Ontology Extraction for Knowledge Reuse: The e-Learning Perspective

Matteo Gaeta, *Member, IEEE*, Francesco Orciuoli, Stefano Paolozzi, and Saverio Salerno

***Abstract*—Ontologies have been frequently employed in order to solve problems derived from the management of shared distributed knowledge and the efficient integration of information across different applications. However, the process of ontology building is still a lengthy and error-prone task. Therefore, a number of research studies to (semi-)automatically build ontologies from existing documents have been developed. In this paper, we present our approach to extract relevant ontology concepts and their relationships from a knowledge base of heterogeneous text documents. We also show the architecture of the implemented system and discuss the experiments in a real-world context.**

***Index Terms*—E-learning, knowledge acquisition, ontology extraction, ontology learning.**

I. Introduction

THE INFORMATION and communication technology community widely acknowledges the importance and usefulness of domain ontologies, particularly in relation to Semantic Web applications [4]. However, the promises of the Semantic Web are still far from being fully implemented. In this scenario, a critical issue is ontology building that includes identifying, defining, and entering concept definitions and their relationships. Indeed, in large complex application domains, this task can be lengthy, costly, and controversial, particularly because people can have different points of view about the same concept. Therefore, finding (semi-)automatic algorithms to extract ontology concepts from existing knowledge bases represents an important activity. However, most approaches have "only" considered one step in the overall ontology engineering process, for example, the acquisition of concepts, the establishment of a concept taxonomy, or the discovering of conceptual relationships, whereas one must consider the overall process when building real-world applications. For this purpose, efforts have been made to facilitate the ontology engineering process, particularly the acquisition of ontologies from domain texts.

In this paper, we describe our approach for ontology extraction from an existing knowledge base of heterogeneous documents. We also show an implementation of the proposed approach in the context of e-learning and present the experimental evaluation.

Manuscript received December 5, 2008; accepted March 25, 2009. Date of publication May 12, 2011; date of current version June 21, 2011. This paper was recommended by Editor W. Pedrycz.

M. Gaeta, F. Orciuoli, and S. Salerno are with the Centro di Ricerca in Matematica Pura ed Applicata, Dipartimento di Ingegneria dell'Informazione e Matematica Applicata, University of Salerno, 84084 Fisciano, Italy (e-mail: gaeta@diima.unisa.it; orciuli@diima.unisa.it; salerno@unisa.it).

S. Paolozzi is with the Dipartimento di Informatica e Automazione, Roma Tre University, 00146 Rome, Italy (e-mail: stefano.paolozzi@gmail.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMCA.2011.2132713

A. Background and Related Works

In the literature, the area of studies addressed by this paper is called *ontology extraction* or *ontology learning*. These terms mean the process of extracting ontological representations starting from extensive amount of unstructured text. Compared to the more general *information extraction*, ontology learning focuses on concepts and relationships between concepts.

Two main approaches have been developed to aid ontology learning. The first one facilitates manual ontology engineering by providing natural language processing tools, including editors, consistency checkers, mediators to support shared decisions, and ontology import tools. The second approach is based on machine learning and automated language processing techniques to extract concepts and ontological relations from structured and unstructured data such as databases and texts. Few systems exploit both approaches. The first approach is predominant in most developed tools such as KAON [36], Protégé [1], Chimaera [22], and many others, but some systems also implement machine learning techniques.

In recent years, there has been an increasing awareness of the potential value of ontologies accompanied by a growing realization of the effort required to manually develop them. As a consequence, there are a lot of research studies which focus on the development of techniques through which ontological knowledge might be extracted from existing sources.

A number of systems have been proposed for ontology extraction from text. We describe some of them in the following.

ASIUM [12] extracts verb frames and taxonomic knowledge, based on statistical analysis of syntactic parsing of texts.

Text-To-Onto [18] combines machine learning approaches with basic linguistic processing such as tokenization or lemmatization and shallow parsing. It is based on the General Architecture for Text Engineering (GATE) framework [8]. The Text-To-Onto system defines a common framework into which extraction and maintenance mechanisms may be easily managed.

OntoLearn [25], [35] is partially supported by the INTEROP Network of Excellence. The main task performed by OntoLearn is semantic disambiguation. Semantic disambiguation is performed using a method called *structural semantic interconnection*, an approach to pattern recognition, that uses graphs to describe the objects to analyze (word senses) and a context-free grammar to detect common semantic patterns between graphs.

OntoLT [29] extracts ontology concepts by term extraction through statistical methods and definition of linguistic patterns as well as convenient mappings to ontological structures.

DODDLE II [40] learns taxonomic and nontaxonomic relations using co-occurrence analysis, exploiting a machine readable dictionary and a domain-specific text.

Most of these systems depend on shallow text parsing and machine learning algorithms to find potentially interesting concepts and relations between them. The OntoLT approach is most similar to the ASIUM system but relies even more on linguistic/semantic knowledge through its use of built-in patterns that map possibly complex linguistic (morphological analysis and grammatical functions) and semantic (lexical semantic classes and predicate–argument structures) structures directly to concepts and relations. A machine learning approach can easily be build on top of this but is not strictly necessary. Additionally, like the Text-To-Onto system, OntoLT provides a complete integration of ontology extraction from text into an ontology development environment but selects for this purpose (unlike Text-To-Onto) the widely used Protégé tool, which allows for efficient handling and exchange of extracted ontologies (e.g., in RDF/S format).

B. Contribution

The main contributions of this paper can be summarized as follows:

1) the definition of a complete methodology for automatic knowledge extraction, in the form of ontological concepts, from a knowledge base of heterogeneous documents;
2) the implementation of the proposed methodology in an integrated system with e-learning purposes;
3) the evaluation and the validation of the implemented system in terms of precision and recall measures.

The rest of this paper is structured as follows. Section II presents our approach for ontology extraction. In Section III, we describe the architecture of the systems that implements the proposed approach. Section IV discusses the case study in which we have tested our approach. In Section V, we report the experiments and their evaluation. Finally, Section VI concludes.

II. Approach for Ontology Extraction

Our approach is based on the idea of domain ontology. A domain ontology tries to reduce or eliminate conceptual and terminological confusion among the members of a particular community who need to share different kinds of documents and information. It is realized by identifying and properly defining a set of relevant concepts that characterize a given application domain (e.g., medical applications, travel, etc.). An ontology specifies a shared understanding of a domain; in other words, it contains a set of generic concepts, together with their definitions and relationships.

Before explaining the single steps of the approach, we must introduce some assumptions on the analyzable documents. They can be summarized as follows.

1) Each document has been assumed to have a partially defined structure. It will be so possible to make assumptions about how to initially model the ontology, simplifying the process of report production.
2) It is also possible to consider nonstructured documents, but they must be preprocessed to make them at least semistructured.
3) Our experimentation has been made considering the PDF, TXT, PPT, and DOC formats, but the approach is extendible to other document formats.

The whole process consists of the following steps.

1) **Preprocessing**: A preliminary work on the available documents is carried out.
2) **First ontology creation**: A first version of the ontology is created.
3) **Concept and relationship creation**: The creation of the whole ontology extracting the concepts and their relations from the text documents is carried out.
4) **Harmonization**: The extracted ontology is "harmonized" through the analysis of other domain ontologies and concept description from other systems (such as Wikipedia[1]).
5) **Refinement and validation**: The resulting ontology is refined and validated.

