Cerno: Light-Weight Tool Support for Semantic Annotation of Textual Documents

Nadzeya Kiyavitskaya[1], Nicola Zeni[3], James R. Cordy[2], Luisa Mich[3] and John Mylopoulos[1,4]

[1]Dept. of Information Engineering and Computer Science, University of Trento, Italy
{nadzeya,jm}@disi.unitn.it

[2]School of Computing, Queens University, Kingston, ON, Canada, cordy@cs.queensu.ca

[3]Dept. of Computer and Management Sciences, University of Trento, Italy
{luisa.mich, nicola.zeni}@unitn.it

[4] Dept. of Computer Science, University of Toronto, ON, Canada
jm@cs.toronto.edu

Corresponding Author: Nadzeya Kiyavitskaya
E-mail: nadzeya@disi.unitn.it
Phone: (+39) 3403283936
Address:
Data and Knowledge Management Laboratory,
Dipartimento di Ingegneria e Scienza dell'Informazione (DISI),
Università degli Studi di Trento,
via Sommarive 14, Povo – Trento, 38100, Italy

Abstract. Enrichment of text documents with semantic metadata reflecting their meaning facilitates document organization, indexing and retrieval. However, most web data remain unstructured because of the difficulty and the cost of manually annotating text. In this work, we present *Cerno*, a framework for semi-automatic semantic annotation of textual documents according to a domain-specific semantic model. The proposed framework is founded on light-weight techniques and tools intended for legacy code analysis and markup. To illustrate the feasibility of our proposal, we report experimental results of its application to two different domains. These results suggest that light-weight semi-automatic techniques for semantic annotation are feasible, require limited human effort for adaptation to a new domain, and demonstrate markup quality comparable with state-of-the-art methods.

Keywords: 8.4 Semantic Web; 23 Natural Language for DKE; 20.1 Re- and reverse engineering; semantic annotation

1. Introduction: The Semantic Annotation Challenge

The Information Revolution has brought about wonderful new opportunities for humans and organizations alike. At the same time, it has created an unprecedented and growing information overload. The Semantic Web initiative [1] aims to address this problem by rendering web data accessible to both humans and software agents. This is to be achieved by enriching web data with *semantic annotations*. These are similar to XML annotations in that they structure text into (usually hierarchically organized) text fragments, as in

```
<person> Paolo Bonino <residence> lives in Trento </residence> and
works at the <work> University of Trento </work> </person>
```

Unlike XML annotations, however, semantic annotations come with definitions of the annotations used, e.g., “person”, “residence”. These definitions may be given in terms of an ontology that employs a rich modeling language such as OWL, or in terms of a conceptual schema using UML class diagrams or the Entity-Relationship model. In the rest of the discussion, we call the collection of definitions of the annotations used a *semantic model.*

Annotations assign a meaning to web text fragments, thereby facilitating their processing, for example, populating a database by extracting data from text. The main thrust of the Semantic Web is exactly this point: web data should be bundled along with their semantics, provided by a semantic model (ontology or conceptual schema). Unfortunately annotating web data to render them “semantic” is an expensive, laborious and error-prone process and only accounts today for a small fraction of web data. Accordingly, much research effort is being invested in developing annotation tools that automatically, or semi-automatically (with a human in the loop) annotate web data with respect to a semantic model.

The main objective of this work is to explore the use of light-weight tools and techniques for the semi-automatic annotation of web data. Our framework is light-weight in two ways. Firstly, the semantic model used is defined in terms of UML class diagrams. As such, it is less expressive than ontologies defined in terms of description logics, such as OWL. Secondly, our framework analyzes text to determine where to introduce annotation by exploiting software source code analysis tools and techniques from Software Engineering (more precisely, Reverse Engineering). Conceptual modeling techniques are less expressive than their AI cousins, but have a huge advantage over their cousins: they have been used in research and practice for decades and have passed the test of usability and scalability. The same applies for code analysis techniques over Natural Language Processing (NLP) techniques. The latter involve large (syntactic) rule sets and special mechanisms for dealing with anaphora and other natural language phenomena. Code analysis techniques, on the other hand, use small sets of context-dependent rules and rely on elaborate, experimentally-determined processes to deliver good results even when their input includes millions of lines of code.

But of course, light-weight techniques have an advantage over their counterparts only when they deliver good enough results. Accordingly, along with the development of the framework, called Cerno[1], much of our work has focused on applications to several case studies with a range of experimental results. Two of these are reported here.

The Cerno framework consists of (i) a systematic process for defining keyword and grammar-based rules for identifying instances of concepts in a text, and (ii) an architecture based on software design recovery for applying the rules to mark up and

[1] The name ‘Cerno’ is derived from the Latin word “cernere”, meaning “to distinguish”, “to separate”, “to sift.”

extract identified instances in a document set. The case studies we report on involve accommodation advertisements in cities of art, and analysis of the content of tourist board web sites. For the first experiment, we used a simple semantic model, derived from a set of user queries. In the second experiment, the annotation process is based on a more elaborate model derived from expert requirements.

Our conclusions from these experiments are positive and suggest that light-weight techniques have a place in the transition from the Web to the Semantic Web, especially so in the annotation of bulk data.

There are several requirements that had to be addressed in designing Cerno:

- Adaptability to a diversity of document formats and domains. In order to make the framework less dependent on domain changes, domain- and document-dependent modules can be factored out from the framework's core.
- Portability of the framework across domains. External data resources, such as manually annotated training corpora, are usually unavailable, and are expensive and laborious to develop from scratch. Accordingly, domain portability has been a major requirement in our work, meaning that we want to be able to port as many of the components as possible from one domain to another.
- Accuracy and efficiency. Our framework is intended for large textual documents that need to be analyzed quickly, but not necessarily very accurately.
- Scalability. Large scale with respect to any dimension – e.g., grammar coverage, vocabulary, domain modeling, adaptation – cannot be achieved without investing large amounts of human and computational resources. We addressed this trade-off in an incremental way, by iteratively generalizing our framework with every new application to a novel domain.
- Evaluation. Performance of a semantic annotation tool needs to be assessed on a standard dataset. However, such benchmarks are not available. Moreover, evaluation is difficult because humans differ on how to annotate the same data. We elaborate on the problems of establishing an evaluation framework and discuss the choices made for assessing the performance of our approach.

As indicated earlier, this paper presents two case studies involving different domains. During these studies, the domain-dependent components were adjusted to reduce human work required to tune the tool for a different domain. Moreover, we have explored the possibility of reusing existing resources, e.g., thesauri and dictionaries, to support humans in constructing domain models needed by the Cerno framework.

The two fields of semantic text annotation and code analysis seem worlds apart, but actually share common goals and approaches. Similarly to the domain of web data, legacy software source code analysis became prominent a few years ago, thanks to the infamous "Year 2000" problem [2]. More specifically, the main goal of the software *design recovery* process is understanding the structure of a software system by interpreting source code fragments in formal terms described by a grammar and a domain model [7]. Accordingly, the two tasks have the following similarities:

- *The need to interpret the source text according to a semantic model.* In fact, in both areas the goal is to classify textual artifacts – a system's components in design recovery and instances of semantic concepts in semantic annotation – with respect to the elements of a semantic model, which describes either system design or web data subject matter.
- *The need for robust parsing techniques.* Real documents and even software code do not always match the grammars of the languages they are written in. For instance, software code understanding can be particularly difficult in such cases as web applications due to language dialects, the presence of multilingual code (e.g., an HTML page containing varied scripts), syntax errors or other unexpected content [35]. Robust parsing techniques can efficiently address such problems, permitting complete processing of a document, even when segments of the document can't be parsed and/or annotated.
- *Semantic clues drawn from a formal model of the domain.* In both areas, identification of facts or concept instances is guided by the semantic model.
- *Contextual clues drawn from the syntactic structure of documents.* In design recovery as well as in semantic annotation, structural elements are recognized by using grammatical knowledge.
- *Inferred semantics from exploring relationships between identified semantic entities and their properties, contexts and related other entities.* Once basic artifacts have been identified, design recovery processes proceed with discovery of relationships between them. A similar approach is followed in semantic annotation methods, where identified information items need to be related through semantic relationships.

Our evaluation of both experimental studies uses a three-stage evaluation framework which takes into account:

- accuracy measures used for evaluation of information extraction systems, such as Recall, Precision, and F-measure;
- productivity, i.e., the fraction of time spent for annotation when the human is assisted by our tool vs. time spent for manual annotation "from scratch"; and
- a calibration technique which recognizes that there is no such thing as "correct" and "wrong" annotation, as human annotators also differ among themselves on how to annotate a given document.

Elements of this research have been presented in a series of conference papers. For instance, an early version of the framework was presented in [19]. Some excerpts of the results of the first case study have been presented in [45], as well as the second case study [20]. This paper extends and integrates these earlier works with further experiments, analyses and discussion to offer a complete account of the Cerno framework.

The rest of the paper is structured as follows. Section 2 describes the challenges of semantic annotation task and provides an overview of work in the area. The semantic annotation method we propose is introduced in Section 3. Sections 4 and 5 discuss the experimental setup and evaluation method, while the evaluation results are presented in Section 6. Finally, conclusions and directions for future work are presented in Section 7.

2. Overview of Semantic Annotation Tools

The semantic annotation problem comprises tasks that have been traditionally researched in diverse areas, such as Information Extraction, Data Mining, and Document Categorization. For this reason, existing semantic annotation approaches differ not only with respect to the text analysis technique used, but also in formulating the annotation task. Dissimilarities – in terms of different semantic model adopted, expected document format, and domain considered – make comparison of semantic annotation tools very difficult. The present survey highlights such differences and focuses only on tools that have a substantial degree of automation, ignoring authoring environments for manually generating semantic annotations.

Several methods use structural analysis to annotate web documents. Among these is SemTag [8], which performs automated semantic tagging of large corpora. SemTag annotates text with terms from the TAP ontology, using corpus statistics to improve the quality of annotations. The TAP ontology contains lexical and taxonomic information about a wide range of named entities, as for instance, locations, movies, authors, musicians, autos, and others. SemTag detects the occurrence of these entities in web pages and disambiguates them using a Taxonomy Based Disambiguation (TBD) algorithm. A large-scale evaluation was fulfilled on a set of 264 million web pages; the total time required for annotation was 32 hours. The performance was estimated based on 200 manually evaluated text fragments. Approximately 79 percent of the annotations were judged to be correct with accuracy of about 82 percent. Human intervention was required at the disambiguation stage to tune the algorithm.

