Logica voor informatica

Matt ter Steege
matttersteege@gmail.com
Universiteit Utrecht
Utrecht, Nederland

Contents

1 Semiotiek

Semiotiek is het zelfde als 'tekenleer'. Denk hierbij aan:

- letters (A-Z)
- Karakters (chinees alfabet)
- Woorden
- Morsetekens/braille
- verkeerdborden, pictogrammern
- gebaren
- voorwerpen (witte vlag, e.d)

Semiotiek houd zich bezeig met elke activiteit, handeling of preces waarbij tekens worden gebruitk. Een teken defineren we als: "Alles dat een booschap communiceert van de zender naar de ontvanger"

1.1 De semiotische ladder

1.2 Fysieke tekens (Technische laag)

Dit houdt alles in van Klanken die je met je mond maakt tot gebaren, letters, Geuren e.d. Tekens kunnen op zichzelf al een betekenis hebben, maar kunnen ook met elkaar een betekenis hebben. Bijvoorbeeld: Het naam "Roderick" is een reeks van tekens (letters) die samen een betekenis hebben. Ofwel, de drager van de boodschap.

1.3 Het empirische niveau (Technische laag)

Dit niveau houd zich bezig met de waarneembare eigenschappen van tekens. Bijvoorbeeld: De letters in het woord "Roderick" hebben een bepaalde vorm, kleur, grootte e.d. Ook de klanken die je maakt als je het woord uitspreekt hebben bepaalde eigenschappen zoals toonhoogte, volume e.d. Dit niveau is vooral belangrijk voor de technische verwerking van tekens, zoals bij spraakherkenning of beeldherkenning. Ofwel, hoe de boodschap wordt overgebracht.

1.4 Syntax

Syntax is de studie van de regels die bepalen hoe tekens gecombineerd kunnen worden om grotere eenheden te vormen. Bijvoorbeeld: In het Nederlands is de volgorde van woorden in een zin belangrijk voor de betekenis. "De kat zit op de mat" heeft een andere betekenis dan "Op de mat zit de kat". Syntax is dus de structuur van tekens en hoe ze samenhangen. Ofwel, hoe de boodschap is opgebouwd.

1.5 Semantiek

Semantiek is de studie van de betekenis van tekens en hoe ze worden geïnterpreteerd. Bijvoorbeeld: Het woord "kat" verwijst naar een bepaald dier, maar in een andere context kan het ook een metafoor zijn voor iets anders. Semantiek gaat dus over de relatie tussen tekens en hun betekenis. Ofwel, wat de boodschap inhoudt.

Objectivisme: Beteknis is vast en objectief. Woorden hebben een vaste betekenis die niet verandert. Denk aan woordenboeken die de betekenis van woorden definiëren.

Constructivisme: Betekenis is subjectief en contextafhankelijk. Woorden kunnen verschillende betekenissen hebben afhankelijk van de context en de interpretatie van de ontvanger. Denk aan hoe straattaal of jargon verschillende betekenissen kunnen hebben in verschillende groepen. (bijv. "cool" kan zowel "koud" als "gaaf" betekenen)

Pragmatisme: Betekenis ontstaat in de interactie tussen zender en ontvanger. Woorden krijgen betekenis door het gebruik ervan in communicatie. Denk aan hoe de betekenis van een woord kan veranderen afhankelijk van hoe het wordt gebruikt in een gesprek.

Determinisme: Betekenis wordt bepaald door externe factoren zoals cultuur, geschiedenis en sociale context. Woorden kunnen verschillende betekenissen hebben in verschillende culturen of tijdperken. Denk aan hoe bepaalde woorden in het verleden een andere betekenis hadden dan nu. (bijv. "gay" betekende vroeger "vrolijk" maar nu wordt het voornamelijk gebruikt om seksuele geaardheid aan te duiden)

1.6 Pragmatiek

Pragmatiek is de studie van hoe tekens worden gebruikt in communicatie en hoe de context de betekenis beïnvloedt. Bijvoorbeeld: Het woord "kat" kan verschillende betekenissen hebben afhankelijk van de situatie waarin het wordt gebruikt. Als iemand zegt "Ik heb een kat", kan dit betekenen dat ze een huisdier hebben, maar het kan ook een uitdrukking zijn van iets anders, afhankelijk van de context. Pragmatiek gaat dus over de praktische aspecten van communicatie en hoe tekens worden gebruikt in de echte wereld. Ofwel, de bedoeling achter de boodschap.

