Applying Software Analysis Technology to Lightweight Semantic Markup of Document Text

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

[1] Dept. of Information and Communication Technology, University of Trento, Italy
{nadzeya.kiyavitskaya,nicola.zeni,luisa.mich}@dit.unitn.it
[2] ITC-IRST, Trento, Italy, and School of Computing, Queens University, Canada
cordy@cs.queensu.ca
[3] Dept. of Information and Communication Technology, University of Trento, Italy, and Dept. of Computer Science, University of Toronto, Canada
jm@cs.toronto.edu

Abstract. Software analysis techniques, and in particular software "design recovery", have been highly successful at both technical and business-level semantic markup of large scale software systems written in a wide variety of programming languages, and in particular have proven efficient and scalable in assisting the resolution of the "year 2000" problem for billions of lines of legacy source code. In this work we describe a first experiment in applying the same technical solutions and tools that have proven so successful in software markup to the more general problem of semantic markup of text documents. In this early report we describe our adaptation of the software analysis techniques, propose a general domain-independent architecture for semantic markup using them, and demonstrate its feasibility in a limited but realistic domain of application by comparison with both raw and tool-assisted human semantic markers.

1 Introduction

Semantic markup [1] is the annotation of world-wide web or other natural language documents to assign explicit real-world semantics to portions of the document in order to allow for rapid identification of documents and parts of documents relevant to a particular question or purpose. Semantic markup represents the essential difference in the vision of the "semantic web" [2].

Given the number and scope of documents on the world-wide web, transition to the semantic web vision cannot be achieved without large-scale efficient automation of semantic markup [3]. It seems clear that full natural language understanding systems will not be ready for this task for some time, and thus lightweight, approximate methods may be our best hope for this immediate and pressing need.

2 Software Analysis and Design Recovery

Another domain in which such an immediate and pressing need for large scale analysis of source texts has been faced is legacy software source analysis, which successfully faced the "year 2000" problem only a few years ago. Some of the most successful techniques for automating solutions to that problem utilized "design recovery" [4], the analysis and markup of source code according to a semantic design theory, to assist in this problem [5].

Design recovery from source code poses many problems in common with natural language text processing: the need for robust parsing techniques, because real documents do not always match the grammars of the languages they are written in; the need to understand semantics of the source text according to a semantic theory or ontology; semantic clues drawn from a vocabulary for the domain; contextual clues drawn from the syntactic structure of the source text; and inferred semantics from exploring relationships between semantic entities and their properties, contexts and related entities.

Formal processes for software design recovery utilize a range of tools and techniques designed and proven to address these challenges for many billions of lines of legacy software source code [6]. One of these is the generalized parsing and structural transformation system TXL [7], the basis of the automated year 2000 system LS/2000 [5].

In this work we propose to leverage the highly efficient methods and tools already proven in the software analysis and markup domain as the basis of a new lightweight method for semantic analysis and markup of natural language texts, in the hope that we can attain similar performance and scalability while yielding good quality approximate results.

3 A New Architecture

The architecture of our solution (Fig. 1) is based on the LS/2000 software analysis architecture [5], generalized to allow for easy parameterization by a range of semantic domains. 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 (object, 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 ontology (entity-relationship schema and interpretation), shown on the right.

The process uses 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 [8], 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

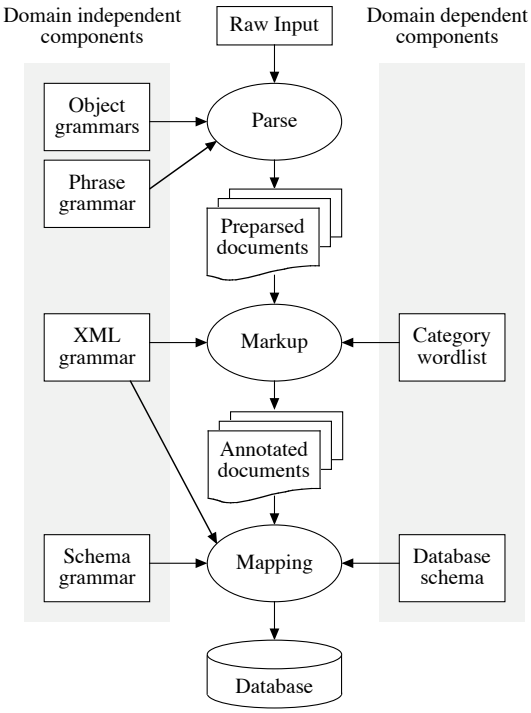

Fig. 1. Architecture of our semantic markup process.

would be done in a programming language parser (Fig. 2). The parse is linear in the length of the input and runs at compiler speeds. In our first experiments this parse is relatively coarse-grained, ignoring language structure below the sentence and verb-clause level.

