Recycling an Information Extraction system to automatically produce Semantic Annotations for the Web

Thierry Poibeau
Thales et LIPN
Domaine de Corbeville
F-91 404 Orsay, France
thierry.poibeau@thalesgroup.com

Alexandre Arcouteil
INALCO
2, rue de Lille
F-75007 Paris, France,
alexandre.arcouteil@inalco.fr

Cyril Grouin
INALCO
2, rue de Lille
F-75007 Paris, France,
cgrouin@msh-paris.fr

Abstract

This paper is intended to show how an Information extraction system can be recycled to produce RDF schemas for the semantic web. We show that this kind of systems has to respect operational constraints like the fact that the information produced must be highly relevant (high precision, possibly low recall). We conclude in reconsidering some tasks like *Question Answering* (Q/A): the production of explicit structured data on the web will lead a better relevance of information retrieval engines.

1 Introduction

Information Extraction (IE) is a technology dedicated to the extraction of structured information from texts. This technique is used to highlight relevant sequences in the original text or to fill pre-defined templates (Pazienza 1997).

With the development of the semantic web, such tools appear to be very interesting to automatically extract semantic information from existing web pages. In this paper, we will not focus on the analysis of semi-structured documents by means of wrappers. Even if HTML is a semi-structured format, most of the information available on the web is located inside unstructured and untagged paragraphs.

This paper is intended to show how an Information extraction system can be recycled to produce RDF schemas for the semantic web. We will see that this kind of systems has to respect operational constraints like the fact that the information produced must be highly relevant (high precision, possibly low recall). We conclude in reconsidering some tasks like *Question Answering* (Q/A): the production of explicit structured data on the web will lead a better relevance of information retrieval engines.

2 Related work

The bases of IE as defined in the introduction are exposed in (Pazienza, 1997). IE is known to have established a now widely accepted linguistic architecture based on cascading automata and domain-specific knowledge (Appelt *et al*, 1993). Several papers mentioned current limitations of MUC-like systems in terms of adaptability and studied the resource development cost (Grishman & Sundheim, 1996). Event'99 was a task intended to evaluate event-level indexing into news stories (Hirschman *et al.*, 1999). The idea is "to minimize the number of event-specific rules" to be produced and to favor a light generic template, informally called a "templette" (Event 99). This paper is based upon the same idea applied to the AFP newswire.

Different systems tried to extract information in analyzing the structure of different kind of texts. For example, (Lacroix *et al.* 98) presents a system able to extract information from the structure of HTML pages. This kind of application will increase with the development of more structured document (like XML documents). Wrapper factories go one step beyond, in connecting together distant pieces of texts and in extracting information from poorly structured documents (Sahuguet and Azavant 98).

To address the problem of portability, a recent research effort focused on using machine learning throughout the IE process (Muslea, 1999). A first trend was to directly apply machine learning methods to replace IE components. For example, statistical methods have been successfully applied to the named-entity task. Among others, (Bikel *et al.*, 1997) learns names using a variant of hidden Markov models. However, a 90% success rate is reached at the cost of tagging manually about half a million words. (Cucchiarelli & Velardi, 1999) propose a more interesting approach: they adopt a hybrid approach mixing a core generic system extended with some learning mechanisms. Their system is able to learn new proper names by generalizing the data extracted by the basic rule-based system. We will adopt a very similar approach, except the fact that we want to stay in a symbolic framework, mainly for readability reasons.

3 Information extraction system

The architecture consists in a multi-agent platform. Each agent performs a precise subtask of the information extraction process. A supervisor controls the overall process and the information flow. The overall architecture is presented in (Poibeau, 2001).

The system can be divided into five parts: information extraction from the structure of the text, the module for named entity recognition (location, dates, etc), semantic filters, modules for the extraction of specific domain-dependent information and modules for the filling of a result template.

