A Semantic Case-Based Reasoning Framework for Text Categorization

Valentina Ceausu[1] and Sylvie Després[2]

[1] CRIP 5 - University of Paris 5,
45, Rue des Saints Pères
Paris 75006, France
ceausu@math-info.univ-paris5.fr
[2] LIPN UMR CNRS 7030 - University of Paris 13,
99 avenue Jean Baptiste Clément
93430 Villetaneuse, France
sylvie.despres@lipn.univ-paris13.fr

Abstract. This paper presents a semantic case-based reasoning framework for text categorization. Text categorization is the task of classifying text documents under predefined categories.

Accidentology is our application field and the goal of our framework is to classify documents describing real road accidents under predefined road accident prototypes, which also are described by text documents. Accidents are described by accident reports while accident prototypes are described by accident scenarios. Thus, text categorization is done by assigning each accident report to an accident scenario, which highlights particular mechanisms leading to accident.

We propose a textual case-based reasoning approach (TCBR), which allows us to integrate both textual and domain knowledge aspects in order to carry out this categorization. CBR solves a new problem (target case) by identifying its similarity to one or several previously solved problems (source cases) stored in a case base and by adapting their known solutions. Cases of our framework are created from text. Most of TCBR applications create cases from text by using Information Retrieval techniques, which leads to knowledge-poor descriptions of cases. We show that using semantic resources (two ontologies of accidentology) makes possible to overcome this difficulty, and allows us to enrich cases by using formal knowledge.

In this paper, we argue that semantic resources are likely to improve the quality of cases created from text, and, therefore, such resources can support the reasoning cycle. We illustrate this claim with our framework developed to classify documents in the accidentology domain.

Keywords: semantic description, ontology, text categorization, case-based reasoning, accidentology.

1 Introduction

Case-based reasoning (CBR), [1] is a problem solving paradigm which solves a new problem by re-using a collection of already solved problem (called source

K. Aberer et al. (Eds.): ISWC/ASWC 2007, LNCS 4825, pp. 736–749, 2007.

cases). This collection represents the case base. Textual CBR, see [2] is an extension of CBR which could be applied in domains where experiences are described by text documents. As many domains produce a large amount of textual data describing problems and their solutions, developing CBR systems able to deal with unstructured or semi-structured text is particularly challenging.

Text documents are unstructured stream of characters, over which only shallow reasoning based on easily observable surface features can be performed.

Thus, cases of TCBR systems are often created by hand or have simplified representations, which can be created by using results of Information Retrieval methods, see [3], [4] or [5].

[6] points out the role of such methods in creating textual cases. Those methods are based on shallow statistical inferences over word vectors, and allow creating a linguistic description of cases, as cases are represented by terms extracted from text. By using Information Retrieval methods, knowledge-poor representations of textual cases are obtained.

This leads to a bottleneck in creating and scaling up TCBR systems, since manual construction of cases often involves inhibitory costs and simplified representations of cases lead to an inefficient reasoning cycle, as little knowledge could be exploited by the cycle.

However, there is a severe gap between the knowledge required for TCBR and the results provided by methods one can perform on textual documents.

Thus, methods, like in particular Information Retrieval, are not sufficient to create knowledge-rich case representations from text. As, among others, [7] points out, the weakness of simple Information Retrieval methods is its lack of exploitation of knowledge about domain objects and relationships.

Since TCBR application is domain specific, descriptions of cases can be improved by using domain ontology. [8] defines an ontology as a formal, explicit specification of a shared conceptualization. Ontology is a formal representation of domain knowledge, providing information about specific objects of the domain and relationships between them. Domain is modeled at conceptual level, in an implementation independent manner. Objects are modeled by concepts, having particular attributes. Relationships between them are modeled as roles, which are binary relations holding between concepts. Each role has a domain and range, both of which are concepts of ontology.

For this work, we assume that an ontology takes into account the linguistic level of entities. Thus, concepts and roles are labeled by terms, which are linguistic manifestation of ontology entities in a specific language (French, English, etc.). Therefore, ontology considered for this work has two levels: a conceptual level, describing domain specific entities (concepts and roles) and a linguistic level, providing linguistic manifestations of those entities in a given language.