In the second stage, initial semantic markup of the document is derived using a wordlist file specifying both positive and negative indicators for semantic categories (Fig. 3). Indicators can be both literal words and phrases (e.g., "air conditioning") and names of parsed entities (e.g., "email"). Phrases are marked up once for each category they match - thus at this stage a sentence may end up with many different (even conflicting) semantic markups. Vocabulary word and entity lists are derived from the ontology for the target semantic domain. At present this is done by hand, but work is underway on automating vocabulary and schema construction from a formal ontology. This stage uses the structural pattern matching and source transformation capabilities of the TXL transformation engine in much the same way as it is used for software markup [9] to yield a preliminary marked-up text in XML form (Fig. 4).

The third stage uses the XML marked-up text to populate an entity-relationship database according to an ER schema approximating the ontology for the domain. The schema is provided as an XML template (Fig. 5) derived (at present by hand) from the ontology for the target domain. Sentences and phrases with

```
% International phone number grammar
tokens
      longnumber  "\d\d\d\d\d\d\d\d*"
      zeronumber  "0\d\d*"
end tokens

define phone
      '+ [anynumber][opt zerocode][opt phone_separator]
         [repeat number_separator+][anynumber]
   |  '+ [anynumber][opt zerocode][opt phone_separator][repeat longnumber+]
   |  '( [opt space][opt '+][anynumber][opt space]')[space_number]
         [repeat separator_number]
   |  '( [opt space][opt '+][anynumber][opt space]')[opt space]
         [repeat longnumber+]
   |  [anynumber] [opt phone_separator] [repeat longnumber+]
   |  [zeronumber][opt phone_separator][longnumber]
   |  [zeronumber] [separator_number] [repeat separator_number+]
   |  [repeat number_separator][longnumber]
   |  [number_separator][number_separator][repeat number_separator][anynumber]
   |  [opt '+][longnumber] [repeat space_number]
end define
```

Fig. 2. Part of TXL grammar for phone number objects.

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

Fig. 3. 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.

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. The result of this stage is an entity-relationship database in XML format suitable for importing into standard database tools such as MySQL or MS Access.

Both the XML marked-up text and the database are products of our process, the former yielding an approximate semantic markup of the original document text and the latter serving as a query answering service yielding relevant sentences and phrases from the document text in response to database queries.

4 A First Experiment

As a first proof-of-concept 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 drawn from online newspapers.

```
3348 <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 347.7894321 </contact>
```

Fig. 4. Example result 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.

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

Fig. 5. Database template schema for accommodation advertisements.

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

In order to make a realistic test of the generality of the method, we restricted ourselves to some constraints: no proper nouns or location-dependent phrases in our vocabulary, raw uncorrected text, and no formatting or structural cues. The human markers against whom we were testing could take full advantage of all of this knowledge in their results, but the tool could not.

In the first instance 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 several categories of semantic information in the advertisements according to a given accommodation ontology, which was reduced by hand to an entity-relationship schema in XML format for input to our system (Fig. 5). The desired result was a database with one instance of the schema for each advertisement in the input, and the marked-up original advertisements (Fig. 4).

5 Early Results

The evaluation of semantic markup presents a particularly difficult problem. 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 opin-

ions. 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 modest but cost-effective three stage validation. At each stage, we were interested in measuring the precision, recall, fallout, accuracy and error (using the definitions of Yang [10]) for the tool's automated markup compared to human opinions. The tool was allowed 38 randomly chosen advertisements as a "training set", although no real training took place - rather, after encoding the target ontology into the tool's vocabulary and schema tables, the tables were allowed to be tuned to do well on this first set by hand.

In the first stage, the tool and each of two human markers were asked to mark 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 (Fig. 6a), and then calibrated against each of the two as definitive (Fig. 6b,c). By comparison with these (widely differing) 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 simple system.

In the second stage evaluation, we were interested in measuring the effect of the initial automated markup of the tool on human markup efficiency. The time

Measure	System vs. Human 2	System vs. Human 1
Recall	92.42%	92.19%
Precision	76.25%	73.75%
Fallout	7.54%	8.27%
Accuracy	92.45%	91.82%
Error	7.55%	8.18%
F-Measure	83.56%	81.94%

(a) System vs. Humans

Measure	Human 2 vs. Human 1	System vs. Human 1
Recall	90.63%	92.19%
Precision	87.88%	73.75%
Fallout	3.15%	8.27%
Accuracy	95.60%	91.82%
Error	4.40%	8.18%
F-Measure	89.23%	81.94%

(b) Calibrated against Human 1

Measure	Human 1 vs. Human 2	System vs. Human 2
Recall	87.88%	92.42%
Precision	90.63%	76.25%
Fallout	2.38%	7.54%
Accuracy	95.60%	92.45%
Error	4.40%	7.55%
F-Measure	89.23%	83.56%

(c) Calibrated against Human 2

Fig. 6. First stage experiment - system vs. unassisted human markup.

