14) have been adapted to take advantage of these semi-structured data.

4.6.1. Named entities detection

In order to take profit of the manually created Wikipedia link structure, those words tagged with internal links have been considered as potential named entities (PNE, Table 7). In this manner, the analysis is reduced to only those terms that are likely to correspond to distinguished entities (that have its own Wikipedia article).

The problem of the PNEs extracted from the internal links are that, on one hand, not all of them are directly related with the analysed entity (ae) and, on the other hand, only a subset of PNE are real NEs.

In order to illustrate these problems, the following fragment of text extracted from Wikipedia corresponding to the Barcelona article is examined. "Barcelona is the capital and the most populous city of Catalonia and the second largest city in Spain, after Madrid, with a population of 1,621,537 within its administrative limits on a land area of 101.4 km²." In this text, there are four terms internally linked with other Wikipedia articles. Three of them are NEs (Catalonia, Spain and Madrid) and they represent instances of cities-regions/countries, whereas the other one is a common noun which represents a concept (capital). Moreover, Madrid is bringing information of general purpose that is not directly related with Barcelona and, in consequence, it is not a relevant feature for describing the entity Barcelona.

Due to these potential problems, the set of extracted PNEs (Table 7) has to be filtered by means of the NE score presented in the general algorithm (see Section 4.3), as it is done with the PNEs obtained from the analysis of raw text. In this case, however, the algorithm starts with a pre-processed set of NEs whose reliability is supported by the community of Wikipedia contributors.

4.6.2. Discovering potential subsumer concepts

In order to extract potential subsumer concepts for each named entity, Wikipedia category links have been used. The idea is to consider Wikipedia categories of articles describing a NE as potential subsumers of the NE. In this manner it is possible to obtain, in a very direct and efficient manner, the set of subsumers that enable the matching of the NE with ontological classes (Table 8).
A problem of these categories, however, is the fact that they are, in many cases, too complex or concrete to be modelled in an ontology. For example, the *Sagrada Familia* article is categorised as *Antoni Gaudí buildings*, *Buildings and structures under construction*, *Churches in Barcelona*, *Visitor attractions in Barcelona*, *World Heritage Sites in Spain*, *Basilica churches in Spain*, etc. These categories are too complex because they involve several concepts or even mix concepts with other NEs. To tackle this problem, we syntactically analyse each category to detect the basic concepts to which it refers. For example in “Churches in Barcelona” the key concept is “Churches” and in “Buildings and structures under construction” there are two important concepts: “Buildings” and “Structures”. To extract the main concepts of each sentence a natural language parser has been used, and all the Noun Phrases have been extracted.

Another limitation of the Wikipedia categories is the fact that they do not always contain enough concepts to perform the matching among them and ontological classes, or those are too concrete to be included in an ontology (and, hence, to enable a direct matching). Fortunately, as mentioned before, Wikipedia categories are included in higher categories. So, we consider two upper-level categories to improve the recall of the ontology matching. As the Wikipedia categorisation does not define a strict taxonomy, it is not advisable to climb recursively in the category graph because the initial meaning could be lost.

### 4.7. Computational cost

The runtime of the proposed method mainly depends on the number of queries, since the Web search engine response time is several orders of magnitude higher (around 500 milliseconds) than any other offline analysis performed by our method (e.g. natural language processing, pattern matching, ontological processing, etc., taking a few milliseconds). There are five different steps in which queries are performed: NE detection, NE filtering, subsumer concepts extraction, semantic disambiguation and class selection. Since the response time of the Web search engine does not depend on the type and complexity of the query (but on the delay introduced by the Internet connection), the runtime for all query types is expected to be equivalent.

Both plain text and semi-structured text analyses have the same cost for NE filtering, semantic disambiguation and class selection. To rank NEs in the relevance filtering step, two queries are required by the NEψ function (4) to evaluate each NE. Hence, a total of 2\(n\) queries are performed where \(n\) represents the number of NEs. Class selection requires computing as many SOC\(_n\) scores (5) as candidates. Hence, being \(c\) the total number of class candidates, this results in 2\(c\) queries. In the semantic disambiguation stage, only one query is needed for each candidate to obtain the snippets relating each candidate with the analysed entity (i.e. \(c\) queries are performed in this step).

