

KnowMan – Een Case-Based Reasoning Tool

Software Review

L.A. Plugge

Onlangs kreeg ik ter evaluatie het softwarepakket KnowMan van de firma Intellix toegestuurd. KnowMan is een case-based reasoning tool, waarmee beslissings-ondersteunende systemen gemaakt kunnen worden die ook op Internet te gebruiken zijn.

Aangezien lezers van *Psychologie & Computers* zich veel bezighouden met het nemen van beslissingen op basis van casussen, en bovendien bezig zijn om de mogelijkheden van Internet te ontdekken, leek mij dit een interessant pakket om te bekijken.

Case-Based Reasoning

Er zijn verschillende manieren om problemen op te lossen met kennissystemen. Een bekende manier is *rule-based*, waarbij ieder geval door een set regels wordt beoordeeld. Die regels worden verkregen van een domein expert en ingebracht in het systeem.

Een andere manier is *case-based*. In dit geval krijgt het systeem geen kant en klare regels ingevoerd, maar voorbeeld casussen. Stel dat het systeem het verschil moet leren tussen vogels en vliegtuigen. In dat geval moet het systeem een aantal voorbeeld casussen voorgeschoteld krijgen met daarin discriminerende kenmerken. In Tabel 1 staat een triviaal voorbeeld: het verschil tussen een vliegtuig en een vogel. Uit het voorbeeld blijkt dat het kenmerk Vleugels niet discrimineert, want zowel vliegtuigen als vogels hebben vleugels. Het kenmerk Motor discrimineert evenmin, want er zijn vliegtuigen zonder motor, namelijk zweefvliegtuigen. Vandaar dat ik die kenmerken heb doorgestreept. De twee overblijvende kenmerken, Bek en Veren, discrimineren beide even goed. Uit de casussen 4 en 5 blijkt tevens dat één van de twee kenmerken voldoende is om de conclusie Vogel te kunnen trekken. Eén kenmerk zou in dit geval volstaan.

Tabel 1 Vliegtuig-Vogel casussen

Casus	Bek	Vleugels	Motor	Veren	Diagnose
1	False	True	True	False	Vliegtuig
2	True	True	False	True	Vogel
3	False	True	False	False	Vliegtuig
4	*	*	*	True	Vogel
5	True	*	*	*	Vogel

Uit Tabel 1 kun je de volgende regels destilleren:

Table 2 Vliegtuig-Vogel regels

ALS X een bek heeft OF X heeft Veren DAN: X = Vogel	ALS X geen bek heeft EN X heeft geen Veren DAN: X = Vliegtuig
---	---

De taak voor de ontwerper is het om te zorgen dat de juiste casussen met de juiste, i.e. discriminerende, kenmerken worden verzameld. Waar geen enkele vorm van redeneren mee overweg kan zijn casussen met dezelfde kenmerken, maar met verschillende diagnoses. Spijtig genoeg is dat binnen de psychologie natuurlijk wel vaak het probleem. Evenals bijvoorbeeld de psychiatrie. Afhankelijk van de beoordelaar word je in hokje X of hokje Y gestopt. Of erger nog, bij één en dezelfde beoordelaar. Bij dat soort casussen zul je dus aan moeten geven dat je het antwoord niet weet. Als je beweert het wel te weten, dan gebruik je een verborgen kenmerk. Wat statistische analyse kan hier uitkomst bieden.

KnowMan

Met KnowMan kun je dus een kennissysteem bouwen dat werkt volgens het hierboven geïllustreerde Vliegtuig-Vogel principe. KnowMan is zelfs in staat enige statistische analyse uit te voeren op de data om statistisch relevante kenmerken te zoeken. Maar daarover later meer. KnowMan werkt volgens als een zogenaamde S.O.U.L. (Self-Optimising Universal Learner). S.O.U.L. is een combinatie van Neurale netwerk technieken en expertsystemen.

KnowMan bestaat uit drie delen: KnowMan Designer, KnowMan Panorama, en KnowMan Guide. De eigenlijke ontwikkelomgeving is KnowMan Designer. (Zie Figuur 1) KnowMan Panorama (zie Figuur 3) en KnowMan Guide (zie Figuur 4) zijn bedoeld voor de gebruiker om het kennisbestand te gebruiken. De drie onderdelen lopen naadloos in elkaar over.

KnowMan Designer

De eerste opgave om een beslissingsondersteunend systeem te bouwen is het vinden van een geschikt onderwerp zoals het voorbeeld met Vogel-Vliegtuig. Om dat te vinden is gewoon goed onderzoek nodig. In tegenstelling tot *rule-based* systemen moet je bij KnowMan beschikken over casussen waaruit KnowMan dan zelf regels destilleert.

