HC 6, 18-12-2019, AI en recht (1)

Het ECHR-algoritme
Elk **voorspelalgoritme** krijgt input. Deze krijgt trainingsdata om een **statistisch model** te maken. Het ECHR-algoritme kreeg de volledige tekst van beslissingen sinds 1959 over art. 3/6/8 ECHR. Daarbij kreeg het als input de volledige tekst van 'Procedure' en 'Facts' over het ECHR. Daar werd bij vermeld dat het Hof over de schending van een van die artikelen heeft beslist. De output was volgens dat algoritme dat het Hof op geen of op tenminste één geschilpunt had beslist dat een artikel was geschonden.

In feite telde het **woordcombinaties**, hoe vaak die voorkwamen in jurisprudentie van het hof en bouwde zo een statistisch model dat de frequentie van woorden statistisch relateert aan 'de uitkomst'. Bij 79% van de zaken wordt door het ECHR-algoritme de 'uitkomst' correct voorspeld. Echter, de 'uitkomst' is niet de echte juridische uitkomst. Het Hof stelt per 'issue' vast of het verdrag is geschonden of niet. Het algoritme kan dat niet doen. Het zegt al 'ja, schending' als een van de issues een schending van het verdrag inhoudt. Daarbij is sprake van toeval. Bij de vraag 'is er schending van het verdrag?' is het antwoord 50/50 goed. Het Hof stelt sowieso in 84% van de zaken vast dat er sprake is van schending. Daarbij moet de uitspraak al grotendeels beschikbaar zijn. Het algoritme kan de uitkomsten juridisch nog niet uitleggen. Het telt alleen maar woordfrequenties. Het systeem maakt **geen juridische of morele afwegingen**.

Dit betekent niet dat het experiment totaal nutteloos is. Er is ook vervolgonderzoek gedaan. Er staat in dit onderzoek een interessante tabel over de meest relevante woordcombinaties. Bij 'schending' waren de voornaamste combinaties 'death penalty', 'that the applicant' en 'the public prosecutor'. Bij 'geen schending' waren dit 'the first applicant', 'district prosecutor office' en 'the urus martan'.

Een 'echt' voorspelalgoritme kan voorspellen of het Hooggerechtshof VS de beslissing van lagere rechter in stand houdt. Dit doet het op basis van de database over het Hof. Informatie over wie de rechters benoemden, wat de persoonlijke gegevens en politieke voorkeuren van rechters waren. Hier werd in 70% van de gevallen correct voorspeld wat de uitspraak zou zijn. Maar bij toeval is dit al in 50% correct. De meeste data gaan niet over inhoudelijke aspecten.

Voorspellen is iets anders dan beslissen. Rechters voorspellen niet, maar beslissen. Dit doen ze op basis van valide rechtsbronnen. Hun beslissingen zijn motiverend en de partijen zijn gehoord. De computer kan dit allemaal niet. De computer kan dit misschien wel ondersteunen.

Voorspellen door inhoudelijke analyse
Statistisch model op basis van automatisch herkennen van relevante factoren in de jurisprudentie valt ontzettend tegen. De computer is nog niet in staat om juridisch relevante feiten te vinden in teksten. Daarbij kan hij ze al helemaal niet begrijpen.

Geschiedenis van AI: cognitief vs. statistisch
AI betekent dingen doen die bekend staan als 'daar moet de mens intelligentie voor nodig hebben'. **Cognitieve AI** is het programmeren van expliciete modellen van aspecten van menselijke cognitie. Het probeert manieren van argumenteren en redeneren in te programmeren. Het brengt ook kennis die nodig is voor het oplossen van een probleem onder in de computer. Dit is een top-down aanpak. **Statistische, datagestuurde AI** betekent het automatisch leren van patronen uit data. Uit grote hoeveelheden data zoekt de computer zelf patronen. Deze twee stromingen wisselen elkaar door de tijd af, maar het interessantst is het wanneer beiden worden gecombineerd.

