Improving Annotation in the Semantic Web and Case Authoring in Textual CBR*

Juan A. Recio-García, Marco A. Gómez-Martín,
Belén Díaz-Agudo, and Pedro A. González-Calero

Dep. Sistemas Informáticos y Programación
Universidad Complutense de Madrid, Spain
{jareciog, marcoa}@fdi.ucm.es, {belend, pedro}@sip.ucm.es

Abstract. This paper describes our work in textual Case-Based Reasoning within the context of Semantic Web. Semantic Annotation of plain texts is one of the core challenges for building the Semantic Web. We have used different techniques to annotate web pages with domain ontologies to facilitate semantic retrieval over the web. Typical similarity matching techniques borrowed from CBR can be applied to retrieve these annotated pages as cases. We compare different approaches to do such annotation process: manually, automatically based on Information Extraction (IE) rules, and completing the IE rules within the rules that result from the application of Formal Concept Analysis over a set of manually annotated cases. We have made our experiments using the textual CBR extension of the jCOLIBRI framework.

1 Introduction

Textual CBR is an increasingly important CBR sub-discipline. Textual CBR techniques can facilitate rapid construction of CBR systems by reducing or eliminating the task of feature-design in domains in which raw cases consist of free or semi-structured text [2]. There are approaches where retaining a textual case representation may be more effective than engineering an intermediate feature representation. However, reasoning with text cases either requires considerable efforts to elicit meaningful features –beyond single words– or remains restricted to weak text retrieval based on information retrieval (IR) methods [14].

Ideally, we would like to find an inexpensive way to automatically, efficiently, and accurately represent textual documents as structured feature-based case representations. One of the challenges, however, is that current automated methods that manipulate text are not always useful because they are either expensive (based on natural language processing, NLP) or they do not take into account word order and negation (based on statistics) when interpreting textual sources. Information Extraction (IE) methods have been typically used for automatically extracting relevant factual information for the process of transforming texts into structured cases [3]. Other approaches have also been proposed aiming to take

* Supported by the Spanish Committee of Education & Science (TIN2005-09382-C02-01).

T.R. Roth-Berghofer et al. (Eds.): ECCBR 2006, LNAI 4106, pp. 226–240, 2006.
© Springer-Verlag Berlin Heidelberg 2006

the domain knowledge into consideration, as the use of Generative Ontologies proposed in [10] or the use of graphs that conserve and convey the order and structure of the source text [5].

"*The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation*" [1]. The Semantic Web aims at machine agents that search and filter the knowledge in the web pages based on explicitly specified semantics of contents. A core technology for making the Semantic Web happen is the field of *Semantic Annotation*, which turns human-understandable content into a machine understandable form [11]. There has been many literature about ontology-based semantic annotation of web pages [7], and there are different tools to help in this purpose [17].

In our ongoing work, we are considering the problem of semantic annotation of web pages (Section 2), and relating this problem to the feature elicitation problem in textual CBR (Section 3). The semantic web provides with such a set of plain Web Pages and Ontologies but is looking for automatic techniques to do such a labeling process. We begin with an initial set of web pages that have been manually annotated according to a certain ontology. Manual annotation is a tedious process that lacks from thoroughness and can not guarantee the uniformity of the tagged texts. So, we propose a semi-automatic process based on manually defined IE rules that results in an uniform labeling process but misses the inherent relationships between the labels that are not explicitly in the texts but exist in the domain and are available in the domain ontology. To solve the problem of connecting sparse information (e.g. a telephone number and an address in the contact information) we use Formal Concept Analysis, a data analysis technique that helps to find dependencies between the tags. Section 4 details the whole process. To show the goodness of our method, we have done an experiment annotating a set of web pages representing restaurants. Section 5 describes the experiment in detail while Section 6 compare the set of labels obtained by the different methods. The annotation process is used in a restaurant recommender system that improves the one presented in [16].

2 Annotation for the Semantic Web

Tim Berners-Lee's great dream of the Semantic Web may be visualized as computers that are able to *understand* what data is available on the Web. However, in a foreseeable future, machines will still be too dumb to understand what people have put on the Web. Therefore, to make this dream come true people must provide computer-understandable data. The building blocks have been elaborated in recent writings [8]: we need standardized languages to describe semantic self describing data and programs to exchange and understand semantic data. However, we are missing the key point here: where and how can we obtain semantic data?

The process of providing semantic data is often referred to as *semantic annotation* [11] because it typically involves the annotation of existing plain text,

that is only understandable by people, with semantic metadata available in ontologies.