De eenvoudigste manier om een kennisbank te maken is het opzetten van een spreadsheet met daarin rijen met casussen, zoals in Tabel 1. De kolommen van iedere casus bestaan dan uit de waarde van de kenmerken. De laatste kolom bestaat uit de conclusie of diagnose, i.e. dat gene wat geadviseerd moet worden.

Als je dat gedaan hebt, dan kun je de spreadsheet importeren via een *tab delimited*¹ of een *comma delimited*² bestand. Designer maakt dan automatisch een compleet kennisbestand aan. Als er waarden ontbreken in de spreadsheet, dan moeten die aangegeven worden met een asterix (*). Let wel, KnowMan ziet dat niet als *missing value*, maar als *irrelevante* waarde voor de conclusie die bij die casus hoort. Het aardige is, dat je dus casussen kunt gebruiken met daarin *missing values*, mits er maar een conclusie aan de casus is verbonden.

Om KnowMan te testen is het verstandig om wat interessante casussen achter de hand te houden. Als je dat niet doet, dan test je namelijk alleen of het systeem casus x kan opzoeken. Om die reden is in het Designer mogelijk om casussen te deactiveren. Een gedeactiveerde casus wordt niet gebruikt om tot een oplossing te komen.

De volgende stap is om het kennisbestand te laten analyseren door Designer. Je kunt zien hoeveel regels er worden gebruikt, welk percentage de voorbeeldcasussen beslaan van de verwachte totale populatie, en nog wat extra zaken. KnowMan heeft ongeveer 10% nodig

¹ Waarden gescheiden door tabs.

² Waarden gescheiden door komma's.

van alle mogelijke casussen om tot een betrouwbare uitspraak te kunnen komen. Dat percentage hangt samen met het aantal kenmerken en hun discriminerend vermogen.

Een interessante bijkomstigheid van Designer is de mogelijkheid om een optimalisatie uit te laten voeren.

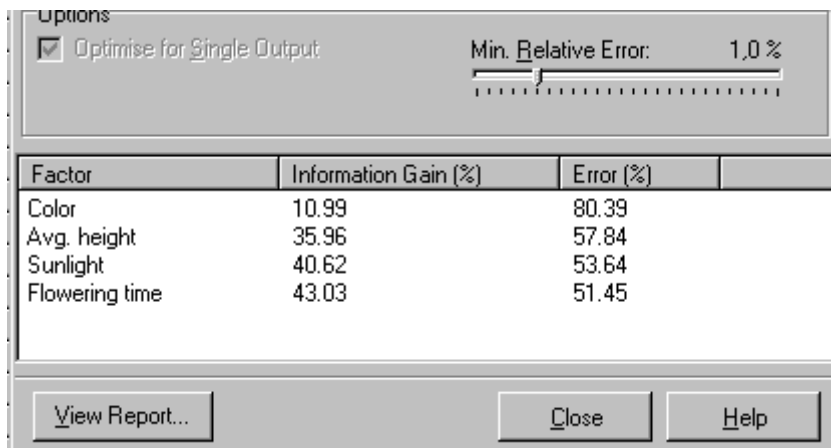
#	Life span	Color	Flowering time	Avg. height	Sunlight	Result
1	perennial	red	spring	12	much	Berginia
2	perennial	blue	spring	12	much	Virginia bluebell
3	perennial	red	summer	12	much	Bleeding heart
4	perennial	*	summer	6	fair	Violet
5	annual	white	spring	6	poor	Candytuft
6	perennial	violet	spring	6	fair	Ajuga
7	annual	violet	spring	24	fair	Honesty
8	perennial	*	summer	30	poor	Delphinium
9	perennial	violet	summer	12	poor	Lavender
10	perennial	red	summer	30	*	Daylily
11	perennial	yellow	summer	30	*	Daylily
12	annual	white	summer	4	poor	Alyssum
13	perennial	red	summer	30	poor	Peony
14	perennial	white	summer	30	poor	Peony
15	perennial	yellow	summer	24	poor	Peony
16	perennial	red	summer	8	poor	Dianthus
17	annual	white	summer	6	poor	English daisy
18	perennial	yellow	summer	30	poor	Yarrow
19	annual	yellow	summer	9	poor	Gazania
20	perennial	*	summer	24	poor	Columbine
21	perennial	*	summer	24	fair	Columbine
22	perennial	blue	summer	30	poor	Japanese iris
23	perennial	white	summer	30	poor	Japanese iris
24	perennial	blue	spring	10	much	Lungwort
25	perennial	yellow	summer	30	poor	Black-eyed susan
26	perennial	*	autumn	15	poor	Hardy Aster
27	perennial	red	summer	18	fair	Cranesbill
28	perennial	blue	autumn	30	fair	Lobelia
29	perennial	red	autumn	20	fair	Lobelia
30	perennial	white	autumn	36	fair	Japanese Anemone
31	perennial	red	summer	30	poor	Poppy

