A case-based reasoning system independant of a representation of cases in terms of features

Sophie ROUGEGREZ
LAFORIA
Université Pierre & Marie Curie
4, place Jussieu
75005 PARIS
FRANCE
e-mail : rougegre@laforia.ibp.fr

Abstract

The main quality that we ascribe to the case-based reasoning is its capacity to resolve a problem even when the application area is badly formalized. The one in which we have interested doesn't even permit to represent cases per a set of important features.
The objective of our system is finding out an evolution from a given date. Matching between evolutions is realized according to a viewpoint. We have hence defined a distributed architecture in which each module "reasons per case" according to different viewpoints on the evolution.
The first originality of our work is the utilization of the case-based reasoning for this kind of problems. The second one, and the most important, is a case representation independent from the analysis of the important features of the case. This one has naturally consequences for the matching technic.That's specifically on this point that our approach is related to techniques being utilized by IRS (Information Retrieval System).

INTRODUCTION

The use of case-based reasoning is especially interesting for domain areas in which the knowledge required for the resolution is insufficient or imprecise. The reasoning steps are based on a case representation under the shape, most often, of a problem statement and the solution associated with it. This reasoning has been used in many areas : cookery [Hammond1990], [Kolodner1987], law [Rissland and Ashley1986], planification [Alterman1986], [Klein *et al.*1988], [Hammond1990], etc.

The different applications realized use a case representation based on the analysis of the important features of the case. Our experience in the conception of a system permitting to determine the follow-up of a process evolution shows that case-based reasoning can be utilized independently of this representation. Our system utilizes a retrieval method of the most interesting case based on the localization of an event sequence in another sequence. This one makes it thus rather related to information retrieval systems.

We describe more precisely the problem proposed in section I : the determination of a process evolution and justify the CBR use. In section II, we detail the retrieval method of the most interesting case. We proceed in the third section by our case description and in section IV by the presentation of one component of our system. We shall terminate in section V by a comparison between our work and IRS.

Problem proposed

We wish to be able to predict the follow-up of an evolution. These ones can be described the following way : a dynamic process whose progress is punctuated with the occurence of events whose we ignore the nature and quantity. They are described by a "trigger" event and the different events that follow it.

An event results from one or several others. Among all the evolutions having these characteristics, we have especially interested ourselves to evolutions for which we have no knowledge about the links between the diverse events which make it up.

A forest fire, an illness are illustrations of this. A forest fire starts and spreads under the action of a high number of environmental factors whose values change in the course of time. Their combination affect the propagation. But we ignore how.
The evolution of an illness results from the successive values of several factors. But we don't know their respective influences.

A fire jump, a change of wind direction, a fall in the blood pressure of a patient... are examples of events which describe, in part, an evolution. From the description of an event succession, the problem to resolve is the description of the events which are going to follow. The use of "classic" reasoning methods as well as expert systems are not considerable.

The approach that we have chosen is based upon the following observation : an event occurence in an evolution is not caused by the hazard. Each of them happened because other events occured before him. Then, if two evolutions undergo the same events, during a given interval, they will know the same events afterwards. That's the reason why we have opted for case-based reasoning.

A best case selection independant of a representation of cases in terms of features

The objective of our CBR is being able to predict the follow-up of an evolution, based on finished evolutions. The hypotheses produced are events too. A case consists thus in the representation of an evolution, described in terms of events having punctuated its "life".

To complete the description of the "target case", the one of which we search for the follow-up of a given event succession, our system searches for a "source case", the description of a finished evolution, which has experienced the same event sequence at a given time. The solution consists then in the set of events that it has experienced afterwards.

The insuffiency of knowledge about the considered domain area doesn't permit the assignation of any importance to events. Our matching algorithm has thus to consider the totality of events describing an evolution, on the assumption that they are all of equal importance. Our cases describing a given evolution, we can't index them. Our reasoning has then to evaluate a similarity with each of the cases in memory. This one is based on the localization of a sequence in another one.

The localization of an event sequence in another one evokes string matching methods. The definition set of the elements being compared has to be finite. But the set of conceivable events is potentially infinite. Such is the case of events describing forest fires. But that is right equally for the medical area.

The second condition on the utilization of a string matching method is the possibility of evaluating the result of the comparison of any two elements of the definition set. But in the evolutions that we have specifically studied, the forest fires, which do besides the object of our system, events are related to changes of the parameter values, like the one of the wind direction, of the air temperature... What can we say about their comparison ?

