

Bootstrapping language

Mart van Santen

19 november 2005



Begeleider: Remko Scha

Inhoudsopgave

1	Inleiding	2
1.1	Overzicht	3
1.2	Uitbreidingen	3
2	Het algoritme	4
2.1	One-word algoritme	4
2.2	Two-word algoritme	4
3	Childes	7
3.1	Data selectie	8
3.2	Data manipulatie	8
4	Frame representatie	8
5	Implementatie	9
5.1	Interne representaties	10
5.2	Database	10
5.3	Processing	10
6	Resultaten	11
7	Discussie & conclusies	11
7.1	Het algoritme	11
7.2	De dataset	12
7.3	Frame representatie	12
7.4	Implementatie	12
7.5	Future work	13
7.5.1	Multi-Word	13
7.5.2	Attention/Priming	13
7.5.3	Feedback	13
7.5.4	Neuraal Netwerk	14
7.5.5	Statistiek & abstracties	14

1 Inleiding

In dit onderzoek heeft als doel om beter inzicht te krijgen hoe de eerste stappen van acquisitie van taal kan plaatsvinden en hoe dit te modelleren valt. Met deze “bootstrapping” wordt het leren van betekenissen van woorden bedoeld en in tweede fase gaat dit over het leren van woordrelaties om daarmee een eerste basis van de structuur van taal te leggen. Dit onderzoek is uitgegaan van de afstudeerscriptie van Mike de Kreek [2]. In deze scriptie worden suggesties gedaan over hoe een dergelijke bootstrapping er uit kan zien, maar is dit niet concreet uitgewerkt en blijft het op een abstract niveau.

De Kreek heeft op basis van andere onderzoeken en literatuur een model ontworpen over welke stappen er bij taalacquisitie plaats vinden. Zijn model begint vanaf het allereerste begin.

1.1 Overzicht

De Kreek gaat er vanuit dat taal een onderdeel is van de totale ontwikkeling. Hij heeft dan ook het hele proces van niets totaan taal bestudeerd en daar een theorie voor ontworpen. Dit proces wordt opgedeeld in verschillende fases. Deze beschrijven hoe een kind met de zintuigen en de informatie die daarop binnen komt om gaat en hoe dit uiteindelijk resulteert tot taal. In onderstaand overzicht staan de verschillende fases genoemd:

- Low-Level: Deze fase resulteert in een corpus bestaande uit situatie specifieke low level concepten. Een kind begrijpt tot op zeker hoogte wat deze waarneemt, echter ziet nog niet relaties. (Het kind snapt niet dat een bal rolt doordat iemand er tegen duwt. Maar het kind snapt wel een rollende bal en iemand die een bal aanraakt).
- Cycle 1: Een kind kan verschillende gebeurtenissen in relatie met elkaar zien, iets wat in de vorige fase nog niet kon. De koppeling tussen een rollende bal en iemand die er tegen duwt kan worden gelegd.
- Cycle 2: De waargenomen gebeurtenissen worden gekoppeld aan een akoestisch signaal. Deze gegevens worden geregistreerd. Tevens kunnen woorden¹ van elkaar worden onderscheiden en is dit niet meer een losstaande lange (ruisachtige) audio stream.
- Cycle 3/One-word: In deze fase kent een kind betekenissen toe aan de woorden die het hoort. Dit gebeurt door een analyse te maken van de geziene situaties en daarbij waargenomen woorden.
- Cycle 4/Two-word: Deze fase resulteert tot een corpus met interpretaties van twee-woord structuren. Dit zijn relaties tussen verschillende type woorden. Bijvoorbeeld dat het woord “vallen” vaak vergezeld gaat met een object.
- Multi-word: De laatste fase is de multi-word fase, in deze uitbreiding kunnen meervoudige structuren worden gemaakt en worden deze begrepen.