Hence, the difference in computational cost between plain text analyses and semi-structured ones is in the steps of NE detection and extraction of subsumer concepts. Although NE detection is different when analysing plain texts and semi-structured resources, neither of them needs to perform queries and its cost is considered constant for both approaches. Referring to the extraction of subsumer concepts, in the first approach six queries are performed to discover subsumer concepts by means of Hearst Patterns (6\(n\)), whereas in the second approach no queries are needed because SCs are directly extracted from the categories of the tagged entities.

In summary, considering the above expressions, the number of queries needed to analyse plain text is 2\(n+6n+2c+1c=8n+3c\), whereas only 2\(n+2c+1c=2n+3c\) are needed when dealing with Wikipedia articles. In both cases the proposed method scales linearly with respect to the number of NEs, but the much lower coefficient for the Wikipedia articles shows how the exploitation of their structure aids to improve the performance of the method.

### 5. Evaluation

This section presents the evaluation of the proposed method, which has been conducted in two directions. In the first part, given a specific evaluation scenario, it is presented a detailed picture of the influence of the different parameters involved in the learning process: thresholds, input ontology and document types. In the second part, in order to give a general view of the expected results, a battery of tests for different scenarios (i.e. different input documents and ontological domains) is presented.

In all cases, the results have been evaluated according to their precision and recall against ideal results provided by a human expert. To do so, the expert has been requested, for each input document representing an analysed entity (\(ae\) and ontology, to manually select which features found in the document directly or indirectly refer to concepts modelled in the ontology. As a result, a list of ideal features corresponding to concepts found in the ontology for each \(ae\) is obtained.

Then, recall is calculated by dividing the number of correct features discovered by our system (i.e. those also found in the ideal set of features) by the amount of ideal features stated by the experts.

<table>
<thead>
<tr>
<th>Pattern structure</th>
<th>Query</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONCEPT such as NE</td>
<td>“Such as Barcelona”</td>
<td>Cities such as Barcelona</td>
</tr>
<tr>
<td>Such CONCEPT as NE</td>
<td>“Such” as Spain</td>
<td>Such countries as Spain</td>
</tr>
<tr>
<td>NE and other CONCEPT</td>
<td>“Ebre and other”</td>
<td>Ebre and other rivers</td>
</tr>
<tr>
<td>NE or other CONCEPT</td>
<td>“The Sagrada Familia or other”</td>
<td>The Sagrada Familia or other monuments</td>
</tr>
<tr>
<td>CONCEPT especially NE</td>
<td>“Especially Tarragona”</td>
<td>World Heritage Sites especially Tarragona</td>
</tr>
<tr>
<td>CONCEPT including NE</td>
<td>“Including London”</td>
<td>Capital cities including London</td>
</tr>
</tbody>
</table>

### Table 6

Patterns used to retrieve potential subsumer concepts.

<table>
<thead>
<tr>
<th>Wikilinks</th>
<th>Correct?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acre</td>
<td>No</td>
</tr>
<tr>
<td>Antoni Gaudi</td>
<td>Yes</td>
</tr>
<tr>
<td>Arc de Triomf</td>
<td>Yes</td>
</tr>
<tr>
<td>Archeology Museum of Catalonia</td>
<td>Yes</td>
</tr>
<tr>
<td>Barcelona Cathedral</td>
<td>Yes</td>
</tr>
<tr>
<td>Barcelona Museum of Contemporary art</td>
<td>Yes</td>
</tr>
<tr>
<td>art</td>
<td></td>
</tr>
<tr>
<td>Barcelona Pavilion</td>
<td>Yes</td>
</tr>
<tr>
<td>Casa Batlló</td>
<td>Yes</td>
</tr>
</tbody>
</table>

### Table 7

Subset of extracted NEs from Barcelona Wikipedia article.
human expert

\[
\text{RECALL} = \frac{\text{Correct features}}{\text{Ideal features}}
\]

