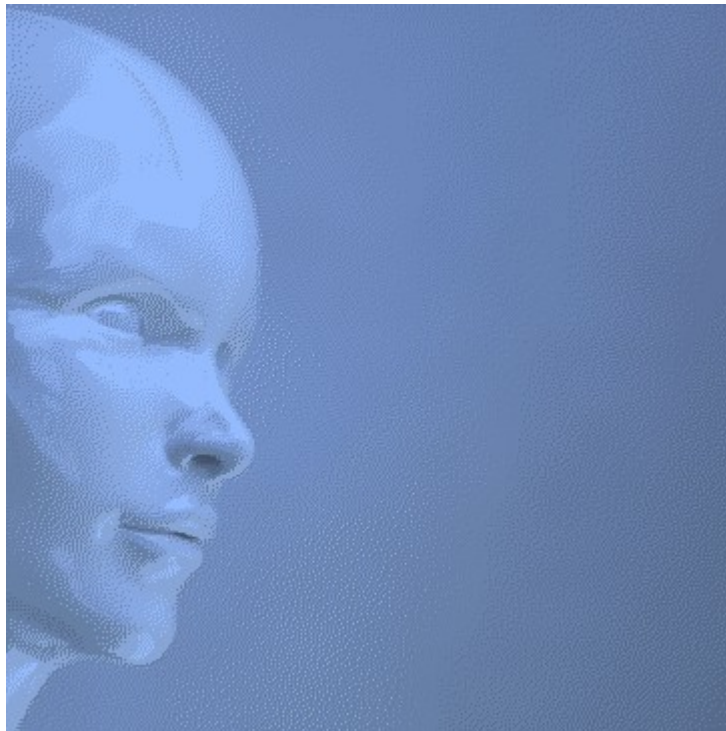


Wetenschappelijke uitdaging:

Versla mijn natuurlijke taal redeneersysteem



Kennistechnologie is gebaseerd op het toepassen van slimme algoritmen op trefwoorden, waardoor de natuurlijke semantiek van niet-trefwoorden wordt genegeerd.

Niet-trefwoorden geven informatie over de structuur van de zin door aan onze hersenen. Maar doordat niet-trefwoorden in de kennistechnologie worden genegeerd, weet dit vakgebied niet wat het aan moet met de "handenvol trefwoorden" en ongestructureerde teksten. Zelfs Watson van IBM is niet in staat om z'n kennissysteem zelfstandig te organiseren. In plaats daarvan heeft Watson brute processorkracht nodig heeft om "de naald in een hooiberg van ongestructureerde teksten te vinden" (quote van IBM). En bij Cycorp structureren experts de kennis al decennialang handmatig.

Predicatenlogica beschrijft de natuurlijke betekenis van het werkwoord "is/zijn", zoals: "Iedere vader is een man" en "Jan is een vader" waaruit kan worden geconcludeerd "Jan is een man". Echter, predicatenlogica beschrijft niet de natuurlijke betekenis van bv. werkwoord "heeft/hebben". Met als gevolg dat kennis met dit werkwoord, direct in een redeneersysteem geprogrammeerd moet worden, zoals: `heeft_zoon(jan,paul)`.

Het direct in een redeneersysteem programmeren – zoals `heeft_zoon(jan,paul)` – is geen generieke oplossing (wetenschap), maar een specifieke oplossing (engineering). Want je moet elk afzonderlijk zelfstandig naamwoord rechtstreeks in het redeneersysteem programmeren (`heeft_dochter`, `heeft_vader`, `heeft_moeder`, enz.), en ook nog eens voor elke nieuwe taal. Met als gevolg dat er geen enkele techniek bestaat om een zin als "Paul is een zoon van Jan" op een generieke manier om te zetten naar "Jan heeft een zoon, genaamd Paul", waarbij de trefwoorden "Paul", "zoon" en "Jan" niet vooraf, rechtstreeks in het redeneersysteem geprogrammeerd hoeven te worden. Dit is slechts het eerste voorbeeld van deze uitdaging.

