Designing and Evaluating Patterns for Ontology Enrichment from Texts

Nathalie Aussenac-Gilles[1] and Marie-Paule Jacques[2]

[1] Institut de Recherche en Informatique de Toulouse (IRIT) - CNRS,
UPS, 118, route de Narbonne, 31062 Toulouse Cedex, France
aussenac@irit.fr
http://www.irit.fr/~Nathalie.Aussenac

[2] Équipe de Recherche en Syntaxe et Sémantique (ERSS) - CNRS,
Maison de la Recherche, UTM, 5, allées Antonio Machado, 31048 Toulouse Cedex, France
marie-paule.jacques@univ-tlse2.fr
http://www.univ-tlse2.fr:8880/erss/index.jsp?perso=mpjacques

Abstract. Pattern-based approaches for knowledge identification in texts assume that linguistic regularities always characterise the same kind of knowledge, such as semantic relations. We report the experimental evaluation of a large set of patterns using an ontology enrichment tool: CAMÉLÉON. Results underline the strong corpus influence on the patterns efficiency and on their meaning. This influence confirms two of the hypotheses that motivated to define CAMÉLÉON as a support used in a supervised process: (1) patterns and relations must be adapted to each project; (2) human interpretation is required to decide how to report in the ontology the pieces of knowledge identified with patterns.

1 Introduction

Relation extraction from texts can be an efficient means to rapidly structure a conceptual model and identify significant domain concepts. Possible approaches to identify relations from corpora include: using existing relations in lexical resources like WordNet [16] [5]; matching lexico-syntactic patterns [9] [10] [16]; learning dependencies between phrases through term distribution analysis [3]. Pattern-based approaches for knowledge identification in texts assume that linguistic regularities always characterise the same kind of knowledge, such as semantic relations. We defined CAMÉLÉON, a method and a supervised tool that supports a knowledge engineer to identify relations and concepts for ontology engineering [15]. CAMÉLÉON provides a set of generic patterns and relations to be adapted and applied on tagged corpora [2].

This paper reports how we built and evaluated a set of 70 generic patterns for the French language. After a presentation of the CAMÉLÉON process (§ 2), we describe how the tool supports pattern definition and evaluation (§ 3). Then we detail the corpora and method used for this experiment (§4), we report its results and discuss them (§5). We conclude by underlying the role of human interpretation to adapt patterns, to evaluate their instances, and later to enrich a conceptual model. This experiment also proves that, rather than generic, patterns should be adaptable and reusable.

S. Staab and V. Svatek (Eds.): EKAW 2006, LNAI 4248, pp. 158 – 165, 2006.

2 Semantic Relation Identification with CAMÉLÉON

CAMÉLÉON is a method and tool to extend an existing network of concepts with new terms, concepts and semantic relations by applying a pattern-based approach [15]. A conceptual model built up with CAMÉLÉON is a semantic network where concepts are associated with a set of terms (synonym terms). This model may be the starting point to design an ontology or it may be considered as a result by itself. Knowledge engineers and linguists are the intended users of the CAMÉLÉON[1] tool. This tool can be one of the components of a natural language processing and modelling chain from texts to ontologies, such as the one proposed in KAON [16], TERMINAE [1] or [7].

2.1 Pattern-Based Knowledge Identification

Patterns are lexical, semantic and/or syntactic characterizations of linguistic contexts in which one expects to find some specific piece of information. Hearst was the first one to experiment a pattern-based approach for the identification of lexical relation and semantic classes [9]. Hearst tested some general patterns mainly expressing definitions or hyperonymy. She noticed that linguistic regularities had to be tuned for each corpus and domain. Over the last ten years, patterns were widely used with success for information extraction or relation extraction like in [13]. To gain efficiency, research has investigated two mains tracks. Firstly, to reduce the cost of pattern definition and tuning, patterns may be learned from manually tagged corpora [5] [6][16]; they may refer to named entities and known semantic classes [8]. Secondly, to reduce the time required to select valid pattern instances and the noise of the overall process, various statistical text analyses have been experimented. Like [8], we consider that an alternative contribution would be to capitalize robust patterns and know-how about their use, together with information about their semantics, their precision and recall in various types of domains and documents.

2.2 Overview of the Approach

For a given project and corpus, CAMÉLÉON suggests a two-steps supervised process.