De eerste fases van dit proces bevatten behalve de acquisitie van taal ook acquisitie over hoe met een omgeving wordt omgegaan en met de verschillende acties die daarbinnen plaatsvinden. Hoewel dit interessant is, staat dit onderdeel meer op zichzelf en is genoemd om een beter totaal overzicht te krijgen. Hoewel de manier hoe dit kan plaatsvinden lijkt op de manier van taalacquisitie, heeft het er verder niet veel mee te maken.

Het Multi-word stadium kan op aansluiten op het DOP-framework [3], dit maakt de hele taalacquisitie van begin tot eind compleet.

1.2 Uitbreidingen

In dit onderzoek worden een aantal onderdelen uit het voorgestelde proces van de Kreek nader onderzocht. Er is geconcentreerd op het first-word en second-word stadium. Dit is opgedeeld in vier onderdelen. Ten eerste is er onderzoek

¹Het is mogelijk dat nog niet alle woorden kunnen worden onderscheiden. Er worden o.a. een voorbeeld genoemd: “eatit”, dit zouden kinderen nog als 1 woord herkennen.

gedaan naar een algoritme wat het first en second woord stadium moeten kunnen modelleren. Ten tweede is er gekeken welke data bruikbaar is voor het testen van het model, en hoe deze vormgegeven dient te worden, er is een mini-wereld gemaakt met data. Het derde onderdeel bestaat uit een ontwerp voor een frame structuur. Deze structuur moet situaties en acties kunnen omschrijven en deze moet simpel verwerkbaar zijn voor een computer programma. Als laatste is een eenvoudige implementatie van het algoritme gemaakt om het algoritme ook daadwerkelijk te kunnen testen.

2 Het algoritme

Het eerste wat ontwikkeld is, is een algoritme wat de voorgestelde theorie kan implementeren. Er is begonnen met de ontwikkeling met een algoritme voor het een-woord stadium. Dit algoritme moet resulteren tot een corpus van woordbetekenissen. Dit is verder uitgebreid naar het een algoritme voor het twee-woord stadium.

2.1 One-word algoritme

Een pseudo-code implementatie van het one-word-algoritme staat in algoritme 1.² Het valt uiteen in twee belangrijke delen, het “Process” deel en het “Analyze” deel.

Het “Process” deel vereist zowel audio als een frame. Deze frame representatie wordt opgesplitst in een set van meerdere frames (zie ook hoofdstuk 4 voor een implementatie model van frames). Deze bestaan uit alle mogelijke combinaties van alle subframes van het originele frame. Vervolgens wordt aan ieder frame van deze set alle geregistreerde woorden toegevoegd en de daarbij behorende teller verhoogt met 1. Op deze manier wordt opgeslagen welke woorden hoe vaak zijn gehoord bij een (sub)frame. Nadat dit gedaan is, wordt het frame gehercalculeerd. Dit betekent dat alle woorden die zelden zijn vernomen in combinatie met de frame worden verwijderd uit de database. Dit beschermt de database van extreme expansie en verkleint de invloed van ruis.

Het tweede deel is het “Analyze” deel. Deze heeft alleen een frame als input. Dit frame wordt op dezelfde manier uiteen gesplitst in meerdere frames. Per frame wordt gekeken of deze eerder is voorgekomen en welke woorden daar toen uitgesproken zijn. Per woord wordt gekeken welke waarde die binnen dat frame heeft. Deze woorden worden in een tabel opgeslagen met hun corresponderende cumulatieve waarde. Uiteindelijk wordt het woord met de hoogste waarde uitgekozen en gebruikt voor de output.

2.2 Two-word algoritme

Het twee-woord algoritme (algoritme 2) is een uitbreiding op het “processing” gedeelte van het een-woord algoritme. Het algoritme moet in staat zijn de structuren tussen twee woorden te ontrafelen. Dit gebeurt door binnen gekomen frames op verschillende manieren te analyseren. Iedere analyse gebruikt een set

²Split() en GetWordsFromAudio() zijn niet verder uitgewerkt, deze zijn qua concept triviaal. De eerste berekend alle combinaties van een frame, de tweede genereert een lijst woorden van een audio stream. Enkele andere triviale functies zijn ook niet verder uitgewerkt.