Measure	Rome (10 ads) Training set	Rome (10 ads) Test set-1	Rome (100 ads) Test set-1	Venice (10 ads) Test set-2
Recall	98.73%	94.20%	92.31%	86.08%
Precision	97.50%	97.01%	96.93%	95.77%
Fallout	0.84%	0.87%	0.93%	1.29%
Accuracy	99.06%	98.00%	97.43%	95.51%
Error	0.94%	2.00%	2.57%	4.49%
F-Measure	98.11%	95.59%	94.56%	90.67%

Fig. 7. Third stage experiment - system vs. assisted human opinions.

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 he human marker was observed to use 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 be even greater than we observed.

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 opinion of the tool. For this test, 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, and a new set of 10 from Venice. The summary of results so far is shown in Figure 7. Accuracy for all of the Rome sets is about 98%, and in the new set from Venice, a completely different location, the accuracy was measured as over 95% with similar precision. A drop in recall to 80% is indicative of locality effects from the original training set - a wider set will be needed to make a general tool for advertisements.

Obviously these tests on this one small domain are insufficient to make any meaningful statements about the generality or applicability of our new architecture and method, and we are presently moving on to large scale tests in both this and other domains, including travel websites, news stories and academic publications in the coming months.

At best what we can say so far is that the results of our small study do validate that there is a real potential for a fast lightweight method based on the software design recovery model. Even without local knowledge and using a very small vocabulary, we have been able to demonstrate accuracy comparable to the best heavyweight methods, albeit thus far on a very limited domain. Performance of our as yet untuned experimental tool is also already very fast, handling for example 100 advertisements in about 1 second on a 1 GHz PC. Performance has also been validated as linear on sets ranging from 38 to 7,600 advertisements (about 2,500 to 500,000 words), at a rate of about 53 kb/sec on a 1 GHz PC.

6 Related Work

Many systems have been shown to do well for various kinds of assisted or semi-automated semantic tagging of large corpora. SHOE [11] and Ontobroker [12] for example are pioneering tools providing machine assistance for manual semantic markup, and AeroDAML [13] automatically generates DAML annotation suggestions for Web pages given an ontology.

Recent work includes fully automated techniques more directly comparable to ours. SemTag [3] for example has been able to process enormous amounts of data, reporting accuracy measures of about 79% in identifying instances of a given set of known entities in web pages. Using compiler-style tokenization of source text followed by a search for entities of the very large TAP taxonomy, SemTag expends much of its effort in disambiguating multiple tags using local context. By contrast our system aims primarily at higher level markup, and tries to minimize ambiguity using a combination of structural information from the parse and contra-indicators in the vocabulary.

The KIM (Knowledge and Information Management) platform [14] is designed for the purpose of implementing the full semantic web vision. Compiler-style tokenization begins the process, followed by a split into sentences and part-of-speech tagging. A gazetteer and ontology-augmented pattern-matching grammars encode rules for markup of a large set of entities of general in interest. In our system phrase structure is identified by the parse, and rules are driven by simple word occurrences.

S-CREAM (Semi-automatic Creation of Metadata) [15] uses a framework that includes a learnable information extraction component. Users hand-annotate a corpus for training the learner, which infers markup rules for a subset of a given ontology. S-CREAM utlilizes the same front end and basic set of steps as KIM, its distinguishing feature being its automated inference of annotation rules from the training set.

Our work differs from all of these approaches in three fundamental ways - first, it uses an extremely lightweight but robust context-free parse in place of tokenization, regular expressions and part-of-speech recognition. Second, it does not use 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. And third, it has already been shown to handle higher-level semantic markup for concepts above and depending on entities rather than just the entities themselves.

Information extraction [16] is a closely related problem to semantic markup with an even larger base of published work. Rather than markup of the documents themselves, the goal of information extraction is the population of a template with slots for information to be extracted from semi-structured documents such as web pages. This corresponds to the third step in our process (population of the database schema after semantic markup).

In general, techniques used for information extraction use patterns based on syntactic and semantic constraints in some ways similar to our initial phrase and object parsing stage. While 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 [17] and unsupervised extraction [18], issues our work does not address, the actual application of the patterns to documents is in many ways similar to our method.

In particular, ontology-based methods such as Embley et al's [19] are in some ways quite similar, using a relational schema as the target structure where we use an entity-relationship schema, and using keyword lists and constraints quite similar to our own vocabularies. The major differences lie in the implementation - whereas Embley's method relies primarily on regular expressions, ours combines high-speed context-free robust parsing combined with simple word search.

Wrapper induction methods such as Stalker [20] and BWI [21] 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.