1) **Defining project-specific patterns:** The user is expected to define a specific set of domain relations and valid patterns for his project and corpus. They may be obtained first by adapting some generic patterns already available in CAMÉLÉON, second by manually defining new patterns for known domain relations, third by defining new relations and patterns after observing the contexts in which pairs of related terms occur (according to Riloff's suggestion [14]). Fixing patterns also includes evaluating the sentences obtained after pattern-matching. The pattern will be modified in order to reduce its noise and increase its precision.

2) **Extending the conceptual model:** The knowledge engineer checks one by one the sentences identified by matching patterns in the corpus. Validated sentences may suggest new concepts and relations. To save time, a default validation is possible. Then, suggestions of relations are presented in CAMÉLÉON ontology browser, when

[1] http://developer.berlios.de/projects/cameleonirit/

editing one of the concepts. The knowledge engineer must decide whether to define a new relation or not, and whether the concepts to be connected and the semantics of the relation are those suggested or other ones.

2.3 Building the Base of Generic Patterns

One of the strengths of CAMÉLÉON is to provide a set of robust and valid generic patterns as a bootstrap. This paper reports how this set was defined and evaluated. By doing so, we used and tested the available functions in the CAMÉLÉON software. It contributes to CAMÉLÉON global evaluation, which would be far more complex. A full evaluation should include the design of a real ontology for a well-determined application. Nevertheless, our experiment contributed to validate two foundational hypotheses (the need for pattern adaptation and human interpretation).

3 How the Tool Supports Relation Extraction

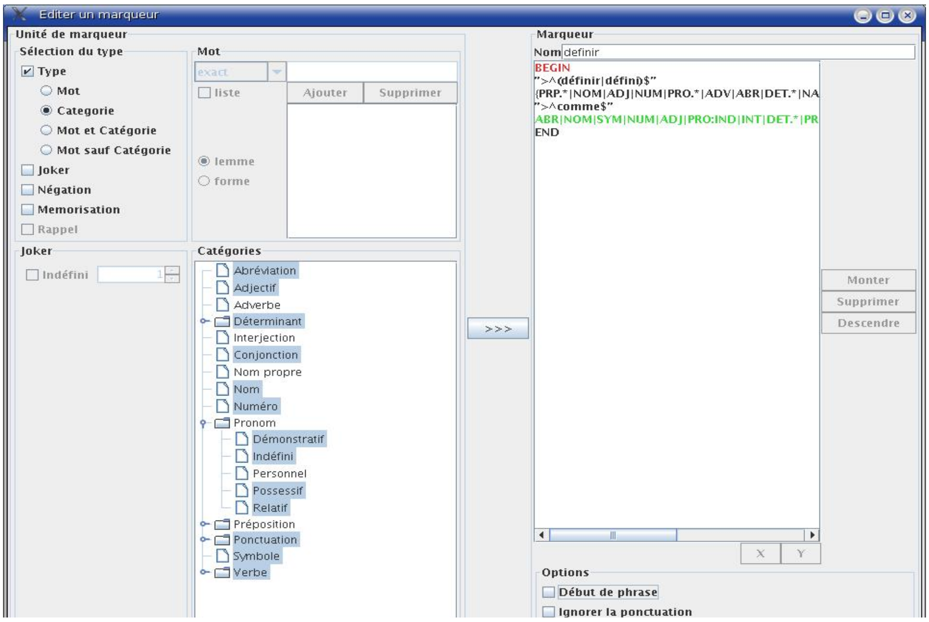
The CAMÉLÉON tool contains a project management interface and two main modules: one supports pattern definition, matching and evaluation; the other one helps to interpret the sentences that contain the patterns and to enrich the conceptual model. The first module includes a concordancer, KESKYA, which matches patterns on texts tagged with a Part Of Speech (POS) tagger. We used Tree-Tagger, but any tagger would do. The second module includes an ontology editor.

A CAMÉLÉON project entails a set of tagged texts - the corpus -, a set of specific patterns and relation types, and a conceptual model. To promote reusability and to avoid starting from scratch, the tool database stores several corpora and a set of generic patterns and relations. A project corpus may include reused corpora and/or tagged new texts. Project patterns are adapted from generic patterns or user-defined.