- Some information is extracted from the structure of the text. Given that the AFP newswire is formatted, some wrappers automatically extract information about the location and the date of the event. This non-linguistic extraction increases the quality of the result by providing 100% good results. It is also accurate when one thinks of the current development of structured text (HTML, XML) via the web and other corporate networks.
- The second stage is concerned with the recognition of relevant information by means of a linguistic analysis. This stage allows the recognition of various named entities (person names, organizations, locations and dates) of the text. New kinds of named entities can be defined according to a new domain (for examples, gene names to analyze a genome database). We use the finite-state toolbox Intex to design dictionaries and automata (Silberztein 1993).

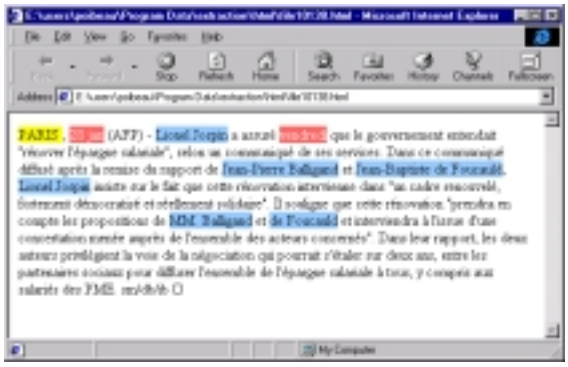

Figure 2: The named entity recognizer

- The third stage performs text categorization from "semantic signatures" automatically produced from a rough semantic analysis of the text. We use an external industrial system implementing a vector space model to categorize texts (the Intuition™ system from the French company Sinequa, cf. Salton (1988)).
- The fourth stage extracts specific information (most of time, specific relationships between named entities). It can be for example the number of victims of a terrorist event. This step is achieved in applying a grammar of transducers (extraction patterns) over the text.
- The next stage links all these information together to produce one or several result template(s) that present(s) a synthetic view of the information extracted from the text. The template corresponding to the text is chosen among the set of all templates, according to the identified category of the text (registered by the system at the third analysis step). A specific template is produced only if some main slots are filled (the system distinguished among obligatory and optional slots).

Partial templates produced by different sentences are merged to produce only one template per text. This merging is done under constraints on what can be unified or not. The results are then stored in a database, which exhibit knowledge extracted from the corpus.

4 Application overview: knowledge extraction from various domains

Various applications have been developed using the above architecture, to ensure its genericity. They concerned different domains:

- Event-based extraction and indexing of the AFP newswire. This multi-domain extraction system is currently running in real time, on the AFP newswire. About 15 templates have been defined that cover about 30% of the stories. From the remaining 70%, the system only extract surface information, especially thanks to the wrappers. The performances are between .55 and .85 P&R, if we do not take into account the date and location slots that are filled by means of wrappers. New extraction templates are defined to prove system scalability.
- Event-based extraction from financial news stories (FirstInvest, a French financial website). This application is very close to the previous one.
- Extraction of gene interactions from the genomics database Flybase. This kind of bases are structured by gene description, but researchers want to find relations among genes. In this context, the IE engine is intended to automatically produce a knowledge base about gene interaction, from the analysis of free texts.
- Customer Request Management application (extracting information from emails relating software problems). This last case poses the problem of email analysis and management. The language used in such texts is not as correct as it can be in news stories. Specific grammar and orthographic relaxations must be applied to achieve relevant results.

The last three applications concern texts from the Internet. FirstInvest is an electronic financial newswire available on the Web. Flybase, like other electronic databases in genomics, is a collection of public data freely available for researchers. The CRM application concerns a currently very popular area, which is also related to Knowledge Management.

5 Semantic annotations and other outputs

The system currently produces various kinds of output, for example:

- XML(/HTML) tagged texts for named entity highlighting in texts.