A. Preprocessing

In this phase, the documents are prepared for the extraction. We can distinguish several subphases, described in the following.

1) *Format conversion*: The documents are converted from the original format to a more suitable one (i.e., an XML version of the document with additional annotations).
2) *Stemming*: It is the process of reducing a term of the analyzed document to its *stem* or root form (e.g., writing → write). However, the stem does not need to be identical to the morphological root of the term; it is usually sufficient that related words map to the same stem, even if this stem is not a valid root. We use the combination of different algorithms to perform this step (e.g., stochastic, lemmatization, i.e., the expansions to synonyms [11], [31], and suffix stripping algorithms [32]).
3) *Part-of-speech tagging (POST)*: It represents the process of marking the terms in the document (including terms composed of several words) in a text as corresponding to a particular part of speech (i.e., names, verbs, adjectives, adverbs, etc.). The POST algorithms that we use rely on the dictionaries and on the context in which the term has been found (i.e., adjacent terms, terms of the sentence or paragraph, etc.).
4) *Stopword list*: In this phase, we remove from the content of the documents all those terms that do not bring useful information for the characterization of the particular domain of interest (such as articles, conjunctions, and verbs).
5) *Synonymous identification*: The problem of synonymy (different terms may express the same concept) may have particular importance, considering the fact that, in some cases, it is possible to have variations of words in different languages. In our approach, we use the WordNet lexical

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

database for the acquisition of the synonyms of a term: The acquired terms are associated to the first term and are taken into account during the text processing.

WordNet [13], [37] is a large lexical database (basically in English) developed under the direction of George A. Miller. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (called synsets), with each expressing a distinct concept. Synsets are interlinked by means of conceptual semantic and lexical relations. This operation has to be performed if we want to update or expand an existing ontology (we will give a more detailed explanation in the next section).

An important aspect that must be stressed is the necessity of identifying the appropriate sense of a term within a particular context. Indeed, natural languages allow several ambiguities of interpretation and for different parts of speech, i.e., a single term can express more than one concept (*polysemy*).

Therefore, the ambiguity of the terms is another potential source of mistakes in the process of concept extraction. To resolve the semantic ambiguity of the terms (word sense disambiguation [17], [30]), in our case, is to automatically determine the most appropriate meaning of a word relying on the context in which it resides.

6) *Terminology extraction*: All the aforementioned subphases are performed to extract the relevant terminology related to a particular domain. We refer to *terminology* as the set of words or word strings which convey a single possibly complex shared meaning within a community. Because of their low ambiguity and high specificity, these words are also particularly useful to conceptualize a knowledge domain.
7) *User intervention*: Considering the fact that the approach can be performed automatically by the implemented system, the human intervention can be useful to improve the preprocessing step. This can be particularly important for the managing of term deletion through the stopword lists.

B. Ontology Creation

In this step, a simple draft version of the ontology is created. From the syntactic point of view, we assume that the elements of interest for the user are constructed on the grounds of some primitive terms. According to this assumption, an ontology consists of primitive classes and compound classes. The primitive classes are the simplest concepts that cannot be assembled from other classes; however, they may be inherited by derived concepts or their children (e.g., subterms). Therefore, in this step, the main task is to create the simple and the compound concepts.

In our approach, we consider the following two types of action in ontology construction.

1) **Ex-novo creation**: The ontology does not exist and must be constructed through the extraction of the concepts from the analyzed documents.
2) **Update/insertion**: We have already an ontology, and the information extraction process is used to augment the existing ontology.

For each extracted term in the previous step, a concept in the ontology will be created. The concept extraction, in this step, takes into account co-occurrences (and linguistic dependences) of terms in the text.

In the case of update/insertion, we have to perform a refinement of an existing ontology through the addition of new concepts from the analyzed documents. An ontology is incrementally updated as new concepts are acquired from the text. Thus, the existing ontology is extended with new concepts and new relations.

C. Concept and Relationship Creation

We have implemented several statistical and data mining algorithms in order to identify the concepts and their relationship in the resulting ontology. We consider, for example, an algorithm that retrieves term frequencies from the text (described in detail in [21]). The output of this algorithm can be used for the creation of concepts in the ontology. To derive the concept hierarchy, we adopted a hierarchical clustering algorithm (see [21] for the complete explanation) that accesses background knowledge from existing ontological entities to label the extracted hierarchy. Moreover, we implemented also an algorithm that is based on frequent couplings of concepts by identifying linguistically related pairs of words in order to acquire conceptual relations (see [20] for the complete description of the algorithm).

D. Harmonization

This is an optional step that is needed when the user wants to "harmonize" the extracted ontology with the available knowledge bases.

With the term *ontology harmonization*, we want to refer to the ability of harmonizing two or more ontologies in a unique ontology in order to improve the available knowledge base. It is strictly related to two main issues: *ontology matching* for the recognition of correspondences between ontologies and *ontology merging* for the actual fusion of those ontologies.

Ontology matching finds correspondences between semantically related entities of different ontologies. These correspondences may stand for equivalence as well as other relations, such as consequence, subsumption, or disjointness between ontology entities [10]. Many different matching solutions have been proposed, but there are no real solutions that can be effectively used in real-world applications.

Ontology merging refers to the process of creating a new ontology from two or more ontologies. In this case, the new ontology will unify and replace the original ontologies. This definition does not explain how the merged ontology relates to the original ontologies, because not all the approaches merge ontologies in the same way. The most prominent approaches are the union and the intersection approaches. In the union approach, the merged ontology is the union of all entities in both source ontologies, where differences in the representation of similar concepts have been resolved. In the intersection approach, the merged ontology consists only of the parts of the source ontology which overlap [23].