Another semantic annotation environment framework is CREAM (CREAtion of Metadata) [16]. Annotations are generated either manually – by typing or in a drag-and-drop manner, associating instances with the concepts appearing in the ontology browser, – or semi-automatically – by using wrappers and information extraction components. OntoAnnotate and OntoMat Annotizer are different implementations of the CREAM framework. In particular, OntoMat Annotizer is a user-friendly tool that exploits the structure of HTML documents to infer annotations and helps the analysts to collect knowledge from documents and Web pages and to enrich an ontology with metadata. OntoMat uses regularity in a collection of documents to quickly induce extraction rules from few human annotations. Therefore, the input collection must be very consistent in its structure. OntoAnnotate is the commercial version of OntoMat. CREAM is well suited for highly structured web documents, while for annotating HTML pages of a less structured nature, SemTag is more appropriate.

Wrapper induction methods such as Stalker [26] and BWI [13] try to infer patterns for marking the start and end points of fields to extract. Wrappers mine the information using delimiter-based extraction rules. When the learning stage is over, complete phrases related to the target concepts are identified. The biggest advantage of wrappers is that they need a small amount of training data. Alternatively, extraction rules for wrappers can be specified manually, as it is done in the World Wide Web Wrapper Factory [34]. However, whether they are generated manually or semi-automatically, wrappers strongly rely on document structure and work best for collections of rigidly structured Web pages.

The semantic annotation task tackles the same class of text analysis problems that Artificial Intelligence- (AI)-based NLP attacked in the early 80's. The next class of tools build on this legacy to offer novel linguistic analysis for generating semantic annotations. To achieve good quality results, such tools employ NLP methods that require large computational and/or memory resources. The KIM (Knowledge and Information Management) platform [17] is an application for automatic ontology-based annotation of named entities. Similar to SemTag, KIM focuses on assigning to the entities in the text links to their semantic descriptions, provided by an ontology. The analysis is based on GATE (the General Architecture for Text Engineering) [15]. KIM recognizes occurrences of named entities from the KIMO ontology that, apart from containing named entity classes and their properties, is pre-populated with a large number of instances. The generated annotations are linked to the entity type and to the exact individual in the knowledge base. Evaluation of KIM was performed over a 1 Mb text corpus annotated by humans with the traditional named entities types: *Organization*, *Location*, *Date*, *Person*, *Percent*, and *Money*. Similar to GATE, that provides separate modules to facilitate integration with different applications, Cerno represents a pipeline modular architecture allowing users to combine tools for each text analysis task they may want to address. The ontology-based gazetteer processing module of GATE supports editing of the gazetteer lists, which reminds creation of an annotation schema in Cerno. However, the nature of Cerno's modules is different, as it doesn't employ any linguistic analysis, such as part of speech recognition or chunk parsing, instead relying on the TXL engine.

Along similar lines, Unstructured Information Management Architecture (UIMA) [11] uses the same kind of linguistic analysis as in GATE to associate metadata with text fragments. OpenCalais [28] goes beyond using natural language processing and machine learning tools to recognize named entities in textual content, and also provides links between related knowledge pieces.

Armadillo is a system for unsupervised automatic domain-specific annotation on large repositories [4]. This tool employs an adaptive information extraction algorithm, learning extraction pattern from a seed set of examples provided, and generalization over the examples. Learning is seeded by extracting information from structured sources, such as databases and digital libraries, or from a user-defined lexicon. Already wrapped information from a database is exploited in order to provide automatic annotation of examples for the other structured sources or free text. Retrieved information is then used to partially annotate new set of documents. Then, the annotated documents are used to bootstrap learning. The user can repeat the bootstrapping process until obtaining the annotations of the required quality. Armadillo has been used in a number of real-life applications: mining web sites of Computer Science Departments, information extraction about artworks by famous artists, and the discovery of geographical information.

Pankow (Pattern-based Annotation through Knowledge on the Web) is an unsupervised learning method for annotation Web documents based on counting Google hits of instantiated linguistic patterns [3]. The approach uses linguistic patterns to identify ontological relations and the Web as a corpus of training data to bootstrap the algorithm and thus overcoming the problem of data sparseness.

Apart from structure- and NLP-based approaches, there is another group of tools that uses pattern-based extraction rules for semantic annotation. KnowItAll is a system for large-scale unsupervised named entities recognition, which uses domain-independent rules to locate instances of various classes in text [10]. The tool aims at extraction and accumulation of basic facts, as person's names, from the large collections of Web documents in an unsupervised, domain-independent, and scalable manner. The tool doesn't need a set of seeds; instead it relies on automatically generated domain-specific extraction rules. The rules are based on a set of keywords, for example, a rule "cities such as" is used to automatically detect the instances of the entity *City*. Then, after having instantiated the initial set of seeds, the tool induces syntax-based extraction rules.

The ontology-based method of Wessman *et al.*, Ontos, is aimed at processing preferably semi-structured Web pages with multiple records and relies primarily on regular expressions [39]. The evaluation of Ontos was performed for the set of obituaries from two newspapers containing a total of 25 individual obituaries annotating both named entity and general concepts. As a result, dispersed values for recall and precision for different concepts were obtained; both overall recall and precision varied from 0 to 100 percent.

In contrast to NLP-based approaches, our approach does not utilize any linguistic patterns, but combines keyword- and structure-based annotation rules. In this sense, our technique is light-weight. Moreover, unlike most of the methods discussed in this section whose the success is largely determined by their focus on identifying and classifying various named entities, our approach is not restricted to any particular set of concepts or to any particular ontology. From our perspective, the semantic model may differ depending on the task or type of document. Therefore, it seems appropriate to develop a method that can be easily adapted to different application areas. In Cerno, the application of patterns to documents is similar to Wessman's approach that relies primarily on regular expressions to identify instances in structured web sites. Our approach combines context-free robust parsing and simple word search to annotate relevant fragments of unstructured text. The method applies a set of rules constructed beforehand to guide the annotation process. Some of these rules are actually generic and therefore reusable. Wrapper induction methods also relate well to our work. When these methods are applied, their effect is quite similar to our results, identifying complete phrases relevant to target concepts. However, we achieve the result in a fundamentally different way – by predicting start and end points using phrase parsing in advance, rather than phrase induction afterwards.

There are tools that use reverse engineering technique to identify transaction service elements in existing electronic services, as for instance SmartGov [38]. The identification of artifacts, such as the input area, group element and their properties, is realized using a set of heuristic rules that strongly rely on the HTML structure of a webpage. Instead, Cerno exploits reverse engineering techniques to deal with unstructured textual content. Artifacts extracted by SmartGov are largely determined by webpage design, thus the system deals with traditional reverse engineering tasks. Cerno, on the other hand, aims at identifying and classifying text fragments that could be previously dealt with only by heavyweight linguistic analysis.

Concluding our summary of related work, we can say that, in contrast to all listed approaches, our method uses context-independent parsing and does not rely on any specific input format.

3. Method

Over several decades, the software source code analysis area has accumulated a wealth of effective techniques for addressing some of the problems that semantic annotation faces now. In order to cope with the Year 2000 problem, some techniques for automating solutions utilized *design recovery*, the analysis and markup of source code according to a semantic design theory [2]. Formal processes for software design recovery utilize a range of tools and techniques designed and proven efficient to address these challenges for many billions of lines of software source code [5]. One of these is the generalized parsing and structural transformation system TXL [6], the basis of the automated Year 2000 system LS/2000 [7]. Given the need for cost-effective tools supporting the semantic annotation process, we propose applying a novel method based on software code analysis techniques for the task of semantic annotation of natural language texts. In this section, we describe the process and the architecture for the processing of documents so that they can be annotated with respect to a semantic model.

3.1 TXL as an Instrument for Semantic Annotation

TXL [6] is a programming language for expressing structural source transformations from input to output text. The structure to be imposed on input text is specified by an ambiguous context free grammar. Transformation rules are then applied, and transformed results are represented as text. TXL uses full backtracking with ordered alternatives and heuristic resolution, which allows efficient, flexible, general parsing. Grammars and transformation rules are specified in a by-example style.

The transformation phase can be considered as term rewriting, but under functional control. Functional programming control provides abstraction, parameterization and scoping. TXL allows grammar overrides to extend, replace and modify existing specifications. Grammar overrides can be used to express *robust parsing*, a technique to allow errors or exceptions in the input not explained by grammar. Overrides can also express *island grammars*. Island parsing recognizes interesting structures, “islands” in a “sea” of uninteresting or unstructured background. TXL also supports *agile parsing* – customization of the parse to each individual transformation task.

Originally, TXL was designed for experimenting with programming language dialects, but soon it was realized it could be useful for many other tasks, such as static code analysis (of which an important application is design recovery), interpreters, preprocessors, theorem provers, source markup, XML, language translation, web migration, and so on. TXL has also been successfully used for applications other than computer programs, for example handwritten math recognition [43], document table structure recognition [44], business card understanding [27], and more.

3.2 The Cerno Architecture

The architecture of Cerno is based partly on the software design recovery process of the LS/2000 system [7], although in that case the documents to be analyzed were computer programs written in formal programming languages, and the markup process was aimed specifically at identifying and transforming instances of Year 2000-sensitive data fields in the programs. The Cerno adaptation and generalization of this process to arbitrary text documents includes four steps: (1) document parse, (2) recognition of basic facts, (3) their interpretation with respect to a domain semantic model, and (4) mapping of the identified information to an external database. Each of the processing steps has been re-implemented according to the peculiarities of the text annotation task. Thus, the Cerno architecture includes three new blocks: *Parse*, *Markup* and *Mapping* that semantically correspond to steps (1)-(2), (3) and (4) of the design recovery process, respectively.

1. Parse. To begin, the system breaks down raw input text into its constituents, by producing a parse tree. In contrast to code design recovery techniques, the parse tree produced by Cerno is composed of natural language document fragments such as document, paragraph, phrase, word, rather than program, function, expression, etc. Any of these fragments may be chosen by the user as an annotation unit, depending on the purpose of annotation. In order to properly recognize sentence boundaries, this step also identifies basic word-equivalent objects such as e-mail and web addresses, phone numbers, enumeration indices, and so on. All input structures are described in an ambiguous context-free TXL grammar using a BNF-like notation (see Figure 1). Note that the standard non-terminal “program” stands for the root element of any input to the TXL engine. The TXL engine automatically tokenizes and parses input according to the specified grammar, resulting in a parse tree represented internally, as in the example shown in Figure 2. If necessary, the recognition of document structure may also involve a different visualization of document elements, for instance, indentation of paragraphs, normalization of monetary amounts, and other similar operations. TXL’s parsing is essentially different from the parsing that NLP tools normally do, because TXL recognizes structures based on identifiers, numbers, and symbols. Linguistic parsing instead recognizes linguistic constructs such as nouns, verbs, adjectives, etc. and therefore needs a dictionary of valid word forms and a disambiguation step. An example of the output obtained using the document grammar recognition of the first stage is shown in Figure 3.

```
% nonterminal types are enclosed in square brackets
define program
   [repeat ad]
end define

% sequences of one or more non-terminals are defined by using repeat
define ad
   [repeat sentence] [repeat newline+]
end define

define sentence
   [repeat word] [fullstop]
end define

% vertical bars are used to indicate alternate forms
define word
   [object] | [number] [shortform]  |  [id]  |  [not word] [token]
end define

define object
     [email] | [money] | [phone] | [webaddress]
end define

% terminal symbols are denoted by a single opening quote
define fullstop
   '.
end define
```

Figure 1. A fragment of the document grammar

```
<program>
 <repeat_ad>
  <ad>
  <repeat_sentence>
   <sentence>
     <repeat_word>
       <word><id>Very</id></word>
       <word><id>elegant</id></word>
       <word><id>apartment</id></word>
        <...>
       <word><id>phone</id></word>
       <word><id>to</id></word>
       <word><object><phone>
             <localnumber>
              <anynumber><number>111</number></anynumber> .
              <longnumber>1111111</longnumber>
             </localnumber>
       </phone></object></word>
   </sentence>
  </repeat_sentence>
  </ad>   <…>
 </repeat_ad>
</program>
```

Figure 2. A part of the TXL parse tree produced by the first stage.