Therefore, we claim that such ontology could help creating cases from text. We illustrate this claim by presenting ACCTOS (ACCident TO Scenarios), a TCBR framework integrating ontologies to create cases from text. Cases of ACCTOS are described at formal level, by concepts and roles of semantic resources. By integrating those resources, formal knowledge could be exploited by the reasoning cycle.

2 Assigning Accident Reports to Accident Scenarios: A Text Categorization Task

This paper deals with automatic assignment of documents describing real road accident to documents describing road accident prototypes. Road accidents are described by accident reports while accident prototypes are described by accident scenarios.

Accident reports are documents created by the police. They include structured paragraphs describing the context of an accident and people involved in, and natural language paragraphs explaining what happened in the accident. Those paragraphs are written by policemen, with the help of witnesses and people involved in the accident.

Accident scenarios are documents created by researchers in road safety. They are prototypes of road accidents and present in a general way facts and causal relations between different phases leading to a collision. Prevention measures aiming to improve road safety are provided for each accident scenario. A first study led by the department Mechanisms of Accidents of INRETS [1] established a first collection of accident scenarios involving pedestrians.

As the tab. 1 shows, there is a number of differences between accident reports and accident scenarios. Thus, accident reports are created by the police, while

Table 1. Accident reports vs. Accident scenarios

	Accident reports	Accident scenarios
created by	policemen	road safety researchers
langage	current language	expert language
contains	description of accidents	expert knowledge
structure	semi structured	free text
goal	identify	prevention of
	legal responsibility	road accidents

accident scenarios are created by researchers in road safety. Therefore, accident reports provide descriptions of road accidents written in current language. This means that a notion is often designated by many synonym terms (i.e. person driving a car: *conducteur, chauffeur, automobiliste (driver)*). Accident scenarios are written in expert language, and the same term is always used to designate a notion (i.e. person driving a car: *conducteur (driver)*).

Assigning an accident report to an accident scenario is twofold: from a domain specific point of view, it allows us to identify particular mechanisms leading to accident; from a linguistic point of view, it allows us to create a bridge between the languages of two different communities (researchers and policemen) of the same domain.

We consider accident scenarios as predefined text categories, as they describe prototypes of road accidents. Therefore, assigning an accident report to an accident scenario is a text categorization task. Moreover, preventions measures are

[1] Institut National de Recherche sur les Transports et leur Sécurité.

provided for each accident scenario. Thus, an accident scenario and his prevention measures can by seen as a problem description (a particular prototype of accident) and his solution (measures proposed in order to avoid this particular prototype of road accident). By consequent, we developed a textual case-based reasoning framework in order to carry out this categorization task. On the following we present this framework.

3 ACCTOS: A TCBR Framework for Text Categorization

ACCTOS is a textual case-based reasoning frame developed to classify textual documents.

3.1 ACCTOS Input/Output Data

The input of the system is a set of accident reports. ACCTOS exploits electronic accident reports, which have been made anonymous by the PACTOL [2] tool. An electronic accident report is a semi-structured document containing structured paragraphs and natural language paragraphs. Structured paragraphs specify a number of variables describing: people and vehicles involved in accident, accident context and accident environment. Natural language paragraphs describe what happened in the accident according to several points of view: police (synthesis), people involved (declarations) and witnesses (testimonies).

The output of the system is set of assignments, where each assignment is composed of a couple *accident report, accident scenario* and a trust assessment.

3.2 Architecture of ACCTOS

ACCTOS adopts a CBR approach. CBR solves a new problem (target case) by exploiting a collection of already solved problems (source cases). The CBR reasoning cycle consists of following phases:

- target case elaboration: creates the target case (problem to solve);
- case retrieval: identifies a number of source cases similar to the target case;
- case adaptation: adapts solutions of source cases (identified by the previous phase) in order to propose a solution for the target case;
- memorization phase: enrich the case base, by adding the target case and his solution.

ACCTOS implements two phases of the CBR reasoning cycle: target case elaboration and case retrieval. To present the architecture of ACCTOS, we use a division into modules, where each of the module addresses a different phase of the reasoning cycle (see Fig. 1).