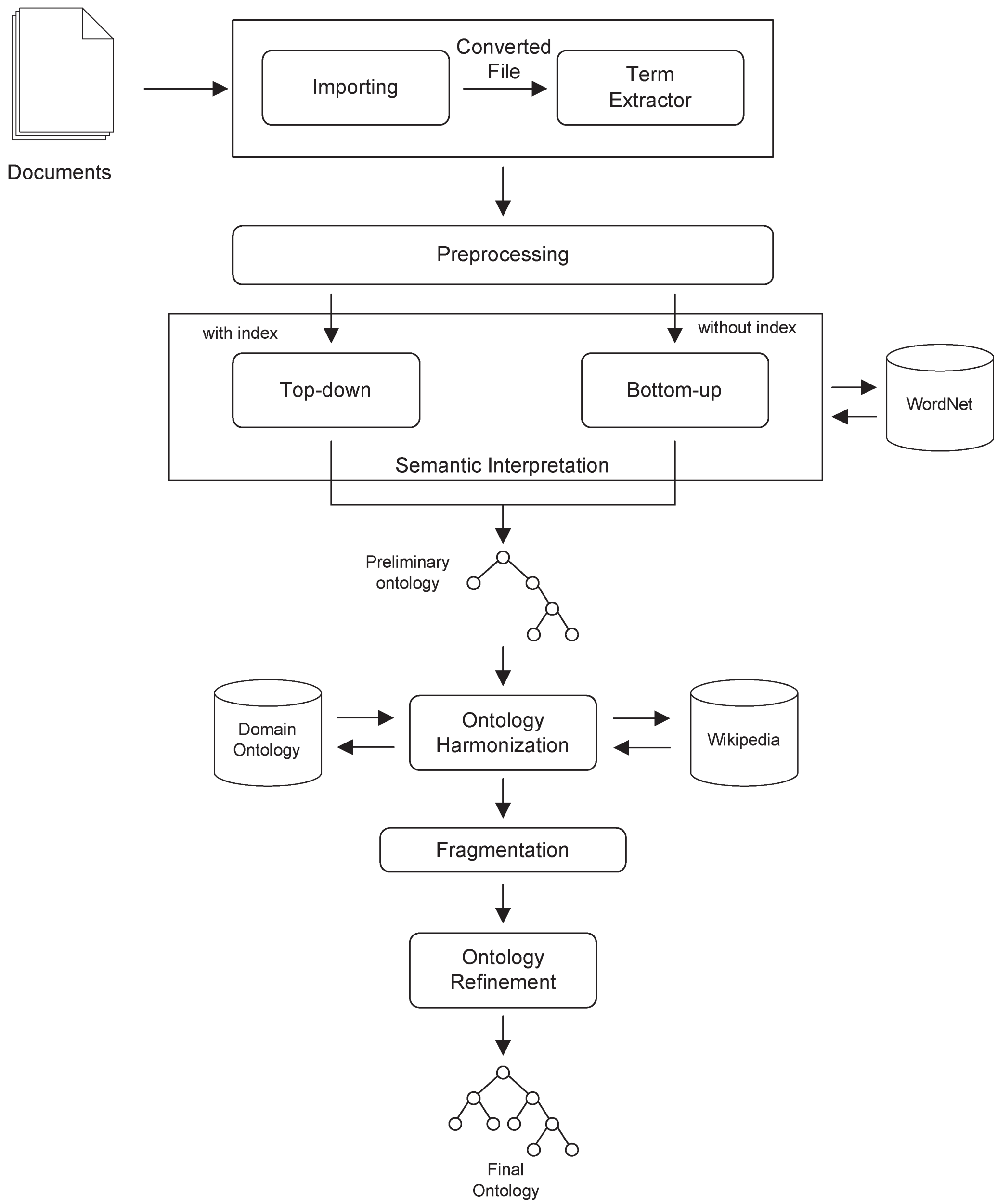

Fig. 1. Architecture of the system.

E. Refinement and Validation

The refinement phase deals with the tuning of the target ontology and the support of its evolving nature. Adapting and refining the ontology with respect to user requirements play a major role for the development of the particular application and its further development.

An important task in this phase is the pruning of irrelevant concepts from the extracted ontologies. Following other approaches, we adopt a pruning strategy which advocates that frequent terms in a text corpus denote domain concepts, while less frequent ones lead to concepts that can be safely eliminated from the ontology [19].

We set an average frequency of the extracted terms as a threshold value and prune all concepts that have a frequency lower than this value.

III. Architecture of the System

Fig. 1 shows the main elements of the architecture of the proposed system.

As we can see in the aforementioned figure, the system computation is realized through the application (in pipeline) of eight main software modules implemented in Java. Each module exploits results coming from the execution of the previously applied module and possibly from user decisions. Therefore, we can define the system input as a set of documents, possibly organized in a hierarchy of folders within a file system. The system output is composed of the following two main results: 1) the domain ontology representing the knowledge underlying the document set and 2) the semantic associations between document sections and concepts into the extracted domain ontology. The proposed system does not offer a completely automatic elaboration but needs human mediation in some execution points.

Before analyzing the details of each module, it is important to give evidence of how the different modules work together. In order to build a flexible, scalable, and easily extensible system, we have adopted *GATE* [16], [34], [39] as base framework. GATE is a Java software toolkit developed at The University of Sheffield. Nowadays, GATE is used by a wide community of scientists, companies, teachers, and students for several natural language processing tasks, including information extraction in many languages. GATE specifies an architecture for language processing software and provides a framework (by means of a software development kit) that implements the architecture. It can be used to embed language processing capabilities in several applications and provides a development environment, built on top of the framework, for developing many types of components. The architecture exploits component-based software development, object orientation, and mobile code. The framework and the development environment are written in Java and are available as open-source software.[2] All modules composing our system are realized as a GATE component and are invoked by a specific workflow defined by means of the text engineering framework.

The *Importing Module* has two main goals. The first one is the harvesting of information about how different documents (composing the system input) are related to each other by a folder structure within the file system. If two documents are related by means of a hierarchy structure, it is reasonable to think that the concepts embedded in these documents are also related. These pieces of information are caught and forwarded along the pipeline in order to be exploited by the *Semantic Interpretation Module*. The hierarchical information structures are extracted using a Java base class library. The second goal of this module is represented by the extraction of text and information structure (e.g., titles, sections, indexes, and so forth) from the input documents. Having an abstract representation of documents in this phase can simplify the design of the other architecture components that do not care about the formats of the original documents. This part of the whole architecture is shown in Fig. 2.

Each document format (e.g., PPT, PDF, DOC, etc.) that we want to use is managed by a specific component (e.g., PPT extractor, PDF extractor, DOC extractor, etc.) in order to extract text and information structure and use them to build the abstract representation of the original document. The system is extensible because we provide a set of Java interfaces and abstract classes that can be implemented and extended in order to construct new extractors for any kind of document format. Clearly, the model of abstract document representation is fixed by the system, and the extractor can use preexisting Java classes in order to build these representations. The implementation of extractors can be supported (this is the case of the already implemented sample extractors) by the use of *Jakarta POI*[3] or

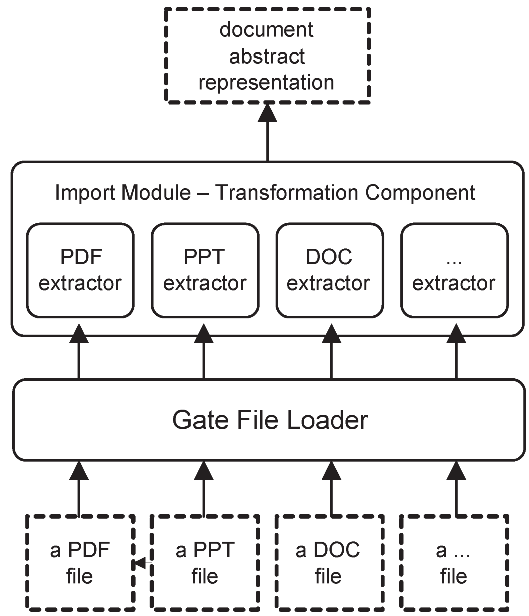

Fig. 2. Transformation of documents in abstract representations.

Open Office API[4] that are open-source class libraries used to manage specific file formats like Microsoft formats.

In Fig. 3, we show the abstract representation skeleton provided in XML.