1.7 Samenvatting

De semiotische ladder bestaat uit vijf niveaus:

- Fysieke tekens (Technische laag): De drager van de boodschap
- Empirische niveau (Technische laag): Hoe de boodschap wordt overgebracht
- Syntax: Hoe de boodschap is opgebouwd
- Semantiek: Wat de boodschap inhoudt
- Pragmatiek: De bedoeling achter de boodschap

2 Kennisbanken

Graafstructuren bestaan uit een verzameling van punten en knopen die met elkaar verbonden zijn door lijnen (ongerichte graaf) of pijlen (gerichte graaf). In een kennisbank worden concepten voorgesteld als knopen en de relaties tussen deze concepten als lijnen of pijlen. Kennisbanken kunnen worden gebruikt om informatie te organiseren, te structureren en te analyseren.

2.1 relationele database

Een relationele database is een type database dat gegevens opslaat in tabellen die met elkaar verbonden zijn door relaties. Elke tabel bestaat uit rijen en kolommen, waarbij elke rij een record vertegenwoordigt en elke kolom een attribuut van dat record (denk aan een spreadsheet). Relaties tussen tabellen worden gemaakt door middel van primaire en vreemde sleutels. Relationele databases worden vaak gebruikt voor het opslaan van gestructureerde gegevens en het uitvoeren van complexe queries.

2.2 SQL

SQL (Structured Query Language) is een programmeertaal die wordt gebruikt voor het beheren en manipuleren van relationele databases. Met SQL kunnen gebruikers gegevens opvragen, invoegen, bijwerken en verwijderen uit de database. SQL biedt ook mogelijkheden voor het definiëren van de structuur van de database, zoals het maken van tabellen en het definiëren van relaties tussen tabellen. SQL is een gestandaardeerde taal en wordt ondersteund door de meeste relationele databasebeheersystemen.

2.3 NoSQL

NoSQL (Not Only SQL) is een type database dat niet gebaseerd is op het relationele model. In plaats daarvan gebruiken NoSQL-databases verschillende gegevensmodellen, zoals documenten, grafieken, kolomgeoriënteerde opslag en sleutel-waardeparen. NoSQL-databases zijn ontworpen om schaalbaarheid, flexibiliteit en prestaties te bieden voor het opslaan van grote hoeveelheden ongestructureerde of semi-gestructureerde gegevens. NoSQL-databases worden vaak gebruikt in big data-toepassingen en real-time webapplicaties.

2.4 Faceted Search

Faceted search is een zoektechniek die gebruikers in staat stelt om zoekresultaten te verfijnen door middel van verschillende categorieën. In een faceted search-systeem worden zoekresultaten georganiseerd op basis van verschillende categorieën, zoals prijs, merk, kleur, grootte, enzovoort. Gebruikers kunnen vervolgens filters toepassen op deze categorieën om de zoekresultaten te beperken tot diegene die aan hun specifieke criteria voldoen. Faceted search wordt vaak gebruikt in e-commerce websites en digitale bibliotheken om gebruikers te helpen snel de gewenste informatie te vinden.

2.5 Term Frequency enInverse Document Frequency ($TF \times IDF$)

Term Frequency (TF) is een maatstaf voor hoe vaak een term voorkomt in een document. Het wordt berekend door het aantal keren dat een term voorkomt in een document te delen door het totale aantal termen in dat document. (bijv. als het woord "kat" 3 keer voorkomt in een document met 100 woorden, is de TF van "kat" 3/100 = 0.03 = 3%) Inverse Document Frequency (IDF) is een maatstaf voor hoe belangrijk een term is in een verzameling documenten. Het wordt berekend door het totale aantal documenten te delen door het aantal documenten waarin de term voorkomt, en vervolgens de logaritme van dat quotiënt te nemen. (bijv. als het woord "kat" voorkomt in 10 van de 1000 documenten, is de IDF van "kat" log(1000/10) = log(100) = 2) De combinatie van TF en IDF, oftewel $TF \times IDF$, wordt gebruikt om de relevantie van een term in een document te bepalen binnen een verzameling documenten. Een hoge $TF \times IDF$-waarde geeft aan dat een term vaak voorkomt in een document, maar zelden in andere documenten, wat suggereert dat de term belangrijk is voor dat specifieke document.

