

Assignment 4

Marius Steller Imingen

Oppgave 1

a)

ICARUS:

Long term conceptual memory: Inneholder kjente konsepter. Input/perception blir matchet opp mot konsepter i LTCM for å skape et mer høy-nivås konsept som blir brukt videre i modellen. Konsepter i LTCM er organisert på en hierarkisk måte. Konsepter som er enkle og mer konkrete ligger lavere i hierarkiet, mens mer komplekse/høy nivås konsepter er øverst i hierarkiet. Konsepter høyere i hierarkiet bygger på foregående konsepter som er lavere i hierarkiet.

Short term conceptual memory: Også kalt belief memory. Alt som ligger STCM er instanser av LTCM. Når et input blir matchet med et lavnivå konsept i LTCM blir det hele tiden gjort om til mer komplekse konsepter. Når det ikke går ann "å gå høyere" i hierarkiet så blir konseptet lagret i STCM. Konseptene i STCM blir koblet sammen med handlinger og mål. Det nåværende målet som agenten prøver å oppnå og den nåværende situasjonen (konseptet fra STCM) blir brukt til å skape en handling.

Long term skill memory:

LTCM består av kjente handlinger som agenten kan ta, og handlingene beskriver hvordan de kommer til å endre miljøet. Hver handling er indeksert i henhold til hvilket mål de oppnår.

Short term skill memory: Også kalt problem memory. Inneholder handlinger som agenten skal utføre.

CLARION:

Action-centered subsystem:

Kontrollerer både interne og eksterne handlinger til agenten. Det vil si at den både kontrollerer hvordan agenten skal for eksempel bevege seg i miljøet, men også hvordan agentens interne operasjoner. Det nedre laget består av "Action Neural Networks" og det øvre laget består av handlingsregler.

Non-action-centered subsystem:

Opprettholder generell kunnskap ved hjelp av semantisk og episodisk minne. Henter ut minner og slutninger basert på action-centered subsystem. Det nedre laget består av nevrale nettverk som representerer implisitt kunnskap og det øvre laget er eksplisitt kunnskap. I det øvre laget er kunnskap delt opp i "chunks" hvor hver chunk kan bli koblet sammen hvis de har eksplisitt sammenheng. Hver chunk representerer konsepter som er linket til mikrofunksjoner(?) i det nedre laget.

Motivational subsystem:

Skaper motivasjon for systemet ved å gi feedback på handlinger slik at systemet vet om handlingen/prosessen var tilfredsstillende.

Meta-cognitive subsystem:

MCS er sterkt knyttet til det motiverende subsystemet og overvåker og modifiserer de andre subsystemene. MCS kan for eksempel sette mål, avbryte og endre pågående prosesser og endre parameteret i de andre subsystemene.

b)

1. Filosofi og motivasjon:

CLARION ble utviklet for å adressere problemer som mange andre kognitive arkitekturer ikke håndterer. Blant annet samspillet mellom kognisjon og motivasjon, implisitt-eksplisitt interaksjon og sammenhengen mellom det kognitive og det metakognitive. Spesielt for CLARION er samspillet mellom det implisittet og eksplisittet og bygger på resultater fra forsvarlig mange psykologiske studier rundt eksplisitt/implisitt læring, hukommelse, oppfatning.

I motsetning til CLARION er ICARUS en eldre arkitektur som bygger på mange av de samme prinsippene som andre arkitekturer, men et stort knippe forskjeller. ICARUS sier blant annet at kognisjon er bygget på oppfatning og handling. ICARUS prøver å bygge arkitekturer på mange funn i psykologisk forskning, men prøver ikke å matche resultatene med menneskelige resultater for å prøve å oppnå en bredere dekning av kognisjon.

2. Memory:

I CLARION er minne representert implisitt og eksplisitt som alt annet i denne arkitekturen. ICARUS har langtidsminnet representert hierarkisk hvor konsepter bygger på konsepter lavere i hierarkiet. I tillegg er korttidsminnet i ICARUS instanser av langtidsminner, men i CLARION er korttidsminnet bare en "buffer"/venteplass mellom NACS og ACS.

3. Syklus:

I CLARION fungerer alt mer parallelt og samtidig. For eksempel kan det motiverende subsystemet avbryte prosesser som foregår i de andre subsystemene og oppdatere parameteret. I ICARUS er det mer sekvensiell prosessering hvor man får input, matcher det mot et konsept, finner hvilken handling man skal utføre, eventuell læring og deretter utførelse av handlingen. I ICARUS så har alle subsystemene sammenheng og er eksplisitt avhengig av hverandre. I CLARION er fortsatt subsystemene avhengige av hverandre, men arkitekturen virker mer modulær og hvert subsystem har interne prosesser som er avhengig av andre interne prosesser.