The need for formal knowledge to create cases from text. Target cases of ACCTOS are created from accident reports. Source cases are created from accident scenarios.

[2] Centre d'Etudes Techniques de l'Equipement de Rouen.

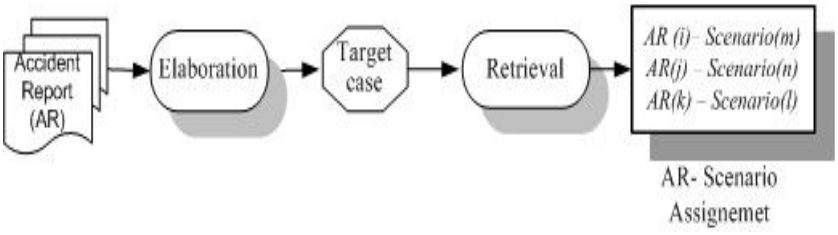

Fig. 1. System architecture

Creating cases from text is a difficult task, as text is unstructured data. To cope with this difficulty, we use semantic resources allowing us to create knowledge-rich representations of cases. As the framework exploits documents created by two communities, we integrate two semantic resources, modeling the accidentology domain according to each community. Thus, the *expert ontology* describes the domain from an expert point of view, while the *facts ontology* describes the domain from a police point of view. Both ontologies are expressed in OWL, [9].

By modeling each community by a semantic resource, it becomes possible to reflect the dynamic of the community. For instance, the expert ontology can be enriched when new scenarios are created by experts in road safety.

3.3 Representation of Cases

We proposed a model to represent cases of ACCTOS. According to this model, a case is described by two types of elements: global variables and agents.
Global variables specify the number of agents involved in accident, the environment in which the accident occurred - such as main road or secondary road - and context of the accident (by day, in intersection, etc.).
A human involved in accident and his vehicle represent an agent (see tab.2).

By using agents, it becomes possible to cope with difficulties related to metonymy between the human involved in accident and his vehicle (i.e. *vehicle stops* vs. *driver stops*). It also allows us to treat the particular case of pedestrian. Each agent is defined by his two components - the human H and the vehicle V - and by his evolution in accident. Each component of an agent is designated by a domain term (ie: driver, car) and has several attributes (ie: age is an attribute of Human). Agent evolution is specified by a set of relations describing interac-

Table 2. Components of an agent

Agent	Humain	Vehicle	Attributes	Evolution
Agent 1	piéton (pedestrian)	no vehicle	age: 35	traverser; courir (crossing; running)
Agent 2	(conducteur) (driver)	Véhicule (car)	age: 60	circuler; tourner (circulate; turning to)

tions between his own components and also between the agent and other agents involved in accident.

3.4 Creating Source Cases by Using the Expert Ontology

Cases of case base are called source cases and have a two parts: the problem and his solution. A set of accident scenarios is used to build the initial case base of the system. The accident scenario represents the *Problem*; measures of preventions assigned to the scenario represent the *Solution*. For this work, the solution part of source cases is ignored, as the adaptation phase of the reasoning cycle is not implemented.

The expert ontology (see [10]) supports the description of source cases. This ontology was built from scratch, by using a corpus composed of accident scenarios and expert knowledge. It models concepts of accidentology and relations holding between them. Concepts are structured in three main classes: concepts describing the *human*, the *vehicle* and the *environment*. Each concept is named by a term and has different attributes.

Concepts are connected by *is-s* relations and roles. *is-a* relations build the hierarchy of domain concepts. Roles describe interactions between concepts and are named by domain specific verbs (i.e. *CirculerSur(Véhicule, Infrastructure*) (CirculateOn(Vehicle,Infrastructure))).

We developed an editor in order to create a source case from an accident scenario. The editor integrates the expert ontology and allows us to describe each accident scenario by a set of concepts and roles of the ontology. The editor also allows us to assign an importance coefficient to each concept or role. For each source case, those coefficients are established by experts. By integrating this ontology, we create source cases having homogeneous descriptions and we can describe them by using formal knowledge (concepts and roles).