3.1 Pattern Design, Adaptation and Evaluation

The internal representation of patterns is the one required by the KESKYA concordancer. Patterns are supposed to be included in a single sentence. They are expressed mainly with lemmas and user defined semantic classes combined with POS tags, and a set of operators like *or* (| symbol), negation or iterations (*joker*). The interface makes it easier to define (or modify) each pattern, chunk after chunk. The user selects one of the listed options and adds it to the pattern (Fig. 1). Patterns characterize linguistic contexts where semantic relations between concepts may appear in texts. So the knowledge engineer must specify which parts of the pattern will refer to the related concepts (*X* and *Y*). Each of these chunks is turned into a particular colour that will be used later on to colour the words that may correspond to the related concepts.

Evaluating a pattern means checking some of the sentences where the pattern appears in each of the corpora (Fig. 2). The goal is to decide whether the pattern is to be rejected, modified or retained as a relevant pattern for this project.



☑ Type
○ Mot
● Categorie
○ Mot et Catégorie
○ Mot sauf Catégorie
□ Joker
□ Négation
□ Memorisation
□ Rappel

exact
□ liste
● lemme
○ forme

□ Indéfini 1

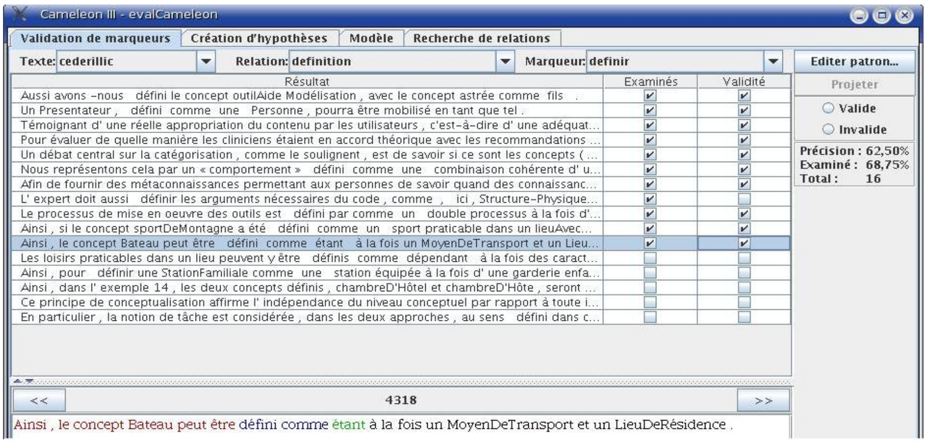
Abréviation
Adjectif
Adverbe
Déterminant
Interjection
Conjonction
Nom propre
Nom
Numéro
Pronom
Démonstratif
Indéfini
Personnel
Possessif
Relatif
Préposition
Ponctuation
Symbole
Verbe

Nom definir

```
BEGIN
">^(définir|défini)$"
(PRP.*|NOM|ADJ|NUM|PRO.*|ADV|ABR|DET.*|NA
">^comme$"
ABR|NOM|SYM|NUM|ADJ|PRO:IND|INT|DET.*|PR
END
```

□ Début de phrase
□ Ignorer la ponctuation

Fig. 1. CAMÉLÉON Pattern Editor. The edited pattern (*définir*) searches for forms like "X is defined as Y". The user preferred not to specify where X exactly could appear in a sentence (*BEGIN* is in the *X* colour), but the list above END constrains how Y could be formulated.