Figuur 1 KnowMan Designer Interface

Die optimalisatie stelt de gebruiker in staat om te zien welke kenmerken het meest bijdragen aan de classificatie, of diagnose. In Figuur 2 is een voorbeeld te zien met kenmerken van tuinplanten. Het kenmerk Kleur levert nog een foutenmarge op van 80%, maar met Avg. Height er bij wordt dat verminderd tot 58%. Bovendien wordt één kenmerk als irrelevant aangemerkt, namelijk Life Span (niet zichtbaar in deze figuren).

Tijdens de optimalisatie wordt het volgende uitgevoerd:

- **Kruisvalidatie:** zowel vertikaal als horizontaal, om vast te stellen welke factoren de meeste informatie opleveren en om een minimum set van factoren te selecteren met de hoogste informatietoename;
- **Ruisonderdrukking:** het verwijderen van data die geen informatie toevoegen, op die manier wordt de complexiteit van het systeem tot een minimum beperkt;
- **Deactiveren van factoren:** al die factoren en waarden die geen extra informatie opleveren worden gedeactiveerd
- **Fine Tuning:** het onderliggende neurale netwerk wordt fijner gekalibreerd. Volgens Intellix levert dat gewoonlijk een verbetering op van 3-7%.

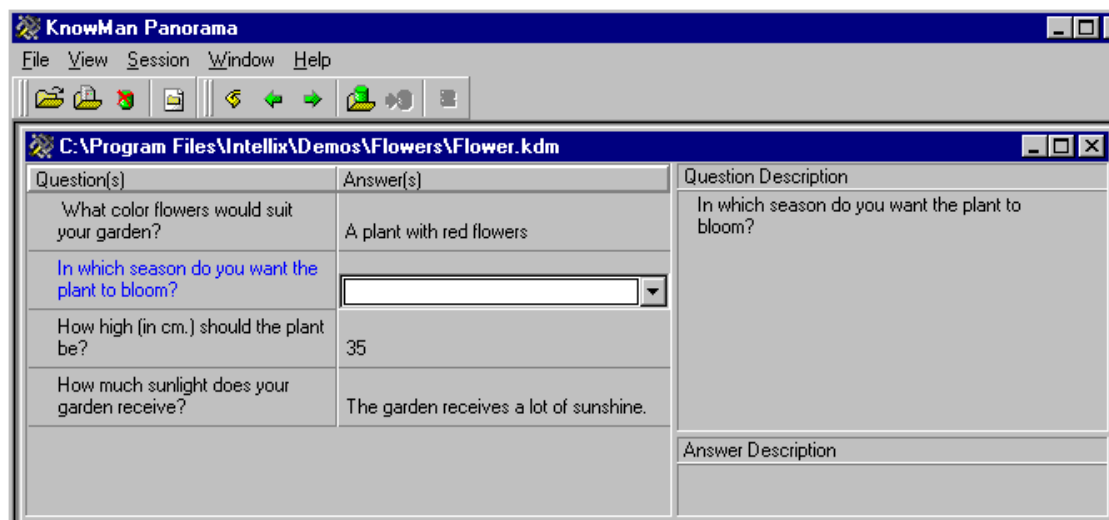


Figuur 2 Optimalisatie

Als dat allemaal gedaan is en de kennisbank is bewaard, dan kan het testen beginnen.

De Panorama Interface

Het testen gebeurt via de Panorama Interface. (Zie Figuur 3) De naam Panorama slaat waarschijnlijk op het feit dat je alle factoren tegelijk kunt bekijken.



