

Silnik sztucznej inteligencji Novamente

Silnik sztucznej inteligencji Novamente, nowatorski system oprogramowania AI, jest krótko omawiany. Novamente to zintegrowany projekt ogólnej sztucznej inteligencji, który integruje aspekty wielu wcześniejszych projektów i paradygmatów AI, w tym symboliczne, probabilistyczne, ewolucyjne programowanie i wzmacnianie uczenia się; ale jego ogólna architektura jest wyjątkowa, czerpiąc z idei teorii systemów dotyczących złożonej dynamiki mentalnej i powiązanych wzorców wyłaniających się. Rozdział omawia zarówno koncepcyjne modele umysłu i inteligencji, które zainspirowały projekt systemu, jak i konkretną architekturę Novamente jako systemu oprogramowania.

Wprowadzenie

Przedstawiamy silnik sztucznej inteligencji Novamente, zintegrowany projekt AGI. Novamente opiera się na ponad dekadzie badań i został opracowany na poziomie koncepcyjnym i projektowania oprogramowania w znacznym stopniu. Przez półtora dekady badań stworzyliśmy teoretyczne podstawy do projektowania systemów AI wyświetlających adaptacyjną, autonomiczną sztuczną inteligencję i jesteśmy w trakcie opracowywania wysoce oryginalnego, bezprecedensowego systemu oprogramowania na tym fundamencie. Novamente łączy aspekty wielu poprzednich paradygmatów AI, takich jak systemy agentów, programowanie ewolucyjne, uczenie się przez wzmacnianie, automatyczne dowodzenie twierdzeń i rozumowanie probabilistyczne. Jest jednak wyjątkowy w swojej ogólnej architekturze, która stawia czoła problemowi tworzenia holistycznego umysłu cyfrowego w sposób bezpośredni, który nie został wcześniej wykonany. Novamente łączy wspólną, przyjazną dla integrującej SI reprezentację wiedzy z szeregiem różnych procesów poznawczych, które współpracują, działając na podstawie tej wiedzy. Ta konkretna kombinacja skutkuje złożonym i unikalnym systemem oprogramowania: autonomicznym, samoadaptacyjnym, uczącym się doświadczalnie systemem, w którym współpraca między procesami poznawczymi umożliwia pojawienie się ogólnej inteligencji. Krótko mówiąc, Novamente jest rodzajem „umysłu cyfrowego”. Jednym ze sposobów, w jaki Novamente różni się od wielu innych podejść do AGI, jest to, że jest rozwijane przede wszystkim w kontekście komercyjnym, a nie akademickim. Chociaż stwarza to pewne wyzwania pod względem priorytetyzacji rozwoju różnych aspektów systemu, uważamy, że było to niezwykle wartościowe podejście, ponieważ oznaczało, że na każdym etapie rozwoju systemu był on testowany w wymagających aplikacjach w świecie rzeczywistym. Dzięki naszej pracy nad komercyjnymi aplikacjami wczesnych, częściowych wersji systemu Novamente, staliśmy się bardzo świadomi pilnej potrzeby Sztucznej Inteligencji Ogólnej w różnych branżach. Wiele mówi się o epoce informacji, odkrywaniu wiedzy i potrzebie narzędzi, które są wystarczająco inteligentne, aby pozwolić ludzkim ekspertom poradzić sobie z nieporęcznymi ilościami informacji we współczesnym świecie biznesu i nauki. Wierzymy, że prawdziwa odpowiedź na te wymagania analityczne leży w AGI, ponieważ obecne wąskie techniki nie są w stanie właściwie zintegrować heterogenicznej wiedzy, wyciągnąć inteligentnych wniosków z tej wiedzy i, co najważniejsze, spontanicznie generować nowej wiedzy o świecie. W momencie pisania tego tekstu system Novamente jest całkowicie zaprojektowany i częściowo wdrożony. Można go stosować do złożonych problemów w określonych dziedzinach, takich jak bioinformatyka i odkrywanie wiedzy, już teraz, i będzie dawał coraz większą funkcjonalność, w miarę jak więcej projektu będzie wdrażanych i testowanych. Oczywiście projekt ciągle zmienia się w szczegółach, zgodnie z nieuchronnie wyciągniętymi wnioskami w trakcie wdrażania. Jednak te ulepszenia na poziomie szczegółów zachodzą w ramach ogólnej struktury projektu Novamente, która – jak dotąd – okazała się dość wydajna i solidna.

System AGI Novamente

Biorąc pod uwagę pilną potrzebę AGI z praktycznego punktu widzenia, zaskakująco niewiele ostatnio prac badawczo-rozwojowych ukierunkowanych konkretnie na problem AGI. Dyscyplina AI rozpoczęła się od marzeń o AGI, ale przez jakiś czas była zdominowana przez różne formy wąskiej AI, w tym AI oparte na wnioskowaniu logicznym, sieci neuronowe, programowanie ewolucyjne, systemy eksperckie, robotykę, widzenie komputerowe itd. Wiele z tych wąskich systemów oprogramowania AI jest doskonałych w tym, co robią, ale łączy je skupienie się na jednym konkretnym aspekcie funkcji umysłowej, a nie na integracji licznych aspektów funkcji umysłowej w celu utworzenia spójnego, holistycznego, autonomicznego, usytuowanego systemu poznawczego. Sztuczna inteligencja ogólna wymaga innego rodzaju skupienia. Autorzy i ich współpracownicy od kilku lat pracują razem nad problemem stworzenia odpowiedniego projektu prawdziwego systemu AGI, mającego na celu w szczególności położenie podwalin pod AGI. Pracowaliśmy razem w latach 1998-2001 nad prototypem AGI o nazwie Webmind, który został zastosowany w domenach zarządzania wiedzą i prognozowania finansowego; a od 2001 roku współpracujemy nad Novamente. Projekt Novamente obejmuje aspekty wielu poprzednich paradygmatów AI, takich jak programowanie ewolucyjne, logika symboliczna, systemy agentowe i rozumowanie probabilistyczne. Jest jednak niezwykle innowacyjny w swojej ogólnej architekturze, która stawia czoła problemowi „tworzenia całego umysłu” w sposób bezpośredni, który nie został wcześniej wykonany. Podstawowe zasady leżące u podstaw projektu systemu wywodzą się z nowej teorii umysłu opartej na złożonych systemach, zwanej „modelem psynef”, który został opracowany przez autora w serii interdyscyplinarnych rozpraw badawczych opublikowanych w latach 1993-2001. To, do czego doprowadził nas model psynef, to nie konwencjonalny program AI ani konwencjonalny system wieloagentowy. Mówimy raczej o autonomicznym, samoorganizującym się, samoewoluującym systemie AGI, z własnym rozumieniem świata i zdolnością do nawiązywania relacji z ludźmi na poziomie „umysł-umysł”, a nie „program-oprogramowanie-umysł”. Projekt Novamente jest obszerny, ale obecnie wdrożona implementacja zawiera już wiele istotnych aspektów. Ze względu na głębię szczegółów w projekcie i bogate doświadczenie w prototypowaniu, jakie zespół inżynierów Novamente zdobył w latach 1997-2004, czas potrzebny na ukończenie implementacji będzie krótszy, niż można by się spodziewać, biorąc pod uwagę skalę zadania: szacujemy, że zajmie to 1-2 lata. Po fazie inżynierskiej nastąpi faza interaktywnego nauczania systemu Novamente, jak odpowiadać na zapytania użytkowników oraz jak użytecznie analizować i organizować dane. Końcowym rezultatem tego procesu nauczania będzie autonomiczny system AGI, zorientowany na pomoc ludziom w kolektywnym rozwiązywaniu problemów pragmatycznych. W tym rozdziale omówiono projekt Novamente AGI i niektóre kwestie związane z jego wdrażaniem, nauczaniem i testowaniem. Po drodze krótko poruszymy również niektóre kwestie praktycznego zastosowania i omówimy sposoby, w jakie nawet wczesne wersje Novamente zapewnią innowacyjne, uderzająco skuteczne rozwiązanie problemu pomocy analitykom ludzkim w zrozumieniu, organizacji i analizie danych w wielu złożonych domenach.

Novamente do zarządzania wiedzą i analizy danych

Sam framework Novamente AGI jest wysoce ogólny i może być stosowany w różnych kontekstach aplikacji. Na przykład można sobie wyobrazić Novamente jako silnik poznawczy zaawansowanego systemu robotycznego; w rzeczywistości opracowano wstępny projekt hybrydyzacji Novamente z „Reference Model Architecture” Jamesa R. Albusa dla robotyki. Początkowo jednak planujemy wdrożyć i wdrożyć Novamente w kontekście zarządzania wiedzą i analizy danych. Uważamy, że Novamente ma pewne ważne korzyści dla tych obszarów aplikacji. Obecna wersja Novamente jest używana do zarządzania i analizy informacji bioinformatycznych, w szczególności baz danych genomicznych i proteomicznych oraz eksperymentalnych zestawów danych; oraz do rozumienia tekstu w dziedzinie bezpieczeństwa narodowego. W ciągu najbliższych kilku lat, kontynuując naszą obecną pracę nad aplikacją, przewidujemy znacznie szerszą inicjatywę zastosowania systemu do zarządzania i analizy

informacji w wielu domenach. Wdrożenie Novamente do zarządzania wiedzą i analizowania jej wymaga zwrócenia uwagi na wiele różnych kwestii, z których większość mieści się w ogólnych kategoriach źródeł danych i interakcji człowiek-komputer. Optymalny sposób radzenia sobie z takimi problemami zależy od domeny. W przypadku zastosowań bioinformatycznych przyjęliśmy podejście kierujące się szczególnymi potrzebami biologów analizujących zestawy danych generowane za pomocą wysokoprzepustowego sprzętu genomicznego i proteomicznego. Jeśli chodzi o źródła danych, po podjęciu decyzji o przyjęciu podejścia integracji wiedzy najtrudniejszą kwestią pozostaje traktowanie danych w języku naturalnym („niestrukturyzowany tekst”). Novamente można stosować na dwa uzupełniające się sposoby:

- Zorientowany na „wyszukiwanie informacji”, w którym tekst jest traktowany jako seria znaków lub seria słów i analizowany statystycznie;
- Zorientowany na przetwarzanie języka naturalnego (NLP), w którym podejmowana jest próba analizy zdań w tekstach i wyodrębnienia ich znaczeń do formy relacji semantycznej.

Podejście do wyszukiwania informacji jest odpowiednie, gdy mamy dużą ilość tekstu i ograniczony czas przetwarzania na jego obsługę. Podejście NLP jest bardziej wyrafinowane i bardziej kosztowne obliczeniowo. Wspólną słabością istniejących algorytmów i ram NLP jest integracja rozumienia semantycznego i pragmatycznego z analizą języka składniowego. Projekt Novamente przewyższa ten problem, przeprowadzając analizę składniową za pomocą unifikacji logicznej, procesu, który automatycznie włącza dostępną wiedzę semantyczną i pragmatyczną do swojego zachowania. W chwili pisania tego tekstu nie zintegrowaliśmy jeszcze komponentu NLP Novamente z unifikacją logiczną, ale nasze doświadczenie z podobną implementacją w systemie Webmind daje nam tutaj uzasadnione zaufanie. Jeśli chodzi o interakcję z użytkownikami ludzkimi, projekt Novamente obsługuje wiele różnych modalności, w tym konwencjonalne zapytania wyszukiwawcze i NLP, formularze internetowe, dynamiczną wizualizację i automatyczne generowanie raportów. Najbardziej innowacyjny projekt, jaki opracowaliśmy do interakcji człowiek-Novamente, obejmuje jednak interaktywną konwersację z wykorzystaniem kombinacji języka naturalnego i języka formalnego. Rzeczywistość dialogu mieszanego języka ludzkiego i formalnego zostanie zdefiniowana poprzez praktyczne eksperymenty z systemem Novamente w każdym innym kontekście. Pod względem interfejsu użytkownika systemy AGI mogą znajdować się na zapleczu konwencjonalnych aplikacji programowych – ale mogą być również używane do obsługi radykalnie nowych form interakcji człowiek-komputer. Jako przykład tego proponujemy opracowanie interfejsu do konwersacji mieszanego języka ludzkiego i formalnego. Hipotetyczny przykład tego rodzaju dialogu podano w Tabeli 3. Ten rodzaj dialogu nie wymaga, aby ludzie nauczyli się myśleć jak komputery, ani nie wymaga, aby komputery próbowały dokładnie naśladować ludzi. Zamiast tego obejmuje on wchodzenie ludzi i komputerów w tryby poznania i opisu drugiego oraz nauczanie i uczenie się od siebie nawzajem w tym procesie.

Technologie oprogramowania wspomagającego

Głębokie koncepcje leżące u podstaw AGI są matematyczne i teoretyczne, niezależne od konkretnych technologii oprogramowania lub sprzętu. Jednak wykonalność inżynierii, testowania i wdrażania systemów AGI oraz stosowania ich w kontekście analizy/zapytań/zarządzania danymi jest mniej abstrakcyjna i zależy krytycznie od obecności odpowiednich technologii wspomagających. Dwie kategorie technologii wspomagających – rozproszona architektura oprogramowania i integracja bazy danych – są szczególnie krytyczne dla praktycznych wdrożeń AGI. Są one przedmiotem niniejszej sekcji.

