

Een tool voor syntactische patroonherkenning voor de Université Joseph Fourier, Grenoble

ir. Marcel J.A. Peek

In dit artikel wordt de ontwikkeling van een tool voor syntactische patroonherkenning, met name handschrift herkenning, beschreven. Dit project werd van juli 1992 tot en met november 1992 werd uitgevoerd door een viertal cursisten van de ontwerpersopleiding Technische Informatica aan de Technische Universiteit Eindhoven voor de Université Joseph Fourier te Grenoble. Deze software ontwikkeling is uitgevoerd zowel in het kader van het ECMI-uitwisselingsprogramma (European Consortium for Mathematics in Industry) als in het kader van een curriculum blok 'systeemtechniek' van de ontwerpersopleiding.

Introductie

'Machine Intelligentie' zal een zeer dominante rol spelen in de jaren 90. Een belangrijk deel van zulke intelligente systemen omvat het gebruik van patroonherkenningstechnieken, die vaak gebruikt worden voor *data preprocessing* en *classificatie* (dit is decision making).

Een van de leuke kanten van patroonherkenning (PH) is dat het zich over meerdere disciplines uitspreidt. PH-technieken worden onder andere gebruikt bij radarsignaal classificatie (defensie, luchtvaart), computer vision (productiebesturing), handschrift herkenning (PTT, bankwezen), vingerafdruk herkenning (Interpol, CRI) en electrocardiografische signaal analyse (medische wereld). Er zijn raakvlakken met bijvoorbeeld automatentheorie, image processing, computer graphics en signaaltheorie. Er zijn grofweg drie klassen van PH-technieken te onderscheiden.

- Statistische PH.
- Syntactische of structurele PH.
- Neurale PH.

Binnen de ontwerpersopleiding Technische Informatica werd afgelopen zomer de mogelijkheid geboden een cursus 'Syntactische Patroonherkenning' te volgen. Deze cursus werd in het ECMI-kader gegeven door prof.dr. C. Bellissant van de Université Joseph Fourier te Grenoble. Bellissant is expert op het gebied van image processing en patroonherkenning, met name in het medische applicatiedomein. Men denke bijvoorbeeld aan celherkenning en image restauration.

Als afronding van deze cursus werd ons gevraagd een tool voor syntactische patroonherkenning te ontwikkelen. Dit tool moest onder andere faciliteiten bieden waarmee het mogelijk moest zijn grafische en tekstuele lineaire patronen in te voeren, op te slaan en op grond daarvan met een aantal geselecteerde leer-algoritmen (*linear inference* algoritmen) het tool te trainen. Daarna zou een gebruiker een nieuw patroon in moeten kunnen voeren en kunnen classificeren met behulp van het tool. Het uiteindelijke doel van dit tool is een demonstratie instrument te worden voor colleges patroonherkenning aan de al genoemde Franse universiteit. Een hoofdeis aan het tool was dat het een aantrekkelijke en makkelijk te gebruiken grafische user interface zal moeten hebben, vanwege de demonstraties. Om het leereffect bij desbetreffende studenten zo groot mogelijk te laten zijn moet het tool duidelijk tonen wat er gebeurt tijdens het classificatieproces. Andere belangrijke eisen waren uitbreidbaarheid van het tool met betrekking tot de leer-algoritmen, een duidelijke user manual en uitgebreide technische maintenance manual.

Ontwikkeling

Het te ontwikkelen tool heeft verschillende ontwerpaspecten. Het tool kan namelijk opgedeeld worden in drie min of meer onderling onafhankelijke delen, namelijk de *grafische user interface* met bijbehorende in- en output functionaliteit, de te gebruiken *leer-algoritmen* plus het classificatieproces en ten laatste de tussenliggende *conversie* van de grafische informatie, plus de opslagstructuur, naar een tekstueel formaat ten behoeve van de leer-algoritmen. Deze drie delen zijn min of

meer parallel ontwikkeld waarbij de interface tussen deze delen van te voren vastgelegd was. Dit bleek zeer belangrijk, het uiteindelijke integratieproces heeft niet meer dan één werkdag gekost.