3 RDF (Resource Description Framework)

RDF is een standaardmodel voor het uitwisselen van gegevens op het web. Het is ontworpen om gegevens te beschrijven in een gestructureerde en machine-leesbare manier. RDF maakt gebruik van een grafenmodel om gegevens te representeren, waarbij gegevens bestaan uit een subject, predicaat en object. Dit maakt het mogelijk om complexe relaties tussen gegevens te modelleren en te analyseren.

```
subject:  http://example.org/person/Alice
predicate: http://xmlns.com/foaf/0.1/knows
object:   http://example.org/person/Bob
```

In dit voorbeeld beschrijft de RDF-verklaring dat Alice Bob kent. RDF maakt gebruik van Uniform Resource Identifiers (URI's) om resources te identificeren, wat zorgt voor een unieke en consistente manier om gegevens te verwijzen.

FOAF (Friend Of A Friend) is ontwikkeld om mensen en hun relaties (met andere mensen of objecten) te beschrijven op het web.

Binnen het RDF heb je ok het RDF Schema (RDFS), dit is een verzameling van klassen en eigenschappen die worden gebruikt om RDF-gegevens te beschrijven. Bijvoorbeeld, je kunt een klasse "Persoon" definiëren en eigenschappen zoals "heeftNaam" en "heeftLeeftijd" om de attributen van een persoon te beschrijven. RDF en RDFS beschrijven alleen de structuur van de gegevens, maar niet de betekenis ervan.

3.1 linked data

Linked Data is een methode om gestructureerde gegevens op het web te publiceren en te verbinden. Het maakt gebruik van RDF om gegevens te beschrijven en URI's om resources te identificeren. Linked Data maakt het mogelijk om gegevens van verschillende bronnen te combineren

en te integreren, waardoor een rijker en meer verbonden web van gegevens ontstaat. Het concept van Linked Data is gebaseerd op vier principes:

- Gebruik URI's om resources te identificeren.
- Gebruik HTTP-URI's zodat resources kunnen worden opgezocht op het web.
- Gebruik RDF om gegevens te beschrijven.
- Verbind je gegevens met andere gegevensbronnen om een web van gegevens te creëren.

Zo is het dus mogelijk om af te leiden dat een persoon een studeert met een x aantal medestudenten:

- <http://example.org/person/MattTerSteege>
- rdf:type foaf:Person
- foaf:name "Matt ter Steege"
- foaf:studiesAt <http://dbpedia.org/resource/Universiteit_Utrecht>
- <http://dbpedia.org/resource/Universiteit_Utrecht>
- dbp:student_count "40000"
- dbp:location "Utrecht, Nederland"

3.2 SPARQL

SPARQL is een querytaal voor het opvragen en manipuleren van RDF-gegevens. Het stelt gebruikers in staat om complexe queries uit te voeren op RDF-datasets en specifieke informatie op te halen. Een query bestaat uit een prefix-declaratie, een select-verklaring en een waar-verklaring. Bijvoorbeeld, de volgende SPARQL-query haalt de namen op van alle personen die studeren aan de Universiteit Utrecht:

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE {
    ?person foaf:studiesAt <http://dbpedia.org/resource/Universiteit_Utrecht> .
    ?person foaf:name ?name .
}
```

* De "?name" betekendt hetzelde als "<http://xmlns.com/foaf/0.1/name>" maar dan korter geschreven.

4 Webnavigatie

4.1 Web usage mining

Dit gaat over het ontdekken van patronen in webdata, zoals gebruikersgedrag op websites. Het doel is om inzicht te krijgen in veel bezochte pagina's en websites of veel voorkomende paden van navigatie. Zo kan je begrijpen welke taken en behoeften gebruikers hebben, wat gebruiksonvriendelijke elementen zijn en hoe je de website kunt verbeteren (UI/UX of snelhied etc.). Het volgende zijn ook voorbeelden:

- Identificeren van advertentielocaties.
- Optimaliseren van menu-desing.
- Herkennen van bots en frauduleuze activiteiten.
- Personaliseren van content en aanbevelingen.
- Voorspellen van de volgende actie van een gebruiker

4.2 gebruikersdata

Gebruikersprofielen: zijn gegevens die door de gebruiker zelf zijn verstrekt, zoals naam, leeftijd, geslacht, locatie en interesses. **Gebruiksdata**: omvat informatie over hoe gebruikers omgaan met een website, zoals bezochte pagina's, klikgedrag, tijd besteed op pagina's en navigatiepaden.

Deze data kan je op verschillende manieren verzamelen, zoals:

- Webserver: Voornamelijk klikgedrag, bezochte pagina's en tijd op pagina.
- Webclient: Data van één gebruiker op verschillende sites, zoals muisbewegingen, scrollgedrag en interacties.
- Proxyservers: Data van meerdere gebruikers, zoals bezochte sites en algemene navigatiepatronen.

4.3 Webserverdata

: Als een gebruiker een website bezoekt, registreert de webserver automatisch verschillende gegevens, zoals het IP-adres van de gebruiker, de tijd van het bezoek, de bezochte pagina's en de duur van het bezoek. Deze gegevens worden opgeslagen in logbestanden die later kunnen worden geanalyseerd om inzicht te krijgen in het gedrag van gebruikers op de website.

Een voorbeeld van een webserverlogbestand:

