On the Definition of Patterns for Semantic Annotation

Mónica Marrero, Julián Urbano, Jorge Morato and Sonia Sánchez-Cuadrado

University Carlos III of Madrid
Department of Computer Science

mmarrero@inf.uc3m.es jurbano@inf.uc3m.es jmorato@inf.uc3m.es ssanchec@ie.inf.uc3m.es

ABSTRACT

The semantic annotation of documents is an additional advantage for retrieval, as long as the annotations and their maintenance process scale well. Automatic or semi-automatic annotation tools help in this matter with the use of patterns. In this paper we analyze the advantages of creating these patterns with standard web languages, as well as the requirements they should meet. We adopt the Speech Recognition Grammar Specification, by the W3C, initially intended for speech recognition in the Web. Our objective is to achieve its full adaptation to the information extraction processes, exploiting its powerful recognition, reuse and flexibility capabilities.

Categories and Subject Descriptors

I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods – *representation languages*.

General Terms

Standardization, Languages

Keywords

Semantic annotation, information extraction, pattern design

1. INTRODUCTION

There are several requirements desired for semantic annotation tools, such as the support for collaborative and automatic annotation [1]. When applied on large collections, the automation is especially important in order to provide the scalability needed to annotate existing documents and reduce the burden of annotating new documents [2]. This automation is typically implemented with Information Extraction (IE) techniques, thanks to some attributes of the text that can provide information for recognition and semantic annotation. These attributes are exploited by certain extraction patterns, whose definition is usually specific of each tool. Thus, the patterns use to be incompatible one to another, and they are not available for their later reuse, let alone in standard formats like OWL or XML.

The pattern models used vary in terms of power. For example, the *Parmenides* system [3] uses context-free grammars (sufficient to recognize virtually every natural language construction), but other automatic annotation tools use less powerful models, based on bag-of-words [4] or regular expressions. Other models make the modification and reuse of patterns very difficult, either for not being human-readable (e.g. neural networks) or for not being partially updatable (e.g. complex patterns created with machine learning). The flexibility upon addition of new attributes or constraints is variable too. For example the tool *Ex* [5], based on extraction ontologies [6], allows for the definition of new constraints in the form of axioms, though the model itself does not provide the means for their interpretation.

ESAIR'10, October 30, 2010, Toronto, Ontario, Canada.
ACM 978-1-4503-0372-9/10/10.

In this paper we analyze the advantages of adopting customizable and shareable models to define patterns with web technologies, and we establish the requirements that such patterns should meet. Based on these requirements, we propose to adapt the Speech Recognition Grammar Specification 1.0 (SRGS), by the W3C (http://www.w3.org/TR/speech-grammar), originally intended for the interaction with web pages by means of voice commands. We present the adjustments implemented in the standard, and discuss their advantages for Information Extraction.

2. DEFINITION OF TEXTUAL PATTERNS

As they appear in a level previous to the annotation itself, extraction patterns are more flexible and effective regarding changes in the documents because only the pattern, rather than all annotations, needs to be modified. To that end, the patterns should not only be easily updatable, but must also have enough power to recognize any element prone to be annotated, supporting several types of attributes and documents (HTML, PDF, etc). The automatic and/or collaborative creation of patterns and their distribution and reuse are also factors to be considered, as well as their relation with ontologies to support the semantics they capture.

However, developing models powerful, reusable and flexible enough to define a wealth of patterns makes these definitions inherently more complex, which could ultimately reduce their adoption. As with the annotation process, the use of standards in the model formalization would certainly facilitate their adoption. Updatable, comprehensive and human-readable models using web languages would make the patterns easier to distribute, promoting their collaborative creation and adoption to be reused partially or completely for different documents. Ultimately, it could be possible to interpret them in the very web pages, which could lead to an infrastructure of services for automatic identification and classification in the Web.

Table 1. Mapping between ABNF (RFC5234) and SGRS main features.

	ABNF	**XML (SGRS)**
Rule definition	A = …	<grammar><**rule** id="A">…</**rule**></grammar>
Alternative	A = a / b A =/ c	<rule id="A"> <**one-of**><**item**>a</**item**>…</**one-of**></rule>
Alt. weight		<item **weight**="n">a</item>
Repetitions	<*min*>*<*max*>a <n>a	<item **repeat**=min-*max*>a</item>
Repetition probability		<item repeat=min-*max* **repeat-prob**="p">a</item>
Non-terminal reference	A = B C	<rule id="A"> <**ruleref** uri="gram#B"/>…</rule>

As an attempt to solve this problem, we decided to adapt the SRGS standard. This standard specifies how to map an ABNF-like grammar (Augmented Backus-Naur Form) to XML, to model the voice commands expected by web users and thus guide the speech recognizers. This standard already meets some of the requirements discussed: thanks to the use of ABNF, metalanguage based on BNF to facilitate information exchange in the Internet,

SRGS defines context-free grammars with Web standards and has a very well defined and accepted DTD to map the constructions of ABNF to XML (see Table 1).

The most remarkable advantages of SRGS over ABNF are 1) the addition of weights for the alternative rules, 2) the addition of probabilities for the repetitions of an item, 3) the expansion of core rules (with predefined rules such as *GARBAGE* to allow any input token, *NULL* and *VOID*), 4) the possibility to reference one grammar with its URI, so that its rules can be used in another grammar, 5) the specification of rule attributes such as the public or private visibility of a rule (to indicate whether it can be used from another grammar or not), and grammar attributes to indicate some meta-information, such as the type of input or language, version, the start rule, etc.