3.5 Target Case Elaboration by Using Semantic Traces

The goal of this phase is to create a target case from an accident report. We create cases from text by using *semantic traces*. On the following we introduce semantic traces and we present the approach proposed to identify semantic traces from text.

Semantic Traces: Definition. Let C be a corpora and O an ontology of the same domain. As entities of O have linguistic descriptions, it becomes possible to identify within C terms similar to those naming entities of O. If e is an entity of O, we define a semantic trace of e as a term t of the C corpora which is similar to the label of e. In other words, semantic traces are terms of C which are similar (from a lexical point of view) to those naming the entities of O.

Discovering semantic traces is based on the following working hypothesis: if synonymy is not considered, then any entity is named by using the same set of characters, called the core set. Consequently, terms which are named by a set of characters close to the core set represent either the same or a similar entity.

Terms expressed by a set of characters completely different to the core set are referring potentially different entities.

Once identified within the corpus, these terms can be labelled by entities of the ontology. A semantic trace of a concept c (role r) is discovered each time a term is labelled by the concept c (the role r).

Discovering Semantic Traces. We have proposed a two steps approach to discover semantic traces within corpora. The first step extracts terms by using an Information Retrieval method. The second one uses string similarity metrics to label extracted terms by ontology entities.

Terms' extraction using Information Retrieval. Terms are extracted from corpus by using an Information Retrieval method, based on lexical patterns, see [11]. We define a lexical pattern as a particular combination of part-of-speech categories. For instance *Noun, Preposition, Noun* or *Verb, Preposition, Noun* are lexical patterns. In order to identify instances of patterns, the corpora is tagged using TreeTagger, see [12], which makes part-of-speech information available. This method consists in defining a set of lexical patterns able to extract potentially valid terms from corpora. Then, a pattern recognition algorithm which we implemented retrieves word regroupings matching lexical patterns[3], see tab. 3. We defined two categories of lexical patterns in order to order to discover

Table 3. Lexical patterns and instances

Pattern	Instance	Note
Noun, Preposition, Noun	ceinture de sécurité (seat belt)	domain term
Noun, Noun	passage piéton (cross road)	domain term
Verb, Preposition	diriger vers (direct to)	verb relation
Verb, Preposition, Noun	venir de i (come from i)	noise

semantic traces: *nominal lexical patterns* are associations of part-of-speech categories which do not include a verb and *verbal lexical patterns* are associations of part-of-speech categories including a verb.

Nominal lexical patterns highlight domain terms, so instances of those patterns could be semantic traces of concepts. Verbal lexical patterns highlight domain relationship expressed by verbs, so instances of those patterns could be semantic traces of roles. The pattern recognition algorithm is applied at sentence level and automatically generates two sets of lexical pattern instances.

Using an ontology to pass from a linguistic description to a formal description of cases. The method described in this section allows identifying terms of corpus by using a basic Information Retrieval method. While creating cases from text, this method could provide a description of cases, as significant terms can be extracted. However, this is a linguistic description, as only terms

[3] Examples of this paper are translated in English, although they are extracted from a French corpus experimentation.

are identified. On the other hand, terms can be considered as linguistic manifestations of concepts or roles, see [13]. Hence, if a domain ontology is available, it becomes possible to label those terms by entities of the ontology. This allows us to enrich description of textual cases, by using concepts and roles by which terms were labelled. In the next section we describe the labelling of terms by entities of ontology.

Semantic labelling of terms. As the previous section shows, the pattern recognition algorithm identifies instances of nominal and verbal lexical patterns. Instances of nominal patterns could be labelled by concepts, as they could highlight domain terms. Instances of verbal patterns highlight relations of the domain, therefore they can be labelled by roles.

To label a nominal instance, the set of concepts is considered. A string similarity coefficient, see [14] is used to calculate the similarity between instances and terms naming ontology concepts. Each instance will be labelled by the concept whose label (term naming the concept) maximizes the value of this similarity, if the maximum similarity is above a threshold value. Otherwise, the instance is labelled as *inconnu*, (unknown). Each nominal instance labelled by a concept represents the semantic trace of this concept.

