Tool-Supported Process for Semantic Annotation: 
An Experimental Evaluation

Nadzeya Kiyavitskaya1, Nicola Zeni1, Luisa Mich2, 
James R. Cordy3, and John Mylopoulos4

1 Dept. of Information and Communication Technology, University of Trento, Italy 
{nadzeya.kiyavitskaya, nicola.zeni}@unitn.it 
2 Dept. of Computer and Management Sciences, University of Trento, Italy 
luisa.mich@unitn.it 
3 School of Computing, Queens University, Kingston, Canada 
cordy@cs.queensu.ca 
4 Dept. of Computer Science, University of Toronto, Canada 
jm@cs.toronto.edu

Abstract. The Semantic Web vision cannot be achieved without large-scale automation in the semantic annotation process. In this paper we describe a tool-supported process and present an experimental evaluation framework as well as extensive experimental results on the effectiveness of the process. Our proposal is based on techniques and technologies traditionally used in software analysis and reverse engineering for large-scale legacy code bases. In this work we first elaborate a process for semantic annotation of web documents based on such reverse engineering technologies. The main results of the paper include an experimental evaluation framework and experimental results based on two case studies adopted from the Tourism sector. The results suggest that our approach can facilitate the semi-automatic annotation of large document bases.

Keywords: semantic annotation, large-scale document analysis, conceptual schemas, software analysis.

1 Introduction

According to the Semantic Web vision, all information available on the web needs to be transformed into machine-readable form. To accomplish this, the Semantic Web initiative proposes to encapsulate semantics along with each web document. This means that documents are annotated with terms derived from a rich semantic model of the domain the document is about. Semantic models are by their very nature domain specific, i.e. they describe a domain such as Medicine or Commerce. Some semantic models are application-independent in the sense that they were not built for a specific information system application. We, and others, call such models ontologies. Alternatively, a semantic model may be application-specific, in which case we call it a conceptual schema, in the conventional sense used in database research and practice since the 1970s.
We are interested in developing methodologies and tools that semi-automate the semantic annotation process for a set of documents with respect to a semantic model (ontology or conceptual schema). Specifically, we are exploring tools that will do most of the work in generating a set of annotations with respect to a semantic model, leaving the fine tuning to a human.

Given the scale of the Web, it is clear that full-scale natural language processing techniques won’t do. Instead, we propose to approach the problem using highly efficient methods and tools proven effective in the software analysis domain for processing billions of lines of legacy software source code [1]. In fact, document analysis for the Semantic Web and software code analysis have striking similarities in their needs:

- robust parsing techniques, given that real documents rarely match given grammars;
- a semantic understanding of source text, on the basis of a semantic model;
- semantic clues drawn from a vocabulary associated with the semantic model;
- contextual clues drawn from the syntactic structure of the source text;
- inferred semantics from exploring relationships between identified semantic entities and their properties, contexts and related other entities.

On the basis of these considerations, we have adapted software analysis techniques to the more general problem of semantic annotation of text documents. Our initial hypothesis is that these methods can attain the same scalability for analysis of textual documents as for software code analysis. Our proposed process for semantic annotation has been presented already [2]. In this work we extend and generalize the process and architecture of the prototype semantic annotation tool presented earlier. The main contribution of this work includes an evaluation framework for semantic annotation tools, as well as two real-world case studies: accommodation advertisements and Tourist Board web sites. For the first experiment, we use a small conceptual schema derived from a set of user queries. For the second experiment, we adopt more elaborate conceptual schemas reflecting a richer semantic domain for our markup process applied to a broader class of documents.

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

- standard accuracy measures, 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” annotations, as human annotators also differ among themselves on how to annotate a given document.

The rest of the paper is organized as follows. Our proposed annotation process and the architecture of our semantic annotation system are introduced in section 2. The two case studies are presented in section 3, and section 4 describes the evaluation setup and experimental results. Section 5 provides a short comparative overview of the state-of-the-art on semantic annotation tools and conclusions are drawn in section 6.
2 Methodology

Our method for semantic annotation of documents uses the generalized parsing and structural transformation system TXL [3], the basis of the automated Year 2000 system LS/2000 [4]. TXL is a programming language specially designed to allow by-example rapid prototyping of language descriptions, tools and applications. The system accepts as input a grammar and a document, generates a parse tree for the input document, and applies transformation rules to generate output in a target format.