Oppgave 2

a)

- **Analogy:** Sammenligning mellom to “ting” hvor man bruker kunnskap og forståelse om den ene tingen til å dra slutninger og skape forståelse om den andre tingen.
- **Base:** Den tingen man allerede vet mye om. Det er kunnskapen og forståelsen av basen som skal “overføres” til “target” som man prøver å forstå
- **Bokstavelig likhet:** Det er helt like aspekter mellom to ting som sammenlignes. Melk er som vann fordi begge er av typen væske er en bokstavelig likhet.
- **Relasjonell likhet:** Det finnes aspekter i begge tingene som sammenlignes som kan relateres. Et atom er som solsystemet. Ikke fordi de er helt like eller kan sammenlignes bokstavelig, men fordi det er en relasjon mellom noen av aspektene i begge tingene for eksempel at planeter snurrer rundt sola og elektroner snurrer rundt nukleusen.

b)

Mapping	Antall attributter mappet til målet	Antall relasjoner mappet til målet	Eksempel
Bokstavelig likhet	Mange	Mange	Solsystemet K5 er som Melkeveien
Analogi	Få	Mange	Et atom er som solsystemet vårt
Abstraksjon	Få	Mange	Atomene er et sentralt kraftsystem
Anomali	Få	Få	Kaffe er som solsystemet vårt

c)

Systematisitets-prinsippet (SP) beskriver hvordan mennesker skaper gode analogier som beskriver sammenheng mellom to domener der fakta i det ene domene vil ha samme relasjon i det andre domenet. En analogi er ikke en samling av uavhengige fakta om hvert domene. For eksempel hvis det er sammenheng mellom to fakta i et domene hvor en bit endrer seg basert på endringer i den andre biten, så vil det være samme relasjon mellom de samme bitene i det andre domenet.

d)

- Discard attributes of objects
- Preserve relations between objects
- Systematicity principle

e)

MAC/FAC:

- En måte å håndtere kompleksiteten rundt “similarity-based retrieval” altså hvordan man henter opp kunnskap fra minne for å forstå eller skape en analogi. MAC/FAC er en måte å filtrere bort uønskede minner som blir hentet frem under mapping prosessen. MAC er den første delen av modellen som er et enkelt filter som filtrerer bort uønskede minner og beholder de minnene som virker relevante. De relevante minnene blir da videreført til FAC delen av modellen hvor de blir prosessert gjennom en mer sensitiv og nøyaktig matcher. Basically er MAC den delen som henter ut alle relevante kandidater for basen i analogi resonnering, mens FAC velger ut en bestemt base ut av alle kandidatene.
- MAC er det samme som Retriever Modulen og FAC er det samme som Analogy Engine Modulen

Oppgave 3

a)

Foreslå løsninger på problemer basert på løsninger fra tidligere problemer som ligner på det problemet vi holder på med nå.

b)

CBR kan knyttes til menneskelig kognisjon ved at det er vanlig at mennesker bruker erfaring med lignende situasjoner til å dra slutninger om en situasjon, forstå noe nytt som ligner på noe man har erfart eller løst tidligere eller forutsi hva som kommer til å skje.

Eksempel 1:

Petter skal lage spaghetti, men har ikke spagettisaus i kjøleskapet. Han vet at spagettisaus hovedsakelig er laget av tomater så han velger å bruke en boks med hermetiserte tomater til å lage en saus. Etter å ha smakt litt på den finner han ut at det kanskje mangler noe i sausen for at det skal ligne på spaghettsausen han vanligvis bruker. Han legger til litt forskjellig krydder og konkluderer med at nå ligner det mer på spagettisaus.

Eksempel 2:

En programmerer som støter på et problem, kan bruke løsninger han tidligere har brukt på lignende problemer og tilpasse løsningen til det nåværende problemet.

c)

MOP = Memory organisation packages.

MOP holder på generell kunnskap og organiserer på en hierarkisk måte spesifikke erfaringer av den generelle kunnskapen kalt "cases".

d)

Indekser bør beskrive uvanlige sider av en situasjon. For eksempel forskjeller mellom en situasjon og mange andre situasjoner. Avvik fra normen.

I henhold til normer kan indekser være et brudd på normen, en spesialisering av normen eller en generalisering av normen.

En indeks bør forutsi brukbarheten til en case i et gitt scenario.