Precision is computed as the number of correct features (as above) divided by the total number of retrieved features by our system

\[
\text{PRECISION} = \frac{\text{Correct features}}{\text{Retrieved features}}
\]

5.1. Influence of input parameters

In this first battery of tests, we focused on documents describing cities (Barcelona and Canterbury), and ontologies related to touristic and geo-spatial features. First, the influence of the algorithm thresholds in the results has been studied. They enable to configure the system behaviour so that a high precision, a high recall or a balance between both of them can be achieved. Then, different types of input documents have been considered. Specifically, our method has been applied to Wikipedia articles, but analysing them as plain text and as Wiki-tagged documents. Finally, different input ontologies covering the same domain of knowledge have been used to evaluate the influence of the ontology design and coverage in the results. Two ontologies have been considered: a manually built Tourism ontology\(^3\), which models concepts related to different kinds of touristic points of interest typically found in Wikipedia articles, and a general ontology modelling geo-spatial concepts (Space\(^4\)) retrieved from the SWOOGLE\(^5\) search engine. A summary of their structure is shown in Table 9.

5.1.1. Learning thresholds

In this section, the influence of the thresholds used to filter named entities (NE_THRESHOLD) and to select annotations (AC_THRESHOLD) is studied. The Wikipedia article of Barcelona, taken as plain text and as a Wiki-tagged document, together with the general Space ontology have been used as input. Fig. 2 shows precision and recall figures when setting one of the two thresholds to 0 and varying the other one from 0 to 1 for the different types of input documents.

Results show that the AC_THRESHOLD has a larger influence in precision/recall. Notice that NE_THRESHOLD is related to the NE_Score, which is calculated by measuring the level of relatedness between the potential named entity and the analysed entity, whereas AC_THRESHOLD goes further by measuring the relatedness between the analysed entity, the potential named entity and the subsumer candidate to be annotated (SOC_Score). This fact implies that the second threshold is more restrictive because the relatedness involves three elements instead of two. Moreover, since the AC_THRESHOLD is considered after the NE_THRESHOLD, its purpose is twofold: (1) it measures the relatedness between the named entity and its subsumer candidate, facilitating the final annotation, and (2) it contextualises the ontology annotation in the domain of the analysed entity, so that it may filter unwanted named entities. This latter aspect is similar to the purpose of the NE_THRESHOLD, which is to select only those NE that are related to the analysed entity. Since all NEs selected according to the NE_THRESHOLD have to pass a second more restrictive threshold, this enables the system to drop irrelevant entities and those that are not related with concepts modelled in the input ontology.

However, NE_THRESHOLD is useful to drop some NE candidates to be analysed in latter stages and, hence, to lower the number of required Web queries associated to the evaluation of their subsumer concept candidates.

From a general perspective we observe an inverse behaviour for precision and recall and for both thresholds. Since the final goal of our method is to enable the application of data analysis methods (such as clustering) a high precision may be desirable, even at the cost of a reduced recall. In this case, more restrictive thresholds can be used. This aspect shows an important difference with related works in which no statistical assessor is used to filter extractions (Cimiano et al., 2005). Since no parameter tuning is possible, the precision of the final results may be limited by the fixed criterion for NE selection. Notice also that, analysing the document as plain text, a total of 202 named entities were detected from which only 146 were already annotated by Wikipedia (i.e. 72%). So, in comparison with methods like the one proposed by Zavitsanos et al. (2010), which limit the extraction to what it is already annotated in input knowledge bases (i.e. WordNet, Wikipedia, etc.), our method is potentially able to improve the extraction and annotation recall.

5.1.2. Plain text vs. Wiki-tagged document

In this second test, we picked up as case studies the Wikipedia articles that describe the cities of Barcelona and Canterbury, analysing them as plain text and as wiki-tagged documents. In all cases, the Space ontology was used. Again, different threshold values were set as shown in Fig. 3.