The architecture of our solution (Fig. 1) is based on the LS/2000 software analysis architecture, generalized to allow for easy parameterization by a range of semantic domains.

Fig. 1. Architecture of our semantic annotation process.

The architecture explicitly factors out reusable domain independent knowledge such as the structure of basic entities (email and web addresses, monetary formats, date and time formats, and so on) and language structures (document, paragraph, sentence and phrase structure), shown on the left hand side, while allowing for easy change of semantic domain, characterized by vocabulary (category word and phrase lists and contra-lists) and semantic model (entity-relationship schema and interpretation), shown on the right.

The process consists of three phases. In the first stage, an approximate ambiguous context-free grammar is used to efficiently obtain an approximate phrase structure parse of the source text using the TXL parsing engine. Using robust parsing techniques borrowed from compiler technology [5] this stage results in a deterministic maximal parse even for badly malformed text. As part of this first stage, basic entities such as email addresses, web addresses, monetary amounts, dates, times and other
word-equivalent objects are recognized grammatically as would be done in a 
programming language parser. The parse is linear in the length of the input and runs 
at compiler speeds. In our first experiment this parse is relatively coarse-grained, 
ignoring language structure below the sentence and verb-clause level.

In the second stage, initial semantic annotation of the document is derived using a 
wordlist file specifying both positive and negative indicators for semantic categories 
(Fig. 2). Indicators can be both literal words and phrases and names of parsed entities.

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

Fig. 2. Prototype category wordlist description for the term concept. This wordlist 
specifies that a phrase or sentence may relate to the term concept if it contains a date 
object and/or one or more of the words and phrases listed, and does not contain any 
objects of the money category or the price concept.

Phrases are marked up once for each category they match – thus at this stage a 
sentence or phrase may end up with many different (even conflicting) semantic 
mrkups. Vocabulary word and entity lists are derived from the semantic model for 
the target domain. This stage uses the structural pattern matching and source 
transformation capabilities of the TXL engine in much the same way as it is used for 
software markup [5] to yield a preliminary marked-up text in XML form (Fig. 3).

<type><location>Very elegant apartment located in Piazza 
Dante, just a walk from Fosse Ardeatine and 10 minutes to 
Colosseum by bus (Bus stop in the square)</location></type>. 
<facility>75 sqm in a charming, and full furnished environment 
</facility>.<facility><price>1.200 euro a month, utilities not 
included </price></facility>. <contact>Write to 
pseudonym@somewhere.it or phone to 123.1234567 </contact>

Fig. 3. Example result of the XML-marked up accommodation advertisement. 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.

The third stage uses the XML marked-up text to populate an XML database 
schema (Fig. 4), derived from the semantic model for the target domain. Sentences 
and phrases with multiple markups are “cloned” using TXL source transformation to 
appear as multiple copies, one for each different markup, before populating the 
 database. In this way we do not prejudice one interpretation as being preferred; rather 
we assume that a single sentence or phrase may in fact be a reasonable answer for all 
of the semantic categories it is marked as.

<ad> 
<location></location> 
<price></price> 
<contact></contact> 
<facility></facility> 
<term></term> 
</ad>
Fig. 4. Database template schema for accommodation advertisements.

The outputs of our process are both the XML marked-up text (explicit *in-line* annotation yielding an inline semantic markup of the original document text) and the populated database (*implicit* annotation extracting and storing instances in an external database). The populated database can be queried by a standard SQL database engine.

3 Experimental Case Studies

Our case studies involve two applications in the Tourism area. Tourism is a sector of economy which can profit greatly from Semantic Web technologies. The need to apply these technologies to Tourism information systems has been emphasized by many authors and a number of typical scenarios have been developed [6]. Moreover, tourism is a very broad sector which comprises many heterogeneous domains: accommodation and eating structures, sports, means of transport, historical sites, tourist attractions, medical services and other areas of human activity. Information available from heterogeneous data sources must be integrated in order to allow effective interoperability of tourism information systems and composition of services for tourist packages. This is where semantic annotations come in handy.