Instances of verbal patterns are labelled in a similar way, by considering the set of roles modelled by the ontology. Each verbal instance labelled by a role represents the semantic trace of this role.

3.6 Using Semantic Traces to Elaborate Target Cases of ACCTOS

The goal of this phase is to create the target case. The target case is created from text by discovering semantic traces entities modelled by the *facts ontology*, see [15]. This ontology was created from a corpora of about 250 accident reports, by using the *Terminae* tool, see [16]. It models concepts of accidentology and relations holding between them according to a police specific point of view. This ontology points out linguistic particularities of this community, thanks to his conceptual and linguistic level.

Each target case is represented according to the model presented in the section 3.3 and is created from an accident report. An accident report is a semi-structured document, composed of specific structures and natural language paragraphs. Specific structures provide data about: people and vehicles involved in accident, accident context and accident environment. Natural language paragraphs provide descriptions of the accident, according to several points of view (people involved, witnesses). Target cases are created by exploiting both specific structures and natural language paragraphs of an accident report, as shown in the following.

Identification of global variables. Values of global variables are identified by automatic procedures exploiting the structure of accident reports.

Identification of agents. To describe an agent involved in accident we need to:

– identify terms naming his components;
– identify values of his attributes;
– identify his evolution.

Terms naming component of an agent and values of his attributes are also identified by automatic procedures exploiting the structure of accident reports.

In order to enrich the description of target cases, terms naming components are labelled by concepts of facts ontology, by using the two steps approach presented in section 3.5. This labelling is always possible, as the ontology was created from accident reports. Moreover, it allows us to pass from a linguistic description of components, to a formal one, as we can see in tab 4. Further, this formal description allows us to identify evolutions of agents, as it follows.

Table 4. Semantic labelling of instances

Type of instance	Instance	*Ontology entity*
Nominal	jeune piéton	*piéton SubConceptOf(Personne)*
	(young pedestrian)	*(pedestrian SubConceptOf(Person)*
Verbal	circuler sur	*circuler(Véhicule, Infrastructure)*
	(circulate on)	*(circulate(Vehicle, Infrastructure))*

Identification of agents' evolution. Evolutions of agents are expressed by a set of verbs appearing in natural language paragraphs (synthesis, declarations and testimonies) of accident reports. The evolution of an agent is identified by discovering traces of facts ontology roles within these paragraphs.

Semantic traces of roles are discovered by using the approach presented in section 3.5. Hence, two lexical patterns are defined: *Verb* and *Verb, Preposition*. Instances of those patterns are identified within paragraphs (previously annotated by TreeTagger). Then, those instances are labelled by roles of facts ontology.

Semantic traces identified consists in a set R of verbs which are similar (from a lexical point of view) to verbs naming roles of facts ontology, see (1).

$$Traces_{evolution} = \{t_1, t_2, ..., t_n | t_i \text{ is a semantic trace}\} \quad (1)$$

In order to identify evolution of agents, each semantic trace is replaced by the corresponding role. By doing so, we end up with a set of roles describing evolutions of all agents involved in accident, see (2).

$$Roles_{evolution} = \{r_1, r_2, ..., r_n | r_i \in Roles_{RTO}\} \quad (2)$$

where $Roles_{RTO}$ is the set of facts ontology roles.

Let a_i be an agent whose components are described by concepts H and V. This agent should identify, among roles of $Roles_{evolution}$, those describing his own evolution. To do so, agent a_i query the facts ontology in order to get roles

of this ontology having H or V as domain. As consequence, a set $Roles_{(H,V)}$ is obtained, see (3).

$$Roles_{(H,V)} = \{r_1, r_2, ..., r_n | r_i \in Roles_{RTO} \text{ having H or V as domain}\} \quad (3)$$

The evolution of agent a_i is given by the intersection of the two sets: $Roles_{(H,V)}$ and $Roles_{evolution}$, see 4.

$$Evolution_{(H,V)} = Roles_{(H,V)} \bigcap Roles_{evolution} \quad (4)$$

The evolution of each agent is expressed by a set of facts ontology roles, whose traces were identified within natural language paragraphs, see fig. 2.