In the example, the input is parsed into a sequence of ads, whereby each "ad" consists of sentences, each "sentence" is a sequence of words. A "word" can be an identifier (non-terminal "id"), number or object (non-terminal "phone"), a shortform ("e.g.", "tel.", "p.m.") or any other input symbol.

```
Very elegant apartment located in Piazza Lante, just a walk from Fosse Ardeatine and
10 minutes to Colosseum by bus(Bus stop in the square). 75 smq in a charming, and
full furnished environment. The apartment has a large and well-lit living room with  sofa
bed a dining area, a large living kitchen with everything you need, a bathroom with tub,
a large double bedroom. TV, hi-fi and a washing machine. <money>1.200
euro</money> a month, utilities not included. Write to
<email>pseudonym@somewhere.it</email>  or phone to
<phone>111.1111111</phone>
```

Figure 3. First stage output: word-equivalent objects are recognized

2. Markup. This stage recognizes instances of concepts, i.e., annotates text units that contain relevant information according to an *annotation schema*. This schema is a structure that includes a list of concept names and the domain-dependent vocabulary, that is, syntactic indicators related to each concept. Cerno assumes that the annotation schema is constructed beforehand, either automatically using learning methods or manually in collaboration with domain experts. Indicators can be single literals, e.g., "Euro" or "tel.", phrases, e.g., "is not required to", or names of parsed entities, for instance, "e-mail" or "money". They also can be *positive*, i.e., pointing to the presence of the given concept, or *negative*, i.e., pointing to the absence of this concept. The systematic process for annotation schema construction is discussed in section 3.3. If one of the indicators from the list is present in a text fragment, the fragment is marked up with a corresponding tag name. If necessary, the decision-making criteria can be refined to specify more complex criteria. For instance, at least *N* words, or *P* percent of the wordlist must appear in a single text unit in order to consider it pertinent to a given concept, or other triggering criteria. The text processing in this stage exploits the structural pattern matching and source transformation capabilities of the TXL engine similarly to the way it is used for software markup to yield an annotated text in XML form.

3. Mapping. In the last stage, which is optional, annotated text units selected from all annotations according to a predefined database schema template are extracted to be stored in an external database. The schema template represents a target structure to accommodate automatic annotations. An example of a DTD for database schema templates is shown in Figure 4. The template is manually derived from the domain-dependent semantic model, as discussed later for case studies in Section 5, and represents a list of fields of a target database. Sentences and phrases with multiple annotations are "cloned", i.e., copied for each field. In this way we do not prejudice one interpretation as being preferred. To recognize and copy the annotated fragments according to the given template, this step uses a schema grammar, which is domain-independent. The final outputs are both the XML marked-up text (Figure 5) and the populated relational database (a fragment of a filled XML-document corresponding to the above database schema template is shown in Figure 6).

```
<ad>
      <location></location>
      <price></price>
      <contact></contact>
      <facility></facility>
      <term></term>
      <type></type>
</ad>
```

Figure 4. Database schema template for accommodation ads.

```
<type><location> Very elegant apartment located in Piazza Lante, just a walk from Fosse
   Ardeatine and 10 minutes to Colosseum by bus (Bus stop in the square) </location></type>.

<facility> 75 smq in a charming, and full furnished environment </facility>.

<type><facility> The  apartment has a large and well-lit living room with sofa bed a dining area, a
   large living kitchen with everything you need, a bathroom with tub, a large double bedroom
   </facility></type>.

<facility> TV, hi-fi and a washing machine </facility>.

<facility><price> 1.200 euro a month, utilities not included </price></facility>.

<contact> Write to pseudonym@somewhere.it or phone to 111.1111111 </contact>
```

Figure 5. Example result XML-marked up accommodation ad

Low-level objects such as *email* and *phone numbers*, while recognized and marked-up internally, are intentionally not part of the result since they are not in the target schema (see Figure 4). Phrases that contain information related to more than one concept are marked-up once for each concept; notice for example *type* and *location* tags in the first sentence.

```
<?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE ads_list SYSTEM "ads.dtd">
<ads_list>
   <ad>
      <location>Very elegant apartment located in Piazza Lante, just a walk from Fosse Ardeatine
   and 10 minutes to Colosseum by bus (Bus stop in the square)</ location >
      <price>1.200 euro a month, utilities not included</ price >
      <contact>Write to pseudonym@somewhere.it or phone to 111.1111111</contact>
      <facility>75 smq in a charming, and full furnished environment
       The apartment has a large and well-lit living room with sofa bed a dining area, a large living
   kitchen with everything you need, a bathroom with tub, a large double bedroom
      TV, hi-fi and a washing machine
      1.200 euro a month, utilities not included
      </facility>
      <term></term>
      <type>Very elegant apartment located in Piazza Lante, just a walk from Fosse Ardeatine
   and 10 minutes to Colosseum by bus (Bus stop in the square)</type>
   </ad>
…
</ads_list>
```

Figure 6. An example of a populated database schema template

Figure 7 illustrates the annotation process (along the center axis) specifying domain-independent and domain-dependent components of the Cerno architecture (on the left and right hand sides respectively).

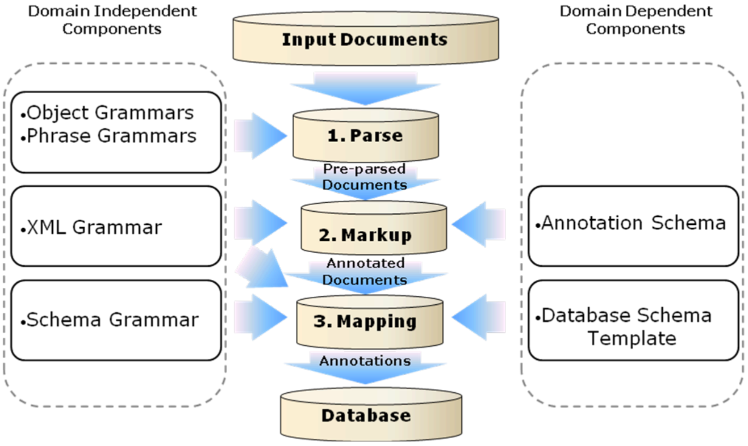


Figure 7. The semantic annotation architecture and process in Cerno

We have designed the method to be general by making the core components of Cerno applicable to different applications and semantic domains. The architecture explicitly factors out reusable domain-independent aspects such as the structure of basic word-equivalent objects, i.e., e-mail and web addresses, monetary formats, date and time formats, etc., and language structures, i.e., document, paragraph, sentence and phrase structure, shown on the left hand side. Components which vary for different domains are represented on the right hand side. They comprise the annotation schema and the corresponding vocabulary containing syntactic indicators.

3.3 Systematic Process for Annotation Schema Construction

Applying Cerno to a specific annotation task requires construction of an annotation schema which effectively represents the requirements of an annotation task. Therefore, to develop such schemas for our case studies, we have been using the systematic process similar to the common requirements engineering approaches:

- Identification of the most relevant concepts by means of discussions with the end users or domain experts;
- Enhancement of the obtained list of main concepts with background knowledge, related concepts, and synonyms using available knowledge sources, both general (for instance, encyclopedias and dictionaries) and domain-specific ones;
- Structuring the collected information in a semantic model;
- Selection of the first-class concepts that must be identified and related keyword-based annotation rules from the entities of the conceptual model.

We also envisage a refinement step in which the expert provides feedback on the quality of the obtained annotation schema to improve its quality. The reason is that the use of complementary knowledge sources may lead to enhancing the schema with too much information and eventual reduction of the annotation accuracy.

Ultimately, the improvement of the annotation schema is realized by tuning the result quality on a small training set of text. Figure 8 shows a fragment of a Cerno annotation schema for *term* concept at two different time points of the refinement phase.

Starting specification for *term* concept:

term : date january february march april may june july august september october november december jan feb febr mar apr jun jul aug sep sept oct nov dec

Final specification for *term* concept:

term : date [rented by] minimum maximum month months short long term terms holidays holiday days lets let period periods

| { money price }

Figure 8. Evolution of the annotation schema for *term* concept

Here "date" is a word-equivalent object whose structure is manually specified using a TXL grammar pattern that recognizes both precise dates, e.g., "10.12.2001", as well as more general date expressions like "November 1999".

3.4 Implementation

The Cerno framework is designed as a multi-stage pipeline process that can be executed from the command line given that the TXL engine is installed on a PC. For instance, the following command executes the markup phrase of Cerno and the mapping phase right after completing the run of markup. The second phase reads the output of the previous phase from the standard input, that is *stdin* in Windows. Once the last phase is executed, the final results are saved in the *Output* folder:

```
txl Input\example.txt markup.txl
  |  txl stdin mapping.txl > Output\example.out
```

Alternatively, the multi-stage processing can be performed in the standard way by consecutive execution of single commands:

```
txl "Input\example.txt" markup.txl > "Temp\example.tmp"
txl "Temp\example.tmp" mapping.txl > "Output\example.out"
```

Here intermediate output files are stored in a temporal folder *Temp*. The second phrase reads its input files from Temp folder. In this way, the process of TXL transformations can be automated via shell scripts. Each processing module accepts as input a file and produces an output file. Thus, the entire process can be monitored and corrected if necessary by controlling intermediate output files and changing processing settings. This approach can be applied during the tuning phase, i.e., when adapting Cerno to a specific type of documents and a semantic model. By looking at intermediate outputs, the user may get some hints on how to improve the grammar and obtain better results.