3.1 Accommodation Ads

As a first full experiment in the application of our new method, we have been working in the domain of travel documents, and in particular with published advertisements for accommodation.

This domain is typical of the travel domain in general and poses many problems commonly found in other text markup problems, such as: partial and malformed sentences; abbreviations and short-forms; location-dependent vocabulary; monetary units; date and time conventions, and so on.

In order to make realistic test of the generality of the method we restricted ourselves to some constraints. In particular, we avoided all proper names and locality-dependent words in our wordlist, and we did not preprocess the text of accommodation descriptions by formatting them or correcting errors. The human annotators in our experiment could take full advantage of local knowledge and corrected structural cues when producing their annotations for the evaluation stage, but the tool could not.

In the first case study we used a set of several hundred advertisements for accommodation in Rome drawn from an online newspaper. The task was to identify and mark up the categories of semantic information in the advertisements according to a given accommodation conceptual schema (Fig. 5), which was reduced by hand to an XML schema for input to our system. The desired result was a database with one instance of the schema for each advertisement in the input, and the marked-up original advertisements. The wordlist for this experiment was constructed by hand from a set of examples.
3.2 Tourist Board Web Pages

In the second case study we pursued two main goals: to demonstrate the generality of our method over different domains, and to verify the scalability of our approach on a richer semantic model and larger documents. For this purpose, we considered the web sites of Tourist Boards in the province of Trentino (Italy)\(^1\) as input documents. This domain presents a number of specific problematic issues: free unrestricted vocabulary; differently structured text; a rich annotation schema covering the content of web sites.

This experiment was run for the E-tourism group of University of Trento, and was motivated in part by the economic interests of the province. From the point of view of experts in tourism, the high-level goal of this case study was to assess the communicative efficacy of the web sites based on content quality or informativity, that is, how comprehensively the web site covers its domain according to the strategic goals of the Tourist Board. In order to assess the content quality we performed semantic annotation of the web pages revealing the presence of information important for a Tourist Board web site to be effective. The list of semantic categories and their descriptions was provided by the experts (Fig. 6).

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

**Local products**
- Local handicrafting
- 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
- Tickets, entrance fees, guides

---
\(^1\) [www.trentino.to/home/about.html?_area=about&_lang=it&_m=apt](http://www.trentino.to/home/about.html?_area=about&_lang=it&_m=apt)
Sport
- Sporting events
- Sport infrastructure
- Sport disciplines

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

Food and refreshment
- Places to eat
- Dishes

Degustation
- Time tables
- How to book

Wellness
- Wellness centers
- Wellness services

Services
- Transport, schedules
- Information offices
- Terminals, stations, airports
- Travel agencies

Fig. 6. Relevant topics for communicative efficacy of a Tourist Board web site.

We adapted our annotation framework to the new domain by replacing the domain-dependent components with respect to this specific task. For this purpose, the initial rough schema provided by the domain experts was transformed into a richer conceptual schema consisting of about 130 concepts systematized into a hierarchy and connected by semantic relations. (See the partial view in Fig. 7, using the visualization tool RDFGravity². This view shows only is-a relations, because this type of relation is essential in guiding the annotation process. The complete model includes many more relations than just taxonomical ones.)

Fig. 7. A slice of the conceptual schema showing semantic (placement in the hierarchy, relationships, attributes) and syntactic (keywords or patterns) information associated with concepts.

Domain dependent vocabulary was derived semi-automatically, expanding concept definitions with the synonyms provided by the WordNet³ database and on-line Thesaurus⁴ and mined from a set of sample documents. The total number of keywords collected was 507 and an additional four object patterns were re-used from previous application to detect such entities as monetary amounts, e-mails, web addresses and phone numbers.

² http://semweb.salzburgresearch.at/apps/rdf-gravity/
³ http://wordnet.princeton.edu
⁴ http://thesaurus.reference.com
To begin this experiment we downloaded the Tourist Board web sites applying a screen-scraping technique. Then two human annotators and the tool were given 11742 text fragments (paragraphs) 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 (Fig. 8).

