

# An Open Architecture for Multi-domain Information Extraction

Thierry Poibeau

Thales Research and Technology  
Domaine de Corbeville, F-91404 Orsay France  
Thierry.Poibeau@thalesgroup.com

Laboratoire d'Informatique de Paris-Nord  
avenue J.-B. Clément, F-93430 Villetaneuse France

## Abstract

This paper presents a multi-domain information extraction system. The overall architecture of the system is detailed. A set of machine learning tools helps the expert to explore the corpus and automatically derive knowledge from this corpus. Thus, the system allows the end-user to rapidly develop a local ontology giving an accurate image of the content of the text, so that the expert can elaborate new extraction templates. The system is finally evaluated using classical indicators.

## 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). A well-known problem of such systems is the fact that moving from one domain to another means re-developing some resources, which is a boring and time-consuming task (for example Riloff (1995) mentions a 1500 hours development).

Moreover, when information is often changing (think of the analysis of a newswire for example), one might want to elaborate new extraction templates. This task is rarely addressed by the research studies in IE system adaptation, but we noticed that it is not an obvious problem. People are not aware of what they can expect from an IE system, and most of the time they have no idea of how deriving a template from a collection of texts can be. On the other hand, if they defined a template, the task cannot be performed because they are waiting for information that is not contained in the texts.

In order to decrease the time spent on the elaboration of resources for the IE system and guide the end-user in a new domain, we suggest to use a machine learning system that helps defining new templates and associated resources. This knowledge is automatically derived from the text collection, in interaction with the end-user to rapidly develop a local ontology giving an accurate image of the content of the text. The experiment also aims at reaching a

better coverage thanks to the generalization process provided by the machine learning system.

We will firstly present the overall system architecture and principles. The learning system is then what allows the learning of semantic knowledge to help define templates for new domains. We will show to what extent it is possible to speed up the elaboration of resources without any decrease in the quality of the system. We will finish with some comments on this experiment and we will show how domain-specific knowledge acquired by the learning system such as the subcategorization frame of verbs could be used to extract more precise information from texts.

## Application Description

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 below.

## Information Extraction System

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 a 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

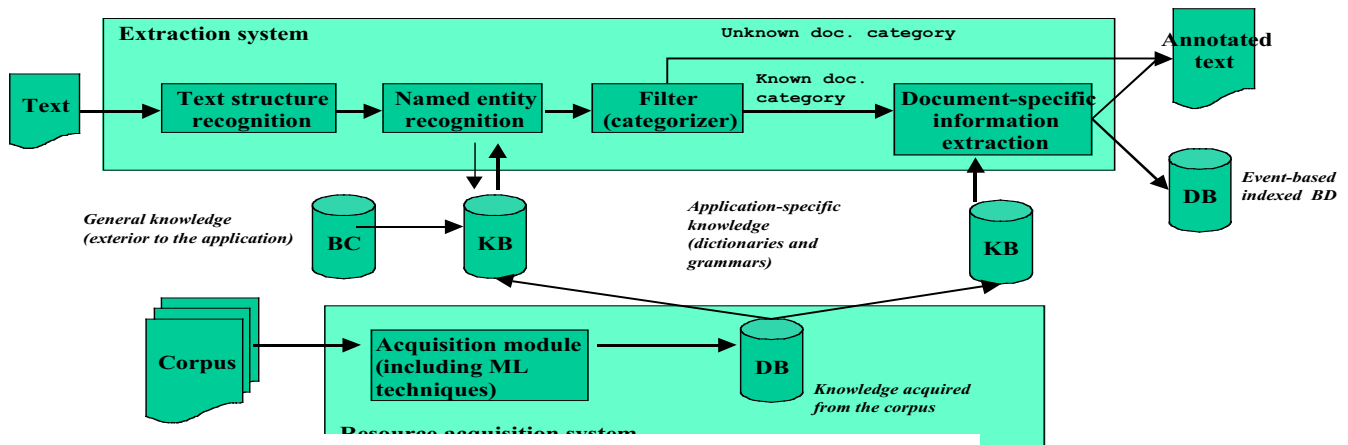


Figure 1: The information extraction system architecture

toolbox Intex to design dictionaries and automata (Silberstein 1993).