User Interface

Het te ontwikkelen tool moest kunnen beschikken over twee soorten invoer, namelijk tekstuele en grafische invoer van lineaire patronen. De tekstuele invoer was relatief eenvoudig te realiseren, voor de grafische invoer moest echter een toepassing bedacht worden. Uiteindelijk is gekozen voor een muisgestuurde grafische invoer waarmee het mogelijk is lijntekeningen in te voeren. Verder moet de user interface onder andere beschikken over het volgende.

- Een optie tussen grafische en tekstuele invoer.
- Een canvas voor de invoer van respectievelijk grafische en tekstuele patronen, plus de bijbehorende editing mogelijkheden van de patronen.
- Een panel waar de al ingevoerde grafische of tekstuele patronen in worden getoond in tekstueel formaat, plus de bijbehorende scroll mogelijkheden, waarbij de patronen ook grafisch getoond worden.
- Een aantal menu's, onder andere menu's voor het kiezen van een leer-algoritme, de in- en uitvoer naar file van de grafische of tekstuele patronen.
- Bij de keuze van het te gebruiken leer-algoritme moet ook gekozen kunnen worden of men met *vaste* dan wel *elastische* patronen wil werken.
- Tekstvelden voor de feedback, zoals het resultaat van de classificatie en het aantal toestanden van de onderliggende eindige automaat (zie de paragraaf over Leer-Algoritmen).

Een vast patroon is het patroon zoals men het invoert (bijvoorbeeld *aaabbcc*). Een elastisch patroon is hetzelfde vaste patroon ontdaan van opeenvolgende herhalende karakters (dus in dit geval

abc). Men kan daardoor een klasse van allerhande schalingen van eenzelfde patroon herkennen.

De uiteindelijke user interface wordt getoond in Figuur 1. Tijdens invoer van een grafisch patroon wordt on-the-fly de conversie berekend naar een tekstueel formaat. De corresponderende tekstuele patronen worden op het linker panel in de figuur getoond. Deze conversie kan desgewenst ingesteld worden, er is default gekozen om lijnstukken van 10 pixels te converteren naar één der 8 gebruikte karakters. Hierbij wordt het karakter bepaald door de richtingshoek van het lijnstuk. Elk der 8 karakters (A t/m H) beslaat dus één der 45° grote symmetrische segmenten.

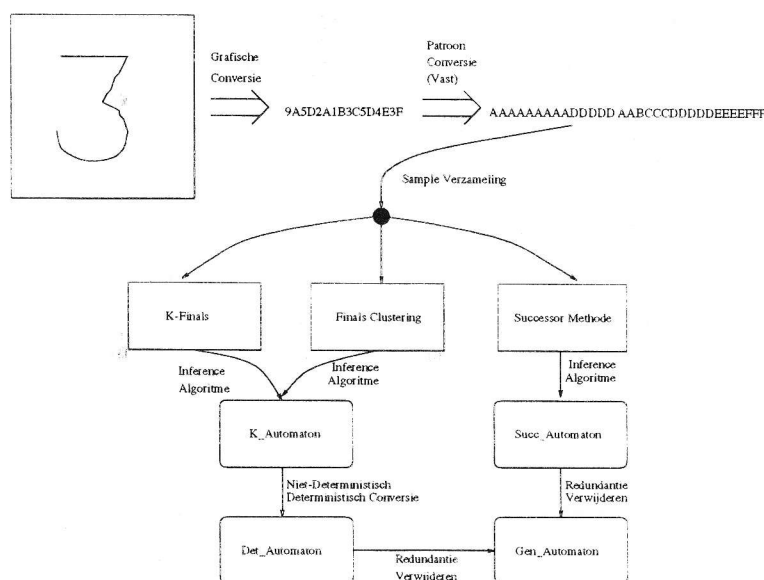
Tijdens de invoer van patronen worden deze, naast getoond te worden in het desbetreffende panel, ook opgeslagen in een zogenaamde patroonlijst. Deze lijst wordt ook gebruikt om in- en uitvoer te regelen naar file en voor de 'communicatie' met de leer-algoritmen. De uiteindelijke user interface is geïmplementeerd met behulp van de XView-toolkit en DevGuide¹.