Rozproszona architektura oprogramowania dla zintegrowanej AI

Marvin Minsky, jeden z ojców AI, zapisał się w historii jako przypuszczenie, że sztuczna inteligencja ogólna na poziomie ludzkim prawdopodobnie mogłaby działać na komputerze 1-megahercowym, gdyby tylko znano odpowiedni projekt. Nasz optymizm w kwestii AGI, choć silny, ma nieco inną naturę. Wierzymy, że technologia obliczeniowa dopiero teraz osiąga punkt, w którym zaawansowane oprogramowanie AGI staje się możliwe. Posiadanie prawidłowego projektu jest nadal najważniejszą rzeczą; ale właściwy projekt bez odpowiedniej platformy sprzętowej i systemu operacyjnego nigdy nie opuści świata teorii. W latach 80. wyspecjalizowany sprzęt superkomputerowy był jednym z tematów dnia. Maszyny przetwarzania wektorowego Craya zrewolucjonizowały fizykę obliczeniową i pokrewne obszary, a architektura równoległej maszyny Connection Machine firmy Thinking Machines Corp. wydawała się gotowa zrobić to samo dla sztucznej inteligencji. Stało się jednak tak, że maszyna Connection Machine nie była w stanie nadążyć za niezwykle szybkim rozwojem konwencjonalnego sprzętu von Neumanna i technologii łączenia tradycyjnych maszyn w sieć. Ostatnia maszyna Connection Machine stworzona przed rozwiązaniem Thinking Machine Corp., CM-5, była mniej radykalna niż jej poprzednicy, ponieważ opierała się na tradycyjnych procesorach połączonych w niezwykle ścisły sposób. Podobnie, najpotężniejsze dzisiejsze superkomputery, IBM, są w rzeczywistości rozproszonymi komputerami pod spodem – są specjalnie skonstruowanymi sieciami stosunkowo konwencjonalnych procesorów, a nie unikalnymi procesorami. Mając czystą kartę, jasne jest, że można zaprojektować znacznie bardziej odpowiednią dla AGI platformę sprzętową niż architektura von Neumanna. Konceptualnie rzecz biorąc, uważamy, że Connection Machine był na dobrej drodze. Jednak nowoczesna technologia sieciowa i rozproszona architektura oprogramowania odsunęły architekturę von Neumanna daleko od jej korzeni i uważamy, że można wykorzystać współczesną technologię do tworzenia rozproszonych platform AI o znacznej mocy i elegancji. Matematyczne struktury i dynamika Novamente mogą być implementowane na wiele sposobów poza DINI; a DINI może być używane jako platforma dla wielu systemów oprogramowania innych niż Novamente. Jednak Novamente i DINI pasują do siebie naturalnie. Kluczowymi komponentami DINI są:

- „Analityczne klastry” maszyn – każdy klaster przeprowadza analizę poznawczą danych i odpowiednio tworzy nowe dane
- Ogromne schronienie danych integrujące wiele baz danych i zapewniające ujednolicony interfejs wyszukiwania
- Proces „Fisher”, wyodrębniający odpowiednie dane z banku danych do klastrów analitycznych
- Procesy „Miner”, wyodrębniające informacje z zewnętrznych baz danych do banku danych
- Pająki sieciowe nieustannie gromadzą nowe informacje
- Proces „Mediator”, łączący wyniki z wielu klastrów analitycznych w banku danych
- Interfejsy do wprowadzania wiedzy przez ludzi
- Interfejsy do prostych i zaawansowanych zapytań
- Oprogramowanie pośredniczące J2EE do komunikacji międzyprocesowej, skalowalności, kontroli transakcji, równoważenia obciążenia, ogólnej adaptacyjnej kontroli systemu

Najsubtelniejszymi procesami są tutaj Fisher i Mediator.

Fisher może odpowiadać na konkretne zapytania o informacje przesłane przez klastry analityczne. Ale musi też być w stanie działać autonomicznie – używać heurystyki, aby zgadywać, jakie dane mogą być interesujące dla klastrów analitycznych, na podstawie podobieństwa do danych o najwyższym priorytecie w klastrach analitycznych. Mediator istnieje z powodu faktu, że różne klastry analityczne,

działające na tych samych danych i myślące o tych samych problemach, mogą generować sprzeczne lub uzupełniające się wnioski. Wymagane jest uzgodnienie tych wniosków w jednym widoku w scentralizowanej bazie danych. Gdy uzgodnienie jest nieprawdopodobne, w scentralizowanej bazie danych przechowywane są liczne widoki. Uzgodnienie odbywa się za pośrednictwem logicznego procesu „rewizji przekonań”, przy użyciu formuł wyprowadzonych ze składnika wnioskowania pierwszego rzędu Novamente.

Integracja bazy danych i integracja wiedzy

W architekturze DINI dużą rolę odgrywa komponent „banku danych”. Większość informacji w banku danych DINI zostanie utworzona przez same procesy AGI. Jednak ogólnie rzecz biorąc, będzie tam również duża ilość danych z innych źródeł. Istnieje ogromna liczba baz danych, utworzonych przez różne organizacje w środowisku akademickim, przemyśle i rządzie – są one przydatne dla AGI w wykonywaniu praktycznych funkcji zarządzania wiedzą, zapytań i analiz, a także w budowaniu jego inteligencji i zrozumienia świata. Jednak informacje w bazach danych rzadko występują w formacie, który można bezpośrednio wprowadzić do AGI, które jest jeszcze w fazie uczenia się. Ostatecznie dojrzały AGI powinien być w stanie strawić bazę danych w stanie surowym, samodzielnie ustalając semantykę struktury schematu. Jednak na obecnym etapie bazy danych wymagają znacznego wstępnego przetwarzania, aby były przydatne dla systemów AGI. To wariant problemu „integracji baz danych”: jak pobrać informacje z wielu baz danych i udostępnić je w sposób ujednolicony. Poprzez badanie podejść do integracji baz danych podejmowanych w różnych domenach, doszliśmy do rozróżnienia czterech różnych ogólnych strategii:

Federacja tworzy wspólny interfejs graficzny użytkownika dla oddzielnych baz danych. Amalgamacja tworzy formalne mapowania między schematami różnych baz danych.

Tłumaczenie schematu tworzy nową bazę danych RDB łączącą informacje z wielu baz danych.

Integracja wiedzy tworzy translatora mapującego zawartość bazy danych na „uniwersalną formalną reprezentację wiedzy”.

Stosowanie systemów AGI do informacji z bazy danych wymaga najbardziej solidnego podejścia: integracji wiedzy. W tym podejściu wiedza jest wyodrębniana z baz danych do niezależnego od schematu języka formalnego. Przykładem tego jest podejście Cycorp do integracji wiedzy, które obejmuje konwersję wiedzy na ich język CycL. Jednak ze względów technicznych uważamy, że podejście CycL nie jest wystarczająco elastyczne, aby obsługiwać podejścia AI skoncentrowane na logice nieformalnej. Jedną z praktycznych i niezwykle elastycznych form integracji wiedzy może być język XML. Stworzyliśmy specjalny XML DTD dla Novamente, który składa się z zestawu tagów odpowiadających wewnętrznej reprezentacji wiedzy Novamente. Aby zintegrować bazę danych z Novamente, podstawowym wymaganym krokiem jest napisanie kodu, który eksportuje zaangażowane tabele danych relacyjnych do XML ustrukturyzowanego przez Novamente DTD. Jednak aby uzyskać najlepsze wyniki, należy wcześniej przeprowadzić znaczący proces „amalgamacji”, aby mieć pewność, że różne nakładające się bazy danych zostaną wyeksportowane do struktur Novamente w sposób w pełni semantycznie kompatybilny. Te same ramy programowe można wykorzystać do obsługi podejść AI innych niż Novamente; trzeba by po prostu utworzyć odpowiednie schematy transformacji XML, aby przetłumaczyć Novamente DTD na DTD odpowiednie dla innego systemu AI.

Czym jest sztuczna inteligencja ogólna?

Aby zrozumieć, dlaczego i w jaki sposób dążymy do świętego Graala AGI, konieczne jest zrozumienie, czym jest AGI i jak różni się od tego, co większość badaczy w dziedzinie AI nazywa „inteligencją”. Gdyby

wąska AI nie istniała, nie potrzebowalibyśmy w ogóle terminu „inteligencja ogólna” – po prostu używalibyśmy terminu „inteligencja”. Kiedy mówimy o inteligencji ludzkiej, w końcu implicite mamy na myśli inteligencję ogólną. Pojęcie IQ pojawiło się w psychologii jako próba uchwycenia czynnika „inteligencji ogólnej” lub czynnika g, abstrahując od zdolności w określonych dyscyplinach. Wąska AI jednak subtelnie zmodyfikowała znaczenie „inteligencji” w kontekście obliczeniowym, oznaczając zasadniczo zdolność do wykonywania dowolnego konkretnego zadania, które jest zwykle uważane za wymagające znacznej inteligencji u ludzi (szachy, diagnostyka medyczna, rachunek różniczkowy i całkowy, . . .). Z tego powodu wprowadziliśmy wyraźne pojęcie Sztucznej Inteligencji Ogólnej, aby odnieść się do czegoś mniej więcej analogicznego do tego, co czynnik g ma mierzyć u ludzi. Kiedy rozróżnia się wąską inteligencję od inteligencji ogólnej, historia dziedziny AI przyjmuje uderzający wzór. AI rozpoczęła się w połowie XX wieku od marzeń o sztucznej inteligencji ogólnej – o tworzeniu programów z możliwością uogólniania swojej wiedzy w różnych domenach, refleksji nad sobą i innymi, tworzenia fundamentalnych innowacji i spostrzeżeń. Ale na początku lat 70. AGI nie osiągnęła nawet zbliżenia się do urzeczywistnienia, a badacze i komentatorzy byli sfrustrowani. AGI zeszła na dalszy plan, z wyjątkiem garstki projektów badawczych. Z czasem AGI zyskało wyraźnie złą reputację, a wszelkie rozmowy o AGI zaczęto traktować z ekstremalnym sceptycyzmem. Jednak dzisiaj sprawy wyglądają nieco inaczej niż na początku lat 70., kiedy AGI straciło swój blask. Nowoczesne sieci komputerowe są nieporównywalnie bardziej wydajne niż najlepsze superkomputery z początku lat 70., a infrastruktura oprogramowania również znacznie się rozwinęła. Technologie wspierające AGI są już dostępne, w znacznie większym stopniu niż w czasach wczesnych niepowodzeń marzenia o AGI. A teraz wiadomo znacznie więcej o matematyce poznania, częściowo dzięki pracom nad wąską sztuczną inteligencją, ale także dzięki rewolucyjnym postępom w neuronauce i psychologii poznawczej. Wierzmy, że nadszedł czas, aby przewyciężyć nagromadzony sceptycyzm wobec AGI i poważnie posunąć się w kierunku AGI. Wniosek jest jasny: te same postępy w technologii komputerowej, które dały nam obecny nadmiar informacji, umożliwiają technologię AGI, która pozwoli nam skutecznie zarządzać nadmiarem informacji, a tym samym przekształcić go w zaletę, a nie frustrację. Uważamy za bardzo sensowne porównanie AGI do obecnie popularnej dziedziny nanotechnologii. Podobnie jak nanotechnologia, uważamy, że AGI jest „zaledwie problemem inżynierskim”, choć z pewnością bardzo trudnym. Nauka o mózgu i teoretyczna informatyka wyraźnie sugerują, że AGI jest możliwe, jeśli dojdzie się do właściwego projektu². Projekt Novamente nie jest jedynym istniejącym wysiłkiem mającym na celu wykorzystanie „właściwego projektu” do stworzenia prawdziwej AGI, ale jest jednym z niewielu takich wysiłków i uważamy, że jest bardziej zaawansowany niż jakikolwiek inny. Ze względu na zagażowaną historię AI, zanim przejdziemy do szczegółów projektu AGI Novamente, uważamy, że warto poświęcić kilka akapitów na wyjaśnienie naszego pojęcia ogólnej inteligencji. Czytelnik jest proszony o pamiętanie, że „inteligencja” jest nieformalnym pojęciem języka ludzkiego, a nie ściśle zdefiniowanym pojęciem naukowym; jej znaczenie jest złożone, niejednoznaczne i wieloaspektowe. Aby jednak tworzyć użyteczne aplikacje AGI, potrzebujemy praktycznej definicji roboczej celu AGI – a nie kompleksowego zrozumienia wszystkich wymiarów pojęcia inteligencji w języku naturalnym.

Czym jest inteligencja ogólna?

Jedną z dobrze znanych charakterystyk ogólnej inteligencji sztucznej jest słynny „test Turinga” Alana Turinga – „napisz program komputerowy, który może symulować człowieka w konwersacyjnej wymianie tekstowej” [67]. Test ten służy do przedstawienia teoretycznego punktu widzenia, że inteligencja jest definiowana przez zachowanie, a nie przez mistyczne cechy, więc jeśli program mógłby zachowywać się jak człowiek, powinien być uważany za tak inteligentny jak człowiek. Jednak test Turinga nie jest przydatny jako przewodnik po praktycznym rozwoju AGI. Naszym celem nie jest stworzenie symulowanego człowieka, ale raczej stworzenie nieludzkiego cyfrowego inteligentnego systemu - takiego, który będzie uzupełniał ludzką inteligencję, wykonując zadania analizy danych i

zarządzania daleko wykraczające poza możliwości ludzkiego umysłu; i takiego, który będzie współpracował z ludźmi w sposób, który wydobywa najlepsze aspekty zarówno ludzkich, jak i cyfrowych smaków ogólnej inteligencji. Podobnie można by pomyśleć, że testy ludzkiego IQ – zaprojektowane w celu oceny ogólnej inteligencji człowieka – mogą być przydatne w ocenie ogólnej inteligencji programów komputerowych. Ale po bliższym przyjrzeniu się okazuje się, że jest to wątpliwa propozycja, ponieważ testy ludzkiego IQ działają całkiem dobrze w obrębie jednej kultury, a znacznie gorzej w różnych kulturach [54] – o ile gorzej będą działać w różnych typach programów AGI, które mogą się różnić tak samo, jak różne gatunki zwierząt? W [27] podano prostą roboczą definicję inteligencji, opartą na różnych pomysłach z psychologii i inżynierii. Matematyczna formalizacja definicji wymaga więcej notacji i maszyn, niż możemy tutaj wprowadzić, ale werbalnie sedno jest następujące:

Inteligencja ogólna to zdolność do osiągnięcia złożonych celów w złożonych środowiskach.