\textless FoodAndRefreshment\textgreater Bread and wine snack in the shade of an elegant park.</FoodAndRefreshment>
\textless FoodAndRefreshment\textgreater Dinner at the "La Luna Piena" restaurant, consisting of the "Il Piatto del Vellutajo"</FoodAndRefreshment>
\textless ArtisticHeritage\textgreater Museo del Pianoforte Antico: guided visit and concert proposed within the "Museum Nights" programme on the 3, 10, 17 and 24 of August.</ArtisticHeritage>

Fig. 8. Example of XML-marked up content of a tourism web site.

4 Experimental Evaluation

4.1 Evaluation Framework

The performance of semantic annotation tools is usually evaluated similarly to information extraction systems, i.e. by comparing with correct markup and calculating recall and precision metrics. Because human opinions on the "correct" markup can vary widely, ideally we should compare our automated results against a wide 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.

In order to evaluate our initial experimental results, we designed a three stage validation process. At each stage, we were interested in measuring standard accuracy rates [7] for the tool’s automated markup compared to manually-generated annotations:

- \textit{Recall} evaluates how well the tool performs in finding relevant items;
- \textit{Precision} shows how well the tool performs in not returning irrelevant items;
- \textit{Fallout} measures how quickly precision drops as recall is increased;
- \textit{Accuracy} measures how well the tool identifies relevant items and rejects irrelevant ones;
- \textit{Error rate} demonstrates how much the tool is prone to accept irrelevant items and reject relevant ones;
- \textit{F-measure} is a harmonic mean of recall and precision (we consider recall and precision weighted equally).

In the first step of our evaluation framework, we compare the system output directly with manual annotations. We expect that quality of manual annotations constitutes an upper bound for automatic document analysis. Of course, this type of evaluation can’t be applied on a large scale for cost reasons, unless a "gold standard" already exists.

In the second step, we check if the use of automatic tool increases the productivity of human annotators. We note the time used for manual annotation of the original textual documents and compared it to the time used for manual correction of the automatically annotated documents. The percentage difference of these two measures
shows how much time can be saved when the tool assists the human annotator.
Finally, in our third step we take into account disagreement between annotators. In order to obtain a realistic evaluation, system performance must be “calibrated” relative to human performance. In this case, we compare system results against the final human markup made by correcting the automatically generated markup.

4.2 Experimental Results

**Experiment 1: Accommodation Ads.** The details of our evaluation for the accommodation ads application can be found in [2]. We only say that as a result of this first experiment, even without local knowledge and using a very small vocabulary and only few TXL rules, we have been able to demonstrate accuracy comparable to some of the best heavyweight methods, albeit on a very limited domain. Performance of our untuned experimental tool was also already very fast, handling for example 100 advertisements in about 1 second on a 1 GHz PC.

**Experiment 2: Tourist Board Web Pages.** Since the semantic model in this experiment was fairly extensive, we could not afford human annotators to handle properly all of the entities of the rich domain schema. Accordingly, we limited the evaluation to a handful of topics (Geography, Sport, Culture, Artistic Heritage, Local Products, Wellness, Accommodation, Food and Refreshment, Services). For these topics we performed simple metrics-based validation (Tables 1a, b, c) and calibration of the results taking into account inter-annotator disagreement (Table 2) for the entire set of 11742 paragraphs.