Figures show that precision tends to remain stable or to improve when taking into account the additional information

### Table 8

<table>
<thead>
<tr>
<th>Articles</th>
<th>Potential subsumer concepts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antoni Gaudí</td>
<td>1852 births, 1926 deaths, architects, roman</td>
</tr>
<tr>
<td></td>
<td>catholic churches, art nouveau architects, catalan</td>
</tr>
<tr>
<td></td>
<td>architects, spanish ecclesiastical architects,</td>
</tr>
<tr>
<td></td>
<td>modernisme architects, 19th century architects,</td>
</tr>
<tr>
<td></td>
<td>20th century architects, organic architecture,</td>
</tr>
<tr>
<td></td>
<td>people, reus...</td>
</tr>
<tr>
<td>Arc de Triomf</td>
<td>Triumphal arches, gates, moorish revival</td>
</tr>
<tr>
<td></td>
<td>architecture, 1888 architecture, public art stubs,</td>
</tr>
<tr>
<td></td>
<td>1888 works, architecture, public art, art stubs...</td>
</tr>
<tr>
<td>Archeology Museum of</td>
<td>Museums, archaeology museums, Sants-Montjuic</td>
</tr>
<tr>
<td>Catalonia</td>
<td></td>
</tr>
<tr>
<td>Barcelona Cathedral</td>
<td>Cathedrals, churches, visitor attractions, basilica</td>
</tr>
<tr>
<td></td>
<td>churches...</td>
</tr>
<tr>
<td>Barcelona Museum of Contemporary art</td>
<td>Museums, art museums, galleries, modern art</td>
</tr>
<tr>
<td></td>
<td>museums, modernist architecture, spots, richard</td>
</tr>
<tr>
<td></td>
<td>meier buildings, el raval, modern art...</td>
</tr>
<tr>
<td>Casa Batlló</td>
<td>Visionary environments, antoni gaudi buildings,</td>
</tr>
<tr>
<td></td>
<td>1907 architecture, world heritage sites, spain,</td>
</tr>
<tr>
<td></td>
<td>visitor attractions, eixample, passieg, gracia,</td>
</tr>
<tr>
<td></td>
<td>outsider art, 1907 works, 1900s Architecture,</td>
</tr>
<tr>
<td></td>
<td>edwardian architecture...</td>
</tr>
</tbody>
</table>

### Table 9

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Taxonomical depth</th>
<th>#classes</th>
<th>Root classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourism</td>
<td>5 levels</td>
<td>315</td>
<td>Administrative divisions, buildings, festivals, landmarks, museums, sports</td>
</tr>
<tr>
<td>Space</td>
<td>6 levels</td>
<td>188</td>
<td>Geographical features, geopolitical entities, places</td>
</tr>
</tbody>
</table>


provided by Wiki-tagged text, that is, the tagged entities and their associated Wiki-categories. On the contrary, recall is, in many situations, higher when analysing the input document as plain text. This is because when using plain text, the whole textual content is analysed. Hence, there are more possibilities to extract named entities that could correspond to representative features than when relying solely on Wiki-tagged entities. On the contrary, since Wiki entities are tagged by humans, the precision of their resulting annotations is expected to be higher than when performing the automatic NE detection proposed by our method. Moreover, for the Wiki-tagged text, humanly edited categories are used to assist the annotation, which may also contribute to provide more accurate results. In any case, results are close enough to consider the plain text analysis (i.e. both the NE detection and the subsumer candidate learning) almost as reliable as human annotations, at least for feature discovery purposes. It is important to note, however, that the analysis of Wiki-tagged text, as discussed in Section 4.7, is considerably faster than its plain text counterpart, since the number of analyses and queries required to extract and annotate entities is greatly reduced.

5.1.3. Input ontologies
The third test compares the results obtained using different ontologies to assist the feature extraction process of a given entity. In this case, the plain-text and wiki-tagged versions of the Barcelona article have been analysed using the input ontologies Space and Tourism. Again, results for different threshold values are shown in Fig. 4.