Mijn fundamentele benadering laat zien dat werkwoord "heeft" complementair is aan werkwoord "is". Dus kan ook werkwoord "heeft" gebruikt worden in de predicatenlogica. Om de natuurlijk-intelligente functie (natuurlijke semantiek) van niet-trefwoorden te kunnen benutten, heb ik [natuurlijke intelligentie](#) gedefinieerd. Daarna heb ik enkele [Natuurwetten van Intelligentie in grammatica](#) geïdentificeerd. Door deze natuurwetten als structureringsalgoritmen te implementeren, kan mijn systeem de kennis zelfstandig structureren.

Spelregels

- Er zijn 5 blokken waarin je de allereenvoudigste basistechnieken van mijn systeem kunt verslaan. Je implementatie dient de hieronder genoemde resultaten van tenminste één blok te leveren;
- Je implementatie mag vooraf geen kennis bevatten. In plaats daarvan dient het systeem de kennis te ontleen van de hieronder genoemde invoerzinnen. Bij voorkeur moeten de gebruikte zelfstandige naamwoorden en eigennamen vooraf onbekend zijn;
- Je implementatie dient zo generiek mogelijk te worden opgezet, zodat alle voorbeelden van deze uitdaging geïntegreerd kunnen worden tot één enkel systeem. (De [redeneer-schermafdrukken](#) van mijn redeneersysteem laten zien dat diverse redeneerconstructies elkaar versterken);
- Je implementatie dient te worden gepubliceerd als open source software, zodat het gecontroleerd kan worden. (Ook mijn systeem is open source: <http://mafait.org/download/>);
- Je implementatie moet aanvaard zijn door een wetenschappelijk comité (conferentie of journaal);
- In het geval dat je resultaten iets anders zijn, dien je uit te leggen waarom je daarvoor hebt gekozen;
- Deze wedstrijd gaat door totdat mijn systeem volledig is verslagen;
- Ik ben de scheidsrechter.

Jouw beloning

- Een klein gebaar van mijn kant: € 200 per blok;
- Je bent de eerste die deze resultaten op een wetenschappelijk geaccepteerde manier heeft bereikt, omdat ik geen enkele techniek uit dit vakgebied gebruik.

Je kunt me bereiken via de [contact pagina](#) van mijn website, en via [LinkedIn](#).

Blok 1

Linguïstische algebra, met cruciale niet-trefwoorden “is”, “heeft”, “genaamd”, “iedere” en “deel van”:

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} van {eigenaam 2}”

is gelijkwaardig aan

“{eigenaam 2} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} genaamd {eigenaam 1}”

“Iedere {enkelvoudig zelfstandig naamwoord 1} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

is gelijkwaardig aan

“{onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} is deel van iedere {enkelvoudig zelfstandig naamwoord 1}”

Implementatie:

> Ingevoerd: “Paul is een zoon van Jan.”

•

• Gegenereerde conclusie:

< “Jan heeft een zoon, genaamd Paul.”

>

> Ingevoerd: “Anna heeft een dochter, genaamd Laura.”

•

• Gegenereerde conclusie:

< “Laura is een dochter van Anna.”

>

> Ingevoerd: “Iedere auto heeft een motor.”

•

• Gegenereerde conclusie:

< “Een motor is deel van iedere auto.”

>

> Ingevoerd: “Een zeil is deel van iedere zeilboot.”

•

• Gegenereerde conclusie:

< “Iedere zeilboot heeft een zeil.”

Blok 2

Linguïstische algebra, met cruciale niet-trefwoorden “**van**”, “**is**”, “**heeft**” en “**genaamd**”:

“{eigennaam 1} **is** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigennaam 2}”

is gelijkwaardig aan

“{eigennaam 2} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigennaam 1}”

“{eigennaam 1} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigennaam 2}”

en

“{eigennaam 1} **heeft** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigennaam 3}”

is gelijkwaardig aan

“{eigennaam 1} **heeft** {aantal: 2} {meervoudsvorm van enkelvoudig zelfstandig naamwoord 1} **genaamd** {eigennaam 2} **en** {eigennaam 3}”

Implementatie:

> Ingevoerd: “**Jan is een ouder van Paul.**”

•

• Gegenereerde conclusie:

< “**Paul heeft een ouder, genaamd Jan.**”

>

> Ingevoerd: “**Anna is een ouder van Paul.**”

•

• Gegenereerde conclusie:

< “**Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.**”

>

> Ingevoerd: “**Paul heeft 2 ouders, genaamd Jan en Anna.**”

•

• Gedetecteerd dat de gegenereerde conclusie bevestigd is:

< “**Paul heeft 2 ouder [meervoud van 'ouder' is onbekend], genaamd Jan en Anna.**”

•

• Gedetecteerd: U heeft meervoudig zelfstandig naamwoord “**ouders**” ingevoerd, welke mij **onbekend was**.