<table>
<thead>
<tr>
<th>Topic Measure</th>
<th>Geography</th>
<th>Local Products</th>
<th>Culture</th>
<th>Artistic Heritage</th>
<th>Sport</th>
<th>Accommodation</th>
<th>Food &amp; Refreshment</th>
<th>Wellness</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>68.23</td>
<td>68.18</td>
<td>72.49</td>
<td>82.28</td>
<td>82.57</td>
<td>83.19</td>
<td>68.29</td>
<td>16.67</td>
<td>76.42</td>
</tr>
<tr>
<td>Precision</td>
<td>85.62</td>
<td>82.19</td>
<td>93.16</td>
<td>97.38</td>
<td>78.35</td>
<td>96.12</td>
<td>94.92</td>
<td>50.00</td>
<td>91.01</td>
</tr>
<tr>
<td>Fallou</td>
<td>0.59</td>
<td>0.34</td>
<td>0.50</td>
<td>0.19</td>
<td>1.50</td>
<td>0.11</td>
<td>0.08</td>
<td>0.03</td>
<td>0.43</td>
</tr>
<tr>
<td>Accuracy</td>
<td>97.88</td>
<td>98.95</td>
<td>97.16</td>
<td>98.39</td>
<td>97.52</td>
<td>99.39</td>
<td>99.26</td>
<td>98.85</td>
<td>98.31</td>
</tr>
<tr>
<td>Error</td>
<td>2.12</td>
<td>1.05</td>
<td>2.84</td>
<td>1.61</td>
<td>2.48</td>
<td>0.61</td>
<td>0.74</td>
<td>0.15</td>
<td>1.69</td>
</tr>
<tr>
<td>F-measure</td>
<td>75.94</td>
<td>74.53</td>
<td>81.53</td>
<td>89.19</td>
<td>80.40</td>
<td>89.19</td>
<td>79.43</td>
<td>25.00</td>
<td>83.08</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Topic Measure</th>
<th>Geography</th>
<th>Local Products</th>
<th>Culture</th>
<th>Artistic Heritage</th>
<th>Sport</th>
<th>Accommodation</th>
<th>Food &amp; Refreshment</th>
<th>Wellness</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>42.19</td>
<td>59.09</td>
<td>74.85</td>
<td>70.57</td>
<td>73.86</td>
<td>68.91</td>
<td>40.24</td>
<td>16.67</td>
<td>59.43</td>
</tr>
<tr>
<td>Precision</td>
<td>69.83</td>
<td>82.54</td>
<td>59.81</td>
<td>59.31</td>
<td>62.24</td>
<td>50.62</td>
<td>55.93</td>
<td>33.53</td>
<td>33.96</td>
</tr>
<tr>
<td>Fallout</td>
<td>0.94</td>
<td>0.29</td>
<td>4.75</td>
<td>4.25</td>
<td>2.94</td>
<td>2.11</td>
<td>0.68</td>
<td>0.05</td>
<td>6.62</td>
</tr>
<tr>
<td>Accuracy</td>
<td>96.27</td>
<td>98.80</td>
<td>93.48</td>
<td>93.71</td>
<td>95.63</td>
<td>97.01</td>
<td>98.08</td>
<td>99.82</td>
<td>91.54</td>
</tr>
<tr>
<td>Error</td>
<td>3.73</td>
<td>1.20</td>
<td>6.52</td>
<td>6.29</td>
<td>4.37</td>
<td>2.99</td>
<td>1.92</td>
<td>0.18</td>
<td>8.46</td>
</tr>
<tr>
<td>F-measure</td>
<td>52.60</td>
<td>68.87</td>
<td>66.49</td>
<td>64.45</td>
<td>67.55</td>
<td>58.36</td>
<td>46.81</td>
<td>22.22</td>
<td>43.22</td>
</tr>
</tbody>
</table>
Table 1c. Evaluating system annotation vs. humans – average scores.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Tool vs. A1</th>
<th>Tool vs. A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>68.70</td>
<td>56.20</td>
</tr>
<tr>
<td>Precision</td>
<td>85.42</td>
<td>56.40</td>
</tr>
<tr>
<td>Fallout</td>
<td>0.42</td>
<td>2.51</td>
</tr>
<tr>
<td>Accuracy</td>
<td>98.52</td>
<td>96.04</td>
</tr>
<tr>
<td>Error</td>
<td>1.48</td>
<td>3.96</td>
</tr>
<tr>
<td>F-measure</td>
<td>75.37</td>
<td>54.51</td>
</tr>
</tbody>
</table>

Table 2. Calibrating system results vs. human annotators.

<table>
<thead>
<tr>
<th>Measure</th>
<th>A2 vs. A1</th>
<th>Tool vs. A1</th>
<th>A1 vs. A2</th>
<th>Tool vs. A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>61.75</td>
<td>68.70</td>
<td>76.47</td>
<td>56.20</td>
</tr>
<tr>
<td>Precision</td>
<td>76.47</td>
<td>85.42</td>
<td>61.75</td>
<td>56.40</td>
</tr>
<tr>
<td>Fallout</td>
<td>1.00</td>
<td>0.42</td>
<td>2.50</td>
<td>2.51</td>
</tr>
<tr>
<td>Accuracy</td>
<td>96.70</td>
<td>98.52</td>
<td>96.70</td>
<td>96.04</td>
</tr>
<tr>
<td>Error</td>
<td>3.30</td>
<td>1.48</td>
<td>3.30</td>
<td>3.96</td>
</tr>
<tr>
<td>F-measure</td>
<td>66.79</td>
<td>75.37</td>
<td>66.79</td>
<td>54.51</td>
</tr>
</tbody>
</table>