```
• 213.6.31.68 - - [01/May/2004:22:38:32 +0200] "GET
  /forsale.html HTTP/1.1" 200 14956
  "http://www.fortepiano.nl/indexforsale.html" "Mozilla/4.0
  (compatible; MSIE 6.0; Windows NT 5.1)

• 213.6.31.68 - - [01/May/2004:22:38:34 +0200] "GET
  /pictures/forsale/bertsche1835-small.jpg HTTP/1.1" 200 5753
  "http://www.fortepiano.nl/forsale.html" "Mozilla/4.0
```

```
(compatible; MSIE 6.0; Windows NT 5.1)
```

Zo'n logbestand bestaat uit alle requests aan een webserver/website. Als je hier data uit wilt halen, dan moet je deze eerst voorbewerken (cleanen).

(1) **Data cleaning**: Verwijder onnodige gegevens, zoals requests voor afbeeldingen, scripts en stylesheets.
(2) **User identification**: Identificeer unieke gebruikers op basis van IP-adressen en user-agent strings. Je kunt dit doen door bijvoorbeeld cookies te gebruiken, maar accounts of headers kunnen ook helpen.
(3) **Session identification**: Groepeer requests van dezelfde gebruiker binnen een bepaalde tijdsperiode tot één sessie. Je kunt hiervoor een tijdslimiet instellen (bijv. 30 minuten van inactiviteit betekent het einde van een sessie).
(4) **Path completion**: Vul ontbrekende pagina's in die mogelijk zijn overgeslagen door caching of directe toegang.
(5) **Robot identification**: Identificeer en verwijder requests van webcrawlers en bots. Dit kan gedaan worden door user-agent strings te analyseren of door bekende IP-adressen van bots te blokkeren. Maar ook door het gedrag te analyseren (bijv. als een gebruiker in een paar seconden 100 pagina's bezoekt, is het waarschijnlijk een bot).

4.4 webclientdata

Webclientdata wordt verzameld via scripts die op de client-side (de browser van de gebruiker) worden uitgevoerd. Deze scripts kunnen informatie verzamelen over het gedrag van de gebruiker op de website, zoals muisbewegingen, scrollgedrag, klikgedrag en tijd besteed op pagina's. Deze gegevens worden vervolgens naar de server gestuurd voor analyse. Populaire tools voor het verzamelen van webclientdata zijn Google Analytics, Hotjar en Mixpanel. Dit is door de GDPR wet aardig in onbruik geraakt.

Conversion tracking: Dit houdt bij hoeveel gebruikers een specifieke actie voltooien, zoals het invullen van een formulier of het doen van een aankoop. Dit werd vaak gedaan met een "spy pixel", een onzichtbaar beeld dat wordt geladen wanneer een gebruiker een bepaalde pagina bezoekt. Om te checken hoe goed emailcampagnes werken of advertenties.

5 Voorspellende modellen

Wat zijn voorspellende modellen

Voorspellende (of probabilistische) modellen proberen niet *zeker* te weten wat een gebruiker gaat doen, maar schatten de kans dat een bepaalde handeling zal plaatsvinden. Alles draait om waarschijnlijkheden. Je kijkt naar eerder gedrag en zegt: *op basis hiervan is dit de meest waarschijnlijke volgende stap.*

Een handeling noemen we (a). De kans dat deze handeling voorkomt ligt altijd tussen 0 en 1:

$$0 \leq P(a) \leq 1$$

Deze kans heet de a-priori kans. Die staat los van context. Sommige handelingen gebeuren vaak (homepage bezoeken), andere zelden (voorwaardenpagina openen).

Voorwaardelijke kansen

Zodra je context meeneemt, verandert het spel. Als handeling (b) vaak gevolgd wordt door handeling (a), dan wordt (a) waarschijnlijker zodra (b) is waargenomen.

Dat schrijven we als:

$$P(a \mid b)$$

Dit lees je als: *de kans op a, gegeven dat b heeft plaatsgevonden.*

Concreet:

$$P(a \mid b) = \frac{\text{aantal keren dat b gevolgd wordt door a}}{\text{aantal keren dat b voorkomt}}$$

Dit is nog steeds simpel tellen, maar nu mét context.