Results obtained for the plain-text version of the Barcelona article show a noticeably higher precision when using the Tourism ontology instead of the Space one. Since the former ontology was constructed according to concepts typically appearing in Wikipedia articles describing cities, it helps to provide more appropriate annotations since, even though extracted named entities are equal for both ontologies, the Tourism one is more suited to annotate those that are already tagged in the Wikipedia article. Moreover, since the Tourism ontology contains more concrete concepts than the more general-purpose Space one, language ambiguity that may affect the subsumer extraction stage and the posterior annotation is minimised, resulting in better precision. The downside is the lowered recall obtained with the Tourism ontology, since solely those named entities that are already tagged in the text (and that, hence, correspond to concepts modelled in the Tourism ontology) are likely to be annotated. Differences between the two ontologies when using Wiki-tagged text are much reduced. In this case, the fact that NEs correspond solely to those already tagged in the Wikipedia article produces more similar annotation accuracies. Recall is, however, significantly higher for the Tourism ontology, since NE subsumers, which correspond to Wikipedia categories in this case, are more likely to match directly (as described in Section 4.4.2.1) to concepts of an ontology that is specially designed to cover topics and entities usually referred in Wikipedia documents. This avoids the inherent ambiguity of the semantic matching process (described in Section 4.4.2.2) that is carried out when a direct matching is not found, which may hamper the results.

5.2. General evaluation
In this second part of the evaluation, general tests have been performed for several entities belonging to two well-differenced domains: touristic destinations and movies. For each domain, eight different entities/documents have been analysed, using their corresponding Wikipedia articles. For touristic destinations, the Tourism ontology introduced above has been used, whereas for movies, a film ontology retrieved from SWOOGLE has been used. This latter one models concepts related with the whole production process of a film, including directors, producers, main actors/actresses starring in the film, etc. The structure of both ontologies is summarised in Table 10.

Table 11 depicts the precision and recall obtained for the different cities analysed in the tourism domain, whereas Table 12 shows the results of the evaluation of films.

Comparing both domains, we observe slightly better results for the film domain, with averaged precision and recall figures of 76.9% and 63.2%, respectively. This is related to the lower ambiguity inherent to film-related entities, in comparison with touristic ones. For example, in the film domain, most NEs refer to

---

person names (directors, producers, stars), which are hardly ambiguous due to their concreteness. Hence, annotation to ontological concepts is usually correct. On the contrary, a NE such as “Barcelona” may refer either to a city or a football team, being both annotations almost equally representative; if both concepts are found in the ontology, the annotation is more likely to fail, in this case.

Analysing individual entities, we observe better precisions and recalls of English/American entities (e.g., American films like First Blood or British cities like London), which usually result in a larger amount of available Web resources for the corresponding entities and, hence, on more robust statistics.

As a general conclusion, considering recall figures, we observe that our method was able to extract, in both cases, more than half of the features manually marked by the domain expert. Moreover, precisions were around a 70–75% in average, which produce usable results for further data analysis. It is important to note that these results, which have been automatically obtained, suppose a considerable reduction of the human effort required to annotated textual resources.

6. Conclusions

One of the basic pillars of the Semantic Web concept is the idea of having explicit semantic information that can be used by intelligent agents in order to solve complex problems of Information Retrieval and Question Answering and to semantically analyse and catalogue the electronic contents. This fact has motivated the creation of new data mining techniques like semantic clustering (Batet et al., 2011), which are able to exploit semantics of data. However, these methods suppose that input contents were annotated in advance so that relevant features can be detected and interpreted.

This work aimed to extract and pre-process data from different kinds of inputs (i.e., plain textual documents, Web resources or...
semi-structured documents like Wikipedia articles) in order to generate the required semantically tagged data for the aforementioned semantic data analysis techniques. In order to reach our goals, several well-known techniques and tools have been used: natural language processing parsers have been useful to analyse texts and detect named entities, Hearst Patterns have been used to discover potential subsumer concepts of named entities, and Web-scale statistics complemented with co-occurrence measures have been calculated to score and filter potential named entities and to verify if the final semantic annotation of subsumer concepts is applicable. The main contribution in comparison with related works, it the proposal of a novel way to go from the named entity level to the conceptual level by using the Web as a general learning source, so that the recall of the annotation can be improved regardless of the coverage limitations of named entities presented by the input ontology or WordNet. This enables a final annotation that minimises the number of queries performed to

**Table 10**

Description of used ontologies.