In particular, part (a) in Fig. 3 reports that a document is divided into several parts managed by the ⟨`resourcePart`⟩ tag. Part (b) in Fig. 3 indicates how it is possible to represent a single document part. In particular, each document part is composed of a title (represented by the ⟨`title`⟩ tag) and a body (represented by the ⟨`body`⟩ tag). Furthermore, both title and body can be decomposed into more granular pieces through the use of the ⟨`textRun`⟩ tag. Each piece of information (textRun) is represented by a raw text (⟨`rawContent`⟩ tag) and style (⟨`style`⟩ tag). The style attributes (`alignment`, `bold`, `italic`, `bullet`, `indentLevel`, and `underline`) are very important in order to infer relevant information structure. The way that a document is divided into parts and the way that a part is divided into more granular pieces are decided by the internal logic of each specific extractor.

For instance, suppose that we have the PPT document in Fig. 4.

A specific PPT extractor could choose to represent each slide as a single part and produce, for each slide, the results shown in Figs. 5 and 6.

The *Term Extractor Module* performs the task of selection of a set of relevant terms from the document abstract representation obtained by the *Importing Module*.

This approach is particularly useful in order to avoid losing of document information structure.

The first operation performed by the *Term Extractor* consists in filtering the set of terms to be analyzed in order to extract the relevant ones. The filtering operation is executed by the use of a stopword list that is useful to eliminate particle words of grammar (e.g., the, of, and, etc.). Obviously, particle words

[2]GATE is available at http://www.gate.ac.uk/

[3]Jakarta POI—Java API To Access Microsoft Format Files, http://jakarta.apache.org/poi/

[4]Open Office API, http://api.openoffice.org/

Fig. 3. Abstract representation skeleton.

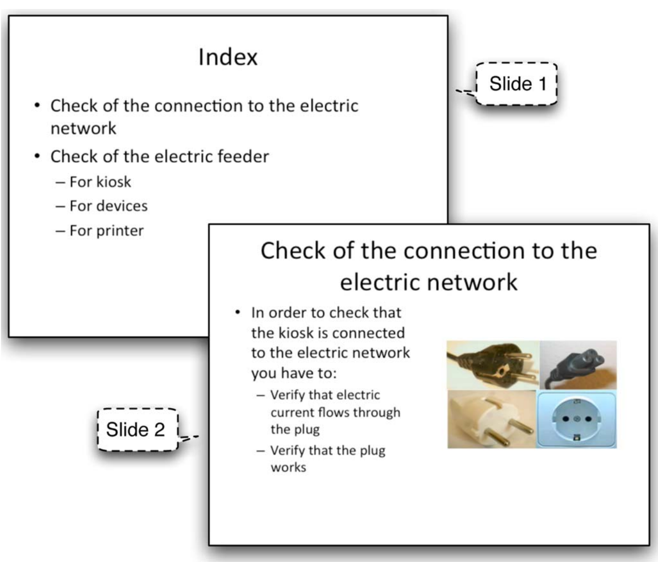

Fig. 4. Sample PPT document.

cannot be considered as relevant terms in a specific domain, so we can eliminate them. The second operation executed by the *Term Extractor* consists in stemming the terms in the filtered set in order to reduce inflected words to their stem, base, or root form. The stemming operation simplifies the relevant term extraction given that it enables to consider terms like "reaction," "reactions," "reactive," "reactivity," and "reactant" as the same term. In order to perform filtering and stemming, we use the Snowball open-source (BSD License) tool.[5] Snowball provides stemmers for several languages (e.g., English, German, Italian, etc.) and is extensible (adding new stemming rules). The main operation of the *Term Extractor* is the selection of relevant terms from the filtered set of stemmed terms. We developed, from scratch, the component that is able to perform the selection operation. The aforementioned component is based on the algorithms proposed in [24] and [33] able to calculate a score for a term relevance in a specific domain. Only the terms with a score over a given threshold are considered relevant.

The *Preprocessing and Semantic Interpretation Modules* identify and execute the right approach to the ontology extraction, respectively. The *Preprocessing Module* analyzes the abstract document representation provided by the previous steps and searches for document parts that are indexes (possibly identified by its title or by the presence of a bullet list) for the document topics. If information structures, like indexes, exist in the document, then the module selects the top-down approach; otherwise, the bottom-up approach is selected.

Once the selection task of the *Preprocessing Module* is finished, the *Semantic Interpreter* starts executing the right approach. In the case of the top-down approach, the *Semantic Interpreter* builds a first draft of the ontology using only the *HasPart* (HP) relation and reproducing the hierarchy structure coming from the index information analysis. In this case, each item in the index becomes a concept within the ontology, and the parent–child relation between items in the index becomes an *HP* relation in the ontology. A concept is described by one or more relevant terms extracted by the title of the document part using a technique that is similar to that used by the *Term Extractor Module*. In the case of the bottom-up approach, a first draft version of the ontology is constructed with an iterative process. In the first iteration, a concept within the ontology is defined for each document part. In the second iteration, the *Semantic Interpreter* tries to cluster couples of concepts in order to construct a new concept and establishes *HP* relations

[5]Snowball is available at http://snowball.tartarus.org/

```xml
...
<resourcePart id="R1">
    <title>
        <textRun>
        <rawContent>Index</rawContent>
        </textRun>
    </title>
    <body>
        <textRun>
            <rawContent>Check of the connection to the electric network</rawContent>
            <style alignment="left" bold="false" bullet="true" indentLevel="0"
                italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>Check of the electric feeder</rawContent>
            <style alignment="left" bold="false" bullet="true" indentLevel="0"
                italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>For kiosk</rawContent>
            <style alignment="left" bold="false" bullet="true" indentLevel="1"
                italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>For devices</rawContent>
            <style alignment="left" bold="false" bullet="true" indentLevel="1"
                italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>For printer</rawContent>
            <style alignment="left" bold="false" bullet="true" indentLevel="1"
                italic="false" underline="false"/>
        </textRun>
    </body>
</resourcePart>
...
```


Fig. 5. XML representation of the first slide.


```xml
...
<resourcePart id="R2">
    <title>
        <textRun>
            <rawContent>Check of the connection to the electric network</rawContent>
        </textRun>
    </title>
    <body>
        <textRun>
            <rawContent>In order to check that the kiosk is connected to the electric
                network you have to:</rawContent>
        <style alignment="left" bold="false" bullet="true" indentLevel="0"
            italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>Verify that electrict current flows through the plug</rawContent>
        <style alignment="left" bold="false" bullet="true" indentLevel="1"
            italic="false" underline="false"/>
        </textRun>
        <textRun>
            <rawContent>Verify that the plug works</rawContent>
        <style alignment="left" bold="false" bullet="true" indentLevel="1"
            italic="false" underline="false"/>
        </textRun>
        <textRun>
        <rawContent/>
        <style alignment="left" bold="false" bullet="true" indentLevel="0"
            italic="false" underline="false"/>
        </textRun>
    </body>
</resourcePart>
...
```


Fig. 6. XML representation of the second slide.

between the new concept (parent) and the couple of existing concepts (children). The clustering operation is based on the possibility (captured by analyzing the text through the application of some specific patterns) that two concepts represent different aspects of the same topic. Once the draft ontology is available, the *Semantic Interpretation Module* uses WordNet in order to identify (within the draft ontology) concepts represented by different terms, but semantically equivalent to each other, and collapse them. Another useful information that we can obtain using WordNet is the semantic correlation between concepts. Unfortunately, this information can only be suggested to the user who decides to translate it or not into *IsRequiredBy* (IRB) or *SuggestedOrder* (SO) relations. At the end of this step, a preliminary ontology is available for the next phases.