Algorithm 1 One-word stadium

```
/* On the event of input, we check if we have audio
 * and/or a frame and descide what we to do */
Event OnInput(audio, frame)
    If (audio && frame) Process(audio, frame) // Train
    If (!audio && frame) Analize(frame)      // Analyze
    If (audio && !frame) NoAction()         // No Action

/* This is the most important function of the one-word
 * algorithm. It uses audio and a frame information to
 * learn word-meaning's */
Process(audio, frame)
    frameset = Split(frame)
    words = GetWordsFromAudio(audio)
    foreach(frame in frameset)
        foreach (word in words)
            // Increase count for this combination in db.
            IncreaseCountFor(frame, word)
        end
        RecalculateFrame(frame)
    end

/* This function recalculates all values for all
 * words attached to this frame and removes these
 * below some thesshold */
RecalculateFrame(frame)
    foreach (word in frame)
        value = GetValue(word, frame)
        if (value < THRESHOLD)
            RemoveWordFromFrame(word, frame)
        end
    end

// Selects the word with maximum value for a frame
Analyze(frame)
    frameset = Split(frame)
    foreach (frame in frameset)
        foreach(word in frame)
            words[word] += GetValue(word, frame)
        end
    end
    // Select the word with the highest value
    word = max(words)
    Output(word)
```

Algorithm 2 Twee-woord stadium

```
Process(audio, input_frame)
  // Frames seen with this audio
  Forach(word in audio)
    audio_frames[word] = GetFramesForWord(word)
  End
  // Analysis 1: True matches
  frames = Split(input_frame)
  Foreach(frame in frames)
    true_matches = FindMatches(frame, audio_frames)
  End

  // Analyses 2: Abstractions
  frames = GetAbstractions(frames)
  Foreach(frame in frames)
    abstract_matches = FindMatches(frame, audio_frames)
  End

  // Analyses 3: Generalizations
  frames = GetGeneralizationsFrames(frames)
  Foreach(frame in frames)
    gen_matches = FindMatches(frame, audio_frames)
  End

  words = Calculate(true_matches, abstract_matches,
                  gen_matches)
  InsertIntoDatabase(input_frame, words)
  InsertIntoDatabase(input_frame, audio)
  Output(words, input_frames)
```

van frames die afgeleid is van de input frame en een set van frames die afgeleid is van de input audio (indien beschikbaar). Deze worden gematched met de database, wat iedere keer resulteert in een lijst van woorden met gevonden frames. In deze hele set worden de waarschijnlijkheid van de woorden en frames ook teruggegeven (de terug gegeven set dient een structuur te zijn). De frames gevonden bij de input audio worden steeds gebruik als een heuristiek bij het matchen, om zo relevantere frames op basis van audio voorkeur te geven.

De eerste analyse van een binnengekomen frame is identiek aan het eenwoord algoritme: alle combinaties van het frame worden berekend. De verkregen frameset wordt gematched met bestaande database (en met gebruik van de audio frames als heuristiek) en resulteert in een result set van “true_matches”. Dit zijn matches van woorden en frames die passen op de input frame.

De tweede analyse maakt abstracties van de frames, deze worden gematched op de database (+ de audio frames). De resultset is een set met abstracte matches op de database. De resultaten komen in abstract_matches.

De laatste stap is identiek aan de vorige, met dit verschil dat er nu frames worden gebruikt met vrije variabelen. Dit maakt het mogelijk om slechts delen

frames de laten matchen, zo kunnen niet eerder voorgekomen situaties worden herkend. In feite is deze laatste stap een ultieme abstractie en het is dan ook niet ondenkbaar dat deze twee stappen bij een implementatie worden samengevoegd tot een geheel.