Domain-independent grammars used at different phases are called from TXL files using a special directive *include* and employed at the compile time, for example:

```
include "Grammars/category.grm"

include "Grammars/schema.grm"
```

Instead, domain-dependent components are read from files on-the-fly. The following line of TXL code creates a new list of categories by reading them from *ads.cat* file:

```
construct MyCategories [repeat category]

     _ [read "Categories/ads.cat"]
```

In order to recognize word-equivalent objects during the Parse phase, we exploit the TXL parse tree of input and infer object annotation based on the type of an object in the tree. This grammatical recognition of some objects of interest is very rapid in TXL. The following two rules embody the first processing step:

```
rule markupObjects
        skipping [markup]
        replace $ [repeat word]
                E [object] Rest [repeat word]
        % object is identified, but its type, like email or phone number,
        %  is further determined by the getType rule
        where
                E [getType]
        % a type of the object is received by using ‘import’ directive
        import TypeId [id]
        % object is marked up with its type tag
        by
                E [markup TypeId] Rest
end rule

function getType
        match * [any]
           A [any]
        construct TypeId [id]
           _ [typeof A]
        % look for an embedded type of ‘object’ in the parse tree
        deconstruct not TypeId
           'object
        export TypeId

end function
```

Figure 9. The first phase processing of the Cerno architecture

Similar rules are used during the Markup phase for annotation of larger text fragments. Recognition of relevant concepts is realized by combining keyword search with grammatical patterns. For example, the annotation schema may incorporate complex patterns, such as:

```
call ( object_phone | object_person | us )
```

where "object_phone" and "object_person" are word-equivalent objects provided by the parse phase, i.e., any phone number and personal name in this case, while "call" and "us" are keywords. This pattern has also a list of alternative choices, thus matching phrases like "call Alice", "call mr. Johnson", "call 000-1111111", "call us". Such combined patterns make Cerno's semantic annotation very powerful.

3.5 Discussion

The proposed process has a number of advantages. Generally speaking, it is domain-independent and doesn't rely on document structure. For instance, in contrast to wrapper induction approaches, our framework uses context-independent parsing and does not require any strict input format.

Compared to machine learning systems, the cost of human attention required is lower. The main reason for this is that there is no need for manual input either in interactive training, or in annotating an initial set of seeds. Rather, the tool is tuned to a particular domain by providing the system some hints, i.e., semantic clues, about concepts. The semantic processing itself is light-weight: wordlist- and pattern-based. Another important benefit of Cerno is that the human engaged to modify or replace domain dependent knowledge, need not be an expert in the TXL language or possess specific programming skills. He or she can quickly realize the necessary modifications and test Cerno to obtain new results.

Given that Cerno doesn't employ full linguistic parsing, this approach renders Cerno tolerant to ungrammatical, erroneous input. At the same time, the shallow processing involved in each step of the process enhances scalability, an issue of particular concern in the Semantic Web.

In addition, Cerno supports the reuse of domain-independent components, factored out by the Cerno architecture, as with some other annotation tools, such as GATE. This feature means that domain-independent knowledge and the core of Cerno remains unchanged when applying the tool to a new domain.

4. Case Studies

Our empirical studies include two experiments: (1) *proof-of-concept experiment*: validation of the feasibility of the new method, performed on the relatively restricted domain of on-line accommodation advertisements; (2) *test-of-generality experiment:* verification of the scalability of our method to larger documents and more complex semantic models; for this experiment we analyzed the contents of tourist board web sites.

Both case studies belong to the tourism sector, a sector that is a broad in the concepts it covers and rich in the data that can be found on the web [21]. The complexity of the tourism sector is reflected in the concept of tourism destination [12] that has to be described as a composition of services belonging to different domains. Besides travel operators, hotels and restaurants, contributing services comprise: sports (covering activities, competitions, courses, facilities), transportation (destinations, transportation means, timetables, terminals), culture and history (history, cultural heritage, places to visit, cultural events, local traditions, holidays, customs), and medicine (medical services and treatments of the resort). Being such a broad sector both in terms of content and challenges, tourism constitutes a rich evaluation testbed for semantic annotation tools.

4.1 Evaluation Challenges

Assessment of output quality poses a great challenge for semantic annotation tools. The output of these tools is often assessed similarly to information extraction systems by comparing results to a “ground truth”, i.e., a *reference annotation*, and calculating standard quality metrics such as recall and precision. Thus, evaluation benchmarks for some of the information extraction tasks were created in the framework of the Message Understanding Conferences (MUC) [25]. To our knowledge, such standard benchmark evaluation tasks and competitions for semantic annotation systems have yet to be established. In case of named entities, such as *Person*, *Organization*, *Location*, we can assume the existence of a standard, shared reference annotation. While for more complicated annotation schemas, we cannot rely on this assumption, because human opinions on the correct markup vary widely. Ideally, we should compare the automated results against a range of high quality human opinions. However, in practice the cost of the human work involved is prohibitive for all but the largest companies and projects.

Manual construction of the reference annotation is complicated because of many issues: (i) annotators must be familiar with the domain of the documents of interest, the language the documents are written in and the annotation model; (ii) there is no guarantee that they are consistent throughout the entire corpus, because different parts can be processed by different people or because an annotator has changed her opinion about some concepts during the process; (iii) the process is costly, time-consuming, and error-prone due to human limitations (tiredness, lack of attention, incapacity to memorize a large number of concepts, etc.). Nevertheless, even annotations accurately crafted by experts in a given domain, native-speakers, well acquainted with the annotation process, may differ because the annotation process includes a healthy element of human discretion.

Another problem is choosing the right evaluation criteria given that an annotation tool can be evaluated in terms of quality of the output, processing speed, and other performance aspects. From an engineering perspective, the most important criteria are the *effectiveness* of the tool in terms of quality of results and the *productivity* of human annotators who use the too. The motivation underlying this assumption is that results of poor quality would require a human expert to manually revise the annotated documents and potentially repeat the entire annotation process. As for productivity, this aspect is used to evaluate the tool in terms of the effort saved by using the tool.

The quality of results provided by a semantic annotation system can be measured by several metrics adopted from Information Retrieval [42]. In most cases, and in the present work, it is assumed that any given text fragment in a collection is either pertinent or non-pertinent to a particular concept or topic. If the annotated fragment differs from the correct answer in any way, its selection is counted as an error. Thus, there is no scale of relative relevance that is considered by some approaches [23],

[24], [14]. For a given concept, retrieving a relevant piece of information constitutes a hit, whereas failing to retrieve a relevant fragment is considered as a miss. Finally, returning an irrelevant fragment constitutes a false hit. Accordingly, in our case studies we were interested in assessing the quality of answers returned by the system using the following six metrics. *Recall* is a measure of how well the tool performs in finding relevant information; *Precision* measures how well the tool performs in not returning irrelevant information; *Fallout* is a measure of how quickly precision drops as recall is increased, it characterizes the degree to which a system's performance is affected by the availability of a large amount of irrelevant information; *Accuracy* is a measure of how well the tool identifies relevant information and rejects irrelevant data; *Error* is a measure of how much the tool is prone to accept irrelevant entities and reject relevant ones; finally, *F-measure* is an aggregate characteristic of performance, expressed in the weighted harmonic mean of precision and recall. In our evaluation, we didn't assume priority of recall over precision or vice versa and used the traditional balanced f-measure with equal weights for recall and precision.

4.2 Evaluation Framework

To assess the results of both experimental studies, we developed a new evaluation framework. In each step of the evaluation process, we were concerned with measuring the quantitative performance measures outlined above for the tool's automated markup compared to manually-generated annotations.

Our evaluation framework consists of three main steps:

1) The first step compares system output directly with manual annotations. We assume that quality of manual annotations constitutes an upper bound for automatic document analysis. However, in order to take into account annotator disagreement and obtain a more realistic estimate of system performance, we introduce an extra step for calibrating automatic results relative to human performance.

2) The second step verifies if the use of an automatic tool increases the productivity of human annotators. We measure the time used for manual annotation of the original text documents and compare it to the time used for manual correction of the automatically annotated documents. The difference between these two measures shows how much time can be saved when the tool assists a human annotator.

3) Finally, the third step compares system results against the final human markup made by correcting automatically generated markup.

The motivation for our *calibration* technique is that the performance of any semantic annotation tool cannot be considered in isolation from corresponding human performance. In fact, a study conducted as part of MUC-5 in a domain involving technical microelectronic documents has estimated that human markers demonstrate only about 82% precision and 79% recall, while the best system achieved about 57% precision and 53% recall on the same information extraction task [40]. Therefore, in order to adequately evaluate the performance of any tool, we propose to calibrate the tool's performance against human performance. In this case, as a reference annotation for the automatic results we used the annotation provided by multiple human markers.

The proposed evaluation schema allows for comprehensive performance assessment of any semantic annotation tool. Each of the proposed stages is important because we are interested to fully assess the capabilities of the semantic annotation tool, and in particular, to what extent the tool can match human performance.

5. Experiments

5.1 Accommodation ads

The first experiment was aimed at validating feasibility of the new method for natural language documents in a limited semantic domain [19]. For this purpose, we worked with accommodation advertisements for tourist cities drawn from online newspapers (see for example Figure 10). These advertisements offer accommodation-related information provided by individual users in a free, unstructured form.

30 square meters studio apt. in Rome center near FAO. Nicely furnished,kitchen corner, bathroom, room with two windows, high floor. nice condo, lift, beautiful and quite area. 1000,00 euros per month including utilities, starting from next March. pseudonym@somewhere.it

Aventino cosy small indep. Bungalow in Green; quiet; residential neighborhood -1 bedroom, living-dining room-fully Furn./equipp. Sleeps 2-4 people Euro 73.00/83.00/93.00 p.n. Euro 825.00 month. E-mail: pseudonym@somewhere.it - Tel.: +39-XXXXXXXXX/ +39-XXXXXXXX

Figure 10. Examples of on-line accommodation advertisements

From a linguistic viewpoint, this application poses a number of problems beyond those normally present in typical natural language documents, for instance:

- partial and malformed sentences, such as "30 square meter studio apt. in Rome center near FAO.";
- abbreviations and shortforms ("Furn./equip.");
- location-dependent vocabulary: names of geographical objects, both proper nouns ("Colosseum") and common nouns ("campo");
- presence of mixed foreign language terms ("via", "Strasse", "Policlinico");
- monetary units ("Euro 73.00/83.00/93.00 p.n.", "€2000");
- varied date and time conventions ("from the 15/20th of July 2006", "from next March").

From a functional viewpoint, such advertisements are present in various kinds of web sites publishing classified ads. Given that these kinds of advertisements are unstructured, searching them is very often a real test of patience for the user who has to look for useful information in long lists of ads. In order to make a realistic test of generality of the method, we restricted our problem. In particular, we avoided using any proper names and location-dependent words; we did not pre-process the text of accommodation descriptions by normalizing their format or correcting errors, and we didn't use any formatting or structural clues to detect semantic categories.

To annotate these ads, we designed a conceptual model that represents the information needs of a tourist looking for accommodation, as shown in Figure 11.