4 Semantic Retrieval

The retrieval phase aims to retrieve source cases similar to the target case. Already solved problems similar to the target case are identified. Therefore, a solution can be proposed to the target case by adapting solutions of those problems. As both target cases and source cases of ACCTOS have semantic descriptions, we propose a retrieval approach supported by the alignment of the experts and facts ontology.

We have proposed an alignemed approach, decribed in [17]. The alignement is given by a similarity function $Sim(e_e, e_f)$ which allows us to estimate similarity between entities (concepts or roles) of experts ontology(e_e) and entities of facts ontology (e_f).

Let T be a target case. Two steps are needed to retrieve similar source cases. *(1) The first step is based on case base indexation.* Global variables are used to index the case base. Values of global variables of the target case are taken into account to identify a set of source cases. The result is a set of source cases having the same context as the target case and involving the same number of agents. *(2) A voting process* is used to improve this first selection. The vote is done by each target case agent to express the resemblance degree between himself and agents of a source case. A note is given by each target case agent to every source case. This note is given by taking into account components of agents and their evolutions. A first similarity measure proposed is given by:

$$Sim(a_i, a_j) = SimComponent(a_i, a_j) + SimEvolution(a_i, a_j) \quad (5)$$

if $SimComponent(a_i, a_j) \neq 0$, otherwise $Sim(a_i, a_j) = 0$, where a_i is an agent of the target case and a_j is an agent of a source case, and : $SimComponent(a_i, a_j)$ expresses resemblances between components of two agents, and is given by :

$$SimComponent(a_i, a_j) = ch_j * sim(H_i, H_j) + cv_j * sim(V_i, V_j) \quad (6)$$

where ch_j and cv_j are importance coefficients established for the source case, and values of $sim(H_i, H_j)$ and $sim(V_i, V_j)$ are given by the alignment of the two resources.

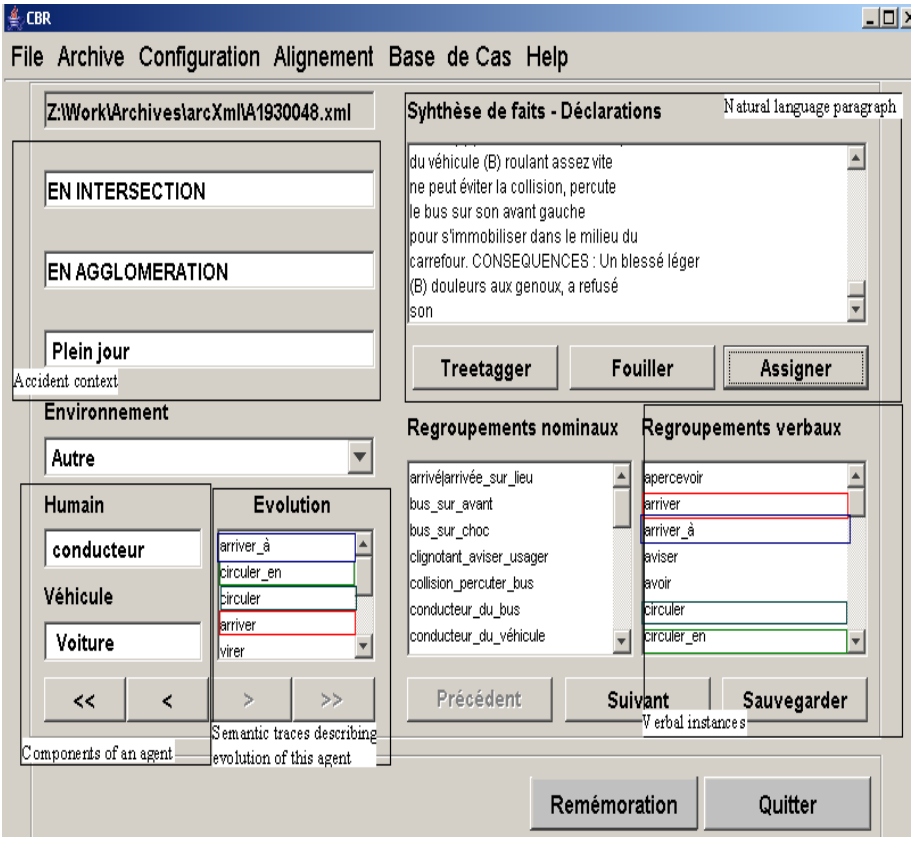

Fig. 2. Identification of evolution

Evolution similarity expresses resemblances between evolutions of two agents :