Voordeel van cognitief is dat het uitleg kan geven van hun uitkomsten. Het is dus transparant. Het blijkt wel heel moeilijk voor de mens om die kennis op betrouwbare manier in de computer onder te brengen. Bij statistische AI leert het zijn eigen kennis automatisch. Dit is wel weinig transparant. Alle AI is gespecialiseerd. Het modelleert geen algemene menselijke intelligentie.

Cognitieve AI (& recht)
Een heel belangrijk begrip hierbij is **expertsystemen** (ook wel **kennissystemen** genoemd). Het heeft aan de ene kant **kennis** over een bepaald probleemgebied en heeft aan de andere kant **redeneermechanismes** om vervolgens problemen op te lossen.

Geschiedenis
In 1950-1970 waren ze al bezig met het modelleren van algemene intelligentie. Bijvoorbeeld Newel & Simon's *general problem solver*. De ambitie zat er toen al in, al was het wat ambitieus. Tussen 1970 en 1990 gingen ze over op het modelleren van kennis en intelligentie van een expert op een beperkt gebied. Een voorbeeld hiervan is het MYCIN (diagnose en behandeling van infectieziekten). Dit bleek het heel goed te doen, beter dan menselijke artsen. Toch had het beperkingen. Het was namelijk moeilijk om betrouwbare kennis in de computer onder te brengen. Maar in het recht speelt dit probleem toch niet? Het recht staat immers in het wetboek en in standaardjurisprudentie. Dit blijkt toch niet zo simpel. Vanaf 1980 begonnen ze met juridische toepassingen. Je ziet twee stromingen binnen de cognitieve aanpak. Dit onderscheid is er, omdat er natuurlijk een verschil is tussen het *common law* systeem en het *civil law* systeem. Het project Taxman (Thorne McCarty, 1977) gaat over **precedenten**. *British Nationality Act* (Marek Sergot, 1985) gaat over **wetgeving**.

Het grote succesverhaal van AI en recht is simpel **regel gebaseerde kennissystemen**. Deze bleken erg nuttig, vooral bij grootschalige uitvoering van bijvoorbeeld bestuursrecht (vergunningen, etc.). Dit werkt vooral om regelgeving die voor de mens erg ingewikkeld is (veel regels en combinaties van regels). De computer is gemaakt om perfect logisch 'na te denken'. Waar de mens vaak regels over het hoofd ziet, doet een computer dit niet.

Juridisch syllogisme
Gegeven een rechtsvraag selecteer je de relevante rechtsregel (**major**), bepaal je de feiten van het geval (**minor**) en pas je de regel toe op de feiten. Dit naïeve standaardmodel moet wel uitgebreid worden. De regel moet namelijk **geïnterpreteerd** worden, de feiten **gekwalificeerd** en de feiten moeten **vastgesteld** worden. Het bewijsprobleem kan behoorlijk fundamenteel zijn.

Standaardarchitectuur
De kennis wordt opgeslagen aan de hand van productieregels. Dit zijn als dan regels met één of meer condities en één conclusie. Bijvoorbeeld: ALS sprake is van materiaal EN het materiaal is afval EN het materiaal wordt gedumpt, DAN is een vergunning nodig. Een manier waarop de machine kan redeneren is '**backward chaining**' (doel gestuurd redeneren). Bijvoorbeeld als doel: is voor iets een vergunning nodig? Hij gaat alle regels waar dit in voorkomt checken en voor alle condities van die regel of die vervuld kunnen worden. Dit kan door voor elke conditie een regel te zoeken met die conditie als conclusie. Als er niet zo'n regel is, kan ook aan de gebruiker gevraagd worden of die conditie geldt of gekeken worden in een gegevenskennisbank.

Voorbeeld: er is een regel over wanneer je een vergunning nodig hebt. (Voor het dumpen van afval heb je een vergunning nodig') Daarbij heb je twee interpretatieregels ('Iets is afval als…' en 'Iets is geen afval als…'). In feite bouwt het systeem een juridisch argument op. In de praktijk is dit erg succesvol.