De gevonden frames dienen een voorkeur te hebben om niet te ver van elkaar af te liggen en zo klein mogelijk te zijn. De analyses hebben ieder een bepaalde waarschijnlijkheid en gewicht. Hoe met deze gewichten om wordt gegaan vindt plaats in de functie `CALCULATE()`. Deze maakt een calculatie op basis van de (sub)frames en de woorden die daar bij horen. De twee woorden die in een (abstract) subframe passen en waarvoor geldt dat zowel dit frame als de woorden een hoge waarschijnlijkheid hebben volgens de statistiek, worden geretourneerd. Omdat de twee woorden in een subframe moeten passen, zouden de twee woorden een onderlinge structuur moeten hebben. Hiermee kan de structuur van een taal woorden gevonden. Uiteindelijk worden de gevonden frames opnieuw ingevoegd in de database.

3 Childes

Childes [1] is een database op internet dit bestaat uit een grote collectie van sessies tussen ouder en kind. Op uitnodiging van het onderzoeksteam wordt een sessie tussen een kind en ouder opgenomen, soms met aanwezigheid van de onderzoeker, maar meestal niet. Tijdens deze sessies spelen de ouder en het kind met het aanwezige speelgoed. Deze sessies duren ongeveer een half uur. De gehele opname wordt vervolgens ingevoerd zodat de conversatie tussen kind en ouder (en eventueel de onderzoeker) digitaal beschikbaar zijn. Hierbij worden soms, maar niet altijd, ook de acties van dat moment ingevoerd, zodat later, zonder gebruik van video materiaal, bekend is wat er plaats heeft gevonden. In figuur 1 staat een fragment uit de childes database. Hierbij is alles wat achter `*MOT` en `*CHI` staat een uiting van de moeder respectievelijk het kind. Alles achter `%ACT` en `%GPX` vertegenwoordigen acties, waarbij `%GPX` gebaren of 'lichaamstaal' is.

```
*MOT: did you see this ball ?
%act: MOT picks up the football
*CHI: 0 .
%gpx: CHI watches MOT
*MOT: 0 .
%act: MOT rolls the ball to CHI
*CHI: 0 [>] .
%act: CHI picks the football up
*MOT: is it a ball
```

Figuur 1: Fragment uit childes database

3.1 Data selectie

De chilles data is gebruikt als basis voor de input voor het uiteindelijke test programma/implementatie. Hierbij is een selectie van de data gemaakt in twee dimensies. Ten eerste is alleen data gebruikt waarvan de kinderen tussen de 0 en 4 jaar zijn en die normaal ontwikkeld zijn; dit omdat dit onderzoek zich concentreert op de first- en secondword stage. Kinderen vanaf ongeveer vier jaar zijn in staat complexere structuren te begrijpen en te produceren en daardoor voorbij de secondword stage. Ten tweede is gekozen voor deze fragmenten data waarin slechts een beperkt aantal objecten betrokken zijn. Dit omdat door een beperking van het aantal verschillende objecten een eenvoudiger (mini)wereld ontstaat waarop het model getest kan worden. Concreet betekend dat in de dataset alleen zinnen en uitspraken terecht zijn gekomen die betrekking hebben op 'ballen', 'blokken', 'stoelen', 'boeken', 'plaatjes (in het boek)' en 'autotjes'. Deze twee dimensies van de dataset hebben geresulteerd tot een mini-wereldje bestaand uit 153 situaties met daarbij 342 uitspraken/zinnen.

3.2 Data manipulatie

Deze selectie is vervolgens herschreven tot een iets ander formaat. Een formaat waarbij acties zijn samengevat tot een enkele zin en waaraan alle behorende uitgesproken zinnen zijn toegevoegd. Ook zijn er soms vereenvoudigingen gemaakt, in sommige sessie werd bijvoorbeeld de term "football" gebruikt en in andere weer "ball". Bij alle instanties is er voor gekozen om alleen de term "ball" te gebruiken en waarnodig de term "football" aan te passen naar "ball". Dit-zelfde is met kleuren gebeurd, zodat slechts enkele kleuren terug komen en niet alle mogelijk kleuren. Deze vereenvoudigingen hebben wederom als doel om de test-wereld klein te houden en daardoor naar verwachting makkelijker te trainen is.