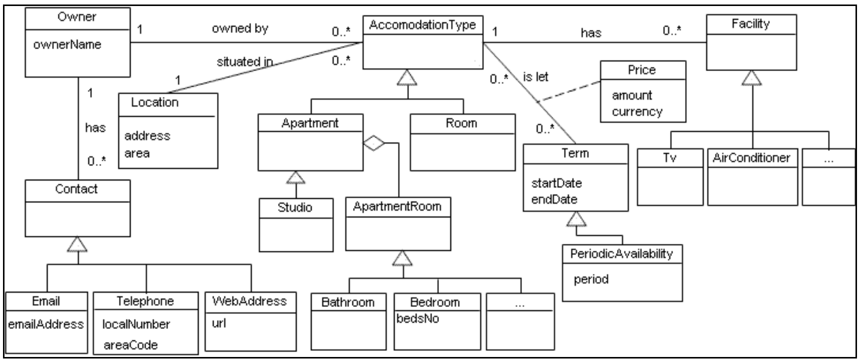


Figure 11. Semantic model for accommodation ads

The annotation schema derived from the semantic model consisted of the concepts *Type*, *Contact*, *Facility*, *Term*, *Location*, and *Price*. This schema contains information typical of the tourism sector in general and is rather standard, given the nature of the documents.

The annotation schema was manually translated into an XML database schema as input for Cerno's Mapping stage (see the template demonstrated in Figure 4). The desired result was a database with one instance of the schema for each advertisement in the input, and the marked-up original advertisements.

To adapt the semantic annotation method to this experiment, domain-related wordlists were constructed by hand on the basis of the semantic model using names of sub-classes of the relevant concepts and from a set of examples. As we described earlier, no real training took place. Rather, after encoding the target schema into the tool's vocabulary, the category wordlist was allowed to be tuned to do well on this first set by hand. The total number of annotation patterns in the vocabulary was 152.

5.2 Tourist Board Web Sites

The second case study pursued two main goals: to demonstrate the generality of the method over different domains, and to verify its scalability on a richer semantic model and larger documents.

For this purpose, we considered a sub-set of the tourist destination web sites of local Tourist Boards in the province of Trentino, Italy, that is the 13 out of 15 web sites of the Aziende di Promozione Turistica di ambito [37] that have an English version (web sites version considered is of 2007). This application presents a number of problematic issues for semantic annotation:

- free unrestricted vocabulary: as the documents are written in a free and informal style, the same concept can be expressed in many different ways;

- differently structured text: apart from being HTML pages, the documents have no common structure. Each of the local web sites uses different design and has its own navigation structure; moreover, the knowledge contained by these web sites is detailed to different extents – from general descriptions of the destination to very specific information such as pricelists, local weather forecasts and ski-bus timetables;
- foreign words even in the English version of the web pages (“malga”, “Gasthaus”).

Under these conditions, we decided that HTML structure was more hindrance rather than help in inferring semantic annotations. Therefore, we extracted plain text from the web sites and conducted our experiments on this text.

This experiment was run in collaboration with the marketing experts of the eTourism group of University of Trento [9]. The high-level goal of the study was to assess the *communicative efficacy* of the web sites. In the area of marketing, the communicative efficacy of a web site is determined by its capability to properly present a tourist destination [20]. This relates, in particular, to the degree to which the web site covers relevant information according to the strategic goals of the Tourist Board. A full description of the assessment of the communicative efficacy is beyond the scope of this paper.

Semantic annotation of the web pages is a necessary step of the project to gather input data for evaluating communicative efficacy. The goal was to identify and annotate the important information for a Tourist Board web site to be effective. To this end, we asked the experts of the eTourism group to provide a list of semantic categories and their descriptions. Then we identified concepts related to these categories and pruned them according to the domain knowledge related to the local tourism strategies, as shown in Figure 12. The final list of semantic categories is shown in Figure 13.

Category	**Description**	**Key concepts**
Geography	Comprises characteristics of the landscape (mountain, lakes, plateaus), or geologic features (type of the rocks), characteristics of the environment (natural resources, parks, protected zones, biotopes) and climate (temperature, number of sunny days, precipitations, quality of the air, altitude)	Climate Weather predictions Land Formation Lakes and Rivers Landscape

Figure 12. A fragment of the expert knowledge for Tourist Board web sites

Geography

- Climate
- Weather predictions
- Land Formation
- Lakes and Rivers
- Landscape

Local products

- Local handcrafting
- Agricultural products
- Gastronomy

Culture

- Traditions and customs
- Local history
- Festivals
- Population
- Cultural institutions and associations
- Libraries
- Cinemas
- Local literature
- Local prominent people

Artistic Heritage

- Places to visit: museums, castles, churches
- Tickets, entrance fees, guides

Sport

- Sport events
- Sport infrastructure
- Sport disciplines

Accommodation

- Places to stay
- How to book
- How to arrive
- Prices
- Availability

Food and refreshment

- Places to eat: malga, restaurant, pizzeria
- Dishes
- Degustation
- Time tables
- How to book

Wellness

- Wellness centers
- Wellness services
- Wellness facilities

Service

- Transport, schedules
- Information offices
- Terminal, station, airport
- Travel agencies

Figure 13. The list of topics related to the comminicative efficacy of a Tourist Board web site

To adapt the annotation framework to this specific task, we needed replace the domain-dependent components of Cerno. For this purpose, the initial domain-specific knowledge provided by the tourism experts was transformed into a rich semantic model. In this process we took advantage of existing two knowledge bases – WordNet [41] and an on-line Thesaurus [36] – to expand relevant domain knowledge. These resources also helped us to derive additional linguistic indicators for Cerno domain modules. The semantic model was constructed using the Protégé 3.0 ontology editor [30] and stored in RDF. The model consisted of about 130 concepts connected by different semantic relationships. A slice of the tourist destination semantic model, visualized using the RDFGravity tool [32], is depicted in Figure 14. The figure shows semantic information, such as concepts (labeled "C"), their properties (labeled "P"), and taxonomic structure. In addition, the figure shows linguistic indicators (labeled "L"), i.e., keywords or object patterns, associated with concepts. These indicators were generated later for the annotation process, but ultimately stored along with the model.

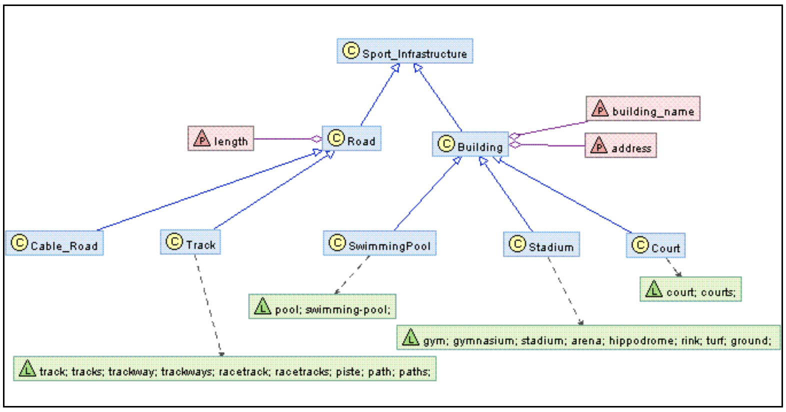

Sport_Infrastructure
building_name
length
Road
Building
address
Cable_Road
Track
SwimmingPool
Stadium
Court
court; courts;
pool; swimming-pool;
gym; gymnasium; stadium; arena; hippodrome; rink; turf; ground;
track; tracks; trackway; trackways; racetrack; racetracks; piste; path; paths;

Figure 14. A slice of the tourism semantic model

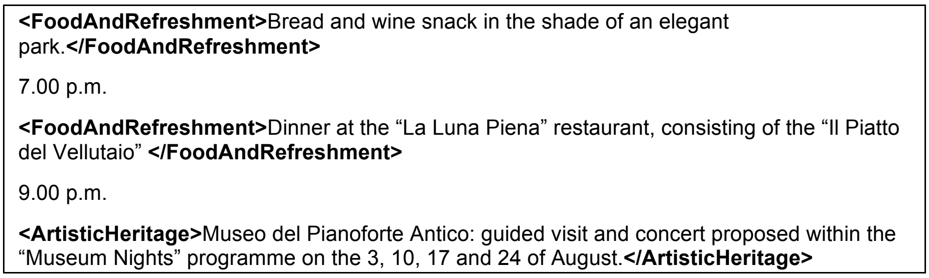
The final annotation schema was essentially a set of concepts at the general level provided by the experts: *Geography, Sport, Culture, Artistic Heritage, Local Products, Wellness, Accommodation, Food and Refreshment,* and *Service*. A database template for mapping system annotations into an external database, shown in Figure 15, was derived straightforwardly from the annotation schema.

```
<Tourism>
    <Geography></Geography>
    <Identity></Identity>
    <Culture></Culture>
    <ArtisticHeritage></ArtisticHeritage>
    <Sport></Sport>
    <Accommodation></Accommodation>
    <FoodAndRefreshment></FoodAndRefreshment>
    <Wellness></Wellness>
    <Service></Service>
</Tourism>
```

Figure 15. Database template schema for tourism web sites

The domain dependent vocabulary for the annotation schema was generated both semi-automatically and manually. Parts of it were mined from a set of sample documents and using sub-terms in the hierarchy of the domain conceptual model, as it was done in the first case study. Additional relevant synonyms were obtained from the definitions provided by WordNet [41] and the on-line Thesaurus [36]. The total number of keywords collected was more than 500. Moreover, four object patterns were reused from the previous application to grammatically recognize monetary amounts, phone numbers, e-mails and web addresses, as these structures frequently occur in the content of web sites.

In this experiment, we downloaded the 13 local Tourist Board web sites which had English versions from a total set of 15 web sites. A set of 11,742 text fragments extracted from these web sites was given to each of the two human annotators and to the tool for annotation. The required result was a database with one instance of the schema for each Tourist Board web site in the input, and the marked-up original text, see Figure 16. Once the automatic annotation was completed, we proceeded to the evaluation stage. The next section analyzes Cerno's performance for different semantic categories.



<FoodAndRefreshment>Bread and wine snack in the shade of an elegant park.**</FoodAndRefreshment>**

7.00 p.m.

<FoodAndRefreshment>Dinner at the "La Luna Piena" restaurant, consisting of the "Il Piatto del Vellutaio" **</FoodAndRefreshment>**

9.00 p.m.

<ArtisticHeritage>Museo del Pianoforte Antico: guided visit and concert proposed within the "Museum Nights" programme on the 3, 10, 17 and 24 of August.**</ArtisticHeritage>**

Figure 16. Example of XML-marked up content for top-level concepts

6. Results

6.1 Experiment 1: Accommodation Ads

According to the evaluation framework described in section 4.3, in the first stage the tool and each of two human annotators marked up a sample set of ten advertisements different from the training set used to tune the tool for the domain. The tool was then compared against each of the human markers for this set separately – see Table 1 – and then calibrated assuming each of the two human annotations as definitive, see Table 2 and

Table 3. By comparison with these two human annotators, the system exhibited a high level of recall (about 92% compared to either human, higher than either human compared to the other), but a lower level of precision (about 75% compared to either human, whereas they each exhibit about 89% compared to the other). However, the system was able to show a 92% accuracy rating compared to either human, extremely high for such a light-weight system.