The process of semantic annotation of these texts is a hard process. There have been different approaches, tools and annotation frameworks to help in this annotation process. Most of the current technology is based on human centered annotation. Typically they comprise methods for completely manual annotation and authoring of documents, where documents and contents are described at the same time. The large majority of annotation tools address the problem of single document annotation. This approach presents visualization and scalability problems, because the tagging knowledge in the ontologies can be huge and distributed and cannot be managed as a whole. The manual approach makes using very large ontologies very difficult. This is the main problem in tools like SMORE, OntoMat Annotizer, COHSE, Ontomat and MnM [17].

There are also semi-automatic annotation approaches based on IE that are trained to handle structurally and/or linguistically similar documents. Examples are KIM, Semantic World and Melita. A problem with this approach is that the process requires writing a large number of wrappers for information sources, and that extraction is limited to highly regular and structured pages. Besides, maintenance becomes a complex problem because when pages change their format, it is necessary to re-program the wrapper [12]. The approach is not applicable to irregular pages or free text documents. Also there is a problem of completeness because there is sparse information that is difficult to connect and there is also subjective information that is impossible to capture within IE rules. In the restaurants example, the *atmosphere* of a restaurant is a tag that reflects the general flavor of the place. Although sometimes we find words in the texts reflecting this feature, this is not the typical case, and the tag depends on the general and knowledge intensive impression of the skilled reader.

3 Textual CBR and Annotation

Textual CBR methods described in the CBR literature often focus on transforming textual data in semi-structured cases that can be used by the usual CBR methods. This process is analogous to the annotation of Semantic Web documents because both processes share the same goals: obtain a structure that allows indexing, retrieval and manipulation of the web documents/cases. The Semantic Web applications will use this structured information to let agents to search and manipulate web pages whereas CBR community will use this data for the CBR systems that work with structured cases.

We have continued our work in the jCOLIBRI framework and its Textual CBR extension presented in [16]. As jCOLIBRI is organized as a Task/Method decomposition system we developed several Problem-Solving Methods (PSMs) that process plain text files and obtain structured cases. Our framework divides CBR applications in three main tasks: precycle, cycle and postcycle. The textual extension implements PSMs that can be used in the precycle to transform plain text cases into structured ones. This way, these structured cases will be

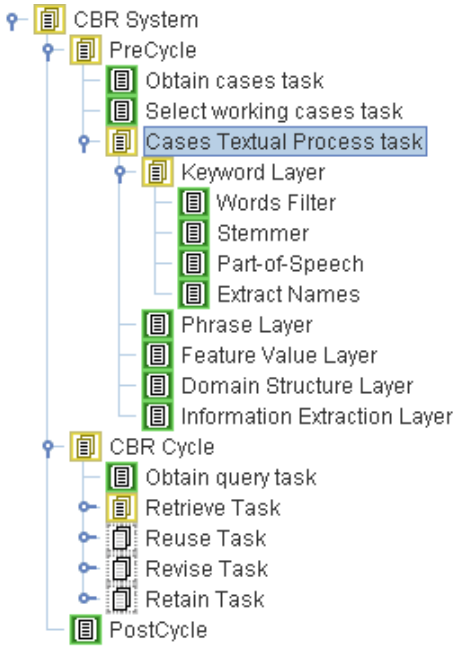


Fig. 1. jCOLIBRI Textual Tasks

manipulated by our library of PSMs that implement the CBR cycle (retrieve, reuse, revise, retain and all their subtasks). Figure 1 shows the task subdivision of the Textual process in jCOLIBRI. Each of these tasks must be solved by a method from our PSMs library.

The implementation of the Textual Extension is based in the theoretic Lenz layers for TCBR [13]. The developed methods described in [16] apply Natural Language Processing algorithms and Regular Expressions to perform the Information Retrieval and Information Extraction processes defined in each layer. After executing these methods jCOLIBRI obtains several syntactic features of the text that can be used as attributes in a structured case.

This paper presents one more step (see Figure 2): the use of ontologies to provide a semantic structure to the extracted features that improve the performance of the CBR cycle. For example, the retrieval task will use the semantic tags to recover semantically similar cases and the reuse phase will use this semantic information to manipulate the cases better. To achieve this goal we have looked at the Semantic Web community because it gives us the two features that we need to enhance the representation of our cases: semantic languages (like OWL) to represent data and repositories of ontologies. This new stage starts with the final structure returned by the set of subtasks shown in Figure 1. This structure contains description features that have been elicited through IE rules. We have slightly adapted the original IE rules to commit a certain domain ontology. With this transformation, the IE process returns pieces of data that correspond