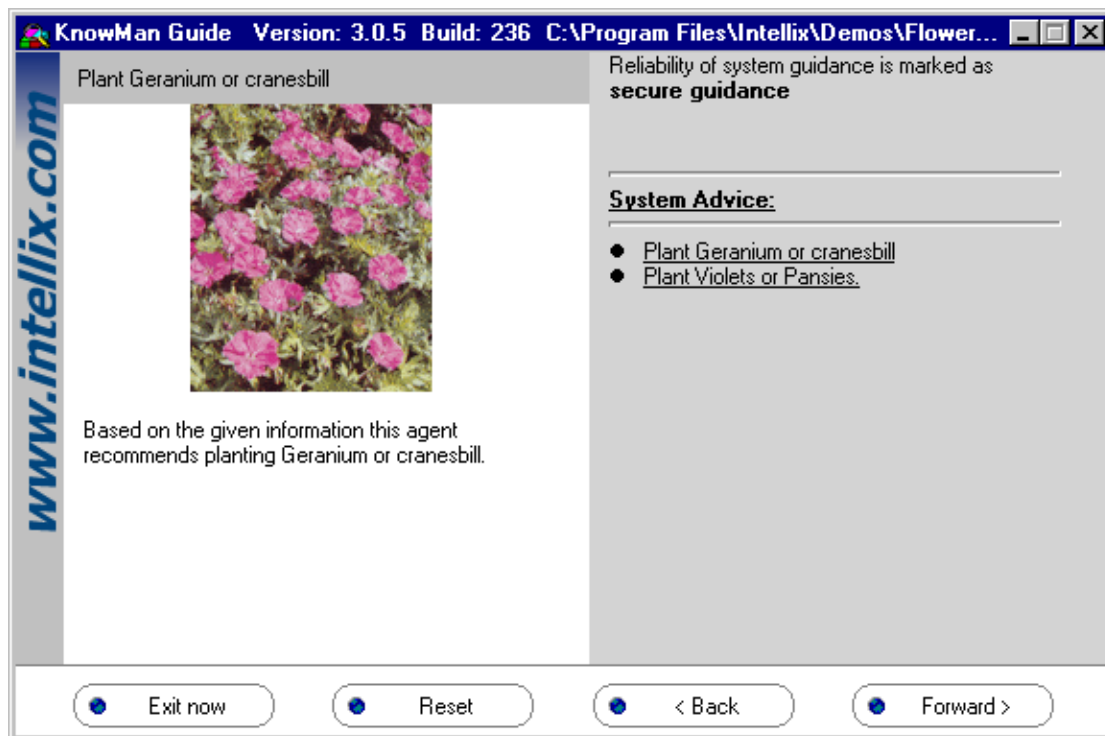
Figuur 3 KnowMan Panorama Interface

In de Panoram Interface kun je zelf bepalen welke gegevens je als eerste wilt geven. Je kunt die beslissing ook overlaten aan KnowMan zelf. Op basis van de hoeveelheid informatie die een gegeven oplevert, maakt KnowMan zelfstandig een selectie. Panorama biedt de mogelijkheid om direct te zien welke resultaten beschikbaar zijn (niet zichtbaar in Figuur 3). Je kunt dus direct de door KnowMan geselecteerde resultaten zien en wat hun waarschijnlijkheid is. Voor een testomgeving is erg prettig.

De Guide Interface

Hoewel de Panorama Interface ook geschikt is voor de eindgebruiker, is de Guide Interface een stuk eenvoudiger om door een leek bedient te worden. Het aardige is dat deze interface ook op Internet (of een Intranet) te gebruiken is. Dat betekent, dat iedereen in staat is om een consultatiepagina te presenteren. Aangenomen dan dat de

gebruikte data valide zijn. In Figuur 4 is een voorbeeld te zien op basis van het bloemenvoorbeeld van Intellix. Iedereen die geïnteresseerd is in andere voorbeelden kan daarvoor terecht bij de Internet pagina's van Intellix: <http://www.intellix.nl/>.



Figuur 4 KnowMan Guide Interface

Systemeisen

Voor de ontwikkelomgeving van KnowMan zijn de systeemeisen niet helemaal duidelijk, maar ik zou zeggen: een Intel machine met Windows 95, 98 of NT met een ruime hoeveelheid geheugen, bijvoorbeeld 32 Mb; een harde schijf van 1 Gb. KnowMan neemt natuurlijk niet zoveel plaats in, maar met alle andere gangbare software erbij moet je voldoende opslagcapaciteit hebben. Bij mij liep het lekker op een 233 MHz machine met 64 Mb geheugen en Windows NT workstation. Samengevat heb je voor KnowMan een simpele PC nodig die tegenwoordig tussen de wasmachines en de magnetrons staan.

Conclusie

KnowMan is een interessant instrument om beslissingsondersteunende systemen te maken als je beschikt over een goede set casussen. Het is bovendien erg eenvoudig in het gebruik. Voor simpele problemen kun je vrijwel direct aan de slag als je de helpbestanden goed leest. Voor wat meer complexe problemen is het misschien aan te bevelen een cursus te volgen bij Intellix.

Meer informatie:

Intellix BV

Dr. Lelykade 22A

Postbus 84037

2508 AA Den Haag

tel: 070- 4161744

fax: 070-4161719