Praca nad silnikiem sztucznej inteligencji Novamente została również zmotywowana ściśle powiązaną wizją inteligencji przedstawioną przez Pei Wanga w jego rozprawie doktorskiej i powiązanych pracach ([69], również w tym tomie). Definicja Wanga zakłada, że inteligencja ogólna to „[Z]dolność systemu przetwarzania informacji do dostosowywania się do swojego otoczenia przy niewystarczającej wiedzy i zasobach”. Definicje Wanga i Goertzela są komplementarne. W praktyce system AGI musi być w stanie osiągnąć złożone cele w złożonych środowiskach przy niewystarczającej wiedzy i zasobach. Badacz AI Shane Legg zasugerował, że tę koncepcję inteligencji należy nazwać „cybernacją”, aby uniknąć uwikłania w niejednoznaczności nieformalnego pojęcia „inteligencji”. Podstawowym aspektem definicji „złożonych celów w złożonych środowiskach” jest mnogość słów „cele” i „środowiska”. Pojedynczy złożony cel nie wystarczy, a pojedyncze wąskie środowisko nie wystarczy. Program do gry w szachy nie jest ogólną inteligencją, ani też nie jest nią silnik eksploracji danych, który nie robi nic poza wyszukiwaniem wzorców w bazach danych informacji konsumenckich, ani też program, który może niezwykle sprytnie manipulować wieloma aspektami mikroświata skonstruowanego przez badacza (chyba że mikroświat jest znacznie bogatszy i bardziej zróżnicowany niż jakikolwiek dotychczas skonstruowany). Ogólna inteligencja musi być w stanie wykonywać wiele różnych zadań w różnych kontekstach, uogólniając wiedzę z jednego kontekstu do drugiego i budując niezależne od kontekstu i zadania pragmatyczne rozumienie siebie i świata. Można również zbadać jeden poziom głębiej niż te definicje, zagłębiając się w subtelność relacji między inteligencją uogólnioną i wyspecjalizowaną. Opierając się na pomysłach z formalnej teorii złożoności (po powiązane, bardziej rygorystycznie rozwinięte pomysły), można zdefiniować system jako w pełni inteligentny ogólnie dla złożoności N , jeśli może osiągnąć dowolny cel złożoności N w dowolnym środowisku złożoności N . I tutaj robi się ciekawie, ponieważ jest jasne, że pełna inteligencja ogólna jest tylko jednym aspektem ludzkiej inteligencji ogólnej. Sposób, w jaki wydaje się działać ludzki mózg, jest następujący:

- część jej architektury jest zorientowana na osiągnięcie pełnej ogólnej inteligencji dla małego N (tj. ludzie mogą rozwiązać każdy stosunkowo prosty problem)
- część jej architektury jest zorientowana na zwiększenie zdolności rozwiązywania problemów dla celów i środowisk o złożoności N tak dużej, że pełna ogólna inteligencja ludzkiego mózgu dla złożoności N jest w zasadzie zerowa.

Na przykład ludzkie poznanie wizualne jest wyspecjalizowane w radzeniu sobie ze środowiskami o dużej złożoności, a ludzki mózg z pewnością nie jest w stanie poradzić sobie równie dobrze ze wszystkimi zjawiskami o porównywalnej złożoności. Ludzki mózg jest wyspecjalizowany w poznaniu wizualnym, nawet jeśli w wielu aspektach wykorzystuje swoją „ogólną inteligencję” do rozwiązania problemu. To samo zjawisko występuje w wielu innych obszarach, od ludzkiego poznania społecznego generowanych twierdzeń matematycznych). Każda użyteczna w świecie rzeczywistym ogólna

inteligencja będzie, podobnie jak ludzki mózg, wykazywać mieszaną metod „pełnej ogólnej inteligencji” skupionych na zwiększaniu pełnej ogólnej inteligencji dla małego N oraz „ogólnej inteligencji wykorzystującej wyspecjalizowane metody inteligencji” (GILSIM), które różnią się od metod wąskiej sztucznej inteligencji tym, że wykorzystują one w szczególności kombinację wyspecjalizowanych heurystyk i metod pełnej ogólnej inteligencji dla małego N. Jak się okazuje, trudną częścią praktycznego problemu ogólnej inteligencji nie jest część małej N pełnej ogólnej inteligencji, ale raczej część GILSIM. Osiągnięcie „małej N ogólnej inteligencji” jest problemem matematycznym, rozwiązywalnym za pomocą algorytmów, takich jak programowanie genetyczne, uczenie się przez wzmacnianie lub algorytm OOPS Schmidhubera. Novamente wykorzystuje tutaj kombinację kilku podejść, co zostanie krótko omówione poniżej. Z drugiej strony, współczesna matematyka ma mniej do zaoferowania, jeśli chodzi o zadanie zbudowania systemu zdolnego do obsługi wielu wyspecjalizowanych inteligencji, które łączą heurystykę odpowiednią do zadania z pełną ogólną inteligencją o ograniczonej złożoności. I to jest centralne wyzwanie projektowania AGI, tak jak to widzimy. To jest wyzwanie, które podejmuje projekt Novamente.

Zintegrowane podejście do AGI

Możliwe są co najmniej trzy podstawowe podejścia do AGI:

1. ścisła emulacja ludzkiego mózgu w oprogramowaniu;
2. koncepcja nowej architektury AGI, bardzo odmiennej od mózgu, a także od wąskich programów AI;
3. podejście integracyjne, syntetyzujące wąskie algorytmy i struktury AI w unikalnym ogólnym schemacie, być może w pewnym stopniu kierowane przez zrozumienie ludzkiego mózgu.

Podejście Novamente leży na kontynuum między podejściem 2 i podejściem 3. Około 2/3 projektu Novamente opiera się na istniejących wąskich podejściach AI, a reszta została pomyślana de novo z myślą o AGI. Novamente zdecydowanie nie należy do Kategorii 1: nie jest emulacją ludzkiego mózgu. Podczas gdy ludzki mózg był koncepcyjną inspiracją dla Novamente, szczególnie we wczesnej fazie projektowania, projekt Novamente podejmuje skoordynowany wysiłek, aby robić rzeczy w sposób, który jest wydajny dla oprogramowania działającego w sieciach maszyn von Neumanna, i często jest to głęboko różne od sposobów, które są wydajne w przypadku oprogramowania neuronowego. Podejście integracyjne opiera się na idei, że wiele wąskich podejść do sztucznej inteligencji ucieleśnia dobre pomysły na temat tego, jak pewien konkretny aspekt inteligencji może być implementowany obliczeniowo. Na przykład sztuczna inteligencja oparta na logice zawiera wiele spostrzeżeń dotyczących natury logicznego rozumowania. Formalne sieci neuronowe ucieleśniają wiele spostrzeżeń na temat pamięci, percepcji, klasyfikacji i wzmacniającego uczenia się procedur. Programowanie ewolucyjne jest doskonałą techniką uczenia się procedur i tworzenia złożonych nowych koncepcji. Algorytmy klastrowania są dobrymi sposobami tworzenia spekulatywnych nowych kategorii w słabo poznanej domenie. Itp. Obserwacja, że wąskie podejścia AI często dobrze modelują konkretne aspekty inteligencji, prowadzi do pomysłu syntezy kilku wąskich podejść AI w celu utworzenia architektury AGI. Tego rodzaju syntezę można przeprowadzić na dwa sposoby: Luźna integracja, w której różne wąskie techniki AI znajdują się w oddzielnych procesach oprogramowania lub modułach oprogramowania i wymieniają się ze sobą wynikami swoich analiz; Ścisła integracja, w której wiele wąskich procesów AI oddziałuje w czasie rzeczywistym na tym samym rozwijającym się integracyjnym magazynie danych i dynamicznie wpływa na parametry i schematy sterowania innych. Podejście „luźnej integracji” przejawia się w DINI jako architektura, w której oddzielne klastry analityczne, ucieleśniające oddzielne wąskie techniki AI, oddziałują na siebie za pośrednictwem centralnego magazynu danych. Podejście „ściślej integracji” przejawia się w kategoriach złożonego klastra analitycznego zawierającego własną lokalną bazę danych, obejmującego wiele wąskich

algorytmów AI nierozzerwalnie ze sobą powiązanych. Ścisłą integrację trudniej zaprojektować, wdrożyć, przetestować i dostroić, ale daje ona możliwość uzyskania większej inteligencji poprzez pojawiające się, kooperacyjne efekty. Novamente opiera się na ścisłej integracji i uważamy, że jest to jedyne podejście, które jest wykonalne dla prawdziwego AGI. Novamente zasadniczo składa się z ram ścisłej integracji różnych algorytmów AI w kontekście wysoce elastycznej reprezentacji wspólnej wiedzy oraz określonego zestawu algorytmów AI stworzonych lub zmodyfikowanych w celu ścisłej integracji w integracyjnym kontekście AGI.

Interaktywne uczenie się oparte na doświadczeniu i adaptacyjna samomodifikacja

Omawialiśmy AGI jako kwestię złożonych systemów oprogramowania ucieleśniających złożone matematyczne algorytmy AI. Jest to ważna perspektywa, ale należy pamiętać, że AGI nie jest po prostu kolejną formą inżynierii – jest to również głęboko filozoficzne i koncepcyjne dążenie. Novamente nie zostało zaprojektowane wyłącznie w oparciu o inżynierię i rozważania matematyczne. Raczej zawdzięcza swoje ostateczne pochodzenie abstrakcyjnej, złożonej teorii systemów psychologicznych/filozoficznych teorii umysłu – „modelowi psynet”, który został przedstawiony przez pierwszego autora w pięciu monografiach badawczych opublikowanych w latach 1993–2001. Opierając się na założeniu, że umysł jest zbiorem wzorców w mózgu, model psynet opisuje określony zestaw struktur i dynamiki wysokiego poziomu dla wzorców umysłu i zakłada, że są one niezbędne dla każdego rodzaju umysłu, ludzkiego lub cyfrowego. Nie są to struktury, które można zaprogramować w systemie; są to raczej struktury, które muszą wyłonić się poprzez usytuowaną ewolucję systemu – poprzez doświadczalne interaktywne uczenie się. Konkretnie struktury i dynamika Novamente ściśle wiążą się z bardziej ogólnymi strukturami i dynamikami postulowanymi przez model psynet. Model psynet zawiera również teorię relacji między uczeniem się a umysłem, która różni się od najczęstszych perspektyw wyrażanych w literaturze AI. Mianowicie zakłada, że:

Samo oprogramowanie i matematyka, bez względu na to, jak zaawansowane, nie mogą stworzyć AGI.

Wierzymy jednak, że oprogramowanie i matematyka mogą stworzyć ramy, w których sztuczna inteligencja ogólna może wyłonić się poprzez interakcję z ludźmi w kontekście bogatego strumienia danych ze świata rzeczywistego. To znaczy:

Inteligencja najbardziej naturalnie wyłania się poprzez usytuowane i społeczne doświadczenie. Oczywiście jest, że ludzka inteligencja nie wyłania się wyłącznie poprzez ludzkie oprogramowanie neuronowe. Ludzkie niemowlę nie jest tak inteligentne, a niemowlę wychowywane bez odpowiedniej socjalizacji nigdy nie osiągnie pełnej ludzkiej inteligencji.

Ludzki mózg uczy się myśleć poprzez naukę i poprzez różnorodne interakcje społeczne. Sugerujemy, że sytuacja będzie nieco podobna w przypadku AGI. Podstawowe algorytmy AGI w Novamente nie są w pełni wystarczające dla praktycznej ogólnej inteligencji, ponieważ dostarczają jedynie „surowców” myśli. Czego brakuje w Novamente „od razu po wyjęciu z pudełka”, to kontekstowo specyficzne mechanizmy kontroli dla różnych mechanizmów poznawczych. System ma zdolność do nauki ich, ale co równie ważne, ma zdolność do nauki, jak się ich uczyć, poprzez interakcje społeczne. Novamente „od razu po wyjęciu z pudełka” będzie znacznie inteligentniejszy niż wąskie systemy AI, ale nie tak solidnie inteligentny jak Novamente, który udoskonalił swoją zdolność do nauki kontekstowo specyficznych mechanizmów kontroli poprzez znaczące interakcje z innymi umysłami. Na przykład, gdy będzie przez jakiś czas wchodził w interakcje ze światem, zyska poczucie, jak rozumować na temat rozmów, jak rozumować na temat danych o włamaniach do sieci, jak rozumować na temat danych bioinformatycznych – poprzez naukę schematów kontroli wnioskowania zależnych od kontekstu dla każdego przypadku, zgodnie z procesem uczenia się schematów dostrojonym przez interakcję doświadczalną. Te rozważania prowadzą nas wprost do koncepcji autonomii, interaktywnego uczenia

się opartego na doświadczeniu i samo-modyfikacji zorientowanej na cel – koncepcji, które leżą u podstaw pojęcia Sztucznej Inteligencji Ogólnej. Aby system oprogramowania mógł zademonstrować AGI, naszym zdaniem musi on zademonstrować:

- spójną autonomię jako niezależny, samopozostający, samokontrolujący się system;
- zdolność do modyfikowania i ulepszania samego siebie na podstawie własnych obserwacji i analiz własnej wydajności;
- zdolność do bogatej interakcji z innymi umysłami (takimi jak umysły ludzkie) i uczenia się od nich.