Texte: cederillic Relation: definition Marqueur: definir

Résultat	Examinés	Validité
Aussi avons -nous défini le concept outilAide Modélisation , avec le concept astrée comme fils .	☑	☑
Un Presentateur , défini comme une Personne , pourra être mobilisé en tant que tel .	☑	☑
Témoignant d' une réelle appropriation du contenu par les utilisateurs , c'est-à-dire d' une adéquat...	☑	☑
Pour évaluer de quelle manière les cliniciens étaient en accord théorique avec les recommandations ...	☑	☑
Un débat central sur la catégorisation , comme le soulignent , est de savoir si ce sont les concepts (...	☑	☑
Nous représentons cela par un « comportement » défini comme une combinaison cohérente d' u...	☑	☑
Afin de fournir des métaconnaissances permettant aux personnes de savoir quand des connaissanc...	☑	☑
L' expert doit aussi définir les arguments nécessaires du code , comme , ici , Structure-Physique...	☑	□
Le processus de mise en oeuvre des outils est défini par comme un double processus à la fois d'...	☑	☑
Ainsi , si le concept sportDeMontagne a été défini comme un sport praticable dans un lieuAvec...	☑	☑
Ainsi , le concept Bateau peut être défini comme étant à la fois un MoyenDeTransport et un Lieu...	☑	☑
Les loisirs praticables dans un lieu peuvent y être définis comme dépendant à la fois des caract...	□	□
Ainsi , pour définir une StationFamiliale comme une station équipée à la fois d' une garderie enfa...	□	□
Ainsi , dans l' exemple 14 , les deux concepts définis , chambreD'Hôtel et chambreD'Hôte , seront ...	□	□
Ce principe de conceptualisation affirme l' indépendance du niveau conceptuel par rapport à toute i...	□	□
En particulier , la notion de tâche est considérée , dans les deux approches , au sens défini dans c...	□	□

○ Valide
○ Invalide

Précision : 62,50%
Examiné : 68,75%
Total : 16

4318

Ainsi , le concept Bateau peut être défini comme étant à la fois un MoyenDeTransport et un LieuDeRésidence .

Fig. 2. CAMÉLÉON Pattern Evaluation Screen. Given a text (*texte*), a relation type (*relation*) and a pattern, the pattern is matched in the text (*projeter*). Results are sentences listed for checking. When selecting a sentence, its full content is displayed in the editor on the bottom. Coloured words correspond to possible related concepts (X and Y). The pattern may be modified (*Editer patron*), rejected or validated (*invalide* or *valide* radio-button). The precision score (on the right) may guide this decision.

3.2 Text Fragment Selection and Model Enrichment

Given the set of project-specific patterns, the user must check each of their occurrences in the corpus. If a relation between concepts can be identified, the user stores the sentence as a relation hypothesis and selects the words that may correspond to related concepts (X and Y), guided by the coloured words.

The next step consists in browsing the conceptual model and the list of terms identified as possible concept labels. When editing a concept, all the available relation hypotheses are shown. The user may decide to define new concepts or relations. This process is quite complex and time-consuming. It requires some know-how in knowledge modelling and a good appreciation of the intended role of the ontology.

4 Corpora and Method

Since CAMÉLÉON is intended to retrieve semantic relations within specific domains, our 8 corpora are all made up of specialized texts. They are grouped into 3 categories:

1. technical writings in the fields of electric networks (GDP), electricity (MOUG) and telecommunication (CRAT);
2. scientific papers in knowledge engineering [4] (IC), archeology (ARCH) and geomorphology (ENC);
3. handbooks of geomorphology (GEO) and of paragliding (PAR).

The patterns which fill in the generic base were not designed from scratch, they were adapted from three sources: 1. a previous experiment on semi-automatic retrieval of definitions [12], also applying to tagged texts and carried out by L. Tanguy and J. Rebeyrolle, who kindly provided us with the patterns they designed; 2. various studies within the framework of knowledge engineering and ontology [11] and 3. the previous version of CAMÉLÉON [15]. The last two both provided patterns devoted to semantic relations such as hyperonymy and part-hood.

In order to build the pattern base, we had to enter the various patterns so as to benefit from tagging. The patterns in the previous version of Cameleon did not include tags, only lexical forms. For instance, a pattern devoted to the relation of inclusion lists the different forms of the verbs bearing such a relation:

inclut|incluent|incluant|intègre|intègrent|integrant (the symbol | means *or*)

Since we could use lemmas and part-of-speech tags (with help of the TreeTagger[2]) to design patterns, a pattern such as the above one has been replaced by a combination of lemmas (*inclure|intégrer*) and tags (present tense or past participle or present participle) which are easy to choose in a list, as shown in Fig. 1.

Each pattern has been sought, if necessary after having been refined, and the contexts have been evaluated. Note that this evaluation has been carried out by only one of the authors. After this, we obtained for each corpus a measure of the precision of each pattern, which was supposed to help us decide which patterns have to be retained to fill in the generic-pattern base.

[2] www.ims.unistuttgart.de/projekte/corplex/TreeTagger/

5 Results and Discussion

5.1 Results

We entered 71 patterns: 19 for definitions, 35 for hyperonymy, 14 for meronymy, 1 for reformulation, 2 'varia'. Due to lack of room, Table 1 below gives only a sample of the precision rates we obtained for the 8 corpora.