4 Frame representatie

Het algoritme levert een basis voor het verwerken van de data en chilles levert een basis voor de data zelf. Het probleem is dat bij een implementatie de software dient te werken met vast omleide structuren. Deze structuren moeten een complete situatie kunnen weergeven, en de onderdelen binnen de situatie/actie kunnen blootleggen. Hiervoor is gekozen een frame per situatie te definiëren. Deze frames bestaan uit een object georiënteerde structuur, zodat uitbreidingen eenvoudig zijn te maken. Een tweede gewenste eigenschap is, is dat de frames de mogelijkheid hebben om abstracties te kunnen toekennen. De directe beschikbaarheid van abstracties in het frame kunnen ambiguïteiten voorkomen en maken het bijhouden van een abstractie-overzicht een stuk eenvoudiger. Een andere mogelijkheid die graag werd terug gezien in het frame ontwerp is de mogelijkheid tot het opnemen van een referentie. Hiermee kan binnen hetzelfde frame (of een later voorkomend frame) naar worden gerefereerd om aan te geven dat het om een specifiek object gaat (het verschil tussen "die bal" en "een bal"). Daar bovenop is er laatste aanvulling gemaakt die de mogelijkheid biedt om gebruik te maken van pad-expressies [4] om een verdieping binnen een (losstaand) object aan te wijzen.

```

action: hit {
  patient: #ref1234:joko {
    abstraction: human;      1
    gender: female;         2
    ...
  }
  actor: john {
    abstraction: human;      3
    gender: male;
    relationship-with: #ref1234; 4
  }
  position {
    relative: true;
    where: #ref1234/head;   5
  }
}

```

Figuur 2: Voorbeeld van een frame-representatie

In figuur 2 staat een voorbeeld van een frame. Dit frame bestaat uit een overkoepelende actie (“hit”) waarbinnen een set van variabele voorkomen. Doordat de waarde van een variabele zelf ook weer een set kan zijn, ontstaat er een (eenvoudige) object structuur. In dit voorbeeld staat onder nummer 1 een waarde met referentie vermelding. De waarde “joko” krijgt hier een referentie nummer mee: 1234, dit nummer wordt later gebruikt om te verwijzen naar “deze” joko. Bij nummer 2 staat een simpele eigenschap van joko: joko heeft het geslacht vrouw. Een voorbeeld van een abstractie staat bij punt 3. De variabele ‘abstraction’ heeft een speciale eigenschap; waarden daarvan maken zelf deel uit van een boom structuur met abstracties. Hierbij is het van belang dat de boom met abstractie geen ambiguïteit vertoont. Bij onderdeel 4 wordt gebruik gemaakt van een referentie binnen een eigenschap. Op deze manier is het duidelijk dat het over dezelfde “Joko” gaat waarmee John een relatie heeft. Onder nummer 5 staat een voorbeeld van een pad-expressie. Het gaat hier om het onderdeel “hoofd” van “ref1234”, kortom het hoofd van Joko.

Deze structuur is compleet genoeg om simpele situaties te definiëren. Het is niet ondenkbaar dat bij complexere situaties deze structuur niet voldoende is. Echter de gebruikte mini-wereld is dusdanig simpel dat deze volledig binnen de voorgestelde structuur past en daarom voldoende voor deze toepassing. Uiteindelijk is de in hoofdstuk 3 gemaakte wereld omgezet naar deze frame structuur. Hierbij bleek het zelfs mogelijk om referenties en pad-expressies niet te gebruiken. Dit heeft uiteindelijk geleid tot een overzichtelijke en simpele dataset met framestructuren.

5 Implementatie

Het algoritme is op een eenvoudige manier geïmplementeerd. Hierbij is gekozen voor de programmeertaal PERL. Deze taal heeft uitgebreide support om tekst en strings te bewerken waardoor een snelle implementatie mogelijk was. Er was helaas te weinig tijd om een uitgebreide implementatie te maken, wat heeft

geresulteerd tot een implementatie van het een-woord stadium in combinatie met elementen uit het twee woord stadium. Dit betekent dat een degelijk twee-woord stadium implementatie niet aanwezig is.