Te ogólne punkty przywołują pewne bardzo konkretne kwestie związane z różnicą między konwencjonalnymi systemami analizy danych i zarządzania wiedzą a systemami AGI stosowanymi do analizy danych, zarządzania nimi i zapytań. Ściśle powiązany, zintegrowany system oprogramowania AI może być wyposażony w określone, zorientowane na cel schematy sterowania i w ten sposób wykorzystywany jako silnik eksploracji danych i/lub przetwarzania zapytań. Takie podejście jest stosowane na przykład w obecnych zastosowaniach silnika Novamente w dziedzinie bioinformatyki. Jednak tego rodzaju wdrożenie oprogramowania Novamente nie pozwala mu rozwinąć się nawet w przybliżeniu do maksymalnego poziomu ogólnej inteligencji. Aby powstała naprawdę znacząca AGI, system oprogramowania musi zostać wdrożony nieco inaczej. Musi zostać wyposażony w ogólne cele, a następnie musi zostać umożliwiony nauczanie się własnych schematów sterowania poprzez wykonywanie dynamiki uczenia się procedur w kontekście interakcji z bogato ustrukturyzowanym środowiskiem i w kontekście rozległych znaczących interakcji z innymi umysłami. Ta ścieżka jest trudniejsza niż droga „sztywnych schematów sterowania”, ale jest konieczna do osiągnięcia prawdziwej AGI. System Novamente, po pełnym zaprojektowaniu i dostrojeniu, zyska inteligencję poprzez przetwarzanie danych istotnych z praktycznego punktu widzenia, odpowiadając na pytania ludzi dotyczące tych danych i dostarczając im raportów podsumowujących zaobserwowane wzorce. Oprócz EIL poprzez interaktywną analizę/zarządzanie danymi, stworzyliśmy specjalny „interfejs użytkownika EIL” o nazwie Shape-World, który obejmuje interakcję z Novamente w kontekście prostego panelu rysunkowego, na którym nauczyciel i Novamente mogą rysować kształty i rozmawiać o tym, co robią i co widzą. Zaprojektowaliśmy również środowisko o nazwie EDEN (EDucational Environment for Novamente), wirtualny świat rzeczywistości, w którym Novamente będzie kontrolować symulowanych agentów, którzy wchodzi w interakcje z agentami kontrolowanymi przez ludzi w symulowanym środowisku. Ten proces „doświadczalnego interaktywnego uczenia się” był jednym z głównych zagadnień w projektowaniu i rozwoju Novamente. Będzie on stale modyfikował nie tylko swoją bazę wiedzy, ale także schematy sterowania w oparciu o to, czego nauczył się ze swojego otoczenia i ludzi, z którymi wchodzi w interakcje. Ostateczne granice tego procesu samodoskonalenia są trudne do przewidzenia – jeśli w ogóle takie istnieją. Warto pamiętać, że sam kod źródłowy jest obiektem formalnym, który można łatwo przedstawić w schemacie reprezentacji wiedzy systemu AGI, takiego jak Novamente. Wnioski dotyczące kodu źródłowego i jego potencjalnych wariantów i ulepszeń wydają się leżeć w domenie obliczeniowo osiągalnego rozumowania probabilistycznego. Wydaje się, że nie ma podstawowego powodu, dla którego system AGI nie mógłby badać swojego własnego kodu źródłowego i wymyślić, jak stać się mądrzejszym. I istnieje atrakcyjna logika wykładnicza tego procesu: im staje się mądrzejszy, tym lepiej będzie się samodoskonalili. Oczywiście realizacja tego rodzaju ostatecznej samoadaptacji leży w pewnej odległej przyszłości. Mogą istnieć znaczne przeszkody, nieprzewidywalne w obecnym momencie. Ale przynajmniej na poziomie koncepcyjnym te idee są naturalnym następstwem procesów samodoskonalenia ukierunkowanego na cel, które wdrożymy w Novamente w niedalekiej przyszłości, jako część procesu dostrajania i nauczania AGI. System Novamente został zaprojektowany z wyraźnym naciskiem na spełnianie krótkoterminowych potrzeb analizy danych, zarządzania i zapytań, ale także z myślą o pełnej okazałości długoterminowej wizji AGI.

Model umysłu Psynet

W tej sekcji zagłębimy się nieco bardziej w model umysłu Psynet, koncepcyjny i filozoficzny fundament systemu Novamente. Na początek musimy wyjaśnić, jak używamy terminu „umysł”. Naszym zdaniem „umysł”, podobnie jak „inteligencja”, jest koncepcją języka ludzkiego, z bogactwem nakładających się znaczeń. Model Psynet nie ma na celu pełnego uchwycenia pojęcia „umysłu” w języku ludzkim. Zamiast tego ma na celu uchwycenie użytecznego podzbioru tego pojęcia, z myślą o ukierunkowaniu inżynierii AGI i analizie ludzkiego poznania. Model Psynet opiera się na tym, co Ray Kurzweil nazywa filozofią „wzorcową”. Opiera się na założeniu, że umysł nie jest ani systemem fizycznym, ani całkowicie oddzielnym od fizycznego – raczej umysł jest czymś związanym z zestawem wzorców w systemie fizycznym. W przypadku inteligentnego systemu obliczeniowego umysł systemu nie znajduje się w kodzie źródłowym, ale raczej we wzorcach obserwowalnych w dynamicznym śladzie, który system tworzy w czasie w pamięci RAM i w rejestrach procesorów komputerowych. Koncepcja wzorca użyta tutaj jest rygorystyczna, co można uzasadnić matematycznie w kategoriach algorytmicznej teorii informacji. W istocie wzór w bycie jest uważany za abstrakcyjny program komputerowy, który jest mniejszy od bytu i może szybko obliczyć byt. Na przykład wzór w obrazie zbioru Mandelbrota może być programem, który może obliczyć obraz ze wzoru. Powiedzenie „umysł jest wzorem” jest zatem równoznaczne z umiejscowieniem umysłu w matematycznej domenie abstrakcyjnych, niefizycznych programów komputerowych. Jak ostrzegaliśmy powyżej, nie twierdzimy tego jako kompletnego wyjaśnienia wszystkich aspektów koncepcji „umysłu” – ale jedynie jako pragmatyczną definicję, która pozwala nam wyciągać wnioski na temat umysłów systemów AGI w użyteczny sposób. Podejście „umysł to wzorec” do teorii AI nie jest samo w sobie oryginalne; podobne idee można znaleźć w myśleniu współczesnych filozofów, takich jak Gregory Bateson, Douglas Hofstadter i Daniel Dennett. Model psynet jednak idzie o krok dalej i pyta, jak zbiór wzorców składających się na umysł jest ustrukturyzowany i jak ewoluuje w czasie. Dąży do zrozumienia umysłu w kategoriach dynamiki wzorców i wyłaniających się struktur wynikających z dynamiki wzorców. Zgodnie z modelem psynet wzorce tworzące umysł funkcjonują jako półautonomiczni „aktorzy”, którzy oddziałują na siebie na różne sposoby. Funkcje umysłowe, takie jak percepcja, działanie, rozumowanie i uczenie się procedur, są opisywane w kategoriach interakcji między aktorami umysłu (które są wzorcami w pewnym podstawowym podłożu fizycznym, np. mózgu lub programie komputerowym). I stawiane są hipotezy dotyczące struktury i dynamiki sieci wzorców umysłu na dużą skalę. Zgodnie z charakterystyką inteligencji „złożonych celów w złożonych środowiskach”, zakłada się, że inteligentny system w danym przedziale czasu ma pewien system celów (który może być wyrażony jawnie i/lub niejawnie w umyśle systemu). Ten system celów może się zmieniać w czasie, albo poprzez „dryfowanie celów”, albo poprzez skoordynowaną aktywność systemu (niektóre cele mogą wyraźnie zachęcać do własnej modyfikacji). Ważne jest, aby inteligentny system miał zarówno cele ogólne, jak i szczegółowe w swoim systemie celów. Ponadto jeden konkretny cel ogólny jest uznawany za krytyczny dla systemu celów każdego inteligentnego systemu: tworzenie i rozpoznawanie nowych wzorców. Mając ten cel w swoim systemie celów, inteligencja będzie dążyć do postrzegania i tworzenia nowych struktur w sobie, podczas gdy zajmuje się osiąganiem swoich innych celów; a ta samopercepcja/tworzenie wzmocni jej inteligencję w dłuższej perspektywie. Dynamika wzorców systemu poznawczego jest rozumiana jako rządzona przez dwie główne „siły”: spontaniczną samoorganizację i zachowanie zorientowane na cel. Bardziej szczegółowo, postuluje się kilka podstawowych zasad dynamicznych, w tym:

Asocjacja, w której wzorce, gdy poświęci się im uwagę, rozprzestrzeniają część tej uwagi na inne wzorce, z którymi wcześniej były w jakiś sposób kojarzone.

Różnicowa alokacja uwagi, w której wzorce, które były cenne dla osiągnięcia celu, otrzymują więcej uwagi i są zachęcane do udziału w tworzeniu nowych wzorców.

Tworzenie wzorców, w którym wzorce, które były cenne dla osiągnięcia celu, są mutowane, aby uzyskać nowe wzorce, i są łączone ze sobą, aby uzyskać nowe wzorce.

Reifikacja relacji, w której nawykowe wzorce w systemie, które są uważane za cenne dla osiągnięcia celu, są wyraźnie wzmacniane i stają się bardziej nawykowe.

Na przykład, proponuje się, że aby system wykazywał znaczącą inteligencję, sieć wzorców obserwowalnych w systemie musi dać początek kilku wielkoskalowym strukturom emergentnym:

Sieć hierarchiczna, w której wzorce są zwyczajowo w relacjach kontroli nad innymi wzorcami, które reprezentują bardziej wyspecjalizowane aspekty samych siebie

Sieć heterarchiczna, w której system zachowuje pamięć o tym, które wzorce były wcześniej w jakiś sposób ze sobą powiązane

Sieć dualna, w której struktury hierarchiczne i heterarchiczne są połączone, a dynamika obu struktur działa harmonijnie

Struktura „Ja”, w której część sieci wzorców tworzy przybliżony (fraktalny) obraz całej sieci wzorców.

Model psynet jest bardzo ogólną konstrukcją. Nie mówi ci, jak zbudować system AGI w sensie inżynierskim; mówi ci tylko, ogólnie, „jak powinien wyglądać system AGI”. Novamente jest trzecim zorientowanym na AGI systemem oprogramowania stworzonym z myślą o modelu psynet i bardzo różni się od dwóch poprzednich projektów. Różnice między tymi systemami można podsumować następująco:

1994: Antimagicians, który był eksperymentalnym programem inspirowanym siecią psynet w czystym duchu systemów samoorganizujących się z bardzo niewielką liczbą wbudowanych struktur i intencją, aby struktury i dynamika umysłu wyłoniły się poprzez doświadczenie. Przewidywane wyłonienie się nie zostało zaobserwowane i zdecydowano się na bardziej zorientowane inżynierskie podejście, w którym więcej początkowych struktur i dynamik jest wszczepianych jako „ziarno” inteligentnej samoorganizacji.

1996-2001: Silnik sztucznej inteligencji Webmind, „Webmind”, opracowany w Webmind Inc., był wielkoskalowym systemem oprogramowania Java, który wywodził swój projekt oprogramowania z modelu sieci psynet w bardzo bezpośredni sposób. Części Webmind zostały pomyślnie zastosowane w domenach prognozowania finansowego i wyszukiwania informacji; wykonano również wiele przydatnych prototypów. Odkryto jednak, że bezpośrednie mapowanie konstrukcji modelu sieci psynet na obiektowo zorientowane struktury oprogramowania prowadzi do poważnych problemów z wydajnością obliczeniową.

Od 2001 r.: Novamente, które reprezentuje zupełnie inne podejście, ucieleśniając wysoce elastyczne, wydajne obliczeniowo ramy AGI, które można wykorzystać do wdrożenia wielu różnych systemów AI. Ramy te obejmują trzy główne aspekty: architekturę DINI, filozofię ściśle powiązanej zintegrowanej AI oraz architekturę Novamente „Mind OS”, która zostanie opisana poniżej. Novamente ucieleśnia również konkretny wybór obiektów oprogramowania w ramach tych ram, których wybór jest w dużym stopniu ukształtowany przez idee w modelu umysłu psynet.