The *Ontology Harmonization Module* is responsible for searching the repository in order to find an ontology (stored in the repository) that covers the same domain, or part of it, of the preliminary ontology previously extracted. Once the searching provides a good result, the preliminary extracted ontology can be harmonized with the existing ontology in order to improve the available knowledge base and to adjust the artifact produced by the *Semantic Interpretation Module*. The *Ontology Harmonization Module* is based on a merging algorithm consisting of four steps. The first step simply consists of the selection (by the user) of two source ontologies. The second step is represented by the matching phase in which couples of concepts, belonging to the two ontologies, with a high similarity level (both lexical and semantical) are identified. A *match* is identified when two terms represent the same concept in two different ontologies. As mentioned earlier, the computation of similarity is transparent to the user. The matching algorithm that we proposed is based on the integration of different techniques that are suitable for application to our ontology model without any predefined document corpus. In particular, we use the following.

1) **String-based technique**. It is a similarity computation applied to all possible pairs of concepts (C_1, C_2), with C_1 belonging to ontology source O_1 and C_2 belonging to ontology source O_2. Similarity metrics like *q-gram* and *Levenshtein distance* [7] and typical operations of text processing like *stemming* are applied in order to establish a first set of *matches* between the two ontologies.
2) **Graph analysis**. Graph analysis techniques are applied to the two ontologies, taking into account also the first set of *matches*, in order to enrich the set using information coming from the structures of the two ontologies.
3) **Semantic analysis**. Normalized Google distance [6] and Wikipedia are used as linguistic resources in order to refine the set of *matches* using semantic similarity information.

In the third step, starting from the set of *matches*, the algorithm generates a list of suggestions containing the operations that can be applied to perform the ontology merging. In the fourth (and final) step, the suggested operations are applied, and moreover, the algorithm determines and resolves redundancies (for relations and concepts) that could be generated. The *Ontology Harmonization Module* is fully interactive because we believe that a completely automatic process is very difficult to obtain and it would not be very effective. Therefore, each step of the merging algorithm can be validated by users that, if needed, can manipulate step results following the ideas proposed with PROMPT (see [26] and [27] for a complete explanation).

The *Fragmentation and Semantic Annotation Module* is responsible for the association of single document parts to concepts within the harmonized ontology produced during the previous steps. The module uses the relevant terms extracted by the *Term Extractor Module* and tries to match these ones with the concepts in the ontology. The matching operation is realized with the same techniques used by the *Ontology Harmonization Module*. At the end of the matching process, each document part is associated with one or more concepts of the harmonized ontology.

The *Refinement Module* is the last component of the proposed architecture. The refinement process can be divided into two phases. The first one is dedicated to the anomaly detection. In this phase, the module automatically identifies possible anomalies in the ontology structure (e.g., the overgrowth deepness of some concepts in the graph with respect to the average depth) and document part associations (e.g., document parts associated to an overgrowth number of concepts with respect to the average number and too many document parts associated to the same concept with respect to the average number). These anomalies can be automatically detected but must be solved by the user exploiting a visual drag and drop user interface provided by the module. In this phase, the user could also do the following: 1) modify the ontology structure and document part associations with concepts and 2) merge more document parts in one part. The second and last phase of the refinement operation is the transformation of ontology and document parts from internal representations into interoperable objects. The ontology is finally represented in OWL [9] that is the most used language for ontology representation. Document parts (abstract representations) are converted into Web contents using convenient tools like Jakarta POI. The produced Web contents are packaged and described with an IEEE Learning Object Metadata schema [28], [38]. Some metadata attributes are automatically filled. Users can fill by hand other attributes using a graphical user interface.

IV. E-Learning Case Study

In this section, we introduce our case study that is based on an integrated semantic virtual learning environment to create and manage personalized e-learning experiences through ontologies.

Before describing the case study, it is necessary to introduce the ontology structure in our system. The main goal of these ontologies is to model the knowledge of disciplinary domains.

An e-learning ontology can be represented with a graph in which nodes are relevant concepts within the educational domain of interest and edges are binary relations between two concepts. Our approach foresees different kinds of relations: *HP* that is an inclusion relation, *IRB* that is an order relation, *SO* that is a "weak" order relation, and *HasResource* (HR) that relates concepts with learning objects. The restricted set of relations is not a knowledge representation limit but is a convenient method to improve the computational complexity of algorithms that have to navigate the graph. Let us illustrate how it is possible to model an e-learning ontology. We have to model the educational domain D, so we try to conceptualize the knowledge of D and to find a set of terms representing relevant concepts in D. The result of this operation is the list of terms $T = C_1, C_2, C_3, C_4$, where T is one of the plausible conceptualizations of D (C, C_1, C_2, and C_3 are ontology concepts). In order to explain the semantics of *HP* relation, we can refer to the ontology shown in Fig. 7.

In Fig. 7, there are three *HP* relations $HasPart(C, C_1)$, $HasPart(C, C_2)$, and $HasPart(C, C_3)$. They mean, in terms

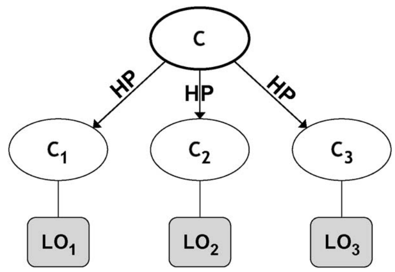

Fig. 7. Simple *HP* relations for an e-learning ontology.

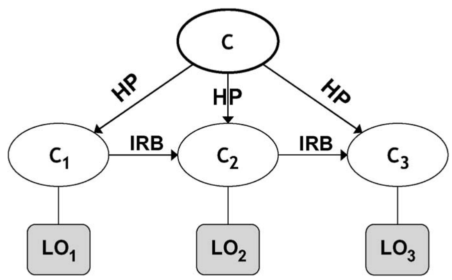

Fig. 8. IWT ontology with *HP* and *IRB* relations.

of e-learning, that, in order to learn concept C, learners have to learn concepts C_1, C_2, and C_3 without considering a specific order. In Fig. 7, the reader can note the existence of elements that are neither concepts nor relations. These new elements are the learning objects LO_1, LO_2, and LO_3. The connection between a concept and a learning object, for instance, C_1 and LO_1, can be interpreted as an *HR* relation. The relation $HasResource(C_1, LO_1)$ means that the educational content packaged in learning object LO_1 explains concept C_1. Therefore, if the learning objective is C_1, then the correspondent assembled e-learning experience is composed only of $[LO_1]$; otherwise, if the learning objective is C, then the assembled e-learning experience will be composed of one of the plausible permutations of $[LO_1, LO_2, LO_3]$.

Now, consider the ontology shown in Fig. 8.

This ontology presents two *IRB* relations, which are $IsRequiredBy(C_1, C_2)$ and $IsRequiredBy(C_2, C_3)$. The two relations mean that C_1 has to be necessarily learned before C_2 and that C_2 has to be necessarily learned before C_3. In this case, if C is the learning objective, learners have to learn the ordered sequence of concepts $[C_1, C_2, C_3]$, and correspondingly, they can join the e-learning experience assembled by the ordered sequence of learning objects $[LO_1, LO_2, LO_3]$. Alternative permutations like $[C_2, C_1, C_3]$ will be invalid.