In the frame of "accurate" string matching methods, the comparison between "the wind direction at 4:00 pm becomes NW " and "the temperature at 5:00 pm becomes 32 degrees" produces for example the result "false" because the two events are different. But given the multiplicity of events which can arise in an evolution, it is then very little likely to find two event sequences rigorously exact. The necessity of using an approximate comparison appears clearly.

We define thus a distance between two events which will be used as a basis for the evaluation of a distance between two event sequences. A distance between two events is evaluated if the two events are comparable. Now, we have defined that two events are comparable if they are related to the change of value of a same entity. For example, the different wind directions are related to the entity "wind".

In the frame of forest fires, we have thus divided up events into categories corresponding to the kinds of parameters. The different winds have a comparable effect, the different vegetations ran along too. The comparison of two events linked to different categories , like the two events above, has thus no sense.

For that reason, we compare two evolutions by considering only one category of events. Among the categories of events that we have identified, some ones are going to be utilized to compare two evolutions according to a viewpoint. In the frame of case-based reasoning, the selected case is the one which best corresponds to the target case, according one viewpoint at least.

The retrieval method of the most interesting case is thus the following :
given a follow-up of events, relating to the evolution E, which occured between instants t_0 and t, search for an evolution E' such as there exists two instants t'_1 and t'_2, between which events that occured according to a viewpoint P, are similar to E between t_0 and t.

A viewpoint thus plays the part of an event filter. But in our case representation, events associated with

different viewpoints are straightaway separated.

Case representation

The case representation is always very dependent upon the domain area. But the representation of evolutions, whatever the domain area considered, forest fires or other, has to permit to represent an undefined number of events relating to a variable number of viewpoints.

We have chosen an object centered representation. A case is thus an object made up of instance variables containing the description of other objects.

An evolution is described by the events which happened during its development. In our system, an event designates a change of the value of a parameter at a given instant. The new value of the parameter is supposed to be valid until an other event related to the same parameter comes to contradict it. Some data not linked to a given instant are events too : a relief, a new vegetation ran along have an influence over the fire development, on the same account that a change of wind direction at a given date. We will see later that we can associate them with time in an indirect manner.

Our reasoning considers cases according to a viewpoint, that is to say according to an event category. To facilitate the extraction of event sequences relating to a given viewpoint, we separate events according to the event category to which they belong. Some of them will be used as viewpoints above an evolution. In one case, an event category is materialized by an instance variable. This instance variable contains the first event, in date, of the event sequence describing the viewpoint. Each event is then linked to its successor (see figure 1).

Our description contains too other instance variables relating to general information about the evolution such as the fire starting date, its duration, its starting conditions... Our system doesn't use these latest data. But we can envisage to exploit them in an other case-based reasoning.

The case representation, such as we described is sufficient if the case-based reasoning is limited to the extraction of cases in which we could localize an event sequence relating to a given viewpoint. But its aim is to generate hypotheses about the fire propagation. These ones are described by events of type "propagation" organized too, as shows the figure 1, in an event sequence associated to an instance variable of the case. The viewpoints utilized to select the best case are described in terms of relief, vegetation, and wind. Events conveying the reasoning results and those used to access to the good case are thus different. A mechanism permitting to link the second ones to the firsts is thus essential.