Relacja między modelem umysłu psynet a Novamente jest w pewnym sensie podobna do relacji między teorią ewolucji a współczesnymi algorytmami programowania ewolucyjnego. Teoria ewolucji dostarcza podstaw koncepcyjnych dla programowania ewolucyjnego, a pierwszy algorytm programowania ewolucyjnego, tradycyjny algorytm GA ciągu bitów, powstał jako dość bezpośrednie próby emulacji ewolucji biologicznej poprzez dobór naturalny [41]. Jednak współczesne podejścia do

programowania ewolucyjnego, takie jak algorytm optymalizacji bayesowskiej i programowanie genetyczne, osiągają wyższą funkcjonalność pragmatyczną, odchodząc dość daleko od modelu biologicznego, i są tylko bardziej pośrednio mapowalne z powrotem do ich inspiracji koncepcyjnej. Podobnie Novamente reprezentuje podstawowe koncepcje zaangażowane w model psynet, ale w formie pośredniej, która w równym stopniu zawdzięcza się kwestiom funkcjonalności pragmatycznej we współczesnym kontekście obliczeniowym.

Projekt AGI Novamente

Silnik AI Novamente („Novamente”) to wielkoskalowy, obiektowy, wielowątkowy system oprogramowania, przeznaczony do działania w ramach DINI. Jest to system oprogramowania C++ z kilkoma zewnętrznymi komponentami napisanymi w Javie. Obecnie rozwój odbywa się głównie w systemie operacyjnym Linux, ale przeniesienie do innych odmian Unix lub Windows nie byłoby problematyczne⁶. W terminologii DINI system Novamente to zbiór klastrów analitycznych, z których większość jest ściśle zintegrowana, a niektóre są prostsze i bardziej wyspecjalizowane. Ucieleśnia ściśle powiązane podejście integracyjne do AGI, w którym szereg wąskich podejść AI jest połączonych z kilkoma innowacyjnymi pomysłami strukturalnymi i dynamicznymi w kontekście wspólnej „uniwersalnej reprezentacji wiedzy”. Struktury i procesy wybrane dla Novamente mają na celu umożliwienie systemowi realizacji abstrakcyjnej dynamiki i wyłaniających się struktur opisanych w modelu umysłu psynet. W tej sekcji przedstawimy projekt Novamente w ogólnych zarysach, ilustrując każdy aspekt omawiany w kontekście analizy danych, zapytań lub zarządzania. Później zagłębimy się w kilka ważniejszych procesów AI w systemie nieco bardziej szczegółowo. Witryna AGIRI zawiera okresowo aktualizowaną stronę, która daje jeszcze większą głębię przedstawieniu, systematycznie wyliczając niektóre z kluczowych struktur i dynamik systemu. Poniżej krótko opisujemy główne aspekty projektu Novamente:

Węzły. Węzły mogą symbolizować byty w świecie zewnętrznym, mogą ucieleśniać proste procesy wykonywalne, mogą symbolizować abstrakcyjne koncepcje lub mogą służyć jako komponenty w sieciach relacji oznaczających złożone koncepcje lub procedury.

Łącza. Łącza mogą być n-arne i mogą wskazywać na węzły lub łącza; ucieleśniają różne typy relacji między koncepcjami, percepcjami lub działaniami. Sieć łączy jest siecią relacji.

MindAgents. MindAgent to obiekt programowy ucieleśniający proces dynamiczny, taki jak aktualizacja ważności, tworzenie koncepcji lub wnioskowanie logiczne pierwszego rzędu. Działa bezpośrednio na poszczególne atomy, ale ma na celu wywoływanie i kierowanie wzorcami dynamicznymi w całym systemie.

System operacyjny umysłu. System operacyjny umysłu, działający w ramach DINI, umożliwia różnym MindAgentom wydajne działanie na dużych populacjach węzłów i łączy rozproszonych na wielu maszynach.

Mapy. Mapa reprezentuje wiedzę deklaratywną lub proceduralną jako wzór rozproszony na wielu węzłach i łączach.

Jednostki. Jednostka to zbiór węzłów, łączy i MindAgentów, żyjących na klastrze maszyn, wspólnie poświęconych wykonywaniu określonej funkcji, takiej jak: przetwarzanie obrazu, generowanie języka, wysoce skoncentrowana koncentracja, . . .

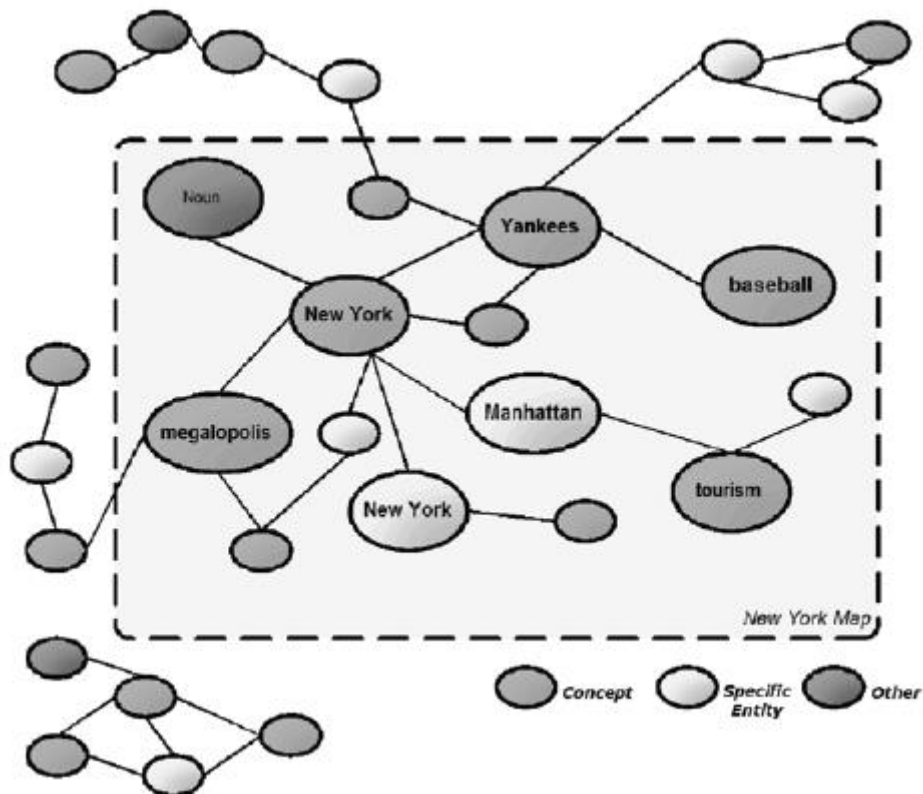
Integracyjna reprezentacja wiedzy

Reprezentacja wiedzy jest jednym z ogromnych, klasycznych problemów AI. Oczywiście jest ona ściśle związana z problemem algorytmów poznawczych – różne algorytmy poznawcze mają różne wymagania dotyczące reprezentacji wiedzy, a różne reprezentacje wiedzy sugerują różne algorytmy poznawcze. Reprezentacja wiedzy Novamente powstała w wyniku poszukiwań najprostszej, najwygodniejszej w obsłudze reprezentacji wiedzy, która byłaby łatwo zgodna ze wszystkimi różnymi procesami AI w systemie Novamente. Podobnie jak sam system Novamente, reprezentacja wiedzy Novamente jest syntezą pomysłów z istniejących wąskich paradygmatów AI – z dodaną znaczną liczbą oryginalnych elementów, aby wypełnić role, których nie obejmują istniejące pomysły (w tym niektóre role, takie jak przydzielanie uwagi w całym systemie, których z natury nie można wypełnić za pomocą wąskich podejść AI). Wiedza jest reprezentowana w Novamente na dwóch poziomach:

Atomy, obiekty oprogramowania, które występują w dwóch gatunkach: Węzły lub Łącza.

Mapy, zbiory atomów, które mają tendencję do wspólnej aktywacji lub mają tendencję do aktywacji zgodnie z pewnym wzorcem (np. oscylacja lub dziwny atraktor).

Ogólnie rzecz biorąc, te same typy wiedzy są reprezentowane na poziomie atomu i na poziomie mapy. Reprezentacja na poziomie atomu jest dokładniejsza i bardziej niezawodna, ale reprezentacja na poziomie mapy jest bardziej podatna na pewne typy uczenia się i pewne typy zachowań w czasie rzeczywistym. Rysunek przedstawia graficzny przykład mapy – mapy dla koncepcji „Nowego Jorku”, jaka mogłaby wystąpić w systemie Novamente.



Ta mapa jest rozmytym zbiorem węzłów zawierającym ConceptNode odpowiadający koncepcji Nowego Jorku, a także wiele innych powiązanych węzłów. Na poziomie atomu podstawową strukturą matematyczną reprezentacji wiedzy Novamente jest hipergraf (graf, którego krawędzie mogą

obejmować $k > 2$ węzłów]). Ten hipergraf nazywamy Atomspace, co oznacza, że jest to hipergraf o specjalnych właściwościach, które:

- węzły i łącza są ważone za pomocą złożonych struktur wagowych (obiekty Truth-Value i AttentionValue)
- węzły i łącza są oznaczone różnymi etykietami „typu”;
- niektóre węzły mogą zawierać obiekty danych (znaki, liczby, wartości kolorów itp.);
- niektóre węzły mogą zawierać wewnątrz małe hipergrafy

Koncepcyjnie rzecz biorąc, dwie struktury wagowe powiązane z atomami Novamente reprezentują dwie podstawowe szkoły badań nad sztuczną inteligencją – logikę (TruthValue) i sieci neuronowe (AttentionValue).

Wartość TruthValue wskazuje mniej więcej, w jakim stopniu Atom poprawnie opisuje świat. Obiekt zawiera:

- wartość prawdopodobieństwa;
- wartość „ciężaru dowodów” wskazującą ilość dowodów użytych do wyprowadzenia prawdopodobieństwa;
- opcjonalnie dodatkowe informacje, takie jak funkcja rozkładu prawdopodobieństwa;
- opcjonalnie specjalne informacje o prawdopodobieństwie Atomu w danym strumieniu percepcji/działania.

Wartość AttentionValue to zbiór informacji mówiący o tym, ile uwagi różnego rodzaju powinien otrzymać i otrzymuje Atom. Obejmuje to:

- Długoterminową wagę (LTI), oszacowanie wartości przechowywania Atomu w pamięci zamiast stronicowania go na dysku;
- Ostatnią użyteczność, miarę tego, ile wartości uzyskano z przetwarzania Atomu ostatnio;
- Ważność, miarę tego, ile czasu procesora powinien otrzymać Atom, która jest oparta na aktywacji, LTI i ostatniej użyteczności.

Ten specjalny hipergraf Atomspace jest używany na wiele różnych sposobów. Na przykład:

1. wszystkie węzły i łącza mają być interpretowane logicznie, przy użyciu logiki probabilistycznej;
2. niektóre węzły i łącza mogą być postrzegane jako odzwierciedlające procesy ausation i są używane do „przypisywania kredytu”, co jest kluczową częścią adaptacyjnego przydzielania uwagi;
3. niektóre węzły i łącza mogą być interpretowane jako programy wykonywalne.

Umożliwienie wszystkich tych interpretacji jednocześnie wymaga pewnej ostrożności.

A co z poziomem reprezentacji wiedzy na mapie? Ponieważ mapy są kodowane w systemie niejawnie, a nie jawnie, można o nich powiedzieć mniej w sposób zwarty. Ale jednym kluczowym punktem jest to, że sieć map w systemie można również postrzegać jako hipergraf – taki, którego węzły są rozmytymi

zbiorami atomów. Połączenia na poziomie mapy są definiowane w sposób naturalny: łącze na poziomie mapy pewnego typu T, między mapą A i mapą B, jest definiowane jako wiązka połączeń typu T przechodząca między atomami w A i atomami w B, które są jednocześnie aktywne. Połączenia na poziomie mapy są definiowane niejawnie przez łącza na poziomie atomu. Reprezentują one bardziej ogólny, rozproszony rodzaj wiedzy, która oddziałuje z wiedzą na poziomie atomu za pośrednictwem złożonego zestawu efektów sprzężenia zwrotnego. W języku modelu psynet mapy są wzorcami, „rzeczą umysłu” odpowiadającą „rzeczą mózgu”, jaką jest kod oprogramowania Novamente i jego dynamiczny obraz w pamięci RAM. Atomy (węzły i łącza) istnieją na interesującym poziomie pośrednim, który nazywamy „konkretnie zaimplementowanym umysłem”. Oznacza to, że atomy nie są materiałem umysłu, ale są częściami materiału mózgu, które są „indeksatorami umysłu”, w tym sensie, że wiele atomów jest powiązanych ze specyficznymi wzorcami w systemie (konkretnymi przypadkami materiału umysłu), a reszta jest bezpośrednio uwzględniana jako składniki wielu wzorców w systemie. Związek między strukturami Novamente a strukturami ludzkiego mózgu jest interesujący. Krótko mówiąc, nie ma korelatu Novamente neuronów i synaps – Novamente nie emuluje mózgu na tak niskim poziomie. Istnieje jednak przybliżone intuicyjne mapowanie między węzłami Novamente a tym, co neurobiolog Gerald Edelman nazywa „grupami neuronowymi” – ściśle połączonymi klastrami 10 000–50 000 neuronów. Łącza Novamente są jak wiązki synaps łączące grupy neuronowe. A mapy Novamente są czymś w rodzaju „map neuronowych” Edelmana. Patrząc na tło współczesnej teorii wąskiej sztucznej inteligencji, reprezentacja wiedzy Novamente nie jest niczym niezwykłym. Łączy ona aspekty sieci semantycznej, sieci neuronowej atraktora i reprezentacji wiedzy w stylu programowania genetycznego. Ale nie łączy tych aspektów w sposób „multimodularny”, który utrzymuje je oddzielnymi, ale oddziałującymi: łączy je razem w nowy schemat reprezentacji, który jest znacznie więcej niż suma jego części, ze względu na specyficzny sposób, w jaki umożliwia współdziałanie różnych jakościowo bardzo różnych, uzupełniających się procesów poznawczych.