As shown in Table 2, for the given annotation schema the task turned out to be difficult both for the system and for the humans due to the vague definitions of the categories. For example, text about local food may be associated with either or both of the Local Products category and the Food and Refreshment category, depending on the context. Explicit resolution of such ambiguities in the expert definition would improve the results. If we compare the difference in scores of F-measure, as the most aggregate characteristic, the overall difference in performances of the system and the humans is approximately 10%. This means that the tool did almost as well as the human annotators.

In the second stage of evaluation, the human annotators were observed to use 72% less time to correct automatically annotated text than they spent on their original unassisted annotations.

In the third stage, when the human annotators corrected automatically marked up documents, the results of comparison to the final human markup are given in Tables 3a, b, c and calibration to human performance in Table 4.

Table 3a. Evaluating system annotation vs. human Annotator 1 as assisted by the tool.

<table>
<thead>
<tr>
<th>Topic Measure</th>
<th>Geography</th>
<th>Local Products</th>
<th>Culture</th>
<th>Artistic Heritage</th>
<th>Sport</th>
<th>Accommodation</th>
<th>Food &amp; Refreshment</th>
<th>Wellness</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>96.88</td>
<td>94.32</td>
<td>97.34</td>
<td>96.91</td>
<td>96.68</td>
<td>94.96</td>
<td>90.24</td>
<td>83.33</td>
<td>93.36</td>
</tr>
<tr>
<td>Precision</td>
<td>100.00</td>
<td>93.26</td>
<td>98.50</td>
<td>100.00</td>
<td>83.21</td>
<td>99.12</td>
<td>100.00</td>
<td>100.00</td>
<td>96.10</td>
</tr>
<tr>
<td>Fallout</td>
<td>0.00</td>
<td>0.16</td>
<td>0.14</td>
<td>0.00</td>
<td>1.28</td>
<td>0.03</td>
<td>0.00</td>
<td>0.00</td>
<td>0.22</td>
</tr>
<tr>
<td>Accuracy</td>
<td>99.85</td>
<td>99.72</td>
<td>99.64</td>
<td>99.74</td>
<td>98.59</td>
<td>99.82</td>
<td>99.80</td>
<td>99.97</td>
<td>99.44</td>
</tr>
<tr>
<td>Error</td>
<td>0.15</td>
<td>0.28</td>
<td>0.36</td>
<td>0.26</td>
<td>1.41</td>
<td>0.18</td>
<td>0.20</td>
<td>0.03</td>
<td>0.56</td>
</tr>
<tr>
<td>F-measure</td>
<td>98.41</td>
<td>93.79</td>
<td>97.92</td>
<td>98.43</td>
<td>89.44</td>
<td>97.00</td>
<td>94.87</td>
<td>90.91</td>
<td>94.71</td>
</tr>
</tbody>
</table>
Table 3b. Evaluating system annotation vs. human Annotators2 as assisted by the tool.