3. USE IN INFORMATION EXTRACTION

We have adopted the SRGS standard to define patterns in a prototypical Information Extraction tool based on another tool developed for the automatic generation of patterns [7], although we had to perform several modifications.

First, we considered the specification of the semantics. SRGS allows the use of semantic tags among the production rules so that they are interpreted in that very place when parsing the text. However, in IE it is also necessary to delimit the element recognized, so we added the attribute *semantic* to the element *rule* to indicate that the text recognized by that rule is actually the text to annotate. This attribute points to the description of a resource, usually through the URI of an external ontology, to indicate the semantics of the annotation. This way, several semantics, nested within the same grammar, can be determined to facilitate the recognition of complex scenarios (e.g. speaker, place and time of a talk) on top of isolated entities (e.g. persons, places and times).

Second, the characteristics most frequently modeled by IE tools are those referred to the syntax, semantics and format of the text. SRGS allows for the specification of regular expressions and even wrappers, where the formatting tags are considered as part of the syntax. Nonetheless, it is not possible to define more than one constraint at the same time (e.g. syntax and semantic tagging), so each of them has to be defined by adding more details to the definition of the pattern (e.g. *[syn = NNP, sem = CITY | COUNTRY | PROVINCE, orth != lowercase]* [3]) or by using several vocabularies (e.g. named entity tags, syntax tags, tokens, lemmas, etc.). The latter alternative is more flexible than the former, as one would not need to modify the model itself just to add new constraints, so we incorporated operations between non-terminals in order to support it. They are specified as child elements of *rule*, and so far we have implemented the boolean operations *AND* and *NOT*, though others can be added without further modifications of the schema. These operations are actually applied upon the text resulted from parsing. For example, a rule C with the operation *A AND NOT B*, where *A* and *B* are non-terminal symbols of the grammar, evaluates to *true* (i.e. it is a valid rule for that text) only if the text is parsed by *A* but not by *B*, using the rule C for that text or not. These operations are especially useful for techniques performing some kind of learning based on positive and negative examples.

But not all restrictions can be modeled syntactically, or just not easily so. For example, an HTML text in bold face can be detected syntactically with the tag <*strong*>. However, this tag might appear far away from the actual text of interest, so that applying a function to detect it is easier than complicating the grammar. Moreover, such formatting may be indicated very differently in other types of document, so altering the grammar for it would not be the best choice. To incorporate this functionality to SRGS, we have added the element *restriction* as child of *rule*. Each non-terminal can thus have several restrictions, each of which points to the URI of a function to be applied. This function may be a Web service or a local procedure, which is applied upon the text parsed by the non-terminal symbol and accepts the rule only if it evaluates to *true*. This way, it is possible to create distributed repositories of frequently used functions for certain types of document (e.g. *bold* function for HTML or PDF, *disambiguation* according to the concrete semantics, etc.) and with sufficient flexibility to create ad-hoc solutions capable of handling infrequent documents or restrictions.

4. CONCLUSIONS AND FUTURE WORK

The annotation processes are little scalable without the aid of techniques for their automation, usually based on the recognition of textual patterns. In this paper we showed how the Speech Recognition Grammar Specification can be adapted to be used for Information Extraction. As a result, the patterns have powerful recognition capabilities, besides the possibility of being edited and reused partially or completely, and flexibility to be adapted to diverse types of document. The use of standards like ABNF and XML brings additional advantages thanks to existing formalisms and tools that facilitate the task. The advantages of this type of definitions in the Web would be immediate too, thanks to the possibility of locating and interpreting the patterns in the own web pages, thus facilitating the maintenance of the annotations.

Although the grammar for such patterns can be easily generated from regular expressions or wrappers, more research should focus on their automatic generation from examples, which would eventually lead to fully automated semantic annotation.

ACKNOWLEDGEMENTS

We acknowledge the National Plan of Scientific Research, Development and Technological Innovation, which has funded this work through the research project TIN2007-67153.

REFERENCES

[1] V. Uren, P. Cimiano, et al. Semantic Annotation for Knowledge Management: Requirements and a Survey of the State of the Art. *Journal of Web Semantics*. 4(1): 14-28, 2006.

[2] L. Reeve and H. Han. Survey of Semantic Annotation Platforms. *ACM Symposium on Applied Computing*, pages 1634-1638, 2005.

[3] F. Rinaldi, A. Vasilakopoulos, et al. *Parmenides Technical Report TR-U4.3.1 CAFETIERE: Conceptual Annotations for Facts, Events, Terms, Individual Entities, and Relations*. 2005. http://www.nactem.ac.uk/files/phatfile/cafetiere-report.pdf.

[4] M. Kudelka, V. Snasel, et al. Semantic Annotation of Web Pages Using Web Patterns. *Advanced Internet Based Systems and Applications*. 2009. 280-291.

[5] M. Labský, V. Svátek, et al. The Ex Project: Web Information Extraction Using Extraction Ontologies. *Knowledge Discovery Enhanced with Semantic and Social Information*. 2009. 71-88.

[6] D.W. Embley. Toward Semantic Understanding - An Approach Based on Information Extraction Ontologies. *Australasian Database Conference*. 27(Winslett): 3-12, 2004.

[7] R. Matallanos, (advisor M. Marrero), Generación Automática de Patrones. *University Carlos III of Madrid*. 2009.