Mind OS

Główną koncepcją projektową Novamente jest implementacja wielu algorytmów poznawczych w ściśle zintegrowany sposób, przy użyciu reprezentacji wiedzy hipergrafu opisanej powyżej, w praktycznym kontekście architektury oprogramowania DINI. Sednem projektu Novamente z perspektywy AI jest wybór algorytmów poznawczych i sposób ich ścisłej integracji. Zanim jednak tam dotrzemy, brakuje jednego ogniwa do wypełnienia – obliczeniowej mechaniki faktycznego zarządzania zbiorem ściśle zintegrowanych procesów AI. Zajmuje się tym komponent oprogramowania, który nazywamy Mind OS, „rdzeniem Novamente” lub po prostu „rdzeniem”. Jak sugeruje nazwa „OS”, Mind OS wykonuje wiele funkcji systemu operacyjnego. W rzeczywistości można go uznać za ogólny serwerowy framework C++ dla systemów wieloagentowych, zoptymalizowany pod kątem złożonych i intensywnych zadań obejmujących masową współpracę agentów. Chociaż jest dostosowany do Novamente AI, podobnie jak DINI, jest szeroko rozszerzalny i może być również używany do wielu innych celów. Mind OS jest sam w sobie rozproszonym frameworkiem przetwarzania, zaprojektowanym do życia w ramach większego rozproszonego frameworka przetwarzania architektury DINI. Jest zaprojektowany do działania w klastrze ściśle powiązanych maszyn, w taki sposób, że węzeł znajdujący się na jednej maszynie w klastrze może mieć łącza łączące go z węzłami znajdującymi się na innych maszynach w klastrze. W DINI Mind OS jest przeznaczony do życia w złożonym klastrze analitycznym. Złożona konfiguracja Novamente będzie obejmować wiele funkcjonalnie wyspecjalizowanych klastrów analitycznych, z których każdy uruchamia Mind OS. Na każdej maszynie w klastrze kontrolowanym przez Mind OS znajduje się tabela atomów (obiekt AtomTable, który zawiera zbiór wyspecjalizowanych indeksów do szybkiego wyszukiwania), a następnie cykliczna kolejka obiektów zwanych MindAgents. MindAgents są cyklicznie przenoszone, a gdy nadchodzi kolej na działanie, działa przez krótki okres, a następnie przekazuje CPU następnemu MindAgent w kolejce. Większość MindAgentów ucieleśnia

procesy poznawcze, ale niektóre ucieleśniają procesy „na poziomie systemu”, takie jak okresowe buforowanie AtomTable na dysku, sondowanie zewnętrznych danych wejściowych (takich jak dane wejściowe z interfejsu użytkownika) lub zbieranie statystyk systemowych. Na maszynie SMP system operacyjny Mind może przydzielać różne MindAgenty różnym procesorom jednocześnie, przestrzegając ustalonej tabeli relacji wykluczeń. Rozróżnienie między MindAgentami a „aktorami umysłu” modelu psynet może być tutaj mylące. To rozróżnienie odzwierciedla subtelność projektu systemu, zgodnie z którym abstrakcyjna struktura matematyczna systemu różni się od struktury implementacji. Agenci oprogramowania (MindAgenty) nie są „aktorami umysłu” modelu psynet; są raczej dynamicznymi obiektami zaprojektowanymi w celu wywołania pojawienia się aktorów umysłu modelu psynet (mapy emergentne). To rozdzielenie między agentami implementacji a agentami emergentnymi jest kompromisem niezbędnym do osiągnięcia akceptowalnej wydajności obliczeniowej. Obecnie komunikacja z Mind OS może odbywać się albo poprzez dostosowaną powłokę Unix o nazwie nmshell, która jest odpowiednia do interaktywnej komunikacji, przesyłania poleceń sterujących i debugowania; poprzez XML, używając Novamente DTD; albo poprzez warstwę middleware Java/J2EE. Trzecie medium komunikacyjne, poprzez specyficzny dla Novamente język programowania funkcjonalno-logicznego o nazwie Sasha, zostało zaprojektowane, ale nie zaimplementowane. Istnieje również język kodowania wiedzy Novamente o nazwie NQL (Novamente Query Language), modyfikacja języka KNOW używanego w systemie Webmind; ale oddziałuje on z rdzeniem pośrednio poprzez nmshell lub XML. Podsumowując, rdzeń Novamente to system „OS” C++ z wieloma agentami, który obsługuje:

- Wielowątkowość
- Elastyczne podłączanie i planowanie heterogenicznych agentów
- Rozproszona wiedza z lokalnymi serwerami proxy i pamięcią podręczną
- Kontrola transakcji
- Komunikacja z zewnętrznymi agentami oprogramowania za pośrednictwem XML i skryptów
- Przetwarzanie zadań i zapytań za pośrednictwem systemu zgłoszeń
- Adaptacyjna kontrola parametrów
- Dynamiczne, adaptacyjne równoważenie obciążenia

Podczas projektowania i wdrażania rdzenia, dołożono wszelkich starań, aby zapewnić wydajność obliczeniową i przestrzeni. Wiele nauczyliśmy się w tym zakresie z wad Webmind AI Engine, rozproszonej architektury AI zaprojektowanej pod koniec lat 90. przez zespół pokrywający się z obecnym zespołem Novamente. Architektura Webmind opierała się na dość bezpośrednim mapowaniu modelu psynet na obiektowo zorientowane struktury oprogramowania. Był to ogromny system wieloagentowy, wykorzystujący reprezentację wiedzy hipergrafu, w którym każdy węzeł był implementowany jako autonomiczny agent oprogramowania. Te węzły-agenci wykonywali wiele takich samych procesów AI, jak w Novamente. Jednak ogromna architektura systemu wieloagentowego okazała się trudna do dostrojenia i optymalizacji. Umiarkowanie duża rozproszona instancja Webmind zawierała miliony autonomicznych agentów oprogramowania (głównie węzłów); podczas gdy umiarkowanie duża rozproszona instancja Novamente będzie miała zaledwie setki (MindAgents). Novamente to nadal system wieloagentowy, ale o innej architekturze – a ta zmiana architektoniczna robi ogromną różnicę w rodzajach optymalizacji wydajności, jakie można wdrożyć, co skutkuje poprawą o trzy rzędy wielkości w szybkości i o dwa rzędy wielkości w wykorzystaniu pamięci. Jesteśmy niezmiernie zadowoleni, że Novamente Mind OS, pomimo swojej złożoności, jest

wystarczająco wydajny i solidny, aby można go było wykorzystać w rdzeniu linii produktów Biomind LLC. Na tym etapie, po wielu latach eksperymentowania z tego rodzaju systemem oprogramowania, jesteśmy w stosunkowo szczęśliwym punkcie, w którym wszystkie praktyczne problemy na poziomie „Mind OS” są rozwiązane i możemy skupić się na naprawę trudnej części, dostrajając ścisłą dynamiczną integrację poznawczych MindAgents.

Typy atomów

Teraz przechodzimy do przeglądu konkretnych typów węzłów i łączy wykorzystywanych w Novamente. Podobnie jak w przypadku wyboru MindAgents, ten zbiór typów węzłów i łączy został wybrany częściowo ze względów pragmatycznych, a częściowo ze względów teoretycznych. Wybraliśmy struktury danych i dynamikę, opierając się głównie na następujących kryteriach:

- wykazana moc w wąskich zastosowaniach AI;
- wzajemna spójność jako integracyjne ramy AGI;
- skłonność do ucieleśniania dynamiki i struktur postulowanych przez Model Umysłu Psynet.

Novamente zawiera kilkadziesiąt typów węzłów i łączy, a niemal kompletna lista znajduje się na stronie internetowej AGIRI. Istnieje jednak warstwa abstrakcji koncepcyjnej między koncepcją „węzłów i łączy” a konkretnymi typami węzłów i łączy. Nazywamy tę warstwę „odmianami węzłów i łączy” – każda odmiana oznacza funkcję koncepcyjną, a nie kategorię matematyczną lub implementacyjną; a każda odmiana może zawierać wiele różnych konkretnych typów. Tabele opisują odmiany węzłów i ogniw obecnie stosowane w Novamente.

Różnorodność węzłów: Opis

Węzły percepcyjne: Odpowiadają one konkretnym postrzeganym elementom, takim jak WordInstanceNode, CharacterInstanceNode, NumberInstanceNode, PixelInstanceNode

Węzły procedur: Zawierają małe programy zwane „schematem” i są nazywane SchemaNodes. Węzły akcji, które wykonują logiczne oceny, nazywane są PredicateNodes. ProcedureNodes służą do reprezentowania złożonych wzorców lub procedur.

ConceptNodes: Reprezentują kategorie węzłów percepcyjnych, akcji lub konceptualnych lub części map reprezentujących takie kategorie.

Węzły psychiki: Są to GoalNodes i FeelingNodes (specjalne rodzaje PredicateNodes), które odgrywają szczególną rolę w ogólnej kontroli systemu pod względem monitorowania stanu systemu i orientowania ogólnego zachowania systemu.

Różnorodność łączy: Opis

Łącza logiczne: Reprezentują symetryczne lub asymetryczne relacje logiczne między węzłami (InheritanceLink, SimilarityLink) lub między łączyami i PredicateNodes (np. ImplicationLink, EquivalenceLink)

Łącza członkowskie: Oznaczają przynależność do zbioru rozmytego

Łącza asocjacyjne: Oznaczają pokrewieństwo generyczne, w tym HebbianLink nauczony poprzez naukę Hebbiana i proste associativeLink reprezentujące relacje pochodzące z języka naturalnego lub z baz danych.

Łącza akcji-konceptu: Nazywane łączykami wykonania i łączykami oceny, tworzą koncepcyjny zapis działań podejmowanych przez węzły schematu lub węzły predykatu

Łącza listy i łączyki listy połączeń: Reprezentują odpowiednio listy utworzone wewnętrznie lub obserwowane zewnętrznie.

Dokładne omówienie nie jest tutaj praktyczne ze względu na ograniczenia miejsca, dlatego podamy tylko kilka krótkich komentarzy na temat semantyki tych typów atomów Novamente. Koniem roboczym systemu jest ConceptNode. Niektóre z nich będą reprezentować pojedyncze koncepcje, inne będą stanowić części większych map konceptualnych. Powiązania logiczne i asocjacyjne są ze sobą powiązane ConceptNodes. Na przykład możemy napisać:

InheritanceLink New York megalopolis, co oznacza, że istnieją ConceptNodes odpowiadające koncepcjom „Nowy Jork” i „naród”, a istnieje InheritanceLink wskazujący od jednego do drugiego (oznaczając, że Nowy Jork jest rzeczywiście megalopolis). Albo możemy napisać: AssociativeLink New York immigration, co po prostu wskazuje na ogólne powiązanie między dwoma oznaczonymi ConceptNodes. Relacja asocjacyjna jest przydatna do rozpraszania uwagi między powiązanimi koncepcjami, a także przydatna jako drogowskaz wskazujący logicznemu wnioskowaniu MindAgents, gdzie szukać potencjalnie interesujących powiązań. Bardziej konkretny związek między Nowym Jorkiem a imigracją, taki jak „wielu imigrantów mieszka w Nowym Jorku”, można przedstawić jako: ImplicationLink lives_in_New_York is_immigrant, gdzie lives in New York i is immigrant to PredicateNodes, a poprzedni predykat podlega relacji, która byłaby zapisana: EquivalenceLink (lives_in_New_York(X)) (lives_in(New_York, X)) w zwykłej logice predykatów i jest zapisana bardziej jako: EquivalenceLink lives_in_New_York (lives_in (New_York)) w bezzmiennej reprezentacji wewnętrznej Novamente. Zarządzanie zmiennymi jest jednym z najbardziej złożonych aspektów systemów AI opartych na logice, a także konwencjonalnych języków programowania; Novamente pomija cały temat, używając bezzmiennej reprezentacji predykatów i schematów, opartej na logice kombinatorycznej. SchemaNodes i PredicateNodes występują w dwóch formach: prostej i złożonej. Każda prosta zawiera pojedynczy elementarny schemat lub funkcję predykatu; każdy złożony zawiera wewnętrzny skierowany graf acykliczny powiązanych ze sobą SchemaNodes i PredicateNodes. Zestaw elementarnych funkcji schematu/predykatu jest w rzeczywistości „wewnętrznym językiem programowania Novamente”, który ma pewne podobieństwo do języków programowania funkcyjnego, takich jak czysty LISP lub Haskell. „Akcje” wykonywane przez SchemaInstanceNodes to nie tylko akcje zewnętrzne, w niektórych przypadkach są to również wewnętrzne akcje poznawcze. Złożone SchemaNodes reprezentują złożone skoordynowane akcje, które są „zamknięte” w pojedynczym węźle; złożone PredicateNodes reprezentują złożone wzorce obserwowane w systemie lub na zewnątrz i uznane za przydatne. ExecutionLinks i EvaluationLinks rejestrują informacje o tym, jakie były dane wejściowe i wyjściowe SchemaInstanceNodes i PredicateInstanceNodes w momencie ich wykonania. Ostatecznie wszystkie procesy AI wykonywane wewnątrz Novamente można by sformułować jako złożone schematy, chociaż w obecnej implementacji rdzenia tak nie jest; podstawowa dynamika AI systemu jest implementowana jako obiekty C++ zwane MindAgents, które są bardziej wydajne niż złożone schematy. Następnie FeelingNodes to węzły „wewnętrznych czujników”, które wyczuwają pewien aspekt ogólnego stanu systemu, taki jak wolna pamięć lub ilość ostatnio przyswojonej wiedzy przez system. Złożone „uczucia” są tworzone przez łączenie FeelingNodes w PredicateNodes i nadają systemowi „poczucie siebie” w praktyczny sposób, który pozwala na wykonanie autonomicznej homeostazy i na to, aby system celowo dostosowywał swoją orientację zadaniową w kierunku zwiększonego poczucia pozytywnego „uczucia”. Wreszcie GoalNodes to wewnętrzne czujniki, takie jak FeelingNodes, ale stan, który wyczuwają, może być czasami mniej globalny; reprezentują wąskie cele systemowe, jak również szerokie holistyczne. System jest

wyposażony w podstawowe cele, tak jak w przypadku podstawowych uczuć, ale złożone i osobliwe cele mogą być budowane w czasie. GoalNodes służą do dostosowywania autonomicznych procesów systemu w celu wsparcia skupienia się na procesach myślowych zorientowanych na cel, a także do tego, aby system mógł świadomie wyszukiwać i analizować istotne informacje w celu osiągnięcia tych celów.