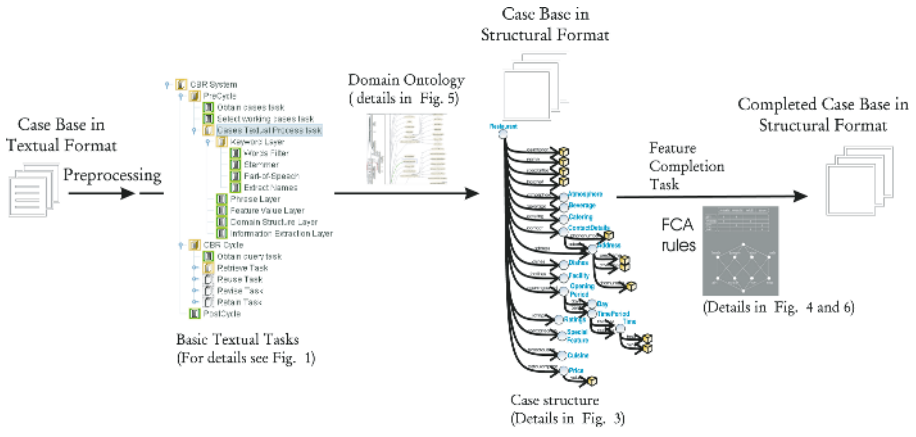


Fig. 2. Case Base refinement process

to concepts of an ontology. But the IE rules by themselves are unable to extract the whole structure of concepts imposed by the ontologies. For example, in the description of a *restaurant* there are *contact details* with *phone numbers* and an *address*; this address will be composed by the description of an *street* that has its *type*, *name* and *door number*. All these names in italics are concepts of the ontology but only some of them (usually the leaves of the grouping structure) can be obtained by the IE rules. In this example, the composite concepts: restaurant, contact details, address and street could not be extracted using simple IE rules.

To solve this problem of connecting sparse information we have applied Formal Concept Analysis (FCA) for completing the representation of the cases. With this new process we can accurately accomplish the transformation of plain text documents into semantically structured cases. These new cases will be based in an ontology that allows us to improve their manipulation in the CBR cycle. Our feature completion method begins with a set of manually tagged texts. We apply FCA as it is describe in Section 4 to extract dependencies between tags. Finally, we use these dependencies to complete the tags inferred by the IE process.

This method is used by our restaurant recommender presented in [16]. This CBR system developed using jCOLIBRI utilizes a case base composed by several texts describing restaurants. Then the IR and IE methods extract the attributes of the cases. The FCA annotation method described in this paper enhances the representation of the cases adding the semantics of the restaurant ontology. This added information improves the indexing, retrieval, and adaptation of the cases obtaining better qualitative results.

4 Annotation Enhancement Based on FCA

Previous sections make clear that both Semantic Web and Textual CBR lack automatic techniques to content annotation (web pages and plain texts).

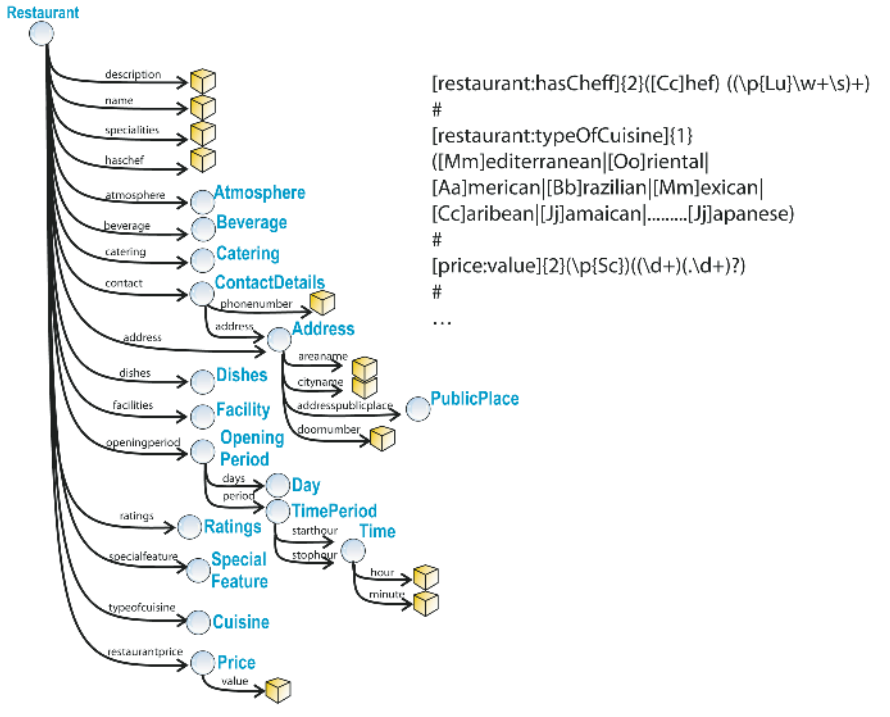


Fig. 3. Case Structure

In this section, we describe our semiautomatic annotation method, useful for both type of contents, and Section 5 and 6 show an experiment and its results.

Our annotation method combines an automatic annotator using Information Extraction rules and Formal Concept Analysis as a mean of obtaining dependencies (association rules) between tags, to provide hints to an expert human in order to facilitate his task of annotating contents.