5.1 Interne representaties

De implementatie kan overweg met de childes data zoals deze in voorgestelde frames is gerepresenteerd. Er worden interne representaties van gemaakt door gebruik te maken van een multi-level hash. Deze frames worden uitgebreid, door ze op te splitsen in alle mogelijke subframes en abstracties. Per frame wordt er bijgehouden in welke mate deze een abstractie is, dit gebeurt door het aantal keer dat een abstractie heeft plaats gevonden bij te houden. Om het zoeken naar frames eenvoudig en snel te maken, wordt er per frame een fingerprint gemaakt. Deze fingerprint is een string die uniek is per frame. De fingerprints worden gemaakt met behulp van MD5 hashing.

5.2 Database

In totaal zijn er drie indexes gebuikt. De eerste bestaat uit een hash tabel met als keys de fingerprints van de frames en bevat de referenties naar de frames zelf. De tweede hash tabel maakt gebruik van dezelfde keys en verwijst naar woordenlijsten die zijn voorgekomen in combinatie met deze frames. Deze woordenlijst bevat ook het aantal keer dat een woord is voorgekomen (icm. de frame). De laatste tabel is een lijst met woorden en hoe vaak deze in totaal zijn voorgekomen, dus ongeacht de in welke frame dat was. Het idee hierachter is dat met de laatste tabel een relevantie kan worden bepaald van een woord bij een frame. Als een woord (zoals “the”) in veel verschillende soorten frames voorkomt, zal de verhouding tussen het totaal aantal keren dat dit woord bij een frame is tegengekomen en hoe vaak het woord in totaal geregistreerd is kleiner zijn dan relevantere woorden zoals “block”.

5.3 Processing

De gehele dataset, in de vorm van de framerepresentaties (zie hoofdstuk 4) wordt ingelezen en toegevoegd aan de verschillende databases. Van ieder frame worden tevens alle abstracties uitgerekend en toegevoegd. Per frame worden de geregistreerde woorden toegevoegd en wordt de teller aangepast. Hierbij wordt verrekend voor de mate abstractie. Als een frame dus een sterke abstractie is, dan worden de woorden slechts voor een deel meegeteld en niet volledig. Dit betekent dat het aantal getelde woorden een floating point waarde kan hebben.

Als deze fase voltooid is, wordt een file gelezen met testdata (dit is een subset van de totale set, met de toevoeging van een aantal handmatig verzonden frames, dit om abstracties te kunnen testen). Van deze frames worden op dezelfde manier abstracties gemaakt. Vervolgens wordt er per frame gekeken welke woorden hier op van toepassing zijn geweest in het verleden. Dit wordt gedeeld door het totaal aantal keer dat een woord voorgekomen is, echter met de eis dat het woord zelf minimaal 10 keer is voorgekomen. Dit laatste voorkomt dat woorden die slechts enkele keren voorgekomen zijn extreem hoog voorkeuren krijgen, dit is een tekortkoming van de beperkte database. Iedere keer wordt per frame de top 10 van meeste relevante woorden getoond met hun waarde erbij.

6 Resultaten

In totaal zijn er 20 test frames gebruikt. In tabel 1 staan de top 5's van drie verschillende (test)frames. De eerste was een frame waar naar een BLOCK bij betrokken was, de tweede waarbij een CHAIR betrokken was en de derde waarbij een BALL betrokken was. De eerste vertoont duidelijk goede resultaten, het woord BLOCK staat helemaal bovenaan. Met een kleine factor twee voorsprong op het eerstvolgende woord. Bij de tweede zit alles veel dichterbij elkaar, maar staat het meest relevante woord nog steeds op plaats twee. Het derde voorbeeld vertoont echter een minder goede prestatie, hier komt het meest relevante woord pas op plek vijf tevoorschijn. Echter de lage waardes geven aan dat dit een frame is waar bestaat uit subframes die niet vaak zijn voorgekomen. Het kan dus goed zijn dat dit aan gebrek van data voortvloeit.