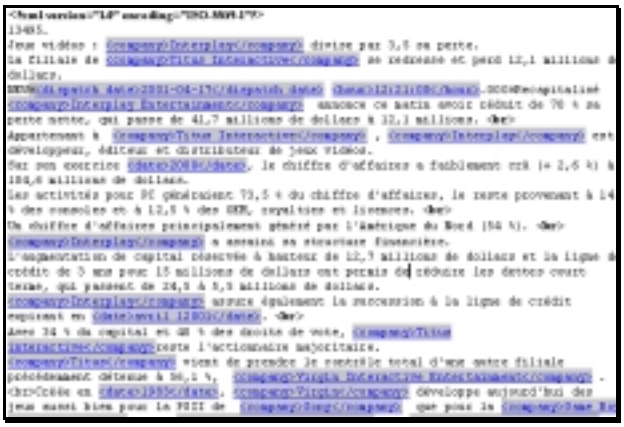

Figure 3: A news story from FirstInvest annotated with XML tags

- Event database for the analysis of the AFP newswire. A new template is produce for each new event.

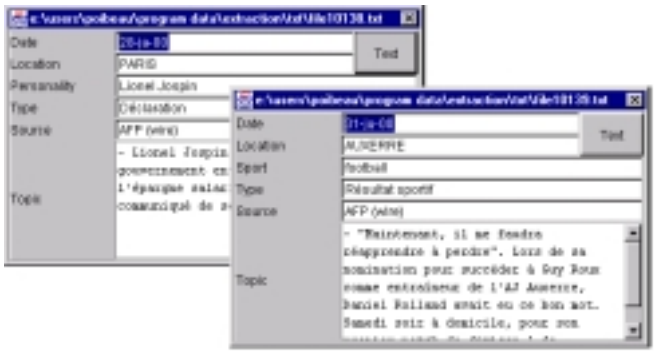

Figure 4: Event-base AFP indexing: each text fills a specific template, given its topic

- A dynamic knowledge base for gene interaction (a query-able knowledge base made of Prolog-like terms)

```
activation(1.28,Dfd)
activation(5-HT1A,C)
activation(ac,E)
activation(Ac13E,G)
activation(Dfd,1.28)
interaction(2R-F,mys)
```

interaction(2R-L,mys)

Figure 5: A part of the knowledge base generated from the analysis of Flybase

The range of performance is generally located between 60 and 80 P&R[1]. However, it is possible to semi-automatically adapt the system so that precision is very high. This point is crucial to produce high quality data for subsequent processing. The system then has a lower recall than classical IE tools (recall between .30 and .50; in the genomics domain, we often see a precision above .95 with a recall of .15, which is not a problem as such since genomics databases are highly redundant. Of course, an effort is made to produce data with the same precision but with a higher recall).

6 Information Extraction and RDF

Expressing structured information using the RDF syntax should provide interoperability between RDF-based applications. Therefore, two kinds of information should be produced by the IE system:

- a structure analysis inferring a RDF schema statement;
- structured data according to this RDF schema.

1.1 Structure analysis

An IE system isolates semantic groups from which a RDF schema is built. As an example, in the case of Flybase, the IE system identifies interactions between genes, the list of genes and the list of interactions. A RDF schema defines an object class and the relations that could exist between several objects from these classes. In our case, these classes are the class "gene" and the class "interaction":

```
<!-- Class Statement -->+
<rdf:Class id="Genes" />
<rdf:Class id="gene" />
<rdf:Class id="Interactions" />
<rdf:Class id="interaction" />
```

Then, the system defines statements about existing constraints on the classes and the properties, according to the syntactic analysis of the document:

[1] P&R is the harmonic means of recall and precision. This metric is classical to measure the performance of filtering and extraction systems.

```
<rdf:Property id="type">

<rdfs:domain
rdfs:Resource="#Interactions" />
<rdfs:range
rdfs:Resource="#Interactions" />
</rdf:Property>

<rdf:Property id="agent">
<rdfs:domain
rdfs:Resource="#Genes" />
<rdfs:range rdfs:Resource="#gene"
/>
</rdf:Property>

<rdf:Property id="target">
<rdfs:domain
rdfs:Resource="#Genes" />
<rdfs:range rdfs:Resource="#gene"
/>
</rdf:Property>

<rdf:Property id="article">
<rdfs:domain
rdfs:Resource="#Interactions" />
<rdfs:range
rdfs:Resource="rdfs:Literal" />
</rdf:Property>

<rdf:Property id="nature">
<rdfs:domain
rdfs:Resource="#Genes" />
<rdfs:range rdfs:Resource="#Genes"
/>
</rdf:Property>
```