The sequence of concepts that are useful to reach a pointed learning objective is called a learning path; the operation used to construct the concrete e-learning experience assembling a learning object sequence is called resource binding.

We have outlined the foundations of our modeling approach; now, we want to refine the approach description. First of all, we state that the same learning object can explain more than one concept within the same ontology. In general, the *HR* relation is represented by a function $HasResource(LO_1; \{C_1, C_2, \ldots, C_n\})$, which means that LO_1 explains all concepts $C_1, C_2, \ldots, C_n$. Otherwise, it is possible to have more than one learning object explaining the same concept. We can have, at the same time, the relations $HasResource(C_1, LO_1)$, $HasResource(C_1, LO_2)$, $HasResource(C_1, LO_3)$, etc. Lastly, suppose that you have a *SO* relation between concept C_1 and concept C_2 that is $SuggestedOrder(C_1, C_2)$; this relation states that the modeler states that it is preferable to explain concept C_1 before concept C_2, but this is not mandatory.

We investigate the problem of ontology extraction in the context of real e-learning activities, building personalized and contextualized learning experiences based on explicit knowledge modeling and exploiting ontologies in order to represent disciplinary domains.

The learning experience definition process is based on ontologies built in the process of knowledge extraction from available text documents of a specific domain.

We have developed a complete e-learning system that, on the basis of the extracted ontologies (representing the disciplinary domain of interest), can be used to define the sequence of concepts needed (by a learner) to acquire a satisfactory knowledge of learning objectives identified (by a teacher) as target concepts of the given ontologies. Therefore, we took a step forward with respect to the information transfer paradigm commonly used in the e-learning practice.

A complete description of the overall e-learning system is beyond the scope of this paper, but our approach proved the importance of ontologies in e-learning systems. To prove the benefits of ontologies, we have performed a small-scale experimentation in the enterprise context. We involved a group of 28 voluntary learners belonging to 7 small and medium enterprises dealing with vocational training on enterprise management.

All the voluntary learners were tested before and after a training phase on the same topics. In all the tests, the learners' skills in the chosen domain were quantified using three ability ranges: low level (0–3 scores), medium level (4–7 scores), and high level (8–10 scores). The group of learners was split into two separate subgroups: The first subgroup was enabled to use all features of our system, and the other subgroup can only access to the traditional e-learning materials (i.e., without the benefits of ontologies).

A complete review of these tests is reported in [5]. However, in Fig. 9, we report a simple diagram that shows a comparison of the learners' skills after the experiment with and without the use of our systems (more details can also be found in [2] and [3]).

We have just introduced the structure of our e-learning ontologies that has been used in the ontology extraction subsystem described here. The reader can find a more detailed dissertation in [15].

V. Experimental Evaluation

Considering our implantation, the overall evaluation of the approach can be a challenging task. Indeed, there are no standard evaluation measures that are specifically suited for the ontology extraction process. Due to the nature of ontology extraction techniques that we implemented, the evaluation is more difficult than, for example, the one in the information retrieval community.

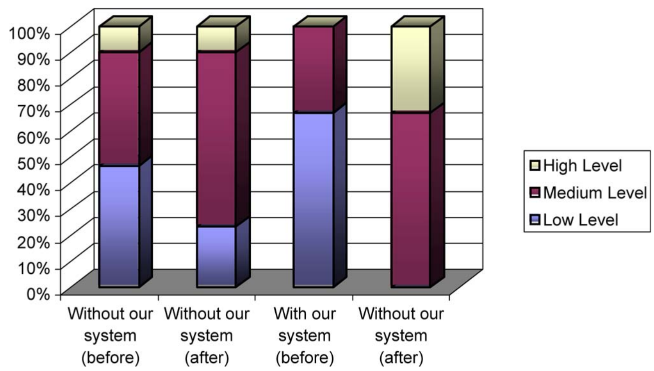

Fig. 9. Experimentation results with our system.

To overcome these hindrances, we propose our evaluation approach following the ideas sketched in [18] and [41]. In practice, we compute the similarity between a manually built reference ontology and an ontology that has been generated by applying our ontology extraction technique. We assumed that a high similarity between the manually built ontology (that is created by an ontology engineer) and the resulting ontology of our approach indicates a successful application of the ontology extraction approach.

We use two straightforward evaluation measures derived from the information retrieval community, namely, *precision* and *recall*.

Precision can be seen as a measure of exactness or fidelity, whereas recall is a measure of completeness. In information retrieval, precision is defined as the number of *relevant documents* (rd) retrieved by a search divided by the total number of *documents retrieved* (dr) by that search, and recall is defined as the number of relevant documents retrieved by a search divided by the total number of existing relevant documents (which should have been retrieved). More formally, we have [14]

$$Precision = \frac{|\{rd\} \cap \{dr\}|}{|\{dr\}|} \tag{1}$$

$$Recall = \frac{|\{rd\} \cap \{dr\}|}{|\{rd\}|}. \tag{2}$$

For our purposes, we slightly modified these definitions (following the ideas of Maedche and Staab [18]) in order to evaluate the ontology extraction process. Our definitions of *precision* and *recall* for the evaluation of ontology extraction are the following:

$$Precision_{OEx} = \frac{|Ref_O \cap Res_O|}{|Res_O|} \tag{3}$$

$$Recall_{OEx} = \frac{|Ref_O \cap Res_O|}{|Ref_O|}. \tag{4}$$

The set Ref_O represents the set of all elements given in the manually built reference ontology. Res_O is the set of elements contained in the resulting ontology given by the ontology extraction process.

For the experiments, we select seven different domains with several documents each (among PDF, DOC, TXT, or PPT documents) as reported in Table I. All extracted ontologies (that have the structure defined in Section IV) have been used to define convenient learning path and learning objects for each e-learning course related to those domains.

TABLE I
DOMAINS USED TO CREATE THE TEST SETS

Domain	N. of documents	Code
Art	46	01
Database	102	02
Human Resources	68	03
Informative Systems	126	04
Problem Solving	26	05
Tourism	85	06
User Modeling	110	07

TABLE II
PRECISION AND RECALL MEASURES FOR ONTOLOGY EXTRACTION

Test set code	Precision	Recall
01	0,78	0,48
02	0,82	0,52
03	0,65	0,44
04	0,83	0,49
05	0,96	0,59
06	0,77	0,62
07	0,89	0,53

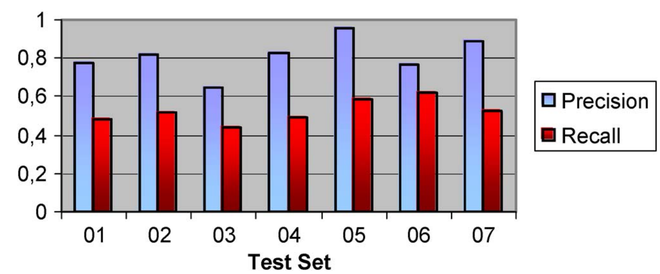

Fig. 10. Precision and recall.

The results of precision and recall for the extraction of ontology from the natural language text documents are reported in Table II and Fig. 10. The results show average values of 81% for precision and 52% for recall. As it is evident in Fig. 10, the well-known tradeoff between precision and recall becomes obvious.