Table 1. Evaluating system annotation vs. humans

Measure	**System vs. A1**	**System vs. A2**
Recall	0.92	0.92
Precision	0.74	0.76
Fallout	0.08	0.08
Accuracy	0.92	0.92
Error	0.08	0.08
F-measure	0.82	0.84

Table 2. Recall and precision scores for each entity

Entity	**Recall (A1)**	**Recall (A2)**	**Precision (A1)**	**Precision (A2)**
Location	0.91	0.91	1	1
Facility	1	0.88	0.48	0.61
Price	1	1	0.88	0.82
Type	1	1	0.60	0.67
Term	0.50	0.67	0.57	0.57
Contact	1	1	1	1

Table 3. Calibrating system results vs. human.

Measure	**A2 vs. A1**	**System vs. A1**	**A1 vs. A2**	**System vs. A2**
Recall	0.91	0.92	0.88	0.92
Precision	0.88	0.74	0.91	0.76
Fallout	0.03	0.08	0.02	0.08
Accuracy	0.96	0.92	0.96	0.92
Error	0.04	0.08	0.04	0.08
F-measure	0.89	0.82	0.89	0.84

If we compare the performance of the tool and one of the annotators, considering as the “ground truth” the markup of the other annotator, we see that the tool retrieves more information than the humans but with lower precision. Indeed, the tool demonstrated a recall of about 92% compared to 91 and 88% for the two humans, while precision was 74–76% compared to 88 and 91% for the humans.

Considering the quality rates for different concepts, we can see that most accurate results were demonstrated for *Contact* and *Price*. This outcome is not surprising, because these concepts are relatively easy to identify, more specifically, e-mail, web address, and phone number in *Contact* and monetary amounts in *Price*. Such constructs easily allow identifying related phrases. Moreover, meaning of this particular concept is unambiguous for human annotators, this way reducing the probability of disagreement to a minimum. This type of construct is now reliably recognized by many information extraction tools, as for instance GATE and UIMA.

The fragments related to *Facility* concept were identified with a 94% average recall, which is quite good for a rather complex semantic entity. For instance, in a new text we can always encounter rarely-provided facilities, for instance “CD player”. Low precision in the results for *Facility* can be interpreted in two ways: (1) there was a large fraction of incorrect answers; or (2) human markers tended to miss information related to this concept, consequently, raising the number of false positive replies for automatic annotation. Taking into account evaluation of system performance against assisted human opinions (its summary is provided later on in Table 4), we discovered that the first hypothesis is not correct, as on these revised annotations the tool showed good performance for the *Facility* concept. Therefore, the second explanation appears more likely in this case. The problem with this concept is that even for humans it is difficult to choose which amenities listed in the ads were in fact accommodation facilities. For example, this problem arises for such phrases as: "Floors, high ceilings, sleeping area on upper level, tasteful decor and fittings.", "Elevator.", "Secure and quiet building with doorman."

Identification of *Location* information shows high precision, but lower recall for both reference annotations, i.e., annotators 1 as well as 2. This can be explained by the fact that the tool did not use any location-dependent vocabulary, containing proper names related to Rome. In short and concise text, as with an accommodation ad, it is often difficult to infer the presence of *Location* information from context. Consider, for example, "Trevi Fountain, charming miniapartm." Therefore, for eventual commercial application, a location-dependent vocabulary can be employed in the tool to achieve a higher recall.

Identification of *Term*-related information turned out to be the most difficult task. The problem is that availability information is not always expressed by exact date, but intended implicitly, therefore is more difficult to predict. For example, phrases like "Studio available for holidays" and "Reductions for long term stay" implicitly contain information about the possible period for rent, but on the other hand may be ignored as not being specific enough. Recognition of temporal information in general is one of the most complex problems in NLP, especially so in the area of question answering [31].

The *Type* concept was identified with high recall and lower precision. The reason for the low precision rate is due to inconsistent assumptions underlying the automated and manual annotations. Human markers actually skipped many items related to this concept. For instance, the word "apartment" related to the type of accommodation often appears in the text of an ad several times, but humans tend to annotate it only once for each ad, ignoring further repetitions of this information. Instead, the tool correctly annotated each *Type* instance across the entire text.

In the second stage evaluation, we were interested in measuring the effect of the initial automated annotation of the tool on human productivity. The time taken by an unassisted human marker to semantically annotate a new sample of 100 advertisements was measured, and compared to the time taken by the same human marker when asked to correct the automated markup created by the tool. In this first evaluation the human annotator was able to annotate 4.5 times faster with the aid of the tool (i.e., used 78% less time to mark up text with assistance than without), a significant saving. Because the system was shown in the first evaluation to be more aggressive than humans in markup, the majority of the correction work was removing markup inserted by the tool. With an appropriate interface for doing this easily, the time savings could have been even greater. Ideally, one should also add to the productivity estimation the time for programming and training Cerno. However, this rate cannot be accurately approximated from these two first case studies, in the context of which Cerno was actually designed.

In the third stage, we gave the human annotators the advantage of correcting automatically marked up text from the tool to create their markups, and compared the final human markup to the original output of the tool. For this experiment, three sets of documents were used in addition to the original training set, one new set of 10 advertisements from the same online newspaper, another set of 100 from Rome (the same city as the original set), and a new set of 10 from Venice. The summary of results is shown in Table 4. Accuracy for all of the Roman sets is about 98%, and in the new set from Venice, a completely different location, the accuracy was measured as 96% with similar precision. A drop in recall to 86% is indicative of locality effects from the original training set.

Table 4. Evaluating system performance vs. assisted human opinions

Measure	**Training set**	**Test set-1**	**Test set-1**	**Test set-2**
	Rome (10 ads)	**Rome (10 ads)**	**Rome (100 ads)**	**Venice (10 ads)**
Recall	0.99	0.94	0.92	0.86
Precision	0.98	0.97	0.97	0.96
Fallout	0.01	0.01	0.01	0.01
Accuracy	0.99	0.98	0.97	0.96
Error	0.01	0.02	0.03	0.04
F-measure	0.98	0.96	0.95	0.91

This experiment is limited because of the small semantic model. However, it is important to note that with limited domain knowledge and a very small vocabulary, we were able to demonstrate accuracy comparable to the best methods in the literature. Computational performance of our – as yet untuned – experimental tool is also already excellent, handling for example 100 advertisements in about 1 second on a 1 GHz PC. The tool has been evaluated on sets ranging from 38 to 7,600 advertisements (about 2,500 to 500,000 words), and found to process text at a rate of about 53 kb/sec on a 1 GHz PC. Thus, the tool scales well to large document datasets.

6.2 Experiment 2: Tourist Board Web Pages

In the second study, for all the categories in the annotation schema we performed a simple metrics-based evaluation, shown in Table 5 and Table 6 over the entire set of 11,742 paragraphs. These tables show estimated rates for each of the concepts defined in the semantic model. Table 7 summarizes the two tables by means of the average rate on all concepts and the total quality rate calculated for all the annotations independently from their semantic category.

Table 5. Evaluating system annotation vs. human Annotator 1

Topic **Measure**	Geography	Local Products	Culture	Artistic Heritage	Sport	Accommodation	Food & Refreshment	Wellness	Service
Recall	0.68	0.68	0.72	0.82	0.83	0.83	0.68	0.17	0.76
Precision	0.86	0.82	0.93	0.97	0.78	0.96	0.95	0.50	0.91
Fallout	0.01	0.00	0.01	0.00	0.01	0.00	0.00	0.00	0.00
Accuracy	0.98	0.99	0.97	0.98	0.98	0.99	0.99	1.00	0.98
Error	0.02	0.01	0.03	0.02	0.02	0.01	0.01	0.00	0.02
F-measure	0.76	0.75	0.82	0.89	0.80	0.89	0.79	0.25	0.83

Table 6. Evaluating system annotation vs. human Annotator 2

Topic **Measure**	Geography	Local Products	Culture	Artistic Heritage	Sport	Accommodation	Food & Refreshment	Wellness	Service
Recall	0.42	0.59	0.75	0.71	0.74	0.69	0.40	0.17	0.59
Precision	0.70	0.83	0.60	0.59	0.62	0.51	0.56	0.33	0.34
Fallout	0.01	0.00	0.05	0.04	0.03	0.02	0.01	0.00	0.07
Accuracy	0.96	0.99	0.93	0.94	0.96	0.97	0.98	1.00	0.92
Error	0.04	0.01	0.07	0.06	0.04	0.03	0.02	0.00	0.08
F-measure	0.53	0.69	0.66	0.64	0.68	0.58	0.47	0.22	0.43

Table 7. Summary of the evaluation results

	Tool vs. A1		**Tool vs. A2**	
Measure	**Average**	**Total**	**Average**	**Total**
Recall	0.69	0.77	0.56	0.65
Precision	0.85	0.90	0.56	0.55
Fallout	0.00	0.00	0.03	0.02
Accuracy	0.99	0.99	0.96	0.96
Error	0.01	0.01	0.04	0.04
F-measure	0.75	0.82	0.55	0.60

As we can observe from these results, for the given annotation schema the task turned out to be difficult both for the system and for the humans. One reason for this is the absence of structural patterns for all of the semantic categories. Another reason is that ambiguities occur frequently in such tourism documents, as they contain arbitrary information from a broad spectrum of topics, which are not always independent. For example, text about local food may be associated with either or both of *Local Products* category and *Food and Refreshment* category, depending on the context. Unfortunately, such overlaps in the semantic model cannot or even should not be resolved due to the nature of ontological modeling. Consequently, a text fragment may relate to more than one entity in the semantic model. A human marker, however, tends to pick only one, most relevant in his or her opinion, annotation tag for such multidimensional instances, whereas the tool will normally choose both.

Table 8 summarizes the comparison of our tool against each of the two human annotators as definitive.

By comparison with these two human annotators, the system exhibits a 65-77% level of recall, whereas the human counterparts each exhibit a recall of about 55-76% compared to each other. These rates indicate a high degree of disagreement between the annotators, for instance, a higher recall of annotator 1 compared to annotator 2 than the inverse ratio shows that annotator 1 provided more annotations than 2. This result can be explained by (a) difference in opinions about the concepts; (b) human factors, such as tiredness when annotating large documents. The first problem relates to the evaluation difficulties in general and was already discussed in section 4.2, whereas the second reason underlines the need for automated support of the semantic annotation task. The system showed a 55-90% level of precision compared to the humans, as good as or better than either human compared to the other.