7 Structured data

According to the RDF schema, the gene and interaction description in Flybase is represented ina new RDF file. Let's take the example of two genes, abd-A and trx:

```
<gene id="abd-A">
<nature rdfs:Resources="#Genes" />
</gene>

<gene id="trx">
<nature rdfs:Resources="#Genes" />
</gene>
```

The interaction between the two genes is represented as follows:

```
<interaction id="5428">
<type
rdfs:Resources="#interaction" />
<agent rdfs:Resources="#abd-A" />
<target rdfs:Resources="#trx" />
```

```
<article
rdfs:Resources="#http://flybase.bi
o.indiana.edu/" />
</interaction>
```

8 Conclusion

In this paper we have shown that a versatile IE system is very appropriate to automatically analyze unstructured texts from the web and produce semantic annotations. Some researches still need to be done to produce more robust IE tools that will be able to deal with various kind of texts. We have proposed some methods (Poibeau, 2001), but large experiments still need to be done. In particular, it is necessary to mix NLP approach with wrappers to make good use of semi-structured texts.

These systems should change a bit the face of the Web. Given that more and more structured and semantically annotated data will be available, Question Answering systems should give more accurate answers to user requests, for example. In this sense, IE systems allow to really extract and structure the semantic of the Web.

Bibliography

Appelt D.E, Hobbs J., Bear J., Israel D., Kameyana M. and Tyson M. 1993. FASTUS: a finite-state processor for information extraction from real-world text. *Proceedings of IJCAI'93*, Chambéry, France, pp. 1172—1178.

Bikel D., Miller S., Schwartz R. and Weischedel R. 1997. Nymble: a high performance learning name-finder, *Proceeding of the fifth Conference on Applied Language Processing*, Washington, USA.

Cucchiarelli A. and Velardi P. 1999. Adaptability of linguistic resources to new domains: an experiment with proper noun dictionaries, *Proceedings of the Vextal Conference*, Venice, Italy, pp. 25—30.

Grishman R. and Sundheim B. (1996) Message understanding conference-6, a brief history. *Proceedings of Computational Linguistics (COLING'96)*, Copenhagen, Denmark, August 1996.

Hirschman L., Brown E., Chinchor N., Douthat A., Ferro L., Grishman R., Robinson P. and Sundheim B. 1999. Event99: a proposed event indexing task for broadcast news, available at `http://www.itl.nist.gov/div894/894.01/proc/darpa99/pdf/dir5.pdf`.

Lacroix Z., Sahuguet A. and Chandrasekar R. 1998. Information Extraction and Database Techniques: A User-Oriented Approach to Querying the Web. *Conference on Advanced Information Systems Engineering*, 1998.

Muslea I. 1999. Extraction patterns for Information Extraction tasks: a survey, *AAAI'99* (available at: `http://www.isi.edu/~muslea/RISE/ML4IE/`)

Pazienza M. T. ed. 1997. *Information extraction (a multidisciplinary approach to an emerging information technology)*, Springer Verlag (Lecture Notes in Computer Science), Heidelberg, Germany.

Poibeau T. 2001. "Deriving a multi-domain information extraction system from a rough ontology". In *Proceeding of the 17th International Conference on Artificial Intelligence*, Seattle, USA. pp. 1264–1270.

Sahuguet A. and Azavant F. (1998) W4F: a WysiWyg Web Wrapper Factory, *Technical report from the Penn Database Research Group*, University of Pennsylvania.

Salton G. 1988. *Automatic Text Processing.* Addison-Wesley, Reading, MA.

Silberztein M. 1993. *Dictionnaires électroniques et analyse automatique des textes*. Masson, Paris.

W3C. 1999. *Resource Description Framework (RDF) Model and Syntax*, W3C Recommendation, 22 February 1999 (`http://www.w3.org/TR/REC-rdf-syntax/}`