Moreover, we consider another popular measure that combines precision and recall; the weighted harmonic mean of precision and recall is the F-measure that can be defined as follows:

$$F_\beta = \frac{(1+\beta^2)\cdot(Precision \cdot Recall)}{\beta^2 \cdot Precision + Recall} \tag{5}$$

where β is a real positive number. With the F-measure, a β lower than one gives more importance to precision, while a β higher than one gives more importance to recall. In our tests, we used $\beta = 0.5$ to underline the importance of precision over recall in our type of application domain. The results are reported in Table III and Fig. 11.

Overall, the results are encouraging. The average result for the F-measure is 73%.

As a result of these experiments, we believe that the proposed algorithms are significant since they can achieve the best result. Indeed, both Figs. 10 and 11 highlight rather good results, even in comparison with the available data in literature, particularly for high-precision results.

TABLE III
F-MEASURES FOR ONTOLOGY EXTRACTION

Test set code	F-measure
01	0,69
02	0,74
03	0,59
04	0,73
05	0,85
06	0,73
07	0,78

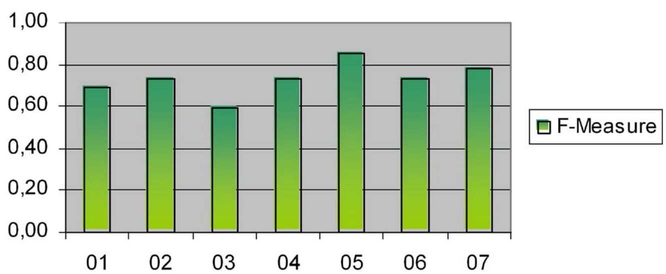

Fig. 11. F-measure.

VI. CONCLUSION

Extracting knowledge from available text documents is one further challenge to ontology engineering. In this paper, we have described our approach to extract ontologies from a knowledge base of heterogeneous text documents. In particular, we used the proposed methodology in a more complex e-learning context, in which the extracted ontologies are used to define contextualized and personalized learning experiences. Moreover, we implemented our approach in a system and also verified that these techniques obtain interesting performance on both precision and recall measures.

The work that we described has several novel features. First of all, the ontology extraction process is general and is not domain dependent. Second, the method has been applied and validated with success in a "real" context of a large e-learning community.

In the future, we will aim at the performance improvement of our approach, investigating also more refined algorithms and addressing other knowledge sources.

REFERENCES

[1] *The Protégé Ontology Editor and Knowledge Acquisition System.* [Online]. Available: http://protege.stanford.edu/
[2] G. Albano, M. Gaeta, and P. Ritrovato, "IWT: An innovative solution for AGS e-learning model," *Int. J. Knowl. Learn.*, vol. 3, no. 2/3, pp. 209–224, 2007.
[3] G. Albano, M. Gaeta, and S. Salerno, "E-learning: A model and process proposal," *Int. J. Knowl. Learn.*, vol. 2, no. 1/2, pp. 73–88, 2006.
[4] T. Berners-Lee, *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by Its Inventor*. San Francisco, CA: Harper San Francisco, 1999.
[5] N. Capuano, M. Gaeta, S. Miranda, F. Orciuoli, and P. Ritrovato, "LIA: An intelligent advisor for e-learning," in *Proc. 1st WSKS*, 2008, pp. 187–196.
[6] R. L. Cilibrasi and P. M. B. Vitanyi, "The Google similarity distance," *IEEE Trans. Knowl. Data Eng.*, vol. 19, no. 3, pp. 370–383, Mar. 2007.
[7] W. W. Cohen, P. Ravikumar, and S. E. Fienberg, "A comparison of string distance metrics for name-matching tasks," in *Proc. IJCAI-03 Workshop Inf. Integr.*, 2003, pp. 73–78.
[8] H. Cunningham, D. Maynard, K. Bontcheva, and V. Tablan, "GATE: A framework and graphical development environment for robust NLP tools and applications," in *Proc. 40th Anniv. Meeting Assoc. Comput. Linguistics*, 2002, pp. 1–8.
[9] M. Dean and G. Schreiber, "OWL Web ontology language reference," W3C Recommendation, Feb. 2004.
[10] J. Euzenat and P. Shvaiko, *Ontology Matching*. Heidelberg, Germany: Springer-Verlag, 2007.
[11] R. Farkas, V. Vincze, I. Nagy, R. Ormándi, G. Szarvas, and A. Almási, "Web-based lemmatisation of named entities," in *Proc. TSD*, vol. 5246, *Lecture Notes in Computer Science*, P. Sojka, A. Horák, I. Kopeček, and K. Pala, Eds. Berlin, Germany: Springer-Verlag, 2008, pp. 53–60.
[12] D. Faure and T. Poibeau, "First experiences of using semantic knowledge learned by ASIUM for information extraction task using INTEX," in *Proc. ECAI Workshop Ontology Learning*, vol. 31, *CEUR Workshop Proceedings*, S. Staab, A. Maedche, C. Nédellec, and P. Wiemer-Hastings, Eds., 2000.
[13] C. Fellbaum, Ed., *WordNet an Electronic Lexical Database*. Cambridge, MA: MIT Press, May 1998.
[14] W. B. Frakes and R. A. Baeza-Yates, Eds., *Information Retrieval: Data Structures & Algorithms*. Englewood Cliffs, NJ: Prentice-Hall, 1992.
[15] M. Gaeta, F. Orciuoli, S. Paolozzi, and P. Ritrovato, "Effective ontology management in virtual learning environments," *Int. J. Internet Enterprise Manage.*, vol. 6, no. 2, pp. 96–123, 2009.
[16] R. Gaizauskas, H. Cunningham, Y. Wilks, P. Rodgers, and K. Humphreys, "GATE: An environment to support research and development in natural language engineering," in *Proc. 8th ICTAI*, Washington, DC, 1996, pp. 58–66.
[17] S. G. Kolte and S. G. Bhirud, "Word sense disambiguation using WordNet domains," in *Proc. Int. Conf. Emerging Trends Eng. Technol.*, 2008, pp. 1187–1191.
[18] A. Maedche and S. Staab, "The Text-To-Onto ontology learning environment," in *Proc. 8th Int. Conf. Conceptual Struct.*, Darmstadt, Germany, 2000, pp. 14–18.
[19] A. D. Maedche, *Ontology Learning for the Semantic Web*. Norwell, MA: Kluwer, 2002.
[20] E. Maedche and S. Staab, "Mining non-taxonomic conceptual relations from text," in *Proc. R. Dieng O. Corby, EKAW00 Eur. Knowl. Acquisition Workshop–LNAI*, 2000, pp. 189–202.
[21] C. D. Manning and H. Schtze, *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press, Jun. 1999.
[22] D. L. McGuinness, R. Fikes, J. Rice, and S. Wilder, "The Chimaera ontology environment," in *Proc. AAAI/IAAI*, 2000, pp. 1123–1124.
[23] P. Mitra and G. Wiederhold, "An algebra for semantic interoperability of information sources," in *Proc. 2nd IEEE Int. Symp. BIBE*, Washington, DC, 2001, p. 174.
[24] R. Navigli and P. Velardi, "Semantic interpretation of terminological strings," in *Proc. 6th Int. Conf. TKE*, 2002, pp. 95–100.
[25] R. Navigli, P. Velardi, and A. Gangemi, "Ontology learning and its application to automated terminology translation," *IEEE Intell. Syst.*, vol. 18, no. 1, pp. 22–31, Jan. 2003.
[26] N. F. Noy and M. A. Musen, "The PROMPT suite: Interactive tools for ontology merging and mapping," *Int. J. Human-Comput. Stud.*, vol. 59, no. 6, pp. 983–1024, Dec. 2003.
[27] N. Fridman Noy and M. A. Musen, "PROMPT: Algorithm and tool for automated ontology merging and alignment," in *Proc. AAAI/IAAI*, 2000, pp. 450–455.
[28] *Draft Standard for Learning Objects Metadata*, IEEE Draft P1484.12.1/D6.412, Jun. 2002.
[29] M. Sintek and P. Buitelaar, "OntoLT version 1.0: Middleware for ontology extraction from text," in *Proc. Demo Session Int. Semantic Web Conf.*, 2004. [Online]. Available: http://iswc2004.semanticweb.org/demos/15/paper.pdf
[30] T. Pedersen, S. Banerjee, and S. Patwardhan, "Maximizing semantic relatedness to perform word sense disambiguation," Univ. Minnesota Supercomput. Inst., Minneapolis, MN, Res. Rep. UMSI 2005/25, Mar. 2005.
[31] J. Plisson, N. Lavrač, D. Mladenič, and T. Erjavec, "Ripple down rule learning for automated word lemmatisation," *AI Commun.*, vol. 21, no. 1, pp. 15–26, Jan. 2008.
[32] J. Savoy, "Light stemming approaches for the French, Portuguese, German and Hungarian languages," in *Proc. ACM Symp. Appl. Comput. SAC*, New York, 2006, pp. 1031–1035.
[33] F. Sclano and P. Velardi, "TermExtractor: A Web application to learn the shared terminology of emergent Web communities," in *Proc. 3rd Int. Conf. I-ESA*, Funchal, Portugal, 2007, pp. 287–290.
[34] V. Tablan, D. Damljanovic, and K. Bontcheva, "A natural language query interface to structured information," in *Proc. 5th ESWC*, Tenerife, Spain, 2008, pp. 361–375.