Theorema van Bayes

Bayes generaliseert dit idee. Niet één vorige handeling telt mee, maar een hele set bewijsstukken (E) (een keten van acties).

De formule:

$$P(a \mid E) = \frac{P(E \mid a) \cdot P(a)}{P(E)}$$

Wat hier gebeurt:

- (P(a)): hoe waarschijnlijk was actie a sowieso al
- $(P(E \mid a))$: hoe goed past het waargenomen gedrag bij actie a
- (P(E)): normalisatie, zodat de kans geldig blijft

In praktijk: gebruikersgeschiedenis weegt mee, maar nooit zonder de basispopulariteit van een actie te corrigeren.

Rankingmethoden op basis van a-priori kansen

Niet alle pagina's zijn gelijk. Sommige pagina's worden structureel vaker bezocht dan andere. Rankingmethoden proberen pagina's te ordenen op basis van herbezoekkans.

Typische observaties:

- Een klein aantal pagina's trekt het merendeel van het verkeer
- Gebruikers keren vaak terug naar recent bezochte pagina's

Dit leidt tot eenvoudige, maar verrassend effectieve methoden.

Last Recently Used (LRU)

LRU kijkt alleen naar *recentheid*.

$$LRU(m_i, I_{m_i}, i_n) = \frac{1}{i_n - i_k + 1}$$

Waar:

- (i_n) de index is van het laatste paginabezoek
- (i_k) de index is van het laatste bezoek aan pagina (m_i)

Hoe recenter het bezoek, hoe hoger de score.

Most Frequently Used (MFU)

MFU kijkt alleen naar *frequentie*.

$$MFU(m_i, I_{m_i}, i_n) = \frac{|I_{m_i}|}{i_n}$$

Pagina's die vaak bezocht zijn in het verleden krijgen een hogere kans, ongeacht hoe lang geleden dat was.

Polynomial Decay (PD)

Polynomial Decay combineert frequentie en recentheid.

$$PD(m_i, I_{m_i}, i_n) = \sum_{j=1}^{|I_{m_i}|} \frac{1}{1 + (i_n - i_j)^\alpha}, \quad 0 < \alpha \leq 1$$

Recente bezoeken tellen zwaarder dan oude. De parameter (α) is bepalend:

- hoge (α): snelle afname, focus op recentheid
- lage (α): langzamere afname, focus op frequentie

Markov-modellen

Markov-modellen veronderstellen dat de volgende actie afhangt van eerdere acties.

Een eerste-orde Markov-model gaat ervan uit dat alleen de *laatste* actie relevant is.

$$P(s_n) = P(s_1) \prod_{t=2}^{n} P(s_t \mid s_{t-1})$$

Dit model kan worden weergegeven als een transitiematrix:

From/To	A	B	C	D
A	3	5	8	16
B	3	7	4	14
C	2	4	6	12
D	1	6	2	9

Elke cel geeft aan hoe vaak een overgang is waargenomen.

Hogere-orde Markov-modellen

Bij hogere-orde modellen hangt de volgende actie af van de laatste (k) acties. Dit is realistischer, maar schaalt slecht. Daarom worden pagina's of gebruikers vaak geclusterd om matrices hanteerbaar te houden.

Associatieregels

Associatieregels kijken niet naar volgorde, maar naar *samen voorkomen*.

Een regel heeft de vorm:

$$X \Rightarrow Y$$

Waarbij (X) en (Y) verzamelingen van items zijn zonder overlap.

Support en Confidence

Support:

$$support(X \Rightarrow Y) = \frac{|t_i \in D : X \cup Y \subset t_i|}{|D|}$$

Confidence:

$$confidence(X \Rightarrow Y) = \frac{|t_i \in D : X \cup Y \subset t_i|}{|t_i \in D : X \subset t_i|}$$

Support is vaak laag, confidence moet hoog zijn om de regel bruikbaar te maken.

A-priori algoritme

Het A-priori algoritme reduceert de zoekruimte drastisch:

- Als een itemset niet frequent is, kan geen enkele superset frequent zijn
- Frequent-itemsets worden iteratief opgebouwd

Dit maakt het vinden van associatieregels praktisch uitvoerbaar.

Praktische realiteit

Niet elk product of elke pagina volgt hetzelfde patroon. Wat werkt voor media en smaakgevoelige producten, faalt bij seizoensgebonden of functionele aankopen. Voorspellende modellen zijn krachtig, maar alleen zolang je accepteert dat gebruikersgedrag niet netjes, stabiel of symmetrisch is.