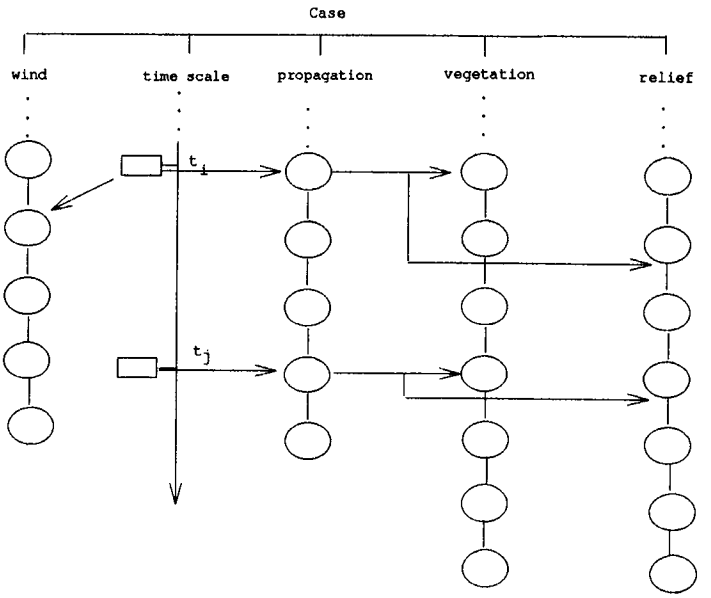

Figure 1: The case structure

This mechanism, that's time that implements it. Indeed, once we have found an event sequence looking like an other event sequence, the reasoning result is described by the events, here of kind propagation, which follow them. To be able to "navigate" between the event categories, we have implemented inside our cases a temporal scale. This one is an "instant" objects succession, letting know a date and an hour and a list of pointers to events which having arosen at this date and this hour. Instants are sorted out in chronological order and constitute so a linked chain whose first link is described by an instance variable "time scale" of the case.

As we already told, it is difficult to associate an instant to events "relief", "vegetation". Nevertheless, the description of fires done by forest fires associates to a place description reached by fire, at a given date, that is to say to a "propagation" event, the points associated in, respectively, the succession of relief and vegetation ran along. It is thus possible to know the date at which some "vegetation and relief events" have happened.

So our representation ensures us that each event can be linked to time because for each event E, whatever the considered viewpoint, there exists at least two events E_1 and E_2 such that :
- E_1 is a predessor of E and E_2 is a successor of E,
- E_1 and E_2 are associated to an instant of the time axis.

We below describe the case-based reasoning steps relating to the relief viewpoint.

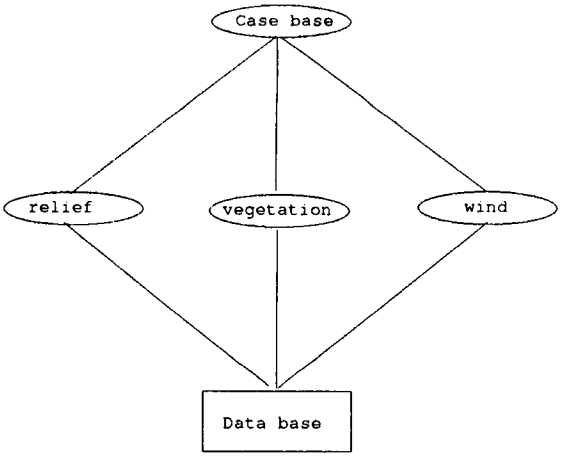

Figure 2: System architecture

The system

This one is applied to forest fires. Its objective is thus being able to predict a running fire evolution.

The consideration of an evolution according to a viewpoint requires a measure of the similarity of two event sequences, potentially different of the one of other viewpoints. Moreover, we think that it would be interesting that the hypotheses about the propagation relating to a viewpoint could be generated in parallel. Therefore we have associated to each one a module of a distributed architecture, constituting itself a case-based reasoning system (see figure 2)

Its working is ruled by the information that the system receives about the fire propagation. Each module associated to viewpoints generates, by access to a shared case base, the hypotheses about the future fire propagation. It adds then them to a data base shared by the set of modules. We describe now the running of the relief module.

Relief module

From an information about the fire propagation, this module searches for a fire in memory which at a given moment, has ran along the same relief, according to our similarity criterion that will be defined later, as the one ran along by the target fire since its outbreak. But this is not sufficient.

The relief that the fire runs along can indeed have long term effects on its propagation. For example, it may have accelerated it. But its effects can be immediate : a propagation stop caused by the descent of a strong slope, for example.