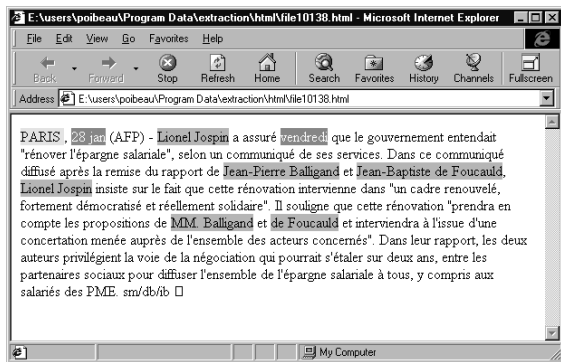


Figure 2: The named entity recognizer

- The third stage performs text categorization from the seek of “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.

The architecture exhibits, outside from the information extraction system in itself, a machine learning module that can help the end-user produce resources for information extraction. The end-user who wants to define a new extraction template has to process a representative set of documents in the learning module to obtain an ontology and some rough resources for the domain he wants to cover. The acquisition module is presented in the next section.

## Template Creation Module

The system can be divided into three main parts:

1. A machine learning engine used to produce semantic clusters from the corpus. These clusters are weighted and are intended to give to the expert a rough idea of the topic addressed in the text;
2. A system to help the creation of extraction template once the relevant topic of the corpus have been identified;
3. The information extraction system in itself, that will use the resources defined at the previous stage to fill the templates.

Figure 3 gives an overview of the overall architecture. The corpus is processed by the machine learning system (1), in order to produce semantic clusters organized in a superficial ontology. The template creation module (2) helps the expert define his own extraction template from the ontology. The lower part of the schema describes the information extraction system in itself (3), processing a text to fill the extraction template.

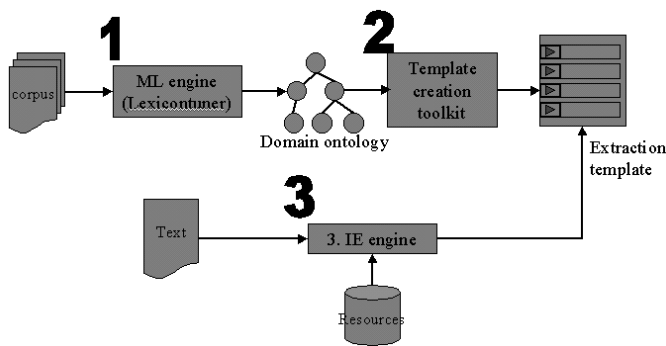


Figure 3: Architecture of the system

Apart from the information extraction system, the machine learning module is intended to help the end-user produce the extraction template. A representative set of documents has to be processed by the learning module to obtain an ontology and some rough resources for the domain he wants to cover. The resources have to be manually completed to obtain a good coverage of the domain.

### Uses of AI Technology: Extraction of Semantic Clusters from Lexical Resources

A mainly automatic method has been elaborated to combine knowledge contained in on-line resources and statistical data obtained from the training corpus of the chosen domain. This method is implemented through the LexiconTuner, which was initially developed for French but can be used for any language if correct resources are provided.

Semantic clusters – that is to say clusters of semantically related words – can be acquired from on-line resources including dictionaries, thesauri and taxonomies (Véronis

and Ide 1990) (Wilks *et al*, 1995), (Aguirre and Rigau 1996). Dictionaries model the meanings of lexical items using textual definitions. Textual definitions are written in natural language and the full information encoded in these definitions can't be extracted easily. However, partial information, which can be used as semantic clusters, can be extracted from the definitions relatively easily. The method is inspired from (Luk 1995) and described in details in (Ecran 1996), (Poibeau 1999) and (Poibeau 2001).

The idea of building networks from definitions was first proposed by Véronis and Ide (1990), along with a propagation technique for computing clusters of related words. Many authors have proposed techniques for deriving some kind of clusters or associations of concepts from graphs of concepts, among others transitive closures and computation of graph “cliques”, simulated annealing, etc. In our approach, the content words in the definitions of a word sense are used as component concepts of the word sense (Luk 1995), that is to say that a concept is a lexical item considered as a member of a semantic cluster.

In the LexiconTuner, the generation of the clusters is carried out in two steps. Firstly, the list of all the concepts occurring in the corpus is generated. For each word in the corpus, the program looks for its textual definition(s) in the lexicon. The words in the definitions are treated as concepts and are recorded. This step allows us to go from a French word (e.g. *opéra*) to some concepts labeled with English words (*dramatic, performance, composition, music*), by means of a bilingual dictionary (*opera* being defined as a “dramatic performance or composition of which music is an essential part”).