[35] P. Velardi, R. Navigli, A. Cucchiarelli, and F. Neri, "Evaluation of OntoLearn, a methodology for automatic population of domain ontologies," in *Ontology Learning from Text: Methods, Applications and Evaluation*, P. Buitelaar, P. Cimiano, and B. Magnini, Eds. Amsterdam, The Netherlands: IOS Press, 2006, pp. 92–105.

[36] R. Volz, D. Oberle, S. Staab, and B. Motik, "KAON server—A Semantic Web management system," in *Proc. (Alternate Track) 12th Int. WWW Conf.*, Budapest, Hungary, 2003, pp. 20–24.

[37] E. M. Voorhees, "Using WordNet to disambiguate word senses for text retrieval," in *Proc. 16th Annu. Int. ACM SIGIR Conf. Res Develop. Inf. Retrieval*, New York, 1993, pp. 171–180.

[38] D. A. Wiley, Ed., *The Instructional Use of Learning Objects*. Bloomington, IN: Agency Instructional Technol.: Assoc. Educ. Commun. Technol., 2002.

[39] R. Witte and T. Gitzinger, "Connecting wikis and natural language processing systems," in *Proc. Int. WikiSym*, New York, 2007, pp. 165–176.

[40] T. Yamaguchi, "Acquiring conceptual relationships from domain-specific texts," in *Proc. 2nd Workshop OL*, 2001, pp. 1–6.

[41] B. Yildiz and S. Miksch, "ontoX—A method for ontology-driven information extraction," in *Proc. ICCSA (3)*, vol. 4707, *Lecture Notes in Computer Science*, O. Gervasi and M. L. Gavrilova, Eds., 2007, pp. 660–673, Berlin, Germany: Springer-Verlag.

Matteo Gaeta (M'05) received the B.S. degree in computer science from the University of Salerno, Fisciano, Italy, in 1989.

He is currently an Associate Professor of information processing systems with the Faculty of Engineering, University of Salerno. His research interests include complex information systems architecture, software engineering, systems of knowledge representation, Semantic Web, virtual organization, and grid computing. He gained large experience in the implementation and design of complex information systems. He is the author of more than 80 papers, published on journals, proceedings, and books. He is the Scientific Coordinator and the Manager of several international research projects. He is the Coordinator of the Ministry of Education, University and Research (MIUR) Working Group. He is a Member of the Panel for the Scientific Assessment of Research and Testing Projects of the MiPAF. He is a Member of the Experts Register, Department for Education and Skills, and the Innovation Technology Experts Register, Department for the Economy Development.

Francesco Orciuoli received the B.S. degree (*cum laude*) in computer science from the University of Salerno, Fisciano, Italy, in 1999.

In the last ten years, he has been working, for the Centro di Ricerca in Matematica Pura ed Applicata (CRMPA), on the definition of innovative e-learning solutions based on knowledge technologies for the provisioning of personalized e-learning experiences. He has been involved in research activities in several European-Union-cofunded projects as a CRMPA Collaborator. He is the coauthor of scientific papers related to e-learning, knowledge, Web, and grid technologies. At present, he is an Assistant Professor with the University of Salerno, and he is focusing his research activities on technology enhanced learning, Semantic Web, and ontology-based systems.

Stefano Paolozzi received the Ph.D. degree in computer science engineering from Roma Tre University, Rome, Italy, in 2009, under the supervision of Prof. Paolo Atzeni.

He collaborated with the Multi Media and Modal Laboratory of the National Research Council of Italy up to November 2008. In the last years, he has worked in collaboration with the Centro di Ricerca in Matematica Pura ed Applicata on the definition of innovative e-learning solutions based on ontologies. He is the coauthor of several scientific papers in international conferences, journals, and books. He is also involved in both national and European Union research projects related to Semantic Web and knowledge management. He is currently a Postdoctoral Fellow with the Dipartimento di Informatica e Automazione, Roma Tre University, and his research is mainly focused on model management, multimodal interaction, multimedia database, ontology alignment, ontology versioning, and semantic data management.

Saverio Salerno received the B.S. degree (*cum laude*) in mathematics from the University of Pisa, Pisa, Italy, in 1974.

From 1990 to 2000, he was a Full Professor of mathematical analysis with the Faculty of Engineering, University of Salerno, Fisciano, Italy, where he has been a Full Professor of operations research since 2000. He is the President of the Scientific Committee in support to the actions of the President of the Province of Salerno and a delegate of the Rector of the University of Salerno for the e-learning. He has been a Foundation Member and the Director, up to October 2002, of the Consortium "Centro di Ricerca in Matematica Pura ed Applicata" and responsible for many European and Italian research projects. He is the author of numerous works published on international journals and proceedings regarding mathematical analysis, simulation, didactic methodologies and technologies, and computer science. He has been an Italian Delegate for the Information Society Technologies program of the European Union 5th Framework Programme and an Expert of the European Commission for Education and Training in the framework of the information and communication technology. He is also the Coordinator of the Centre of Excellence on Learning and Knowledge and a Member of the Technical-Scientific Committee for the Industrial Research at the Ministry of Education, University and Research (MIUR).