$$SimEvolution(a_i, a_j) = \frac{\sum_r c_r * sim(rSource_r, rTarget_r)}{\sum_r c_r} \tag{7}$$

where coefficients c_r expresses the importance of $rSource_r$ role for the considered source case. Values of $sim(rSource_r, rTarget_r)$ are given by alignment of the two resources.

Each agent of the target case evaluates his resemblance to agents of the source case by using the presented approach. A similarity vector is obtained. The note $note_i$ given by the $agent_i$ to the source case is the maximum value of this similarity vector. Based on notes given by agents, the similarity between the target case and a source case is estimated by the average value:

$$Sim(target, source) = \frac{\sum_{i=1}^{N_a} note_i}{N_a} \tag{8}$$

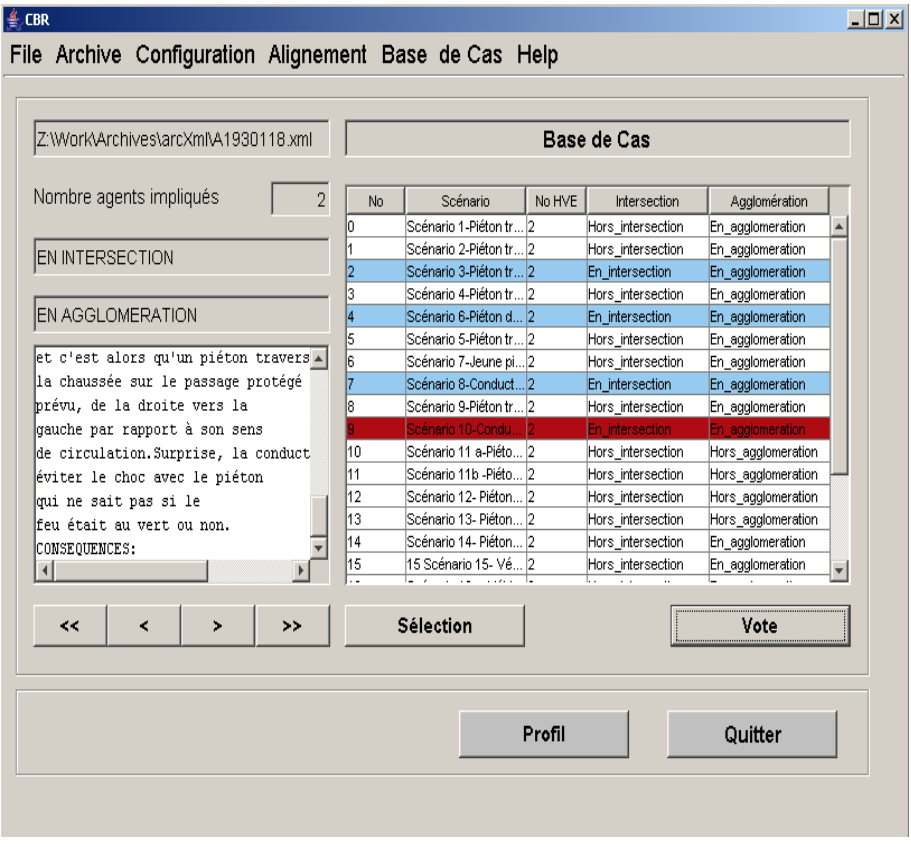

Fig. 3. Case retrieval

where $note_i$ is the note granted by the agent $agent_i$, and N_a is the number of agents of the considered target case. Case base indexation allows a fast identification of source cases that are similar to the target case. By voting, the most similar cases are selected among the cases retrieved by the first selection. The retrieval process is driven by the description of source cases whose importance coefficients are taken into account by similarity measures. Fig. 3 shows cases selected by case base indexation (light gray) and by vote (dark gray).