Secondly, semantic clusters are generated by clustering related concepts in the list. The program uses a semantic net as its semantic knowledge source to determine the semantic relatedness between the concepts. The net is

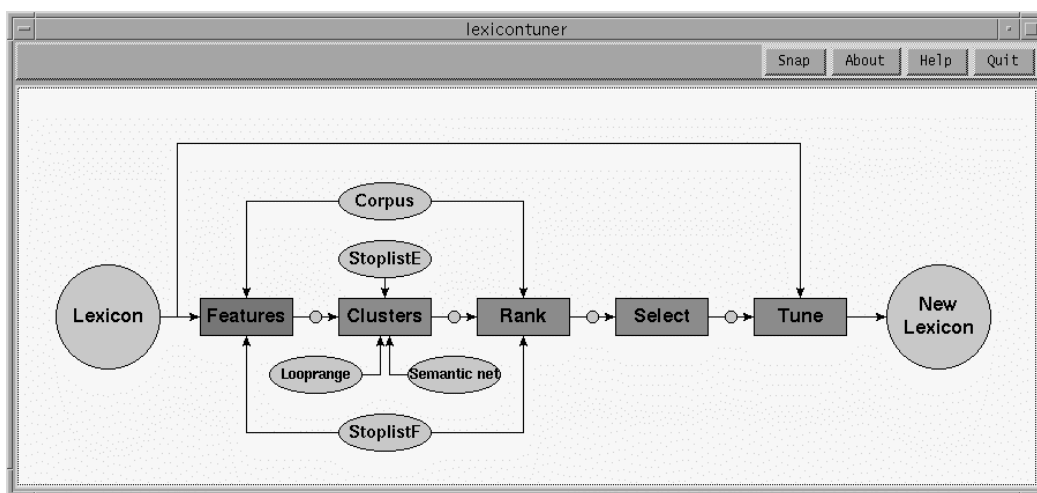


Figure 4: The LexiconTuner architecture

derived from the *Oxford Advanced Learner's Dictionary* (OALD). Each node represents a word, and for any two words  $w1$  and  $w2$ ,  $w1$  is linked to  $w2$  if  $w2$  appears in the definitions of  $w1$  in OALD (*music* will be linked to *composition* if *music* appears in the definition of *composition*). The link from  $w1$  to  $w2$  is weighted inversely proportionally to the product of the number of senses of  $w1$  as defined in OALD and the number of words in the definition of  $w1$  that contains  $w2$ . The clusters are formed in 3 steps:

- Circles that share one or more nodes are merged. Thus, (*opera, performance, dramatic*) and (*dramatic, composition, music*) are merged into a single set (*opera, performance, dramatic, composition, music*). This is called a "core cluster".
- Lastly, peripheral words (words which are part of a circle which length is inferior to the user-defined number) are related to the core cluster. If two words like *dramatic* and *comic* are related, then *comic* will be added to the cluster, being related to *dramatic*.
- Every circle in the semantic net which length is equal to a user-defined number is identified. For example, if the number is 3, the system will generate all the sets consisting of three related words according to the dictionary definition. For *opera*, the following associations will be considered: (*opera, performance, dramatic*) and (*dramatic, composition, music*).

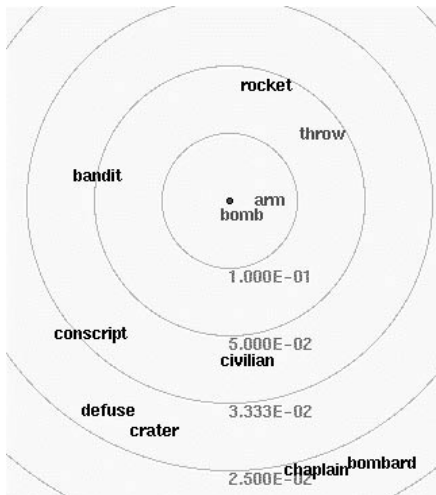


Figure 5: A cluster generated by the LexiconTuner

Lastly, the membership of a cluster is "weighed". The weight of a core member is considered as the inverse of the number of senses of the word as defined in OALD. The weight of a non-core member is considered as the mean of the weights of the core members of the cluster to which it is related. Then, semantic clusters can capture notions

appearing in texts independently from the lexical items expressing these notions.

## Template Design

Semantic classes produced by the LexiconTuner are proposed to the end-user, who chooses which clusters are of interest to him. Once he has chosen one cluster, the system automatically proposes him an interface to refine the cluster, aggregate a part of the closest clusters and develop a hierarchy. This hierarchy can be assimilated to a local ontology, describing a part of the world knowledge related to the event of interest for the end-user.