Case studies Marlies van Eck
Vorig jaar is haar proefschrift verschenen. Dit ging over het volautomatisch afhandelen van verzoeken van burgers. Het ging om de bepaling van kinderbijslag en om de bepaling van iemands fiscaal inkomen. Ze vond hier een aantal problemen. Het lijkt alsof een systeem erg transparant is, maar in de praktijk bleek dit een stuk minder duidelijk. Zowel het systeem als de beslissing was **slecht uitlegbaar**. Er was slechte of ontbrekende **documentatie**. Er zijn altijd vage en abstracte termen die geïnterpreteerd moeten worden. De ontwerpers hebben die ad-hoc interpretatieregels dan zelf bedacht. Bijvoorbeeld als je niet meer dan twee weken te laat bent heb je een 'redelijk excuus'. Dit is natuurlijk dubieus. Een ander probleem is dat **bijzondere gevall**en niet in de standaardisatie passen. Daarbij zijn fouten of herzieningen moeilijk door de ketens heen te herstellen zijn. Vergelijk dit met Lessig's architectuur.

Voorbeeld: Er is een regel die luidt: 'geen voertuigen in het park'. Omtrent de feiten kunnen hier bewijsproblemen optreden. De algemene termen zijn ook lastig en regeltoepassing ook in verband met uitzonderingen. Juridisch redeneren is namelijk argumentatie.

Beperkingen van regel gebaseerde systemen
Regel: voor het dumpen van afval is een vergunning vereist. Feit: dempen van een sloot met puin. De interpretatie van de HR was 'wat naar algemeen spraakgebruik als afval wordt beschouwd is afval'. De kwalificatie van puin was 'wordt naar algemeen spraakgebruik als afval beschouwd'. De conclusie zou dan zijn: voor het dempen van een sloot met puin is een vergunning vereist. Echter, de interpretatie van de Kroon van de term 'afval' was 'wat niet meer gebruikt wordt'. Wanneer puin wordt gebruikt om de sloot te dempen, wordt het wel gebruikt. Dan is het dus geen afval en is geen vergunning nodig. Dit is een voorbeeld van **botsende interpretaties**.

Van Mesdagh had bedacht dat je met elkaar botsende regels in een systeem kunt opnemen en dat het systeem dan alternatieve argumenten voor en tegen de beslissing kon construeren. In dit geval kan het systeem zowel de interpretatie van de HR als die van de Kroon construeren. Het echte probleem kon niet echt worden opgelost, namelijk welke interpretatie het beste is. Het systeem kan niet écht **argumenteren** zoals een jurist. Daarbij moeten beide interpretaties moeten al **beschikbaar** zijn. Een mens moet die al hebben ingevoerd. Het systeem kan niet zelf de verschillende interpretaties bedenken. Er is dus absoluut geen sprake van creativiteit.

Automatiseren van routinebeslissingen
Frits Bakker wilde graag routinebeslissingen automatiseren. De voorspelalgoritmes zijn ongeschikt, omdat die niet naar redengevende verbanden, maar naar statistische verbanden kijken. Maar kan het dan met regel gebaseerde kennissystemen? Dit kan alleen als het probleem **klein en goed gedefinieerd** is. Daarbij moeten alle **feiten uit procesdossiers of databases** gehaald kunnen worden.

Fundamenteel
Redeneren gebeurt vaak **reden gebaseerd/ factor gebaseerd**. Er zijn vaak geen duidelijke regels, maar slechts redenen voor of tegen een conclusie. Redenen worden gewogen in rechtszaken en die worden dan precedenten. Maar hoe wegen de rechters die redenen en wat als een nieuwe zaak niet perfect matcht met een precedent?

Factor gebaseerde kennis: wettelijk (art. 9 Wbp)
De vraag is of je als je legitiem persoonsgegevens hebt gekregen hier iets mee mag doen wat niet precies binnen de doelomschrijving valt. Dan zegt de wet dat je persoonsgegevens niet mag verwerken op een wijze die onverenigbaar is met de doeleinden waarvoor ze zijn verkregen. Daarna noemt de wet een aantal voorbeelden van waar rekening mee moet worden gehouden bij het bepalen van onverenigbaarheid. De wet presenteert hier geen logische eenduidige regel, maar is redengevend.