Formal Concept Analysis is a mathematical approach to data analysis. It was first introduced in [18], and has been extensively used in many areas. See [19] for a gentle introduction.

FCA distinguishes between *formal objects* (or entities) and *formal attributes* (or features). We consider every text (or case) manually tagged as an object, and every possible tag as an attribute. The input of FCA is a binary relation called *formal context* that relates formal objects and formal attributes. The context is usually represented as an incidence table, with rows representing objects and columns representing attributes. Cells contain a cross when the object of that row has the attribute of that column.

In our case, we consider texts (or web pages) as *formal objets*, and tags as *formal attributes*. The formal context created has as many objects as texts, and as many attributes as distinct tags on them. A text (*formal objet*) is related with those tags (*formal attributes*) that appear in the manually annotated text.

Alegria
3510 Sunset Blvd.
Silver Lake
(323) 913-1422
The best food here revolves around ...

(a) Plain restaurant description

```
...
<restaurant:contact>
<contact:ContactDetails>
<contact:phoneNumber
   rdf:datatype="string">
(323) 913-1442
</contact:phoneNumber>
<contact:address>
<address:Address>
<address:areaName
   rdf:datatype="string">
Silver Lake
</address:areaName>
<address:addressPublicPlace>
<address:Boulevard
   rdf:ID="Sunset_Blvd."/>
</address:addressPublicPlace>
<address:cityName
   rdf:datatype="string">
Silver Lake
</address:cityName>
<address:doorNumber
   rdf:datatype="string">
3510
</address:doorNumber>
</address:Address>
</contact:address>
</contact:ContactDetails>
</restaurant:contact>
...
```

(b) Tagged restaurant

	contact	contactDetails	phoneNumber	addressPublicPlace	Boulevard	cityName	doorNumber	price	startHour	TakeAwayFacility	atmospheres	...
Alegria	■	■	■	■	■	■	■	■	■	■		...
Alex	■	■	■	■		■	■	■			■	...
Antique	■	■	■	■		■	■				■	...
Beverly	■	■	■			■						...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

(c) Formal context(tags' prefixes have been omitted for clarity)

Fig. 4. FCA example in restaurant context

Figure 4 shows an excerpt of the plain description of a restaurant, its annotated version and a section of the formal context where that restaurant appears.

With the formal context, FCA is able to build a set of *formal concepts* (or briefly *concepts*). Formally speaking, a concept is a pair (A, B), where A is a set of objects (known as *extent*) and B the set of the common attributes of these objects (*intent*). Formal concepts represent maximal groups of texts (or cases) with shared properties. The concepts of a given context can be ordered using the subconcept–superconcept relation and can be represented as a lattice, like the one showed in Figure 6a.

Though the formal concepts and lattice structure could be useful on their own [6], we use the capacity of mining association rules from it. An association rule is an expression $A \rightarrow B$ where both A and B are sets of attributes. They means that objects having all the attributes in A will probably have those attributes in B.

Association rules are characterized by two parameters: confidence and support. Confidence express the *probability* of that rule to hold, or in other words,

the percentage of objects that, having all the attributes in A also have those in B. On the other hand, support indicates the number of objects where the rule is applicable, formally speaking, the number of objects with attributes in A and B divided by the total number of objects.

Rule extraction algorithms based on FCA are able to efficiently extract all the association rules that have a confidence above a threshold. There are several algorithms, though we have used Duquenne–Guigues [9] to extract exact association rules (100% of confidence) and Luxenburguer [15] for non-exact ones.

Our annotation method starts with a set of texts or cases (C_1) that we annotate manually. Now ahead, we call the set of tags (labels) created manually for every text from C_1 as $L_M(C_1)$ (that stands for *l*abels *m*anually extracted) and it is composed of every text and its set of tags. With them, we then construct the formal context as described above (see Figure 4). Next, we apply FCA to extract the association rules between attributes (or tags). The set of rules, $R = fca(L_M(C_1))$, will be used later on the annotation process.

As an example, R can include a rule like

```
address:Address -> restaurant:contact
```

because all texts in C_1 that have `address:Address` tag also have the annotation `restaurant:contact`.

When our method receives a new text T to be annotated, it first uses Information Extraction rules to obtain a first version of its tags, $L_{IE}(T)$. IE is not expected to extract all the tags because of the limitations stated in Section 3. To enhance the results, we apply the rules R to the tags. Association rules will discover those tags that have not been discovered by the IE process, and we get our final set of tags, $L_{FCA} = Apply(R, L_{IE}(T))$.

Following the previous example, if $L_{IE}(T)$ has `address:Address` but lacks `restaurant:contact`, the application of R to the set of tags will discover that this tags has to be added.