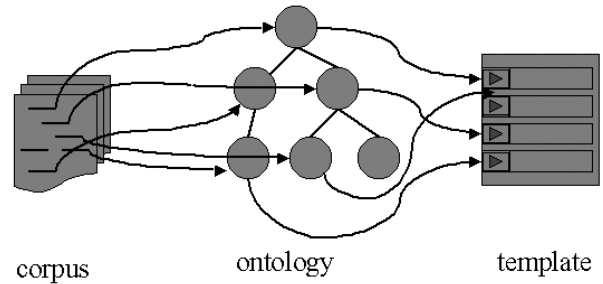


Figure 6: From corpus to template, through the ontology...

The semantic clusters produced by the LexiconTuner give to the expert a rough idea of the topics addressed in the corpus. The expert has to navigate among these clusters to select relevant topics and establish a candidate template. The chosen topics has to be named and to be associated with a slot in the template. The system automatically generates the corresponding information into the database: the creation of a new template leads to the creation of a new table and each new slot in the template corresponds to a new column in the table. This technical part of the template creation process can be compared to the Tabula Rasa toolkit developed at New Mexico State University to help end-users define their own templates (Ogden and Bernick 1997).

e:\users\poibeau\program data\extraction\txt\file10138.txt	
Date	28-ja-00
Location	PARIS
Personality	Lionel Jospin
Type	Déclaration
Source	AFP (wire)
Topic	- Lionel Jospin a assuré vendredi que le gouvernement entendait "rénover l'épargne salariale", selon un communiqué de ses services.

Figure 7: A specific template

The evaluation of the template creation task is not obvious since it necessitates domain knowledge and text ma-

nipulation from the experts. No clear reference can be established. The only way to evaluate the contribution of the semantic clusters is to ask the expert firstly to manually elaborate templates from a corpus and secondly to do it with the help of the Lexicontuner.

The expert we worked with made the following comments:

- The semantic clusters produced by the LexiconTuner give an appropriate idea of the overall topic addressed by the texts.
- These clusters help the elaboration of templates and allows to focus on some part of information without reading large part of texts.
- However, the elaboration of the template itself remains largely dependant of the domain knowledge of the expert because (a) he knows what kind of information he wants to find in relation with a given topic and (b) the clusters are too coarse-grain to directly correspond to slots.

When the template corresponds to an event, the information to be found generally refers to classical *wh-questions*: who, what, where and when. Some additional slots can be added but most of the time they correspond to classical associations of ideas. For example, if one wants to extract information about football matches, he will immediately create a slot corresponding to the score of the match. However, this comment is due to the fact that the system is analyzing news stories, for which one can associate stereotypic reduced templates (sometimes called *templates* in the IE community).

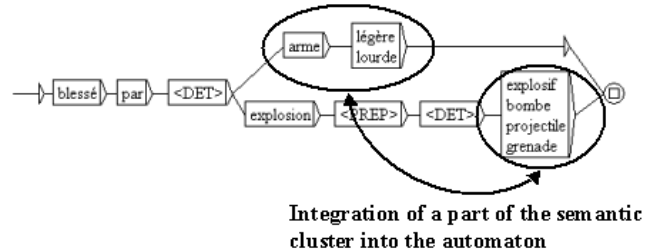
We observed that the template creation task is frequently a problem in more technical domains for which no clear *a priori* schema exists. In this experiment, the LexiconTuner can be seen as a tool to explore the corpus and give a rough idea of tentative templates rather than a tool designed to help the creation of the content of the templates themselves. However, the LexiconTuner results is also useful for the creation of the resources of the system (next section).

## Experiment Description and Evaluation

We asked an expert to define a new template and the associated resources, using the tools we presented above. We chose the terrorist event domain from the AFP newswire, because it is a well-established task since the MUC-4 and MUC-5 conferences. Moreover, similar experiments has been previously done that give a good point of reference (see Poibeau (1999) and Faure and Poibeau (2000)).

Homogeneous semantic clusters learned by the LexiconTuner are refined: a manual work of the expert is necessary to exploit semantic classes (merging of scattered classes, deletion of irrelevant elements, addition of new elements, etc.). About five hours have been dedicated, after the acquisition process, to the refinement of data furnished by LexiconTuner. Merging and structuring

classes incrementally develop a local ontology, which nodes are related to slots in the extraction template. This knowledge is also considered as a resource for the finite-state system and is exploited either as dictionaries or as transducers, according to the nature of the information. If it is a general information that is not domain specific, the development guidelines advise the user to use dictionaries that can be reused, otherwise, he designs a transducer.



**Figure 8:** a part of the “weapon” automaton

The elaboration of linguistic knowledge, from clusters produced by the LexiconTuner, to the design of the Intex resources, spent about 15 hours. This duration has to be compared with the two weeks (about 80 hours) needed for the manual resources development for a previous experiment on the same subject.