Blok 3

Linguïstische algebra, met cruciale niet-trefwoorden “**van**”, “**was**”, “**heeft**”, “**had**” en “**geen [...]** **meer**”:

“{eigennaam 1} **was** {bepaald of onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigennaam 2}”

is gelijkwaardig aan

“{eigennaam 2} **heeft geen** {enkelvoudig zelfstandig naamwoord} **meer**”

“{eigennaam 1} **was** {bepaald of onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **van** {eigennaam 2}”

is gelijkwaardig aan

“{eigennaam 2} **had** {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} **genaamd** {eigennaam 1}”

Implementatie:

> Ingevoerd: “**Johan was de vader van Michael en Peter.**”

•

• Gegenereerde conclusies:

< “**Michael heeft geen vader meer.**”

< “**Michael had een vader, genaamd Johan.**”

< “**Peter heeft geen vader meer.**”

< “**Peter had een vader, genaamd Johan.**”

Blok 4

Linguïstische algebra, met cruciale niet-trefwoorden “iedere”, “is”, “of”, “en” en “geen”:

“Iedere {enkelvoudig zelfstandig naamwoord 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

is in conflict met

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} en {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“Iedere {enkelvoudig zelfstandig naamwoord 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

en

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

waaruit kan worden geconcludeerd

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}”

is gelijkwaardig aan

“Is {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 3}?”

“Is {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}?”

en

“{eigenaam 1} is geen {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

waaruit kan worden geconcludeerd

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

“Is {eigenaam 1} {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1} of {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}?”

en

“{eigenaam 1} is geen {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 2}”

waaruit kan worden geconcludeerd

“{eigenaam 1} is {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord 1}”

Implementatie:

> Ingevoerd: “Ieder persoon is een man of een vrouw.”

> Ingevoerd: “Anne is een man en een vrouw.”

!

! Gedetecteerd conflict: Deze zin is niet geaccepteerd, omdat het in conflict is met zichzelf.

!

> Ingevoerd: “Anne is een persoon.”

•

• Gegenerateerde vraag:

< “Is Anne een man of een vrouw?”

>

> Ingevoerd: “Anne is geen vrouw.”

•

• Gedetecteerd dat de gegenereerde vraag beantwoord is:

< “Is Anne een man of een vrouw?”

•

• Gegenerateerde conclusie:

< “Anne is een man.”

Blok 5

Linguïstische algebra, met cruciale niet-trefwoorden “de/het”, “is”, “heeft”, “genaamd”, “vorig/vorige” en “nieuw/nieuwe”:

“{eigenaam 1} is {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} van {eigenaam 2}”
is gelijkwaardig aan

“{eigenaam 2} heeft {onbepaald lidwoord + enkelvoudig zelfstandig naamwoord} genaamd {eigenaam 1}”

“{eigenaam 1} is {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} van {eigenaam 2}”
en

“{eigenaam 3} is {bepaald lidwoord + enkelvoudig zelfstandig naamwoord} van {eigenaam 2}”
is gelijkwaardig aan

“{eigenaam 2} heeft {onbepaald lidwoord} nieuw(e) {enkelvoudig zelfstandig naamwoord} genaamd {eigenaam 3}”
en

“{eigenaam 2} heeft {onbepaald lidwoord} vorig(e) {enkelvoudig zelfstandig naamwoord} genaamd {eigenaam 1}”

Implementatie:

> Ingevoerd: “Bill Clinton is de president van de Verenigde Staten.”

•

• Gegenereerde conclusie:

< “De Verenigde Staten heeft een president, genaamd Bill Clinton.”

>

> Ingevoerd: “Barack Obama is de president van de Verenigde Staten.”

•

• Gegenereerde conclusies:

< “De Verenigde Staten heeft een nieuwe president, genaamd Barack Obama.”

< “De Verenigde Staten heeft een vorige president, genaamd Bill Clinton.”