To probe the enhancement of our annotation method, we have run it through a set of restaurant texts. We have compared the set of tags of the manual version (L_M) with the tags extracted by the information extraction rules (L_{IE}) and the final set of tags after the application of association rules (L_{FCA}). Section 5 details the experiment, and Section 6 shows the results.

5 Experiment Description

To run an experiment, we need an ontology and a set of web pages (or texts) to be annotated with it. Section 5.1 describes the ontology and Section 5.2 describes the set of texts we have used.

5.1 Ontology

Text annotation is made using a domain ontology. We have reused external ontologies created by the Agentcities Project[1]. These resources where originally

[1] http://www.agentcities.org

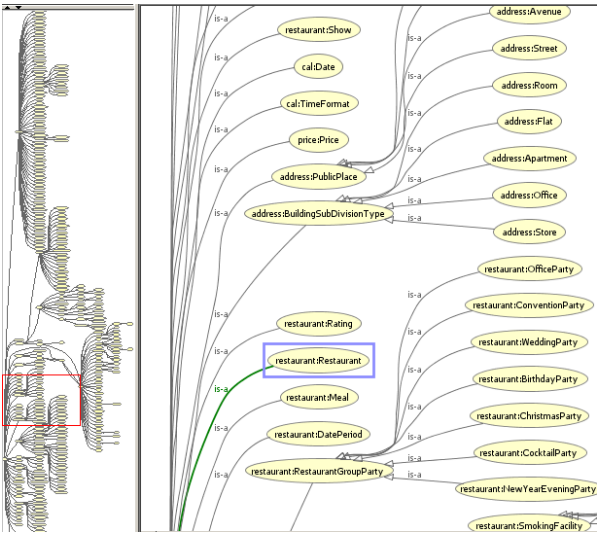


Fig. 5. Restaurant ontology

written in DAML+OIL [4]. We firstly translated them to OWL and then composed them properly.

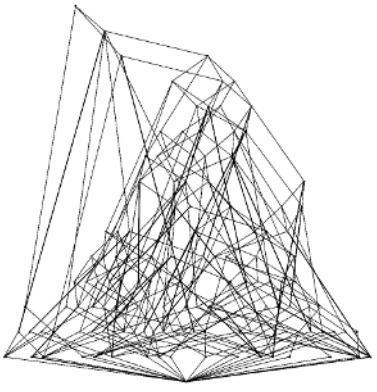
Our final restaurant ontology combines several sub-ontologies (address, price, calendar and food), and it has more than 1000 concepts, though only a few of them are used in the tagging process. Figure 5 shows a partial view of it. The complete version is available at `http://gaia.fdi.ucm.es/ontologies/restaurants.owl`.

5.2 Test Case Bases

We originally started from a case base of 268 textual cases with information about restaurants extracted from http://www.laweekly.com/eat-drink. To manage these textual cases we removed all the html tags of the original web pages obtaining only the plain text descriptions about restaurants.

Our goal was to compare our annotation method with the completely manual one. This way we had to manually annotate the texts describing restaurants with the ontology tags. But, this manual method is really complex and time consuming, so finally we did our experiment with a subset of 30 restaurants. On the other hand, the development of the IE rules adapted to the ontology cost about 4 times less that the manual process. So, we realized that if our annotation method had similar results to the manual one it would improve greatly the annotation process. Now ahead we will refer this set of 30 texts as C and its manually tagged version as $L_M(C)$.

We have duplicated the experiment, performing the annotation method and studying its accuracy twice in experiments A and B. In both of them, we split the set of restaurants C in two different sets, $C_{1\{A,B\}}$ and $C_{2\{A,B\}}$, having 20



Confidence	Set A	Set B
100%	137	117
95%	138	121
90%	168	166
85%	183	176
80%	192	180

(a) Lattice of training set A (b) Number of association rules

Fig. 6. FCA results

and 10 restaurants respectively. We have applied FCA to C_1 and applied the association rules extracted together with Information Extraction to C_2. Finally we compare the resulting tags ($L_{FCA}(C_2)$) with the manually annotated versions of C_2 ($L_M(C_2)$).

Both training sets A and B have been selected on purpose to reflect the best and the worst scenario. Set C contains several irregular descriptions that don't contain the same information that the other ones because some data and therefore tags like the address, price or type of food has been skipped. We have chosen sets A and B to contain these descriptions in C_1 or C_2. This way, experiment A contains the irregular descriptions in the set where we apply FCA: C_{1A}. On the other side, experiment B has these irregularities in the set of manually annotated restaurants that are extended with the FCA rules: C_{2B}. With this split we have intended to check how the noisy training examples can affect the accuracy of our method.