A hundred texts have been used as "training corpus" and a hundred different texts have been used as "test corpus". Texts are first parsed with our system, and then some heuristics allow to fill the extraction template: the first occurrence of a number of victims or injured persons is stored. If a text deals with more than one terrorist event, we assume that only the first one is relevant. Thanks to the nature of the channel, very few texts deal with more than one event.

Our results have been evaluated by two human experts who did not follow our experiment. Let  $Pos$  be the total number of good answers and  $Act$  the number of solutions proposed by the system. Our performance indicators are defined as:

- (Ok) if extracted information is correct;
- (False) if extracted information is incorrect or not filled;
- (None) if there were no extracted information and no information has to be extracted.

Using these indicators, we compute two different values for each slot:

- Precision ( $P_{rec}$ ) is defined as  $Ok/Pos$ .
- Recall ( $R_{ec}$ ) is defined as  $Ok/(Act)$ .

Slot name	Precision	Recall	P&R
Event date	.95	.96	.95
Event location	.88	.92	.90
Nb of killed people	.83	.73	.77
Nb of injured people	.80	.69	.74
Weapon	.85	.82	.83
<b>Total</b>	.86	.82	.83

The performances are good according to the state-of-the-art and to the time spent on resource development. However, we can analyze the remaining errors as follows:

The date of the story is nearly fully correct because the wrapper uses the formatted structure of the article to extract it. The errors for the location slot are due to two “contradictory” locations found by the system. A more complete linguistic analysis or a database providing lists of cities in different countries would reduce this kind of errors. The errors in the number of dead or injured persons slot are frequently due to silence: for example the system fails against too complex syntactic forms. The silence for the weapon slot is frequently due to incompleteness of semantic dictionaries.

## Conclusion and Future Work

The experiment that has been described is based on an external knowledge base derived from a dictionary. It is thus different from (Faure and Poibeau 2000) which tries to acquire knowledge directly from the text. The use of an external database allows to work on middle-size corpora that are not as redundant as technical texts. We also think that using a general dictionary is interesting when dealing with general texts like a newswire. Clusters contain words that were not contained in the training part of the corpus, allowing a better coverage of the final result.

The 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 (about one new template per week). We hope to reach the number 50 templates towards summer 2001.

## Acknowledgement

A part of this study re-used some pieces of software developed in the framework of the ECRAN project (1996-1999). I would like to thank Alpha Luk and other people having participated to the development of the Lexicon Tuner. I am also indebted to David Faure, Tristelle Kervel, Adeline Nazarenko and Claire Nedellec for useful comments and discussions on this subject.

## References

- Aguirre E. and Rigau G. 1996. Word Sense Disambiguation using conceptual density. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING)*, Copenhagen.
- Ecran. 1996. D-2.5.1 - Methods for Lexical Items Modification/Creation and D-2.6.1 - Heuristics for Automatic Tuning. ECRAN Project Deliverable.
- Faure D. and Poibeau T. 2000. First experiments of using semantic knowledge learned by Asium for Information Extraction task using Intex. In *Proceedings of the workshop on learning ontologies*, during *ECAI'2000*, Berlin.
- Luk A. K. 1995. Statistical Sense Disambiguation with Relatively Small Corpora using Dictionary Definitions. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*.
- Ogden W. and Bernick P. 1997. “Tabula Rasa Meta-Tool: Text Extraction Toolbuilder Toolkit”. Technical Report MCCS-97-305. Las Cruces: Computing Research Laboratory.
- 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. 1999. “A statistical clustering method to provide a semantic indexing of texts”. In *Workshop on machine learning for information filtering*, during *IJCAI'1999*, Stockholm.
- 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.
- Riloff E. 1995. “Little Words Can Make a Big Difference for Text Classification”. In *Proceedings of the 18th Annual International Conference on research and Development in Information Retrieval (SIGIR\_95)*.
- Salton G. 1988. *Automatic Text Processing*. Addison-Wesley, Reading, MA.
- Silberztein M. 1993. *Dictionnaires électroniques et analyse automatique des textes*. Masson, Paris.
- Véronis J. and Ide N. M. 1990. “Word Sense Disambiguation with Very Large Neural Networks Extracted from Machine Readable Dictionaries”. In *Proceedings of the 13th International Conference on Computational Linguistics (COLING'90)*, Helsinki, Finland.
- Wilks Y., Slator B. and Guthrie L. 1995. *Electric words: dictionaries, computers and meanings*, MIT Press, Cambridge, MA.