Mapy Novamente

Wiele atomów jest znaczących i wartościowych samych w sobie, ale niektóre zyskują znaczenie tylko poprzez skoordynowaną aktywność obejmującą inne atomy, tj. ich zaangażowanie w „mapy”. Mapy występują w wielu kształtach i rozmiarach; ogólna charakterystyka map Novamente byłaby trudna do uzyskania. Jednak Tabela wymienia kilka mniej więcej zdefiniowanych „kategorii map”, które naszym zdaniem są przydatne do zrozumienia Novamente na poziomie mapy, w sposób ogólny.

Różnorodność map: Opis

Mapa pojęć: Mapa składająca się głównie z węzłów koncepcyjnych

Mapa percepcji: Mapa składająca się głównie z węzłów percepcyjnych, która pojawia się nawykowo, gdy systemowi prezentowane są bodźce środowiskowe pewnego rodzaju

Mapa schematu: Schemat rozproszony

Mapa predykatów: Predykat rozproszony

Mapa pamięci: Mapa składająca się głównie z węzłów oznaczających określone byty (a zatem powiązanych za pośrednictwem MemberLinks i ich krewnych z bardziej abstrakcyjnymi węzłami) i ich relacje

Mapa pojęć i percepcji: Mapa składająca się głównie z węzłów percepcyjnych i koncepcyjnych

Mapa pojęć i schematów: Mapa składająca się głównie z węzłów koncepcyjnych i SchemaNodes

Mapa pojęć i schematów: Mapa składająca się zasadniczo z węzłów percepcyjnych, koncepcyjnych i SchemaNodes

Mapa zdarzeń: Mapa zawierająca wiele linków oznaczających relacje czasowe

Mapa odczuć: Mapa zawierająca FeelingNodes jako istotny składnik

Mapa celów: Mapa zawierająca GoalNodes jako istotny składnik

Interesujący przykład relacji między atomami i mapami w Novamente przedstawiono, przyglądając się implementacji satysfakcji w systemie. Novamente ma FeelingNode, które są „wewnętrznymi czujnikami” raportującymi aspekty bieżącego stanu systemu. Niektóre z nich są elementarne, a niektóre są kombinacjami danych wejściowych z innych FeelingNode. Jednym z ważnych FeelingNode jest Satisfaction FeelingNode, który podsumowuje te czynniki, które system jest początkowo zaprogramowany, aby uznać za „pożądane”. Odnosi się to do MaximizeSatisfaction GoalNode, który jest centrum systemu motywacyjnego Novamente. Na pierwszy rzut oka FeelingNodes wyglądają jak symboliczne reprezentacje uczuć systemowych w stylu AI. Jednak, aby kontynuować analogię do ludzkiego umysłu, te FeelingNodes są w rzeczywistości bardziej podobne do podstawowych bodźców mózgowych układu limbicznego lub w inny sposób chemicznie indukowanych niż do bogatych w teksturę, wysokopoziomowych ludzkich uczuć. W umyśle ludzkim satysfakcja jest o wiele bardziej złożona niż chwilowa przyjemność. Obejmuje oczekiwania satysfakcji w różnych skalach czasowych i obejmuje wnioski na temat tego, co może dać satysfakcję, szacunki, jak bardzo inni będą zadowoleni z

danego kierunku działania, a zatem ile przyjemności dana osoba będzie czerpać ze swojej satysfakcji itp. Przyjemność biologiczna jest w pewnym sensie źródłem ludzkiej satysfakcji, ale relacja ta nie jest relacją tożsamości. Zmiany w biologii przyjemności zazwyczaj skutkują zmianami w doświadczaniu satysfakcji – dowodem jest inna subiektywna struktura ludzkiej satysfakcji w okresie dojrzewania w przeciwieństwie do dzieciństwa lub dojrzałości w przeciwieństwie do wczesnej dorosłości. Ale szczegóły tych zmian są subtelne i indywidualnie różne. Tak więc w tym przykładzie mamy paralelę między bytem na poziomie atomu, Pleasure FeelingNode, a wyłaniającą się mapą umysłu, meta-węzłem, uczuciem satysfakcji w całym systemie lub „szczęścia”. Istnieje znaczne podobieństwo między tymi dwoma równoległymi bytami istniejącymi na różnych poziomach, ale nie ma tożsamości. Satysfakcja jest ucieleśniona w:

- dużym, niejasno zdefiniowanym zbiorze węzłów i połączeń („mapa”);
- dynamicznych wzorcach w systemie, które są indukowane, gdy ten zbiór staje się wysoce aktywny („dynamiczny wzór mapy”).

Satisfaction FeelingNode jest jednym z elementów mapy powiązanych z ogólną satysfakcją systemu lub „szczęściem”. I jest to szczególnie krytyczny element tej mapy, co oznacza, że ma wiele ważnych połączeń z innymi elementami mapy. Oznacza to, że aktywacja przyjemności prawdopodobnie – ale nie jest gwarantowana – spowoduje szczęście.

Mind Agents

Sednem inteligencji Novamente są MindAgents, które dynamicznie aktualizują Atomy w systemie na bieżąco. Niezależnie od tego, jakie dane wejściowe trafiają do systemu lub jakie wymagania są mu stawiane, MindAgents nadal pracują, analizując informacje w systemie i tworząc na ich podstawie nowe informacje. Istnieje kilka MindAgents „konserwacji systemu”, zajmujących się takimi rzeczami, jak zbieranie statystyk systemu, okresowe buforowanie Atomów na dysku, aktualizowanie pamięci podręcznych związanych z przetwarzaniem rozproszonym, obsługa kolejek zapytań od użytkowników i innych maszyn w tym samym klastrze analitycznym lub innych klastrach analitycznych Novamente. Nie będziemy ich tutaj dalej omawiać, ale ograniczymy się do „poznawczych MindAgents”, które działają poprzez modyfikację AtomTable. Tabele 7 i 8 krótko wspominają o kilku istniejących i możliwych MindAgentach, podczas gdy strona internetowa AGIRI podaje kompletną listę MindAgentów z krótkimi komentarzami na temat funkcji każdego z nich na poziomie atomu i mapy.

Agent: Opis

Wnioskowanie pierwszego rzędu: Działa na logicznych łączach pierwszego rzędu, tworząc nowe logiczne łącza ze starych, używając formuł logiki terminów probabilistycznych

Logiczne eksplorowanie łączy: Tworzy logiczne łącza z nielogicznych łączy Ewolucyjne uczenie się predykatów: Tworzy węzły predykatów zawierające predykaty, które przewidują przynależność do węzłów koncepcyjnych

Klastrowanie: Tworzy węzły koncepcyjne reprezentujące klastry istniejących węzłów koncepcyjnych (umożliwiając w ten sposób działanie na klastrze jako na zunifikowanej całości za pomocą precyzyjnych metod wnioskowania, w przeciwieństwie do mniej dokładnej dynamiki na poziomie mapy)

Aktualizowanie znaczenia: Aktualizuje zmienne „znaczenia” atomu i ich powiązane ilości, używając specjalnie wdrożonego wnioskowania probabilistycznego

Tworzenie pojęć: Tworzy spekulatywne, potencjalnie interesujące nowe węzły koncepcyjne

Optimalizacja ewolucyjna: „usługowy” agent MindAgent, używany do uczenia się schematów i predykatów oraz ogólnej optymalizacji parametrów systemu

Agent: Opis

Wnioskowanie wyższego rzędu: Wykonuje operacje wnioskowania na logicznych łączach, które wskazują na łącza i/lub węzły predykatów

Unifikacja logiczna: Wyszukuje atomy, które wzajemnie spełniają parę węzłów predykatów

Tworzenie schematu/predykatu: Tworzy spekulatywne, potencjalnie interesujące nowe węzły schematu

Tworzenie skojarzeń hebbowskich: Buduje i modyfikuje łącza między atomami, w oparciu o specjalne wdrożenie wnioskowania probabilistycznego, które w przybliżeniu emuluje (ale znacznie przewyższa pod względem dokładności) regułę uczenia się przez wzmacnianie hebbowskie

Ewolucyjne uczenie się schematu: Tworzy węzły schematu, które spełniają kryteria, np. które mają spełniać dane GoalNodes

Wykonanie schematu: Ustanawia aktywne SchemaNodes, umożliwiając systemowi wykonywanie skoordynowanych ciągów działań

Enkapsulacja mapy: Skanuje AtomTable w poszukiwaniu wzorców i tworzy nowe Atomy ucieleśniające te wzorce

Rozszerzenie mapy: Przyjmuje schematy i predykaty ucieleśnione w węzłach i rozszerza je na wiele węzłów i łączy w AtomTable (przekształcając w ten sposób złożone Atomy w mapy prostych Atomów)

Adaptacja parametrów homeostatycznych: Stosuje programowanie ewolucyjne w celu adaptacyjnego dostrojenia parametrów systemu

Dynamika mapy

Duża część znaczenia MindAgentów Novamente leży w implikacjach, jakie mają dla dynamiki na poziomie mapy. Tutaj relacja między mapami Novamente a koncepcjami matematycznej teorii układów dynamicznych jest wysoce istotna. Intuicyjna koncepcja mapy jest prosta: mapa to zbiór atomów, które działają jako całość. Mogą działać jako całość w celach poznawczych, percepcyjnych lub działania. A działając jako całości, mogą się ze sobą odnosić, tak jak atomy mogą się ze sobą odnosić. Relacje między mapami nie przyjmują formy pojedynczych linków; przybierają formę wiązek łączy łączących atomy wewnątrz jednej mapy z atomami wewnątrz innej. Dynamika map jest nieco „mniej płynna” do omówienia niż dynamika atomów, ponieważ mapy nie są jawnie projektowane – one się pojawiają. Aby dowiedzieć się, jakie atomy są obecne w systemie w danym momencie, wystarczy wydrukować AtomTable. Aby dowiedzieć się, jakie mapy są obecne, trzeba wykonać pewne zaawansowane rozpoznawanie wzorców w przestrzeni atomów, aby określić, które zestawy węzłów działają w rzeczywistości jako skoordynowane całości. Jednak mapa nie musi być jawnie identyfikowana przez nikogo, aby wykonać swoje zadanie. Mapy istnieją niejawnie w dynamicznym systemie Novamente, wyłaniając się z dynamiki na poziomie atomów, a następnie kierując tą dynamiką. W terminologii systemów dynamicznych istnieją dwa rodzaje map: mapy atraktorów i mapy przejściowe. Mapy schematów i predykatów są na ogół przejściowe, podczas gdy mapy pojęć i perceptów są na ogół atraktorami; ale nie jest to twarda i szybka reguła. Inne rodzaje map mają większą wewnętrzną różnorodność dynamiczną, na przykład będą pewne mapy odczuć związane z dynamiką przejściową, a inne z dynamiką atraktora. Sens, w jakim termin „atraktor” jest tutaj używany, jest nieco