We have performed each experiment in four steps:

- The first one consists on the analysis of both training sets, C_{1A} and C_{1B} using FCA. As we have explained previously, association rules extraction has the minimum confidence as a parameter. Instead of just fixing it at 100% (exact association rules) we have used different levels of it to be able to infer how this parameter affects to the final results. Concretely, we extract the set of association rules from 100% (R_{A100} and R_{B100}) to 80% (R_{A80} and R_{B80}) of confidence using a decrement of 5%. Just to show the complexity of the case base, Figure 6 shows the lattice associated with training set A (C_{1A}) that has 135 formal concepts and the number of association rules we got.
- The second step takes the other 10 restaurants in their plain version (without manual annotation) and uses the IE rules to annotate them. These rules are a slight adaptation of the original rules to commit the restaurant ontology

used in [16]. Thereby, most of this task has been done reusing previous work in the jCOLIBRI textual methods.
Using the same notation as in Section 4, we call $L_{IE}(C_{2A})$ the set of tags we get in this step for experiment A and $L_{IE}(C_{2B})$ for B. We will write $L_{IE}(C_2)$ meaning the tags extracted applying IE to the plain texts. As we will see in Section 6, the number of tags in $L_{IE}(C_2)$ was about 40% of the number tags in the manual version, $L_M(C_2)$.
- In the third step, we apply the FCA rules of the first step in the restaurant annotated with IE. Obviously, we use R_{Ax} to enhance $L_{IE}(C_{2A})$ and R_{Bx} against $L_{IE}(C_{2B})$.
- The last step of our experiment compares the number of suggested annotations using FCA rules against the number of tags contained in M_1. Section 6.2 details this comparison.

Though it has only a theoretical value, we have perform an extra experiment, just to compare it with the other ones. It is what we will call in the next section "Complete Set". We have applied the association rules extractor to the complete set of manually annotated restaurants (C). Then we have applied IE to the same set of restaurants and enhance the results with the rules. Briefly speaking, we have applied our annotation method to the same set of cases that we used to train it. This experiment has no meaning in practice but in theory tells us the upper limit of the recall of the method. If our process was perfect this experiment should give us a perfect recall and precision because it checks the results with the same set used to train the system.

6 Experimental Results

We have performed several experiments to compare the accuracy of the Information Extraction and its improvement with the FCA rules. To measure the experiment we have used the two typical quality values:

- *Precision = Correctly extracted tags / total extracted tags*
  Tells if the extracted tags are correct (belong to the training set).
- *Recall = Correctly extracted tags / total correct tags*
  Represents the amount of correct tags that have been extracted.

6.1 Comparison Between L_M and L_{IE}

We have compared the tag set obtained using Information Extraction and the manual annotated set. The IE rules can only extract the tags corresponding with the leaves or final attributes of each restaurant annotation. The reason is that the upper concepts of the annotation tree are abstract concepts that cannot be extracted directly from the text. This problem of connecting sparse information was explained in Section 3.

Thereby, if the representation of a restaurant utilizes about 40 concepts and properties we have created IE rules for 20 of them. With these rules our IE

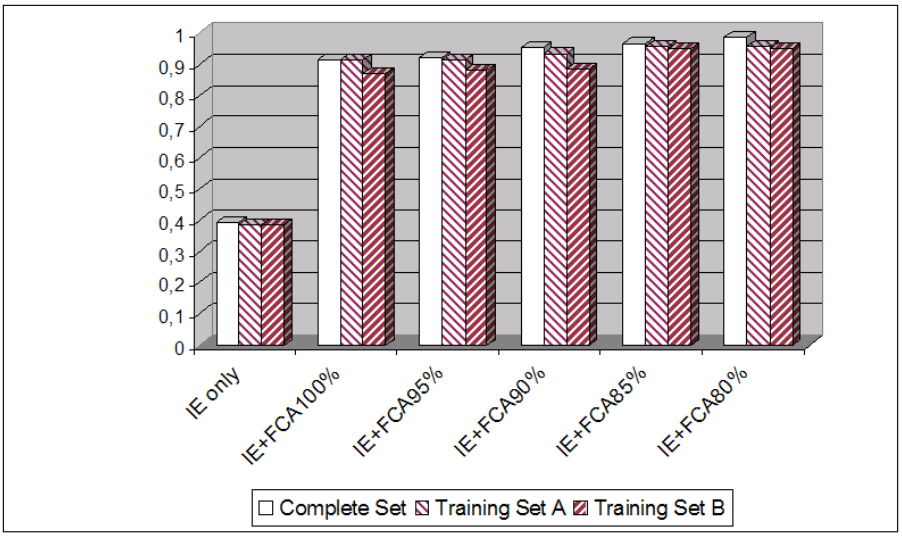


Fig. 7. Recall

process could obtain at most 50% of the total tags used in a restaurant annotation. The experimental results show that we extract 40% (in average) of the total tags. This value is represented in the first group of columns (IE only) of Figure 7 that shows the recall values obtained in our experiment. This value could be interpreted as a low value because other IE systems have a better performance, but in our approach we have not focused on the generation of high-quality IE rules. Our idea consists on developing the IE rules quickly and complete them with the FCA rules, saving time and effort in the whole annotation process. The precision of this comparison (tags extracted by the IE process that are also in the manual annotation) remained above the 98% of the total tags (see first group of columns of Figure 8). The rest of the tags (less than 2%) are the so called "false positives" returned by our IE module.