7 Conclusions and Future Work

Obviously this work is only beginning. At most we have thus far demonstrated that applying software design recovery techniques to semantic markup of documents is feasible and has potential. It is also clear that these techniques can retain their efficiency in this new domain, exhibiting very fast linear performance even without tuning, and it seems likely that they could provide high levels of markup accuracy.

However, the work set out for us now is clear - testing and validation of our method on large corpora and richer conceptual spaces so that a more meaningful comparison with the state of the art can be done. While our method has done well for our small but realistic first domain of application, it is by no means clear that it will retain such high levels of accuracy as we scale to larger and richer domains.

There are still a number of techniques used in software analysis that we have not taken advantage of - alias resolution, unique naming, architecture patterns, markup refinement and so on. In future we hope to explore these other techniques to improve our semantic annotation architecture as well.

References

1. Daconta, L., Orbst, L., Smith, K.: The Semantic Web: A guide to the future of XML, web services and knowledge management (2003)
2. Berners-Lee, T., Hendler, J., Lassila, O.: The Semantic Web. Scientific American **284** (2001) 34–43
3. Dill, S., Eiron, N., Gibson, D., Gruhl, D., Guha, R., Jhingran, A., Kanungo, T., McCurley, K., Rajagopalan, S., Tomkins, A., Tomlin, J., Zien, J.: A case for automated large-scale semantic annotation. J. Web Semantics **1** (2003) 115–132

4. Biggerstaff, T.: Design recovery for maintenance and reuse. IEEE Computer **22** (1989) 36–49
5. Dean, T., Cordy, J., Schneider, K., Malton, A.: Experience using design recovery techniques to transform legacy systems. In: Proc. 17th Int. Conference on Software Maintenance. (2001) 622–631
6. Cordy, J., Dean, T., Malton, A., Schneider, K.: Source transformation in software engineering using the TXL transformation system. J. Information and Software Technology **44** (2002) 827–837
7. Cordy, J.: TXL – a language for programming language tools and applications. Proc. 4th Int. Workshop on Language Descriptions, Tools and Applications, Electronic Notes in Theoretical Computer Science **110** (2004) 3–31
8. Dean, T., Cordy, J., Malton, A., Schneider, K.: Agile parsing in TXL. J. Automated Software Engineering **10** (2003) 311–336
9. Cordy, J., Schneider, K., Dean, T., Malton, A.: HSML: Design-directed source code hotspots. In: Proc. 9th International Workshop on Program Comprehension. (2001) 145–154
10. Yang, Y.: An evaluation of statistical approaches to text categorization. J. Information Retrieval **1** (1999) 67–88
11. Sean, L., Lee, S., Rager, D., Handler, J.: Ontology-based web agents. In: Proc. 1st International Conference on Autonomous Agents. (1997) 59–68
12. Decker, S., Erdmann, M., Fensel, D., Studer, R.: Ontobroker: Ontology-based access to distributed and semi-structured information. In: Proc. 8th Working Conference on Database Semantics. (1999) 351–369
13. Kogut, P., Holmes, W.: AeroDAML: Applying information extraction to generate DAML annotations from web pages. In: Proc. KCAP-2001 Workshop on Knowledge Markup and Semantic Annotation. (2001)
14. Popov, B., Kiryakov, A., Ognyanoff, D., Manov, D., Kirilov, A.: KIM: a semantic platform for information extaction and retrieval. J. Web Semantics **10** (2004) 375–392
15. Handschuh, S., Staab, S., Ciravegna, F.: S-CREAM: Semi-automatic CREAtion of Metadata. In: Proc. 13th Int. Conference on Knowledge Engineering and Management. (2002) 358–372
16. Muslea, I.: Extraction patterns for information extraction tasks: A survey. In: Proc. AAAI-99 Workshop on Machine Learning for Information Extraction. (1999) 1–6
17. Nobata, C., Sekine, S.: Towards automatic acquisition of patterns for information extraction. In: Proc. International Conference on Computer Processing of Oriental Languages. (1999)
18. Etzioni, O., Cafarella, M.J., Downey, D., Popescu, A.M., Shaked, T., Soderland, S., Weld, D.S., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. Artificial Intelligence **165** (2005) 91–134
19. Wessman, A., Liddle, S.W., Embley, D.W.: A generalized framework for an ontology-based data-extraction system. In: Proc. 4th Int. Conference on Information Systems Technology and its Applications. (2005) 239–253
20. Muslea, I., Minton, S., Knoblock, C.A.: Active learning with strong and weak views: A case study on wrapper induction. In: Proc. 18th Int. Joint Conference on Artificial Intelligence. (2003) 415–420
21. Freitag, D., Kushmerick, N.: Boosted wrapper induction. In: Proc. 17th National Conference on Artificial Intelligence. (2000) 577–583