Table 1. Sample of the evaluation results (N= Number of contexts; P= Precision percentages)

	GDP		IC		GEO		MOU	
	N	P	N	P	N	P	N	P
définir	3	*100*	43	*98*	0		2	*100*
être-un	258	*17*	489	*18*	641	*23*	120	*8*
et Adv	10	*10*	15	*7*	56	*30*	6	*17*
sorte de	0		7	*57*	3	*67*	0	
inclure	75	*51*	32	*41*	16	*50*	18	*61*
partie de	0		0		7	*0*	0	
situé dans	40	*53*	63	*38*	38	*24*	4	*50*
c-à-dire	6	*67*	37	*54*	40	*80*	3	*100*
	ENC		PAR		ARCH		CRAT	
	N	P	N	P	N	P	N	P
définir	2	*100*	1	*0*	-		-	
être-un	375	*15*	62	*40*	181	*29*	-	
et Adv	66	*5*	2	*0*	13	*38*	19	*58*
sorte de	1	*100*	0		0		4	*100*
inclure	29	*62*	2	*100*	27	*19*	267	*48*
partie de	1	*100*	1	*0*	1	*0*	11	*18*
situé dans	55	*24*	4	*75*	36	*56*	291	*59*
c-à-dire	14	*29*	2	*100*	8	*63*	11	*64*

To give an example of pattern, 'définir' is 'lemma of verb *définir* (to define) followed by a joker followed by lemma of *comme* (as)': <définir> 1 <comme>. It yields a context[3] such as:

Un Projet Logiciel peut **se définir comme** un Processus de Développement.
A software project may be defined as a development process.

The major comment on Table 1 is that patterns differ considerably from each other regarding numbers of contexts and precision. Furthermore, the results of a one pattern may vary to a great extent as far as the corpus is concerned. To give but one example, the *inclure (include)* pattern ranges from 2 to 267 contexts yielded and from 19% to 100% in terms of precision.

Our experiment gives rise to two major issues: issues related to the elaboration of patterns itself and issues related to the results.

[3] Original sentences are in French, and we give a translation below. Bolded parts of the sentence are those that match the pattern.

5.2 Pattern Elaboration

In our experiment, the point of departure was a set of already-existing patterns which had to be adapted by replacing lexical forms with lemmas combined with tags. We could then see that a tagset offers a convenient method for designing patterns in that it facilitates the expression of abstract features while avoiding tedious entries of lists of forms. However, the accuracy of the tagset must represent a trade-off between the need for precision and manageability: the more accurate the tagset, the more difficult it is to understand the tags, especially when the user is not attuned to dealing with morpho-syntactic categories, and the more difficult the handling of the tagset.

Another point is the adaptation of the patterns to the different corpora. A given pattern is seldom convenient for each corpus; it is therefore necessary to modify it, generally to reduce irrelevant contexts. For example, the pattern NP1 <être> 1 DEF_ART NP2 DEF_ART (plus|moins) captures the following context:

La méthode KOD en **est** l'exemple **le plus** frappant
The KOD method is the most striking example of this

which does not express hyperonymy. To avoid it, we needed to specify that NP2 must not have *exemple, cas* or *résultat* as its head, which is an *ad-hoc* constraint.

Generally, it must be kept in mind that the so-called 'generic' patterns capture the most frequent or the most widespread constructions for a given relation. To a certain extent, it would be unrealistic to hope to take such a pattern and use it without modification. In this sense, one may wonder whether some patterns are really generic.

5.3 What Is a "Generic Pattern"?

The results presented in section 5.1, together with the above observations, challenge the notion of "generic pattern". If a generic pattern is the lexico-syntactic formulation of a semantic relation, which is said to invariably retrieve the same number of relevant contexts, whatever the corpus is, then we can conclude from our experiment that a generic pattern does not exist. Even the *is-a* pattern shows a huge difference between corpora, although it is acknowledged to be as generic as possible, in the sense that it "occurs frequently and in many text genres" [9: 540]. If one tests this pattern only on the PAR corpus, one will conclude this pattern is worth retaining since it has a 40% precision rate; while if the same pattern is tested only on the MOUG corpus, it is likely to be rejected, for the precision rate is 8%. If one wants to enhance the results for each corpus, one will have to introduce new constraints and to "fine-tune" the patterns, which is the contrary of what would be expected for a "generic" pattern.