<table>
<thead>
<tr>
<th>Ontology</th>
<th>Taxonomic depth</th>
<th># Classes</th>
<th>Main classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tourism</td>
<td>5 levels</td>
<td>315</td>
<td>Administrative divisions, buildings, festivals, landmarks, museums, sports</td>
</tr>
<tr>
<td>Film</td>
<td>4 levels</td>
<td>20</td>
<td>Person, product, company</td>
</tr>
</tbody>
</table>

**Table 11**

Evaluation results from Wikipedia descriptions of different cities using Tourism ontology.

<table>
<thead>
<tr>
<th>Articles</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cracow</td>
<td>80</td>
<td>36.3</td>
</tr>
<tr>
<td>London</td>
<td>75.8</td>
<td>66.6</td>
</tr>
<tr>
<td>New York City</td>
<td>69.2</td>
<td>75</td>
</tr>
<tr>
<td>Poznan</td>
<td>66.6</td>
<td>44.4</td>
</tr>
<tr>
<td>Saint Petersburg</td>
<td>66.6</td>
<td>57.1</td>
</tr>
<tr>
<td>Stockholm</td>
<td>73.6</td>
<td>56</td>
</tr>
<tr>
<td>Vancouver</td>
<td>45.4</td>
<td>50</td>
</tr>
<tr>
<td>Warsaw</td>
<td>79.1</td>
<td>70.3</td>
</tr>
</tbody>
</table>

Average 69.5 57.0

Fig. 4. Influence of domain ontologies.
the Web, since the final matching is performed at a conceptual level (Cimiano et al., 2004). Moreover, statistical assessors have been tuned to better assess the suitability of extraction and annotation at each stage, minimising the inherent language ambiguity of Web queries. Our method has been also designed in a general way so that it can be applied to different kinds of inputs and exploit semi-structured information (like Wikipedia annotation) to improve the performance. Finally, being unsupervised and domain-independent, the implemented methodology can be applied in different domains and without human supervision. 

The evaluation performed for different entities and domains produced usable results that, by carefully tuning the algorithm thresholds, reach the high precisions needed for applying them to data mining algorithms. The evaluation has also the benefits of using semi-structured inputs (Wikipedia articles), producing quality results with a lower computational cost.

As further work, it is a priority to study the quality of the extracted features when using them in semantic data mining algorithms (Batet et al., 2011; Martínez et al., 2012). Other important line of work is to study how to reduce the number of queries (e.g., using only a subset of Hearst Patterns) to Web search engines, as they are the slowest part of the algorithm and introduce a dependency on external resources. Finally, the annotation process could be extended in order to take profit of tools for concept detection and classification from heterogeneous data sources. Int. J. Hum.—Comput. Stud. 66, 759–788.


Table 12

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Articles</td>
<td>Precision (%)</td>
<td>Recall (%)</td>
</tr>
<tr>
<td>Batman Begins</td>
<td>75</td>
<td>52</td>
</tr>
<tr>
<td>First Blood (Rambo)</td>
<td>87.5</td>
<td>77.7</td>
</tr>
<tr>
<td>Inception</td>
<td>81.2</td>
<td>72.2</td>
</tr>
<tr>
<td>Matrix</td>
<td>87.5</td>
<td>57</td>
</tr>
<tr>
<td>Oldboy</td>
<td>57.1</td>
<td>66.6</td>
</tr>
<tr>
<td>Pulp Fiction</td>
<td>78</td>
<td>61</td>
</tr>
<tr>
<td>Requiem for a dream</td>
<td>60</td>
<td>50</td>
</tr>
<tr>
<td>The Fountain</td>
<td>88.8</td>
<td>69.2</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>76.9</strong></td>
<td><strong>63.2</strong></td>
</tr>
</tbody>
</table>