If we wish to generate hypotheses about the future

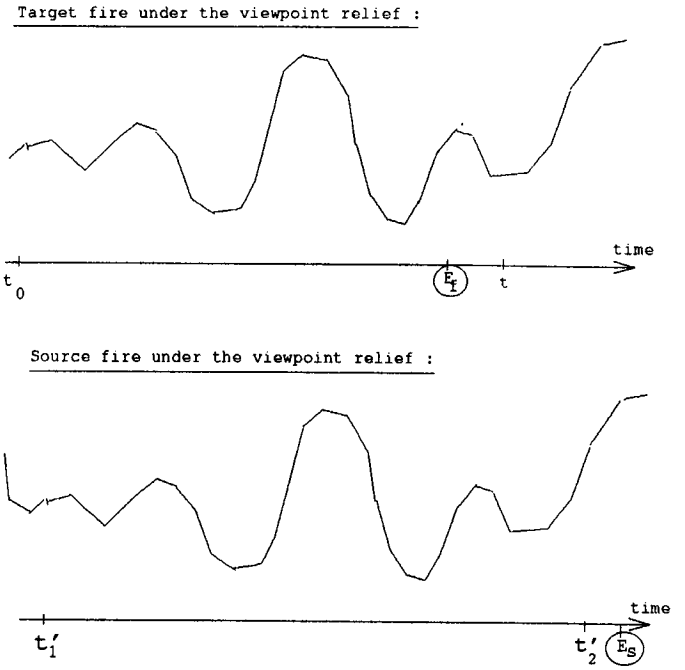

Figure 3: A comparison between two cases under the viewpoint relief

fire behavior, we have to consider, not only all the relief that it ran along, but the relief that it's going to run along, too. Therefore, we have to anticipate the relief that will be ran along.

The figure 3 presents the information utilized to calculate the propagation of the target fire.

E_f and E_s are events describing the fire propagation. The "target fire" is the fire which is occuring. We have described above the profile of the relief that it ran along since its outbreak, that is to say until the t instant. We have represented the "anticipated relief" too, represented here after the t instant. We have associated to the time axis events corresponding to the information that we have on the fire propagation. E_f is the latest event. In our representation, we separate the different kinds of information. As it is shown in the diagram, the time axis permits yet to link them.

The principle of our method is to retrieve a fire in memory which, under the relief viewpoint, is similar to the target fire. We search then for a fire which has ran along at a given moment the same relief : the source fire above has well went across a similar relief, between the instants t'_1 and t'_2. The event E_s has occured when the fire has gone across the relief we have anticipated : it results therefore of all the relief previously ran along. This is such an event that we are going to utilize to describe the hypotheses about the target fire propagation.

We descrive below the reasoning steps :
- transformation of the relief under a suited shape,
- retrieval of a fire having ran along the same relief,
- retrieval of informations about its propagation,
- transfer and adaptation of those informations towards the target fire,
- adding of the hypotheses carried out to the database.

1. Relief transformation
The relief called up in the fires is represented under the form of a curve, or a profile too. In our system, by contrast with the description of fires, as it is done by the experts, it consists into a succession of points faraway from each other of a certain distance and located at a given altitude. The relief module is going to try to retrieve a fire which has spread over the same relief. The mathematical methods grounded on curves differences are not useful here : theses ones consider a pixel follow-up. However, we try to compare slope successions. The likeness criterion of two profiles is indeed the sense of the slope (ascendant, descendant), the order in which they have been went across, and too the eventual pass over of relief accidents more complicated : such as a pass, a valley, etc.

From the numerical relief representation under a numerical form, we build a symbolic representation. This is equivalent to slipping the curve into a succession of segments, to associate to those a slope ; and if possible, to replace a slope follow-up by a "shape" : for example, a valley. We dispose then of two representation levels : slopes and shapes. The transformed relief will be represented thanks to these two levels.

In the experts language, we speak about "weak slope", "strong slope", etc. This vocabulary has led us to classify the set of slopes : to each group of slopes, we associate a representative, a prototype in form recognition [Simon1984]. We dispose hence of models of slopes, either ascendant, or descendant. The degree associated to each is symbolic. We have indeed used the multi-set theory [Akdag *et al.*1990] according to which an element belongs to a set to a certain degree.

A first transformation consists then in, from a set of points, generating slopes under the form of symbols to which we associate a degree. From these symbolic slopes, we try to generate forms, associated to a degree too. This latest is a vector constituted of degrees associated to slopes constituting the generated form.