<table>
<thead>
<tr>
<th>Topic Measure</th>
<th>Geography</th>
<th>Local Products</th>
<th>Culture</th>
<th>Artistic Heritage</th>
<th>Sport</th>
<th>Accommodation</th>
<th>Food &amp; Refreshment</th>
<th>Wellness</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>100.00</td>
<td>97.73</td>
<td>99.11</td>
<td>100.00</td>
<td>99.17</td>
<td>99.16</td>
<td>100.00</td>
<td>66.67</td>
<td>98.10</td>
</tr>
<tr>
<td>Precision</td>
<td>94.58</td>
<td>97.73</td>
<td>90.79</td>
<td>73.14</td>
<td>84.45</td>
<td>72.39</td>
<td>89.13</td>
<td>80.00</td>
<td>92.41</td>
</tr>
<tr>
<td>FallOut</td>
<td>0.30</td>
<td>0.05</td>
<td>0.95</td>
<td>3.31</td>
<td>1.20</td>
<td>1.19</td>
<td>0.26</td>
<td>0.03</td>
<td>0.46</td>
</tr>
<tr>
<td>Accuracy</td>
<td>99.72</td>
<td>99.90</td>
<td>99.05</td>
<td>96.96</td>
<td>98.82</td>
<td>98.82</td>
<td>99.74</td>
<td>99.92</td>
<td>99.46</td>
</tr>
<tr>
<td>Error</td>
<td>0.28</td>
<td>0.10</td>
<td>0.95</td>
<td>3.04</td>
<td>1.18</td>
<td>1.18</td>
<td>0.26</td>
<td>0.08</td>
<td>0.54</td>
</tr>
<tr>
<td>F-measure</td>
<td>97.22</td>
<td>97.73</td>
<td>94.77</td>
<td>84.49</td>
<td>91.22</td>
<td>83.69</td>
<td>94.25</td>
<td>72.73</td>
<td>95.17</td>
</tr>
</tbody>
</table>

Table 3c. Evaluating system annotation vs. humans – average scores.

<table>
<thead>
<tr>
<th>Measure</th>
<th>Tool vs. A1</th>
<th>Tool vs. A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>93.78</td>
<td>95.55</td>
</tr>
<tr>
<td>Precision</td>
<td>96.69</td>
<td>86.07</td>
</tr>
<tr>
<td>FallOut</td>
<td>0.20</td>
<td>0.86</td>
</tr>
<tr>
<td>Accuracy</td>
<td>99.62</td>
<td>99.16</td>
</tr>
<tr>
<td>Error</td>
<td>0.38</td>
<td>0.84</td>
</tr>
<tr>
<td>F-measure</td>
<td>95.05</td>
<td>90.14</td>
</tr>
</tbody>
</table>

Table 4. Calibrating system results vs. humans assisted by the tool.

<table>
<thead>
<tr>
<th>Measure</th>
<th>A2 vs. A1</th>
<th>Tool vs. A1</th>
<th>A1 vs. A2</th>
<th>Tool vs. A2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>80.99</td>
<td>93.78</td>
<td>92.54</td>
<td>95.55</td>
</tr>
<tr>
<td>Precision</td>
<td>92.54</td>
<td>96.69</td>
<td>80.99</td>
<td>86.07</td>
</tr>
<tr>
<td>FallOut</td>
<td>0.19</td>
<td>0.20</td>
<td>1.00</td>
<td>0.86</td>
</tr>
<tr>
<td>Accuracy</td>
<td>98.88</td>
<td>99.62</td>
<td>98.88</td>
<td>99.16</td>
</tr>
<tr>
<td>Error</td>
<td>1.12</td>
<td>0.38</td>
<td>1.12</td>
<td>0.84</td>
</tr>
<tr>
<td>F-measure</td>
<td>86.02</td>
<td>95.05</td>
<td>86.02</td>
<td>90.14</td>
</tr>
</tbody>
</table>

This second experimental case study was much more difficult to set up and evaluate than the first for the following reasons:
- Ambiguity in annotations: the large conceptual model of the domain allows significant overlap between some topics.
- Difficulty in identifying fragments for annotation: web documents contain various complex text structures, such as tables, menu labels, free text.
- Lack of a “gold standard” for correct annotation.

However, as a result of this experiment we can say that our semantic annotation framework was able to demonstrate reasonable results on the more general documents and the richer domain while maintaining fast performance.

5 Related Work

A number of tools have been shown to do well for various kinds of assisted or semi-automated semantic annotation of web content.
SMT [8] is semi-automatic tool for markup of documents combining commercial text extraction tools and manual annotation based on predefined templates to produce consistent OWL\(^5\) annotations. Templates are database slots representing entities of the OWL ontology. The user reads a preprocessed document, chooses a relevant template and fills it in, possibly assisted by the previous extraction stage. The main feature of SMT is that mapping of annotations to OWL is handled automatically, although the annotation process involves a high degree of user assistance.