Woord	Waarde	Woord	Waarde	Woord	Waarde
block	42.8	look	150.2	that	12.4
a	23.7	chair	144.7	chair	11
chair	21.3	the	137.4	look	11.1
you	16.9	that	120.9	the	7.7
is	13,3	ball	115.5	ball	7.3

Tabel 1: Drie frame, met een 'block', 'stoel' en een 'bal'

Tabel 2 toont de resultaten van alle test frames. De tabel laat zien dat voor de twintig testframes 19 keer het meest relevante woord naar voren is gekomen in de top 10. Drie keer is het meest relevante woord in de top 1 geïndigd. Een frame heeft geen relevante woord teruggegeven.

Relevant woord in top:	Aantal voorgekomen
1	3
2	6
3	8
4	11
5	13
6	13
7	15
8	18
9	19
10	19

Tabel 2: Totaal resultaten

7 Discussie & conclusies

7.1 Het algoritme

Het lijkt erop dat het voorgestelde algoritme inderdaad in staat is woord betekenissen te leren door input van situaties en vergezeld audio/woorden. Hiermee lijkt dat het one-word stadium in ieder geval is te modelleren op basis van het

algoritme. Echter het blijkt wel dat niet altijd relevante woorden tevoorschijn komen. Dit duidt erop dat er nog veel kan worden verbeterd aan het algoritme en de statistiek waar deze gebruik van maakt. Het twee woord algoritme is helaas niet uitgebreid getest, maar de elementen die dat wel zijn lijken correct te werken (abstracties). Het is niet ondenkbaar dat er nog aanpassingen nodig zijn om dit algoritme goed functionerend te krijgen.

7.2 De dataset

Hoewel childes een zeer uitgebreide database heeft, is het niet eenvoudig en een langdurige proces om deze data aan te passen voor dit gebruik. Met name het omzetten van acties naar frame-representaties is een tijdrovende klus. Dit maakte het voor dit onderzoek, met beperkte tijd, niet mogelijk een zeer uitgebreide set te verkrijgen wat eigenlijk nodig is om de implementatie en het algoritme goed te testen. Een tweede probleem met de childes database is dat dit een set is van verschillende kinderen. Het zou beter zijn van enkele kinderen meerjarige sessies te hebben. Immers, niet ieder ouder gebruikt dezelfde manier van communiceren. Helaas is er op dit moment niet een dergelijk uitgebreide dataset beschikbaar en is er slechts een kleine kans dat deze ooit komt.

Een andere opmerking is dat er in de gebruikte dataset veel ruis zit, woorden die niet veel te maken hebben met de huidige situatie. Maar juist omdat deze ruis ook veel voorkomt bij verschillende situaties zou dit voor het algoritme geen probleem moeten zijn, omdat de ruis dan geen aanzienlijk aandeel deel vertegenwoordigen binnen de (sub)frames ten opzichte van de totale registratie van deze ruis.

7.3 Frame representatie

De framerepresentatie zoals hier voorgesteld voldeed aan de huidige dataset. Voor dit stadium van taalacquisitie lijkt het niet nodig om een uitgebreidere representatie te definiëren. Wel kan de vormgeving van een abstractie-boom een goede aanvulling zijn en meer inzicht geven over relaties tussen verschillende frames. Ook is het misschien een zinnige aanvulling om relaties tussen opeenvolgende frames te kunnen vastleggen (zie ook 7.5.2).

7.4 Implementatie

De implementatie van dit onderzoek is een vereenvoudigd model. Het is duidelijk dat daar nog veel aan verbeterd kan worden, echter is het voornaamste doel, een proof-of-concept, is redelijk gelukt. Zeker voor het one-word stadium. Het is gebleken dat de abstracties een positief invloed hebben op de resultaten. Hiermee wordt een kleine basis gelegd voor de verdere implementatie van het twee-woord stadium.