6.2 Comparison Between L_M and L_{FCA} Rules

Our experiments show that the FCA rules that complete the tags obtained by the IE module increase the recall from 40% to 90%. These results are shown in Figure 7 where recall increases as we decrease the confidence of the rules. Each group of columns in this figure represents the same experiment with different levels of confidence (IE+FCA100%, IE+FCA90%, ...).

On the other hand, as we increase the confidence we obtain a lower precision. Obviously, this is a direct effect of the confidence because it means more general rules that are more prone to generate "false positives". As Figure 8 shows, a confidence below 90% decreases too much the precision so the best configuration for our method in this experiment should use a value above 90%.

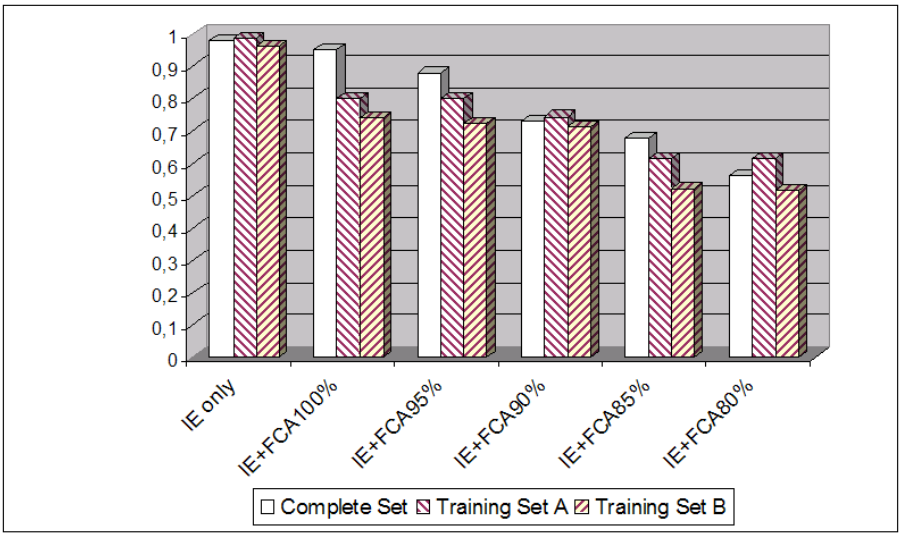


Fig. 8. Precision

In both Figures 7 and 8 the "Complete Set" column shows the value with the complete training set. The "Training Set A" and "Training Set B" columns represent the values when using the C_{1A} and C_{1B} training sets.

As we explained in Section 5.2 the "Complete Set" is a theoretic indication of the accuracy of our method and the results show that it is always higher than the practical experiments. It is important to note that the recall values are really close although the precision has significant differences. This result means that the main advantage of our method is that it retrieves nearly all the tags in the training set. Contrary, the main drawback of our method consists on retrieving too many incorrect tags besides the correct ones. This is specially meaningful in the experiments using confidence below 90%.

Using the division of the training examples in A and B we can obtain one more conclusion. Experiment A contains the noisy examples in the set used to extract the FCA rules C_{1A} whereas experiment B has these examples in the set enhanced with the FCA rules C_{2B}. As the results in A are better than in B we can conclude that the generation of FCA rules hides the errors produced by the irregular descriptions. Experiment B has worse results because its FCA rules are similar to A and C_{2B} has the noisy descriptions that, even enhanced with the rules, return a worse accuracy.

The global conclusion of the experiment is that FCA rules improve greatly the accuracy of the IE process. In theory, our scenario restricts the Information Extraction performance to obtain only 50% of the tags. In practique we obtain a value of 40% using simple and quickly developed IE rules. Completing the tagging with the FCA rules we increase automatically this value to 90% (loosing only 15% of the precision).

The recall increase indicates that our system extracts most of the concepts in the ontology that can not be obtained using Information Extraction. This way the system could automatically propose the concepts of the ontolgy inferred from the text that might be used during the semantical tagging process.

7 Conclusions

The aim of the research conducted is to investigate the relation between the problem of semantic annotation of web pages and the feature elicitation problem in textual CBR. Both processes share the same goals: obtain a structure that allows indexing, retrieval and manipulation of the web documents/cases.

The semantic web provides the CBR community with a very good field of experimentation. It provides with a lot of Web Pages (texts) that can be annotated with the knowledge on Ontologies. The underlying goal is to let machine agents to search and filter the knowledge in the web pages based on explicitly specified semantics of contents. From our point of view, this process can be understood and solved using CBR techniques where the cases are the annotated Web Pages.