Table 8. Comparing system results vs. human annotators, total scores

Measure	**A2 vs. A1**	**Tool vs. A1**	**A1 vs. A2**	**Tool vs. A2**
Recall	0.55	0.77	0.76	0.65
Precision	0.76	0.90	0.55	0.55
Fallout	0.01	0.00	0.02	0.02
Accuracy	0.97	0.99	0.97	0.96
Error	0.03	0.01	0.03	0.04
F-measure	0.64	0.82	0.64	0.60

In the second stage of evaluation, the human annotators were observed to use 75% less time to correct automatically annotated text than they spent on their original unassisted annotations.

In the third stage, where the human annotators corrected automatically marked up documents, the results of comparison to the final human markup are given in Table 9 and Table 10, while Table 11 summarizes both tables by estimating the average metrics on all concepts and the overall quality metrics calculated for all the retrieved annotations. Calibration to human assisted performance is evaluated in Table 12.

Table 9. Evaluating system annotation vs. human Annotator 1, assisted by the tool

Topic **Measure**	Geo-graphy	Local Prod-ucts	Cult-ure	Artistic Heritage	Sport	Accom-moda-tion	Food & Refresh-ment	Well-ness	Ser-vice
Recall	0.97	0.94	0.97	0.97	0.97	0.95	0.90	0.83	0.93
Precision	1	0.93	0.99	1	0.83	0.99	1	1	0.96
Fallout	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
Accuracy	1	1	1	1	0.99	1	1	1	0.99
Error	0.00	0.00	0.00	0.00	0.01	0.00	0.00	0.00	0.00
F-measure	0.98	0.94	0.98	0.98	0.89	0.97	0.95	0.91	0.95

Table 10. Evaluating system annotation vs. human Annotator 2 as assisted by the tool

Topic **Measure**	Geo-graphy	Local Prod-ucts	Cult-ure	Artistic Heritage	Sport	Accom-moda-tion	Food & Refresh-ment	Well-ness	Ser-vice
Recall	1	0.99	0.99	1	0.99	0.99	1	0.67	0.98
Precision	0.95	0.98	0.91	0.73	0.84	0.72	0.89	0.80	0.92
Fallout	0.00	0.00	0.01	0.03	0.01	0.01	0.00	0.00	0.00
Accuracy	1	1	0.99	0.97	0.99	0.99	1	1	0.99
Error	0.00	0.00	0.01	0.03	0.01	0.01	0.00	0.00	0.01
F-measure	0.97	0.98	0.95	0.84	0.91	0.84	0.94	0.73	0.95

Table 11. Summary of the evaluation results for human assisted opinions

Measure	Tool vs. A1		Tool vs. A2	
	Average	Total	Average	Total
Recall	0.94	0.96	0.96	0.99
Precision	0.97	0.96	0.86	0.85
Fallout	0.00	0.00	0.01	0.01
Accuracy	1	1	0.99	0.99
Error	0.00	0.00	0.01	0.01
F-measure	0.95	0.96	0.90	0.91

Observing the calculations vs. human assisted opinions, we note a high performance of the tool with respect to both precision and recall. Fallout is very low for all items, thus demonstrating that the tool was not misled by the large quantity of irrelevant information; the error rate is also fairly low; and accuracy reaches almost 100% rate.

Table 12. Comparing system results vs. humans assisted by the tool

Measure	A2 vs. A1	Tool vs. A1	A1 vs. A2	Tool vs. A2
Recall	0.82	0.96	0.96	0.99
Precision	0.96	0.96	0.82	0.85
Fallout	0.00	0.00	0.01	0.01
Accuracy	0.99	1	0.99	0.99
Error	0.00	0.00	0.00	0.01
F-measure	0.89	0.96	0.89	0.91

For both human annotators, comparison of the tool's and the human's results shows that the tool outperforms humans for all quality metrics. We conclude that such large-scale applications as Tourist Board web sites can greatly benefit from employing Cerno for semantic annotation of documents. Moreover, even human annotators are not able to accurately perform tasks of this size without tool support. The results suggest that efficiency of human annotation increases substantially if the annotator works with the output provided by Cerno, rather than conducting the annotation task manually from scratch.

The time required to handle the documents containing from 6,143 to 24,810 words ranged from 1.19 to 5.14 seconds on a 1 GHz PC with 512 Mb of memory running Windows XP. Thus, the tool has demonstrated scalability to the large document sizes and given the bigger semantic model of the domain.

As a result of this experiment, we can say that the semantic annotation framework can demonstrate reasonable results on more general documents and richer domain while maintaining high performance. Taking into account that the experiment involved analysis of large textual documents where human expertise is particularly expensive and difficult to obtain, we can say that Cerno had also allowed to minimize the costs of the assessment of the communicative efficacy of the web sites. The tool therefore can be especially useful in such applications where input needs to be analyzed quickly, but not necessarily very accurately.

6.3 Comparative Assessment of the Results

Although it is not really feasible to directly compare the quality of different tools because of unavailability of implementations, semantic models and, most importantly, comparable data sets, a possible solution is to approximate this comparison. For this purpose we use the author-reported performance previously summarized in [33] (Table 13). We also provide the type of data sets used for evaluation and show if a tool primarily relies on named entities recognition. The results for Cerno, apart from the two case studies presented in this paper, take into consideration performance reported for other experiments where Cerno has been applied [46], [18]. In general, Cerno was able to provide essential aid in generating semantic annotations that were in some cases equal to perfect recall and precision rates.

Table 13. Quality performance rates for different tools

Framework	**Precision**	**Recall**	**F-measure**	**Data sets**	**Reliance on named entities**
Armadillo	91.0	74.0	87.0	web sites of Computer Science Departments	Yes
KIM	86.0	82.0	84.0	HTML	Yes
Ont-O-Mat: Pankow	65.0	28.2	24.9	Varied web documents	Indifferent
SemTag	82.0	n/a	n/a	Varied web sites	Yes
Cerno	*90.6*	*90.8*	*90.7*	*Ads, APT web sites, academic papers, legislations*	*Indifferent*

From our experiences in applying Cerno to different domains, we can say that the effort of adapting the tool to a new task is relatively small and does not require any specific linguistic or programming expertise apart from general computer skills. In particular, we found that each new application required human effort ranging from one person-day to a couple of weeks to tune Cerno.

In terms of factors that influence Cerno's performance, our experiences suggest that domain homogeneity and document structure are most important. The first means that the results are better for the annotation schema that contains a limited and unambiguous set of concepts, as in the case of accommodation ads. The broader the domain is, the more difficult it becomes to draw a discriminative set of annotation rules for each concept. We also observed that the quality of the results do not improve with a richer semantic model. For many applications, ER-models are sufficient for representing the semantic knowledge that will then be used by the annotation schema.

Instead, domain expertise is very crucial for building a good quality annotation schema.

The presence and regularity of structure in input documents may facilitate the annotation process by providing some format-based annotation hints. For instance, in [46] we used information about the document structure to recognize title and abstract in academic papers. Alternatively, one can consider giving different weights to indicators identified in different structural elements. For instance, in HTML web pages annotations found in title and heading tags can be assigned higher ranks compared to those found in paragraphs or lists.

7. Conclusions and Lessons Learned

This paper addresses the problem of semantic annotation of textual documents using light-weight analysis techniques. We emphasized the need for robust and scalable tools to help automate this process in the context of web data processing. To address this need, we presented and evaluated the *Cerno* framework for semantic annotation of web documents.

The results of two experimental studies of the use of Cerno in two different semantic domains lead us to the following observations:

- As concerns the impact of the tool on human productivity, observing the time gains obtained in both experiments, we conclude that in comparison to manual annotation, the usage of an automatic tool can provide significant improvements in human performance, while at the same time improving the overall quality of the results.
- In terms of required resources, Cerno does not necessarily need gazetteers, linguistic knowledge bases, proper name vocabularies, or other knowledge sources. Though, these resources can be utilized to facilitate construction of the domain-dependent knowledge for a different application. For instance, synonyms provided by a thesaurus can be added as additional linguistic indicators to wordlists under corresponding categories.
- Cerno has demonstrated high processing speed and scalability to large documents.
- Because Cerno requires only limited computational resources, it can be easily adapted to light-weight interfaces to access tourist information or to online real-time applications.
- Another important feature of the tool is that it is not limited to a certain type of entities to be annotated. Two applications demonstrate that Cerno can cope with very different concepts. Compare, for instance, phone numbers and information about artistic heritage of a region.
- Based on our experience in adapting Cerno to several different semantic domains, we can claim that the required effort is minimal compared to tools that require accurately annotated training corpora.

- In addition, the results of the experiments represent useful data for the designer of a new semantic annotation application, demonstrating how to tune the tool to obtain good quality results for a particular task.

In summary, this research provides concrete evidence that a light-weight semantic annotation approach derived from software code analysis methods can be effective in semantic text document annotation. Our experiments with Cerno suggest that such an approach can provide acceptable performance and scalability, while yielding good quality results.

Apart from the experiments presented in this work, the feasibility of Cerno has been demonstrated in several additional studies. One of these is Biblio, an application developed for information mining from large collections of published research papers [46]. In [18], Cerno was used to support requirements extraction from system descriptions in natural language. The Gaius T. tool – based on Cerno – identifies legal requirements in legislative documents [22]. In that application, apart from the annotation of legal concept instances such as rights and obligations, we also identified relationships between concept instances and associated constraints. Cerno is also currently being evaluated in the context of the European Project Papyrus [29], where the proposed semantic annotation process is employed for the annotation of textual news content.

Acknowledgements

The authors are grateful to the anonymous reviewers whose comments led to many improvements to the results and the presentation of this work.

This research was conducted within the context of the STAMPS project, funded by the Province of Trento (Italy). It was also funded in part by the European Papyrus project and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] T. Berners-Lee, J. Hendler, O. Lassila, The Semantic Web, Scientific American 284 (5) (2001) 34–43.

[2] Biggerstaff, T., Design recovery for maintenance and reuse. IEEE Computer 22 (1989) 36–49.

[3] P. Cimiano, S. Handschuh, S. Staab, Towards the Self-Annotating Web, in: Proc. of the 13th WWW Conference, ACM, New York, May 2004, pp. 462–471.

[4] F. Ciravegna, S. Chapman, A. Dingli, Y. Wilks, Learning to Harvest Information for the Semantic Web, in: Proc. of the 1st European Semantic Web Symposium (ESWS 2004), LNCS, Vol. 3053, 2004, pp. 312–326.

[5] J. R. Cordy, T. Dean, A. Malton, K. Schneider, Source transformation in software engineering using the TXL transformation system, Information and Software Technology Journal 44 (2002) 827–837.

[6] J. R. Cordy, TXL – a language for programming language tools and applications, in: Proc. 4th Int. Workshop on Language Descriptions, Tools and Applications, Electronic Notes in Theoretical Computer Science, Vol. 110, 2004, pp. 3–31.