2. Retrieval of a fire having ran along the same relief
Traditionaly in CBR, the selection of the case the most comparable to the source target is realized by the determination of a set of "candidate" cases, then among those the selection of the best one. We consider that all the cases are at the beginning candidate. We evaluate the likeness with the target case and only then we choose the best one.

```
ltarget <- symbol list describing the relief stemming from the target case
lsource <- symbol list describing the relief stemming from the source case
cpt <- 0
while ltarget and lsource are not empty do
   ftarget <- first (ltarget). ltarget <- rest (ltarget).
   fsource <- first (lsource). lsource <- rest (lsource).
   if ftarget and fsource describe the same kind of slope (ascendant
   or descendant) or the same kind of form then
      cost <- 0
   else
      search for a kind of slope or the kind of form the most like
      fsource and such that it is in ltarget.
      The found object is aux. We take it off from ltarget.
      cost <- the distance, in number of positions, from the aux
      aux position in ltarget + the "virtual distance" which exists
      between fsource and aux.
   end if
   cpt <- cpt + cost
end while
result <- cpt.
```

Figure 4: Matching algorithm under the viewpoint relief

Our matching algorithm is hence put in charge of matching lists of symbols representing slopes or forms, associated to a membership degree which can be a symbol, or a vector of symbols. For example : (ascendant slope, strong)-(descendant slope, weak)- (descendant slope, very weak), (valley, (weak, strong)).

The matching step compares then two symbol follow-up associated to a membership degree. Two follow-up look like each other if :
- they are constituted of the same elements,
- the elements follow each other in the same order.
These two matching criterion constitute two different methods, two viewpoints, which are going to be mutually utilized to evaluate this one. From a matching results then a couple of costs : the first value comes from the comparison between successions as sets of symbols. If the sets are composed of the same elements, the matching cost is null. The second one considers the positioning of the elements in the two successions respectively. We describe in figure 4 the corresponding algorithm.

The determination of the most similar slope or shape is realized from a graph whose values are symbols. Each value is related to the "most proximate" symbols per bow stamped with a "virtual distance" wich separate them. We have attributed those distances to bows in function of an evaluated similarity of their effects on the fire.

We dispose at this stage of a list of costs couples. As theses measures are done for each case in memory, there are as many as there are cases in memory. The best case is the one whose two costs are lower than the

costs of all the others. In the absence of a case presenting this feature, we'll take the one whose one of cost is lower than all the other costs, independently of the couple it belongs to.

3. Search of information about the propagation
The best case being chosen, we can proceed to the calculus of the fire evolution. This one is realized from information about the propagation of the source fire which has been selected. Events in a case being classified in chronological order, we extract from a case the following events : in this case the event E_s in the above figure.

4. Transfer and adaptation of events to the target case
Once these events have been selected, we have to adapt them to the target case. We consider that each event is represented in a plan guided by a time axis and a distance axis, whose origin corresponds to the location and starting date of the fire. The transfer and adaptation of an event consist then in a transposition of a guiding into another one.

5. Adding of the generated hypotheses to the database
The hypothetical events carried out describe the succession of the running evolution. All the data on the database are organized along the structure of a case defined in appendice. At every moment of the reasoning, the data base is then quite readable. To store the new case in memory, we only have to suppress all the hypothetical data.

There is another step in case-based reasoning systems that doesn't exist in all systems. It is the evaluation phasis. This one depends indeed on the available knowledge to validate the yeld results. When the reasoning is directed towards a single, or several goals, it is sufficient to verify that those are reached [Hammond1990]. But the domain area in which we are working doesn't permit an evaluation, by the system, of its own results. A comparison only between the prediction realized and the actual evolution is conceivable.

Related work

Our work is the first one in the case-based reasoning. which to our knowledge, permits searching the best case according to criterions independant of a representation of cases in terms of features. Our approach is thus rather similar to information retrieval systems which base their search of interesting documents on the presence of a string of characters, or terms...

Here, we search for an event sequence in another sequence. Comparison between two sequences is approximate. For that, we have used the notion of distance between two events. This one is based, for the relief module, on a predefined distance between the elements permitting to describe them. MEDLARS [Forsyth and Rada1986] uses equally this approach to evaluate the distance between two term sequences.

Our cases contain the complete description of finished evolutions. Those evolutions consist in events related, directly or indirectly, to time. Rather than describing an evolution per the succession of events which occured, as we describe a text by the follow-up of words that make it up, we have structured our cases according to the different event categories that we have identified. The object representation of our cases permits us to structure the events of different categories differently, which because of their nature, may need an ad hoc representation. This object structuration permits us equally to select cases by a method local to each viewpoint. This representation has equally been utilized in some IRS to represent data of different kinds. The views in [Anick *et al.*1991] may lead to the selection of a data base part according to a criterion related to the wanted type of document. The perspectives in [Tissen1991] consist in the selection, not of documents as previously, but in a part of each document. It equates to the selection done by our viewpoints.