6 Conclusion

We have presented a tool and an approach for supervised relation and concept identification. Our experiment shows that the performance of the semantic patterns used to retrieve conceptual relations within texts is highly corpus-dependent and that human supervision is therefore needed at various stages: pattern definition, sentence evaluation and model enrichment. Hence, the generic pattern base that comes with the CAMÉLÉON tool is thought of as a 'bootstrap' for elaborating and adapting convenient patterns and is not intended to be used "as is". Therefore, future work must be devoted to facilitating pattern elaboration and to browsing the resulting contexts. Firstly, we must ensure that "human-made" patterns actually surpass machine-learning approaches, which we would expect because of the complexity of their lexico-syntactic structures. Secondly, we must reduce the number of contexts the user has to check by filtering them via statistical methods. Thirdly, we must test how easy users find the overall pattern creation or adaptation task with CAMÉLÉON and improve on it.

References

1. Aussenac-Gilles N., Biébow B., Szulman S.: Revisiting Ontology Design: a method based on corpus analysis. Knowledge engineering and knowledge management: methods, models and tools. R Dieng and O. Corby (Eds). LNAI 1937. Berlin: Springer Verlag. (2000) 172-188
2. Aussenac-Gilles, N.: Supervised Text Analysis for Ontology and Terminology Engineering. Proc. of the Dagstuhl Seminar 05071 on "Machine Learning for the Semantic Web" (2005)
3. Bourigault, D.: Upery : un outil d'analyse distributionnelle étendue pour la construction d'ontologies à partir de corpus, Actes de Traitement Automatique des Langues Naturelles, Nancy (France) (2002) 75-84
4. Charlet, J., Zacklad, M., Kassel, G., Bourigault, D. (Eds): Ingénierie des connaissances Evolutions récentes et nouveaux défis. Paris : Eyrolles (2000).
5. Cimiano, P., Pivk, A., Schmidt-Thieme, L., Staab, S., Learning Taxonomic Relations from Heterogeneous Evidence. ECAI-2004 WS on Ontology Learning and Population, Valencia (Spain) (2004) 59-73
6. Faure, D., Poibeau, T.: First experiments of using semantic knowledge learned by ASIUM for information extraction task using INTEX. ECAI-2000 WS on Ontology Learning, Berlin (Germany) (2000) 7-12
7. Gillam, L., Tariq, M., Ahmad, K.: Terminology and the construction of ontology. Terminology, Volume 11, Number 1, (2005) 55-81
8. Girju, R., Moldovan, D.: Text Mining for Causal Relations. AAAI Conference (2002)
9. Hearst, M. Automatic Acquisition of Hyponyms from Large Text Corpora. In Proc. of the 15th Inter. Conf. on Computational Linguistics (COLING-92), Nantes (F) (1992) 539-545
10. Kavanagh, J.: The Text Analyzer: a Tool for Extracting Knowledge from Text. Master's of computer science Thesis, Univ. of Ottawa Canada (1996)
11. Marshman, E., Morgan T., Meyer I.: French patterns for expressing concept relations. Terminology, 8 (1) (2002) 1-30
12. Rebeyrolle, J., Tanguy, L.: Repérage automatique de structures linguistiques en corpus : le cas des énoncés définitoires. Cahiers de grammaire, 25 (2000) 153-174
13. Reinberger, M.-L., Spyns, P.: Discovering Knowledge in Texts for the learning of DOGMA-inspired ontologies. ECAI-2004 WS on Ontology Learning and Population, Valencia (Spain) (2004) 19-24
14. Riloff, E.: Automatically Generating Extraction Patterns from Untagged Text. Proc. of the 13th National Conference on Artificial Intelligence (AAAI-96). Portland (1996) 1044-1049
15. Séguéla, P.: Construction de modèles de connaissances par analyse linguistique de relations lexicales dans les documents techniques. Mémoire de thèse en Informatique, Université Toulouse 3, France (2001)
16. Staab, S., Maedche, A.: Ontology Learning for the Semantic Web, IEEE Intelligent Systems, Special Issue on the Semantic Web, 16(2) (2001) 72-79