Uit de resultaten blijkt dat relevante woorden vaak in de top 10 terug komen. Hier dient wel nog de opmerking worden gemaakt, dat het meest relevante woord nu bepaald is door een persoon met volledige taal kennis en dit weinig zegt over de relevantie van de woorden in de miniwereld. Een goede manier om te bepalen welk woord echt relevant is, is niet aanwezig.

7.5 Future work

Bij dit onderzoek zijn een aantal aspecten aan het licht gekomen waarvan kan worden verwacht dat deze een positieve bijdrage kunnen leveren aan de implementatie en het algoritme. Deze worden in de komende paragrafen kort toegelicht.

7.5.1 Multi-Word

In dit onderzoek is het multi-word stadium nog niet grondig bekeken. Het lijkt realistisch dat dit stadium veel weg heeft van de recursieve versie van het two-word stadium. Op deze manier zouden er complexere boomstructuren kunnen ontstaan. Uiteindelijk kan dit model aansluiten bij het DOP-framework [3] om over te kunnen gaan naar complexe zinsstructuren.

7.5.2 Attention/Priming

Veel situaties bestaan uit een opeenvolgende reeks van acties waarin dezelfde objecten betrokken zijn. Bijvoorbeeld als er met een bal gespeeld wordt, dan zijn er verschillende acties die hiermee samenhangen, zoals het weggrollen van de bal, het oppakken ervan of het overgooien. Het huidige algoritme gaat steeds uit van een moment opname en gebruikt alleen de huidige situatie om een analyse uit te voeren. Om tot betere resultaten te komen, kan het zin hebben om de omliggende acties, en de objecten waar op dat moment de attentie ligt mee te nemen. Op deze manier kan de moment opname worden uitgebreid tot een tijdsspanne die de situatie in zijn totaliteit beter kan omschrijven. Dit kan positieve invloed hebben op de resultaten.

In het verlengde hiervan ligt priming. Het valt op dat in de chilles database het vaak voor komt dat er een uitspraak wordt gedaan waarna pas een actie op van toepassing is. Bijvoorbeeld kan een ouder zeggen “pak de bal” en daarna pas naar de bal wijzen. Het algoritme kan momenteel alleen overweg met uitspraken die volgen na een actie. Dit zou op een vergelijkbare manier opgelost kunnen worden.

7.5.3 Feedback

Uit de chilles database blijkt dat ouders vaak gebruik maken van positieve en negatieve feedback. Als een ouder vraagt om de bal te pakken en het kind doet dit ook, dan is het geen uitzondering dat de ouder dit beantwoord met een “very good!”. Het kan een verbetering opleveren door deze feedback mee te nemen in de analyse, bijvoorbeeld door een zwaarder gewicht aan de gebeurtenis toe te kennen als deze positief wordt beantwoord. Het is dan wel van belang dat de begrippen “goed” en “slecht” bekend zijn bij het kind.

Op latere leeftijden gebeurd het vaak dat een ouder vraagt wat een bepaald object is. Als het kind dit goed antwoord, dan wordt dit meestal met een positieve feedback beantwoord. Ook hiervoor geldt dat deze feedback een rol kan spelen bij de analyse.

7.5.4 Neuraal Netwerk

Enkele eigenschappen, met name die van de statistiek die voorkomt in de model vertonen overeenkomsten met neurale netwerken. Het lijkt niet onmogelijk om (een deel van) de statistiek door een neuraal netwerk te laten afhandelen. De analogie met de menselijke hersens wordt dan ook groter.

7.5.5 Statistiek & abstracties

De statistiek die gebruikt wordt om waarschijnlijkheden te bepalen berust nu op een lineair systeem met een aantal gewichten. Dit heeft tot zekere resultaten geleid maar er kan met alle redelijkheid worden verwacht dat op dit onderdeel nog behoorlijke winst te behalen moet zijn.

Referenties

- [1] Childes. <http://childes.psy.cmu.edu/>.
- [2] Mike de Kreek. Language acquisition and virtual grammars. Master's thesis, University of Amsterdam.
- [3] Bod et al. Rens bod, remko scha. data oriented language processing: an overview. 1996.
- [4] R. Scha. Pad expressies voor taal.