Leer-Algoritmen

Het uiteindelijke trainen van het te ontwerpen systeem gebeurt, in dit geval, met *lineaire inference algoritmen*. De term lineair slaat op de één-dimensionale patronen die herkend kunnen worden, dit zijn patronen die als karakterstring gerepresenteerd kunnen worden. De bedoeling is om iteratieve regulariteit te ontdekken en uit te buiten. Een voorbeeld is een sample set $R^+ = \{a, ab, abb\}$ met terminalset $V_T = \{a, b\}$ te extrapoleren naar de taal $L = \{ab^n \mid n \geq 0\}$. De algemene lijn in dergelijke methoden, en ook de methoden die gekozen zijn voor het te ontwerpen tool, bestaat uit de volgende stappen.

1. Neem een eindige sample set R^+ en terminalset V_T en construeer een bijbehorende grammatica of eindige automaat die deze sample set precies als taal accepteert.
2. Extrapoler generieke karakteristieken (iteratieve regulariteit) van de patronen uit R^+ door middel van het structureel aanpassen (infereren) van de eindige automaat of grammatica.

¹DevGuide en XView zijn trademarks van Sun Microsystems, Inc



Figuur 2: De algoritmische laag en zijn omgeving.

Conclusie

Medio december volgde het vierdaagse bezoek per TGV aan de Universit  Joseph Fourier in Grenoble. Deze universiteit wordt bezocht door ± 45.000 studenten, verdeeld over alle α -, β - en γ -richtingen, die ongeveer 10% van de stadsbevolking vormen. De universiteit is   n der grootsten in Frankrijk en vervult een centrale plaats in Oost-Frankrijk. Tegen het jaar 2000 wil men de universiteit uitbreiden zodat er plaats zal zijn voor ± 60.000 studenten. De installatie en demonstratie van het tool verliepen zonder problemen. De demonstratie werd vooral gegeven om de user interface in zijn geheel te laten zien. De uiteindelijke acceptatietest werd uitgevoerd op de verscheidene algoritmes met ter plaatse bedachte testinvoer van de mensen aldaar. Sterke punten bleken de grafische user interface, het filemanagement met betrekking tot de eindige automaten, de robuustheid van de implementatie van de algoritmes, en de uitgebreide documentatie. Na onze demonstratie werden er demonstraties gegeven over enkele research projecten in de groep van Bellissant, onder andere een celherkennings-systeem voor het herkennen van bijvoorbeeld kankercellen, waarbij nauw wordt samengewerkt met het lokale medisch centrum. Een dag later werd het instituut ICP bezocht, een onderdeel van de Universit  Polytechnique in Grenoble, dat zich bezighoudt met onderzoek naar onder andere spraakherkenning/synthese en automatisch vertalen. Daar werden demonstraties gevolgd over spraakherken-

ningssystemen.

Concluderend menen wij een ervaring rijker te zijn, het was zeer de moeite waard om eens in de Franse universitaire wereld rond te kijken. Het was verrassend dat de groep waar wij een bezoek aan brachten een zeer diverse bezetting had, van biologen tot informatici, van wiskundigen tot medici. Tijdens de ontwikkeling van het tool, genaamd LIT (Linear Inference Tool), hebben de ontwerpers het zeer leerzaam gevonden diverse aspecten van het gehele ontwerptraject mee te maken in een *kleine* groep. Men denke aan user interfaces, projectmanagement, testen en documenteren. $\square$



Ir. Marcel J.A. Peek volgt op dit moment de ontwerpopleiding Technische Informatica.