We propose an annotation method that is based on three components: a set of IE rules, a domain ontology and a set of rules automatically extracted by the application of FCA to an initial set of manually annotated pages.

In this paper we have compared the accuracy of the annotation process. We have concluded that FCA allows finding dependency rules to solve the problem of connecting sparse information in the texts, and to find additional tags that depends on previously assigned tags. We have shown the results of comparing the set of labels obtained by the different methods. The annotation process is used in a restaurant recommender system that improves the one presented in [16].

References

1. T. Berners-Lee, J. Hendler, and O. Lassila. The semantic web, 2001.
2. L. K. Branting and R. Weber, editors. *Textual Case-Based Reasoning Workshop, at the 6th International Conference on Case-Based Reasoning*, Chicago, IL, USA, August 2005.
3. S. Brüninghaus and K. D. Ashley. The role of information extraction for textual CBR. In *Proceedings of the 4th International Conference on Case-Based Reasoning, ICCBR '01*, pages 74–89. Springer, 2001.
4. D. Connolly, F. v. Harmelen, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein. *DAML+OIL Reference Description*. World Wide Web Consortium, March 2001.
5. C. Cunningham, R. Weber, J. M. Proctor, C. Fowler, and M. Murphy. Investigating graphs in textual case-based reasoning. In P. Funk and P. A. González-Calero, editors, *Proceedings of Advanced in Case-Based Reasoning, 7th European Conference on Case-Based Reasoning, ECCBR 2004*, volume 3155 of *Lecture Notes in Computer Science*, pages 573–587. Springer, 2004.
6. B. Díaz-Agudo and P. A. González-Calero. Formal Concept Analysis as a Support Technique for CBR. In *Knowledge-Based Systems, 14 (3-4)*, pages 163–172. Elsevier, June 2001.

7. M. Erdmann, A. Maedche, H. P. Schnurr, and S. Staab. From manual to semi-automatic semantic annotation: About ontology-based text annotation tools. *ETAI Journal - Section on Semantic Web (Linköping Electronic Articles in Computer and Information Science)*, 6(2), 2001.
8. Y. Gil, E. Motta, V. R. Benjamins, and M. A. Musen, editors. *The SemanticWeb, 4th International SemanticWeb Conference, ISWC 2005*, volume 3729 of *Lecture Notes in Computer Science*, Galway, Ireland, November 2005. Springer.
9. J.-L. Guigues and V. Duquenne. Familles minimales d'implications informatives resultant d'un tableau de données binaires. *Math. Sci. Humanies 95, 1986, 5-18.*
10. K. M. Gupta. The role of generative ontologies in textual CBR. In *Invited Talk in the Textual CBR Workshop at the 6th International Conference on Case-Based Reasoning, Chicago, IL*, 2005.
11. S. Handschuh and S. Staab, editors. *Annotation for the semantic Web.* IOS Press, 2003.
12. N. Kushmerick, D. Weld, and R. Doorenbos. Wrapper induction for information extraction. In *Fifteenth International Joint Conference on Artificial Intelligence, IJCAI'97*, pages 729–737, Nagoya, Japan, 1997. Morgan Kaufmann.
13. M. Lenz. Defining knowledge layers for textual case-based reasoning. In *Proceedings of the 4th European Workshop on Advances in Case-Based Reasoning, EWCBR-98*, pages 298–309. Springer, 1998.
14. M. Lenz. Knowledge sources for textual CBR applications. In M. Lenz and K. Ashley, editors, *AAAI-98 Workshop on Textual Case-Based Reasoning*, pages 24–29, Menlo Park, CA, 1998. AAAI Press.
15. M. Luxenburguer. Implications partielles dans un contexte. *Mathématiques, Informatique et Sciences Humaines*, 113(29):35–55, 1991.
16. J. A. Recio, B. Díaz-Agudo, M. A. Gómez-Martín, and N. Wiratunga. Extending jCOLIBRI for textual CBR. In H. Muñoz-Avila and F. Ricci, editors, *Proceedings of the 6th International Conference on Case-Based Reasoning, ICCBR 2005*, volume 3620 of *Lecture Notes in Artificial Intelligence, subseries of LNCS*, pages 421–435, Chicago, IL, US, August 2005. Springer.
17. Semantic web annotation and authoring. `http://annotation.semanticweb.org/tools/`.
18. R. Wille. *Restructuring Lattice Theory: an approach based on hierarchies of concepts.* Ordered Sets, 1982.
19. K. E. Wolff. A first course in formal concept analysis. How to understand line diagrams. In F. Faulbaum, editor, *SoftStat, Advanced in Statistical Software*, pages 429–438. Gustav Fischer Verlag, 1993.