Running example factors: misuse of trade secrets
HYPO was ontwikkeld door Ashley en CATO door Aleven. Hier ging het over het misbruik maken van bedrijfsgeheimen. Er worden verschillende factoren gegeven die meespelen in de vraag of er misbruik wordt gemaakt van bedrijfsgeheimen. Sommige factoren zijn pro misbruik en andere zijn con misbruik. Ashley en Aleven hebben de jurisprudentie doorgespit om erachter te komen wat nou belangrijke factoren waren op dit gebied. Voorbeelden pro: omkoping, beveiligingsmaatregelen, uniek product. Voorbeelden con: informatie 'reverse-engineerable' (je kunt zelf bedenken hoe je iets produceert als je goed naar het product kijkt), eiser in onderhandelingen al wat info had prijsgegeven, waiver of confidentiality en of de info daadwerkelijk reverse engineered was.

HYPO (Ashley en Rissland 1987-1990)
De kennis is vrij simpel. Van de precedenten over misbruik maken van bedrijfsgeheimen zijn de beslissingen bekend (p wint of d wint). Daarbij zijn de p-factoren (pro plaintiff) en d-factoren (pro defendant) bekend. Van een nieuwe zaak zijn de p- en d-factoren bekend. Het redeneermechanisme is niet de bedoeling om het geschil te beslissen, maar meer om **een debat te creëren**. Zodat advocaten

hier wat aan konden hebben als hulp. Als je een nieuwe zaak krijgt, ga je een precedent citeren op grond van gelijkenissen met de nieuwe zaak, die in jouw voordeel zijn. Dan kan de andere kant van het debat de citatie onderscheiden op grond van verschillen met de nieuwe zaak. De andere kant kan ook met een tegenvoorbeeld komen: een ander precedent met een andere uitkomst.

Voorbeeld met factoren
Stel een kennisbank bestaat uit twee precedenten C1 (P wint) en C2 (D wint). Binnen C1 heb je pro plaintiff factoren (P1, P2, P3) en pro defendant factoren (D1, D2). Binnen C2 heb je P1, D2 en D3. In de nieuwe zaak wil je beargumenteren dat de plaintiff (P) moet winnen. Binnen deze nieuwe zaak heb je factoren P1, P3 en D3. Als advocaat van P kun je dan aan de hand van C1 laten zien dat daar ook sprake was van P1 en P3. De gedaagde (D) gaat vervolgens onderscheiden door te zeggen: Ja, maar P2 was er ook en daarbij is er in deze zaak D3. Dit maakt de nieuwe zaak zwakker voor de eiser dan precedent C1. Bovendien kan de gedaagde een tegenvoorbeeld citeren. D gaat dan laten zien dat P1 en D3 overeenkomen. Vervolgens zal P weer onderscheiden door te wijzen op D2.

Voorbeeld Mason
Het taalvoorbeeld gaat over precedent Bryce en nieuwe zaak Mason vs. Jack Daniels. Nu is de vraag wat eiser Mason kan doen om te beargumenteren dat Jack Daniels misbruik heeft gemaakt van bedrijfsgeheimen. Er zijn in casu twee overeenkomende p-factoren. Vervolgens laat Jack Daniels de verschillen tussen de twee zaken zien. Bovendien is er een nieuwe d-factor aanwezig in de nieuwe zaak. Daarbij komt Jack Daniels met een tegenprecedent: Robinson. Hier was een d-factor aanwezig die ook in zaak Mason aanwezig is. Vervolgens laat Mason de verschillen zien tussen Robinson en Mason. De output van het HYPO-systeem is niet de uitkomst, maar het debat tussen Mason en Jack Daniels.

In de praktijk zijn factoren niet vaak wel of niet aanwezig. Ze kunnen in **meerdere of mindere mate aanwezig** zijn. Iets anders is dat **onderliggende waarden** in het geding kunnen zijn. Deze kunnen gebruikt worden om afwegingen te maken en wat kan het systeem daarmee? Als sprake is van een factor is er al een soort **juridische interpretatie** aan de gang. Dit in een systeem zetten kost daarbij erg veel geld en moeite.