SemTag [9] is an application that performs automated semantic tagging of large corpora. It is based on the Seeker platform for large-scale text analysis. It tags large numbers of pages with terms from TAP ontology, using corpus statistics to improve the quality of tags. The TAP knowledge base contains lexical and taxonomic information about popular objects such as music, movies, authors, sports, autos, health, and others. SemTag detects the occurrence of these entities in web pages and disambiguates them.

The KIM platform [10] is an application for automatic ontology-based named entities annotation, indexing and retrieval. In KIM, as well as in SemTag, semantic annotation is considered as the process of assigning to the entities in the test links to their semantic descriptions, provided by ontology. The platform is based on GATE (General Architecture for Text Engineering). The main contribution of KIM is recognition of named entities with respect to the KIMO ontology.

Another tool that has been used on a large-scale is SCORE [11], which integrates several information extraction methods, including probabilistic, learning, and knowledge-based techniques, then combines the results from the different classifiers.

Our approach fundamentally differs from all these tools: it uses a lightweight but robust context-free parse in place of tokenization and part-of-speech recognition; our method does not have the learning phase, instead it has to be tuned manually when being ported to a particular application, substituting or extending domain dependent components\(^6\); and it does not necessarily require a gazetteer or knowledge base of known proper entities, rather it infers their existence from their structural and vocabulary context in the style of software analyzers. This advantage helps make our tool faster and less dependent on the additional knowledge sources.

Much of the work in the information extraction community is aimed at “rule learning”, automating the creation of extraction patterns from previously tagged or semi-structured documents [12] and unsupervised extraction [13]. The approach of Pivk et al. [14] supports automatic population of ontologies from table-like structures converting HTML tables into semantic frames. The algorithm is based on structural representation of the table, identifies attribute- and instance-cells, and uses WordNet to find hyponyms for similar subtypes. Another tool for semi-automatic annotation of the documents is ASSAM [15], which exploits supervised machine learning techniques to assign to WSDL document fragments one or more semantic categories from a domain ontology.

---

\(^5\) http://www.w3.org/2004/OWL/

\(^6\) As we already have the experience of adaptation of our annotation framework to three different applications we can estimate this process takes about a couple of weeks of one person work.
While learning issues are not addressed by our work, the application of patterns to documents is in many ways similar to our method, in particular the ontology-based method of Embley et al. [16]. The major differences lie in the implementation – whereas Embley’s method relies primarily on regular expressions, our approach combines high-speed context-free robust parsing with simple word search. Embley’s approach is intended for processing preferably semi-structured web pages with multiple records: the more structured the page, the better the annotation results. Another approach making use of structural patterns to automatically propose annotations for the HTML documents is presented by Mukherjee et al. [17], who use the heuristic that semantically related items often have similar representation style and spatial position.

Wrapper induction methods such as Stalker [18] and BWI [19] which try to infer patterns for marking the start and end points of fields to extract, also relate well to our work. When the learning stage is over and these methods are applied, their effect is quite similar to our results, identifying complete phrases related to the target concepts. However, our results are achieved in a fundamentally different way – by predicting start and end points using phrase parsing in advance rather than phrase induction afterwards. The biggest advantage of wrappers is that they need small amount of training data, but on the other hand they strongly rely on contextual clues and document structure. In this case if the source document would be reorganized, the tool should be retrained on the newly annotated examples. In contrast, our method uses context-independent parsing and does not require any strict input format.

6 Conclusions and Future Work

We have presented and evaluated a tool-supported process for the semantic annotation of web documents. The evaluation of our proposal included two case studies and the experimental results suggest good performance on the part of the semantic annotation tool. More importantly perhaps, the results suggest that productivity of a human annotator can increase substantially if the annotator works with the output of our tool, rather than conduct the annotation task manually. Our experiments also suggest that the tool is scalable when used with larger document sets. Apart from the experimental evaluation, we also consider the evaluation scheme itself as a novel contribution in that it measures not only the quality of the annotation, but also productivity improvements for human annotators. Our evaluation framework also takes into account inter-annotator disagreements to calibrate the quality scores for the tool.

Our future research plans include tackling the problem of automatically generating inputs to the annotation process, such as object grammars and category keywords. We also propose to conduct experiments adapting other techniques used in software analysis to improve the quality of annotations and to accommodate different annotation granularities.
7 References