5 Conclusion and Future Work

This paper presents a semantic case-based reasoning framework for text categorization. Cases of the framework are created from natural language documents provided by two different communities: accident reports written by the police and accident scenarios created by road safety researchers.

Semantic resources are used to cope with heterogeneity and difficulties related to case elaboration from natural language documents. Two ontologies are used to create source and target cases of the system. By integrating semantic resources, we can create knowledge-rich descriptions of cases, as cases are described by concepts and roles of two different ontologies. The advantage is that this knowledge can be used by the reasoning cycle. Hence, the retrieval phase is supported by aligning the expert and the facts ontology.

The development if the framework is finished. There now remains to evaluate his results and to identify different ways to improve them. As for now, an expert evaluation of the system is ongoing. This evaluation is carried out in collaboration with road safety experts, able to validate *acident report, accident scenarios* assignments provided by ACCTOS. This validation will allows us to evaluate the precision of ACCTOS results.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: Foundational issues, methodological variations and system approaches. AICom - Artificial Intelligence Communications 7(1), 39–59 (1994)
2. Lamontagne, L., Lapalme, G.: Raisonnement à base de cas textuel: état de l'art et perspectives futures. Revue d'intelligence artificielle 16(3), 339–366 (2002)
3. Wiratunga, N., Koychev, I., Massie, S.: Feature selection and generalisation for retrieval of textual cases. In: Proceedings of the 7-th European Conference on Case-Based Reasoning (2004)
4. Gupta, K., Aha, D., Sandhu, N.: Exploiting taxonomic and causal relations in conversational case retrieval. In: Proceedings of the Sixth European Conference on Case-Based Reasoning (2002)
5. Bergmann, R.: On the use of taxonomies for representing case features and local similarity measures. In: Proceedings of the 6th German Workshop on Case-Based Reasoning (1998)
6. Bruninghaus, S., Ashley, K.D.: The role of information extraction for textual cbr. In: Aha, D.W., Watson, I. (eds.) ICCBR 2001. LNCS (LNAI), vol. 2080, pp. 74–89. Springer, Heidelberg (2001)
7. Lenz, M.: Textual cbr and information retrieval - a comparison. In: Proceedings of the 6th German Workshop on Case Based Reasoning (1998)
8. Gruber, T.: A translation approach to portable ontology specifications. Knowledge Acquisition, 199–220 (1993)
9. Smith, M., Welty, C., McGuinness, D.: Owl web ontology language guide. Technical report, W3C, W3C Proposed Recommendation (2004)
10. Després, S.: Contribution à la conception de méthodes et d'outils pour la gestion des connaissances. In: Habilitation à diriger des recherches, Université René Descartes (2002)
11. Seguela, P.: Adaptation semi-automatique d'une base de marqueurs de relations sémantiques sur des corpus spécialisés. In: Terminologie et Intelligence Artificielle (1999)
12. Schmid, H.: Probabilistic part-of-speech tagging using decision trees. In: Proceedings of the International Conference on New Methods in Language Processing (1994)

13. Ville-Ometz, F., Royauté, J., Zasadzinski, A.: Filtrage semi-automatique des variantes de termes dans un processus d'indexation controlée. In: Proceedings of Colloque International sur la Fouille de Textes (2004)
14. Cohen, W., Ravikumar, P., Fienberg, S.: A comparison of string distance metrics for name-matching tasks. In: IJCAI 2003. Proceedings of the International Joint Conference on Artificial Intelligence, Workshop on Information Integration on the Web pages (2003)
15. Ceausu, V., Desprès, S.: Towards a text mining driven approach for terminology construction. In: Proceedings of the 7th International conference on Terminology and Knowledge Engineering (2005)
16. Biébow, B., Szulman, S.: A linguistic-based tool for the building of a domain ontology. In: Proceedings of the International Conference on Knowledge Engineering and Knowledge Management (1999)
17. Ceausu, V., Després, S.: Alignement de ressources sémantiques à partir de régles. In: EGC 2007. Dans la revue RNTI (Revue des Nouvelles Technologies de l'Information), numéro spécial (2007)