Whereas the goal of an IRS is to present all the documents related to a given subject or containing a given piece of information, case-based reasoning systems search for the case the most similar to the problem proposed. The fact that we are unable to realize a global matching, we utilize at most one case according to a given viewpoint. The different predictions realized during our system working result then in the set of all selected cases according to a similarity, to a given event sequence, criterion.

Finally, an IRS presents the union, in a mathematical sense, of the sets of documents relating to one at least of the request terms. Our system utilizes the best case relating to one of the viewpoints. But as the evolution occures, other cases can be selected because the event sequences that we utilize as search criterion change with the same rythm. A prediction is only a phasis of our system working. But in IRS too, we move towards a dynamical request formulation, done by the user, or automatically with the help of a thesaurus [Gaugh1991].

CONCLUSION

The determination of the follow-up of a succession of data has already been the object of research. The ones of [Dietterich and Michalski1989] for example, are based on the detection of regularities among the provided data.

The determination of evolutions, much more general, has constituted the target of our research. Our aim was to conceive a system being able to determine its

"future". An evolution is described thanks to a set of events about which we have little information : we don't know how they interact and then how they influence the evolution. We only know only that they are of different kinds.

We have introduced the notion of viewpoint on an evolution : we do hypotheses on its future along an event category. Hence, we have defined a distributed architecture in which each module reasons per case according to a viewpoint of the evolution.

Our approach is original because each of our cases constitutes a direct and integral transcription of the evolutions representations on which we have based our system. It has yet an inconvenient which is the necessity to consider all the cases. But the use of an index requires a case interpretation in terms of features. Given the lack of knowledge about the considered area, this one would have all the chances to be erroneous. That's what we firmly wanted to avoid.

We study now the feasability of the combination of hypotheses generated by each of the modules. That's not an easy thing to do and its justification is not obvious because the events we use to build evolution hypotheses result "intrinsically" of the influence of all the kinds of events. Moreover we obtained them by considering only one of them. In short, we should retire from hypotheses carried out all the influence of events which don't have been voluntarily considered. An other solution, much more reasonable, would consist in controling the choice of cases by each module : if we succeed in finding a case which is the best along all the viewpoints, the problem is actually resolved.

References

[Akdag *et al.*, 1990] H. Akdag, M. De Glas, and D. Pacholczyck. Towards a qualitative theory of uncertainty. Technical Report 8/90, LAFORIA - Univ. Paris 6, 1990.

[Alterman, 1986] R. Alterman. An adaptative planner. In *Proceedings of AAAI*, pages 65–69, 1986.

[Anick *et al.*, 1991] P. G. Anick, R. A. Flynn, and D. R. Hanssen. Adressing the requirements of a dynamic corporate textual information base. In *Proceedings of ACM/SIGIR*, 1991.

[Dietterich and Michalski, 1989] T. G. Dietterich and R. S. Michalski. Learning to predict sequences. In *Machine learning*, volume II. Morgan Kaufman, 1989.

[Forsyth and Rada, 1986] R. Forsyth and R. Rada. *Machine learning : applications in expert systems and information retrieval.* Ellis Horwood series, 1986.

[Gaugh, 1991] S. Gaugh. Evaluation of an expert system for searching in full text. In *Proceedings of ACM/SIGIR*, 1991.

[Hammond, 1990] K. J. Hammond. Case-based planning : a framework for planning from experience. *Cognitive science*, 14:385–443, 1990.

[Klein *et al.*, 1988] G. A. Klein, L. A. Whitaker, and J.A. King. Using analogues to predict and plan. In *Proceedings of the second workshop on CBR - Pencola Beach, FL*, 1988.

[Kolodner, 1987] J. L. Kolodner. Extending problem solver capabilities through case-based inference. In *Proceedings of the 4th international workshop on machine learning*, pages 167–178, 1987.

[Rissland and Ashley, 1986] E. L. Rissland and K. D. Ashley. Hypotheticals as heuristic device. In *Proceedings of AAAI*, pages 289–297, 1986.

[Simon, 1984] J. C. Simon. *La reconnaissance des formes par algorithme.* Masson, 1984.

[Tissen, 1991] A. Tissen. A case-based architecture for a dialogue manager for information-seeking processes. In *Proceedings of ACM/SIGIR*, 1991.