[7] T. Dean, J. R. Cordy, K. Schneider, A. Malton, Experience using design recovery techniques to transform legacy systems, in: Proc. 17th Int. Conference on Software Maintenance, 2001, pp. 622–631.

[8] S. Dill, N. Eiron, D. Gibson, D. Gruhl, R. Guha, A. Jhingran, T. Kanungo, K. S. McCurley, S. Rajagopalan, A. Tomkins, J. A.Tomlin, J. Y. Zien, A Case for Automated Large-Scale Semantic Annotation, Journal of Web Semantics 1(1) (2003) 115–132.

[9] e-Tourism group web site, www.economia.unitn.it/eTourism

[10] O. Etzioni, M. J., Cafarella, D. Downey, A. M. Popescu, T. Shaked, S. Soderland, D. S. Weld, A. Yates, Unsupervised named-entity extraction from the web: An experimental study, Artificial Intelligence 165 (2005) 91–134.

[11] D. Ferrucci, A. Lally, Uima: an architectural approach to unstructured information processing in the corporate research environment, Natural Language Engineering 10(3-4) (2004) 327–348.

[12] W. Framke, The Destination as a Concept. A discussion of the business-related perspective versus the socio-cultural approach in tourism theory, Scandinavian Journal of Hospitality and Tourism 2 (2) (2002) 92–109.

[13] D. Freitag, N. Kushmerick, Boosted wrapper induction, in: Proc. 17th National Conference on Artificial Intelligence, 2000, pp. 577–583

[14] D. Freitag, Machine Learning for Information Extraction in Informal Domains, PhD thesis, Carnegie Mellon University, 1998.

[15] The GATE project web site, http://gate.ac.uk

[16] S. Handschuh, S. Staab, R. Studer, Leveraging metadata creation for the Semantic Web with CREAM, in: R. Kruse et al. (eds.), KI 2003 - Advances in Artificial Intelligence, Proc. of the Annual German Conference on AI, Vol. 2821, Springer, Berlin, 2003, 19–33

[17] A. Kiryakov, B. Popov, I. Terziev, D. Manov, D. Ognyanoff, Semantic Annotation, Indexing, and Retrieval, Journal of Web Semantics 2(1) (2005) 49–79.

[18] N. Kiyavitskaya, N. Zannone,. Requirements model generation to support requirements elicitation: the Secure Tropos experience, Automated Software Engineering 15 (2) (Jun. 2008), 149–173.

[19] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, J. Mylopoulos, Applying software analysis technology to lightweight semantic markup of document Text, in: Proc. ICAPR 2005, 3rd International Conference on Advances in Pattern Recognition, Bath, UK, 2005, pp. 590–600.

[20] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, J. Mylopoulos, Text Mining through Semi-Automatic Semantic Annotation, in: Proc. PAKM 2006, 6th International Conference on Practical Aspects of Knowledge Management, Vienna, Springer-Verlag, 2006, pp. 143–154.

[21] N. Kiyavitskaya, N. Zeni, L. Mich, J. R. Cordy, J. Mylopoulos, Annotating Accomodation Advertisement using CERNO, in: Proc. ENTER 2007, Ljubljana, Slovenia, January 24–26, 2007, pp. 389–400.

[22] N. Kiyavitskaya, N. Zeni, T. D. Breaux, A. I. Antón, J. R. Cordy, L. Mich, J. Mylopoulos, "Automating the extraction of rights and obligations for regulatory compliance, in: Proc. of 27th International Conference on Conceptual Modeling (ER2008), LNCS (5231/2008) 154–168, Berlin/Heidelberg: Springer.

[23] A. Lauriston, Criteria for Measuring Term Recognition. European Chapter Meeting of the ACL, in: Proc. of 7th Conf. on European chapter of the Association for Computational Linguistics, Dublin, Ireland, 1995, pp. 17–22.

[24] D. Maynard, W. Peters, Y. Li, Metrics for Evaluation of Ontology-based Information Extraction, in: the 4th Int. Evaluation of Ontologies for the Web Workshop (EON 2006), the 15th Int. World Wide Web Conference, Edinburgh, United Kingdom, May 22, 2006.

[25] R. Grishman, B. Sundheim, Message Understanding Conference - 6: A Brief History, in: Proceedings of the 16th International Conference on Computational Linguistics (COLING), Kopenhagen, 1996, 466–471.

[26] I. Muslea, S. Minton, C. A. Knoblock, Active learning with strong and weak views: A case study on wrapper induction, in: Proc. 18th Int. Joint Conference on Artificial Intelligence, 2003, pp. 415–420.

[27] C. Oertel, S. O'Shea, A. Bodnar, D. Blostein, Using the Web to Validate Document Recognition Results: Experiments with Business Cards, in: Proc. of the SPIE, Vol. 5676, 2004, pp.17–27.

[28] OpenCalais: http://www.opencalais.com/

[29] European Project Papyrus – Cultural and historical digital libraries dynamically mined from news archives – official web site, www.ict-papyrus.eu

[30] The Protégé platform web site, http://protege.stanford.edu/

[31] J. Pustejovsky, J. Castaño, R. Ingria, R. Saurí, R. Gaizauskas, A. Setzer, G. Katz, TimeML: Robust Specification of Event and Temporal Expressions in Text, in: Proceedings of Fifth International Workshop on Computational Semantics IWCS-5 (2003).

[32] The RDF-gravity web site, http://semweb.salzburgresearch.at/apps/rdf-gravity

[33] L. Reev, H. Han, Survey of semantic annotation platforms, in SAC'05: Proceedings of the 2005 ACM Symposium on Applied Computing, pp. 1634–1638, New York, NY, USA, 2005. ACM Press.

[34] A. Sahuguet and F. Azavant, Building intelligent web applications using lightweight wrappers, Data & Knowledge Engineering, Vol. 36 (3) 283–316, March 2001.

[35] N. Synytskyy, J. R. Cordy, T. R. Dean, Robust multilingual parsing using island grammars, in: Proceedings of the 2003 Conference of the Centre For Advanced Studies on Collaborative Research, Toronto, Ontario, Canada, October 6-9, 2003, IBM Centre for Advanced Studies Conference. IBM Press (2003) 266–278.

[36] On-line Thesaurus web site, http://thesaurus.reference.com

[37] Trentino Marketing Company web site,
www.trentino.to/home/about.html?_area=about&_lang=it&_m=apt

[38] C. Vassilakis, G. Lepouras, A. Katifori , A heuristics-based approach to reverse engineering of electronic services, Information and Software Technology 51(2) (2009) 325–336.

[39] A. Wessman, S. W. Liddle, D.W. Embley, A generalized framework for an ontology-based data-extraction system, in: Proc. 4th Int. Conference on Information Systems Technology and its Applications, 2005, pp. 239–253.

[40] C. A.Will, Comparing human and machine performance for natural language information extraction: results for English microelectronics from the MUC-5 evaluation, in: Proc. of the 5th Message Understanding Conference, 1993, pp. 53–67.

[41] The WordNet web site, http://wordnet.princeton.edu

[42] Y. Yang, An evaluation of statistical approaches to text categorization, Journal of Information Retrieval 1 (1/2) (1999) 67–88.

[43] R. Zanibbi, D. Blostein, J. R. Cordy, Recognizing Mathematical Expressions Using Tree Transformation, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (11) (2002) 1455–1467.

[44] R. Zanibbi, D. Blostein, J. R. Cordy, A Survey of Table Recognition, Models, Observations, Transformations, and Inferences, International Journal of Document Analysis and Recognition 7(1) (2004) 1–16.

[45] N. Zeni, N. Kiyavitskaya, L. Mich, J. R. Cordy, J. Mylopoulos, Annotating Accommodation Advertisements using Cerno, in Proc. of ENTER 2007, IFITT 14th International Conference on Information Technology and Travel & Tourism, Ljubljana, Slovenia, January 2007, pp. 389–400.

[46] N. Zeni, N. Kiyavitskaya, L. Mich, J. Mylopoulos, J. R. Cordy, A Lightweight Approach to Semantic Annotation of Research Papers, Natural Language Processing and Information Systems (2007) 61–72, LNCS, Springer Berlin / Heidelberg.

Vitae

Nadzeya Kiyavitskaya

Nadzeya Kiyavitskaya is post doctorate student in the Department of Information Engineering and Computer Science, University of Trento, Italy. She received her MSc in 2002 from the Department of Mechanics and Mathematics of the Belarusian State University, Minsk, Belarus. Since then, Dr. Kiyavitskaya joined the data and knowledge management group of the University of Trento, Italy, and received her PhD in Information and Communication Technology in 2006. She then continued her research work at the Information Engineering and Computer Science Department by participating in a national project called TOCAI.it and European project Papyrus.

Nicola Zeni

Nicola Zeni is research fellow in the Department of Information Engineering and Computer Science, University of Trento, Italy. Dr. Zeni received his MSc in Economics and Business Administration in 1999. Afterwards he changed the topic of his studies to information technologies by attending professional courses in Information Systems, Informatics Fundamentals, Databases, Languages and Translators. Thus, in 2007 he completed his PhD studies in Information and Communication Technology. Besides his research activities, for many years Dr. Zeni has been a teaching assistant in several courses on information systems engineering.

James R. (Jim) Cordy

Jim Cordy is Professor and past Director of the School of Computing at Queen's University, Kingston, Canada. Dr. Cordy received his BSc in computer science and mathematics from the University of Toronto in 1973 and his MSc in computer science in 1976. After serving several years as chief programmer and senior research associate at the Computer Systems Research Institute of the University of Toronto, he returned to school and received his PhD from the University of Toronto in 1986. He found Legasys Corporation in 1995 where he was vice president and chief research officer until his return to Queen's in 2001.

Luisa Mich

Luisa Mich is Associate Professor of Information Systems and Web Engineering and senior researcher in the eTourism group of the Department of Computer and Management Sciences, University of Trento, Italy. She received her MSc degree in physics in 1983. Her research interests include Web site quality, requirements engineering and semantic annotation, focusing on conceptual modeling and on the role of natural language and creativity techniques in information systems analysis. She authored the 7Loci meta-model for the evaluation of web site quality and contributed to the WTO/IFITT joint project by developing an evaluation and benchmarking scheme for destination web sites. Prof. Mich has led the Cerno project at the University of Trento since 2002.

John Mylopoulos

John Mylopoulos received his BEng degree from Brown University in 1966 and his PhD degree from Princeton in 1970, the year he joined the faculty of the University of Toronto. His research interests include information modeling techniques, covering notations, implementation techniques and applications, knowledge based systems, semantic data models, information system design and requirements engineering. Dr. Mylopoulos is the recipient of the first Outstanding Services Award given by the Canadian AI Society, a fellow of the American Association for AI and the elected president of the VLDB Endowment Endowment (1998-2003). He has served on the editorial board of several international journals.