niestandardowy. W teorii układów dynamicznych atraktor zwykle oznacza podzbiór przestrzeni stanów systemu, który jest: niezmienny, gdy system znajduje się w tym podzbiorze przestrzeni stanów, nie opuszcza go; atrakcyjny, gdy system znajduje się w stanie zbliżonym do tego podzbioru przestrzeni stanów, będzie się przemieszczał coraz bliżej przyciągającej podprzestrzeni. W Novamente podzbiór przestrzeni stanów odpowiadający mapie jest zbiorem stanów systemu, w których ta mapa jest wysoce ważna. Jednak w dynamice Novamente te podzbiory przestrzeni stanów prawie nigdy nie są prawdziwie niezmiennie. Wiele map jest atrakcyjnych, ponieważ dynamika aktualizacji ważności Novamente zachowuje się mniej więcej jak sieć neuronowa atraktora. Gdy większość mapy jest bardzo ważna, reszta mapy otrzyma dużo aktywacji, co uczyni ją bardzo ważną. Z drugiej strony, atomy połączone z elementami mapy za pomocą połączeń hamujących otrzymają mniej aktywacji i staną się mniej ważne. Ale mapy nie są niezmiennie: gdy mapa jest aktywna, nie ma gwarancji, że pozostanie aktywna na zawsze. Raczej funkcja aktualizacji ważności, regulująca dynamikę Novamente, gwarantuje, że w większości przypadków, po tym jak mapa była ważna przez jakiś czas, stanie się mniej ważna, ponieważ procent nowych rzeczy, których się o niej nauczono, stanie się mniejszy niż procent nowych rzeczy, których się nauczono o czymś innym. Ta kombinacja atrakcyjności i tymczasowej niezmienności, którą widzimy w związku z mapami Novamente, została zbadana przez fizyka Michaiła Zaka, który nazwał podzbiory przestrzeni stanów z tą właściwością atraktorami końcowymi. Stworzył proste matematyczne układy dynamiczne z atraktorami końcowymi, używając funkcji iteracyjnych zawierających osobliwości matematyczne. W ten sposób zbudował kilka interesujących modeli sieci neuronowych. Równania rządzące Novamente niewiele przypominają równania Zaka, ale intuicyjnie rzecz biorąc, wydają się one dzielić własność prowadzenia do atraktorów końcowych, w luźnym sensie podzbiorów przestrzeni stanów, które są atrakcyjne, ale są niezmiennie tylko przez pewien okres czasu. Wiele map pojęć będzie odpowiadać atraktorom mapy punktów stałych – co oznacza, że są one zbiorami atomów, które, gdy staną się ważne, będą miały tendencję do pozostawania ważnymi przez jakiś czas ze względu na wzajemne wzmacnianie. Z drugiej strony, niektóre mapy pojęć mogą odpowiadać bardziej złożonym wzorcom dynamicznym map. A mapy zdarzeń mogą czasami manifestować wzorec dynamiczny imitujący zdarzenie, które reprezentują. Ten rodzaj reprezentacji wiedzy jest dobrze znany w literaturze dotyczącej sieci neuronowych atraktorowych. Wracając do schematów, pojedynczy SchemaNode niekoniecznie reprezentuje cały schemat o jakimkolwiek znaczeniu mentalnym – może to robić, szczególnie w przypadku dużego schematu hermetyzowanego; ale częściej będzie częścią rozproszonego schematu (co oznacza, że SchemaNode można by dokładniej nazwać LikelySchemaMapComponentNode). I oczywiście rozproszony schemat czerpie swoje znaczenie z tego, co robi, gdy jest wykonywany. Rozproszony schemat jest rodzajem mapy myśli – mapy, która wykracza poza SchemaInstanceNode i SchemaNodes, wprowadzając inne węzły, które są zwyczajowo aktywowane, gdy SchemaInstanceNodes na mapie są wykonywane. Należy zauważyć, że to zachowanie systemu może wykraczać poza działania wyraźnie ucieleśnione w SchemaNode zawartym w rozproszonym schemacie. Wykonywanie tych SchemaNodes w określonej kolejności może mieć powszechne skutki uboczne w całym systemie, a te skutki uboczne mogły zostać wzięte pod uwagę, gdy schemat został nauczony, stanowiąc kluczową część „dopasowania” schematu. Następnie, percepty – elementy danych – trafiające do systemu niekoniecznie są reprezentowane przez poszczególne węzły percepcyjne. Na przykład, wystąpienie słowa, które pojawiło się w systemie podczas procesu czytania, będzie reprezentowane na wiele równoczesnych sposobów. Może istnieć WordInstanceNode, ListLink CharacterInstanceNodes itd. W systemie wyposażonym w wizję, reprezentacja obrazu słowa będzie przechowywana. Będą one ze sobą powiązane i połączone z innymi węzłami percepcyjnymi i koncepcyjnymi, a być może z SchemaNodes ucieleśniającymi procesy mówienia słowa lub tworzenia liter zaangażowanych w słowo. Ogólnie rzecz biorąc, percepty są bardziej prawdopodobne, aby były ucieleśniane przez mapy, które są skoncentrowane na indywidualnych węzłach percepcyjnych (w tym przypadku WordInstanceNode), ale nie będzie to

koniecznie i powszechnie prawdą. Linki mają również swoje korelaty na poziomie mapy i w wielu przypadkach najlepiej je traktować jako nasiona, które dają początek relacjom między mapami. Na przykład InheritanceLink reprezentuje relację częstotliwości między węzłami lub linkami, ale relacje dziedziczenia między mapami również istnieją. Relacja dziedziczenia między dwiema mapami A i B nie będzie na ogół ucieleśniona w pojedynczym łączu, będzie ona niejawną w zestawie łącz dziedziczenia obejmujących atomy należące do A i atomy należące do B. To samo dotyczy wszystkich innych odmian relacji logicznej. Co więcej, reguły wnioskowania pierwszego rzędu z Probabilistic Term Logic, systemu rozumowania Novamente, przenoszą się naturalnie na łącza logiczne na poziomie map.

Specjalizacja funkcjonalna

Teraz wracamy do architektury DINI i jej konkretnego zastosowania w Novamente. MindAgents Novamente są zaprojektowane tak, aby były ściśle zintegrowane, tak aby duża kolekcja MindAgents działała na dużej populacji Atomów w sposób przeplatany. Ten zestaw Atomów może znajdować się na jednej maszynie lub w klastrze połączonych maszyn. Ten rodzaj ścisłej integracji jest niezbędny do działania zintegrowanej AGI. Jednak zgodnie z projektem Novamente wymagana jest również inna warstwa, warstwa luźnej integracji na wierzchu ściśle zintegrowanej warstwy. System Novamente składa się z luźno zintegrowanej kolekcji „klasterów analitycznych” lub „jednostek”, z których każda ucieleśnia ściśle powiązaną kolekcję procesów AI, obejmującą wiele różnych typów Atomów i MindAgents, i dedykowaną określonemu przetwarzaniu poznawczemu w określonej domenie lub o określonym ogólnym charakterze. Różne klastry analityczne oddziałują na siebie za pośrednictwem DINI; wszystkie pobierają dane z tego samego systemowego magazynu danych i umieszczają je w nim. W niektórych przypadkach mogą również wzajemnie się przeszukiwać. A parametry MindAgents wewnątrz różnych klastrów analitycznych mogą być dostosowywane i optymalizowane globalnie. Najprostszy wieloklastrowy Novamente ma trzy jednostki, mianowicie:

1. podstawową jednostkę poznawczą;
2. jednostkę myślenia tła, zawierającą wiele innych węzłów z tylko bardzo ważnymi relacjami między nimi, istniejącą tylko po to, aby dostarczać podstawowej jednostce poznawczej rzeczy, które uznaje ona za istotne;
3. jednostkę AttentionalFocus, zawierającą niewielką liczbę atomów i wykonującą na nich bardzo intensywne przetwarzanie zasobów.

W tym przypadku specjalizacja dotyczy intensywności przetwarzania, a nie zawartości przetwarzania. Aby Novamente mógł intensywnie oddziaływać ze światem zewnętrznym, powinien mieć dwa dedykowane klastry dla każdego „kanału interakcji”:

- jeden do przechowywania schematów kontrolujących interakcję;
- jeden do przechowywania „pamięci krótkotrwałej” odnoszącej się do interakcji.

„Kanał interakcji” to zbiór narządów zmysłów jakiejś formy, wszystkie postrzegające mniej więcej ten sam segment rzeczywistości zewnętrznej. Każdy człowiek ma tylko jeden kanał interakcji. Jednak Novamente nie naśladuje ściśle ani ludzkiego ciała, ani mózgu, więc może łatwo znaleźć się w takiej sytuacji, wchodząc w interakcje z ludźmi w różnych miejscach na świecie. Przetwarzanie percepcyjne, takie jak przetwarzanie obrazu lub dźwięku, najlepiej wykonywać w specjalnie dedykowanych jednostkach, z wartościami parametrów dostrojonymi do modalności. Przetwarzanie języka wymaga również wyspecjalizowanych jednostek, zajmujących się konkretnie aspektami przetwarzania języka, takimi jak analiza składniowa, mapowanie semantyczne i ujednoznacznianie. Ludzki mózg zawiera ten rodzaj specjalizacji funkcjonalnej w dużym stopniu. W rzeczywistości wiemy więcej o specjalizacji

różnych części mózgu niż o tym, jak faktycznie wykonują one swoje wyspecjalizowane zadania. Każdy wyspecjalizowany moduł mózgu wydaje się wykorzystywać mieszaną tych samych reprezentacji danych i procesów uczenia się. Wiele systemów AI zawiera podobną strukturę modułową, ale każdy moduł zawiera wiele wysoce sztywnego, wyspecjalizowanego kodu. Podejście tutaj jest zupełnie inne. Zaczyna się od zbioru aktorów, którzy w sposób wyłaniający się zapewniają ogólną zdolność poznawczą, a następnie kształtuje się dynamiczne wzorce ich interakcji poprzez specjalizację funkcjonalną.

Novamente i ludzki mózg

Po przejrzaniu kluczowych aspektów projektu Novamente, powrócimy teraz na krótko do wcześniej wspomnianego tematu, relacji między Novamente a ludzkim mózgiem. Chociaż Novamente nie próbuje naśladować struktury ani dynamiki neuronowej, istnieją jednak pewne znaczące paralele. Na poziomie strukturalnym paralele są dość bliskie: funkcjonalnie wyspecjalizowane płaty Novamente są mniej więcej analogiczne do różnych obszarów mózgu. Na poziomie pośrednim węzły Novamente są mniej więcej analogiczne do grup neuronowych w mózgu, jak wspomniano powyżej; a połączenia Novamente są jak wiązki synaps łączące grupy neuronowe. Mapy Novamente są jak mapy neuronowe Edelmana, a także w niektórych przypadkach jak atraktory neuronowe postulowane przez teoretyków, takich jak Daniel Amit i Walter Freeman. Paralele słabną jednak, gdy zwracamy się ku dynamice. Bardzo niewiele wiadomo o dynamice mózgu na poziomie pośrednim. Zasadniczo wiemy, jak działają neurony, ale niewiele wiemy o dynamice wzajemnych powiązań poziomów różnych typów neuroprzekazników w różnych obszarach mózgu, ani o dyfuzji ładunku pozakomórkowego, ani nawet o dynamicznym zachowaniu złożonych kolektywów prawdziwych neuronów. Novamente ma szereg specyficznych dynamik poznawczych (np. wnioskowanie probabilistyczne), które nie mają znanych analogów w dynamice mózgu; ale to niewiele znaczy, ponieważ dynamika mózgu na poziomie pośrednim jest tak słabo poznana.

Struktury wyłaniające się

Dynamika systemu Novamente jest w dużej mierze kontrolowana przez strukturę hipergrafu Atom, a struktura hipergrafu Atom jest silnie kierowana, a częściowo wyraźnie kształtowana, przez dynamikę systemu. To strukturalno-dynamiczne sprzężenie zwrotne może prowadzić do wszelkiego rodzaju złożonych struktur emergentnych – niektóre istniejące w systemie w danym czasie, niektóre manifestujące się jako wzorce w czasie, a niektóre o charakterze czasoprzestrzennym. Mapy są jednym z przejawów tego sprzężenia zwrotnego; ale istnieje również wyższy poziom organizacji, w którym sieć map osiąga pewne wzorce emergentne. Wśród tych wzorców emergentnych znajdują się te zidentyfikowane w modelu umysłu psyne: dualna sieć i jaźń.

Sieć podwójna

Sieć podwójna w Novamente przyjmuje dość prostą i bezpośrednią formę:

- aspekt heterarchiczny składa się z podsieci zdefiniowanej przez symetryczne łącza logiczne i/lub łącza asocjacyjne;
- aspekt hierarchiczny składa się z podsieci zdefiniowanej przez asymetryczne łącza logiczne i łącza asocjacyjne oraz podsieci zdefiniowanej przez schematy i ich relacje kontrolne (schemat A kontroluje schemat B, gdy A modyfikuje parametry B znacznie bardziej niż odwrotnie).

Pomijając schematy, statyczny aspekt sieci podwójnej jest dość prosty. Na przykład węzły koncepcyjne odpowiadające różnym narodom mogą być połączone łączami podobieństwa i łączami asocjacyjnymi: jest to mała „sieć heterarchiczna”, podzbiór ogólnej sieci heterarchicznej w danej przestrzeni atomów

Novamente. Te węzły reprezentujące poszczególne narody mogą dziedziczyć po węźle koncepcyjnym narodu (łączy dziedziczenia jest asymetrycznym łączem logicznym). To prosty, statyczny przykład podwójnej struktury sieciowej: elementy, które są powiązane heterarchicznie, są również blisko siebie w swoich hierarchicznych relacjach. Ten aspekt podwójnej struktury sieciowej wypada dość naturalnie z wewnętrznej semantyki podobieństwa, dziedziczenia i skojarzenia. Aspekt kontroli podwójnej sieci jest mniej oczywisty i pojawi się tylko wtedy, gdy różne MindAgents będą działać razem prawidłowo. Na przykład rozważmy rodzinę schematów, z których każdy zajmuje się rozpoznawaniem części mowy: rzeczowników, czasowników, przymiotników itd. Te schematy będą miały podobieństwa i powiązania ze sobą. Wszystkie odziedziczą po garstce bardziej ogólnych schematów do analizy słów i ich właściwości. Będą jednak również kontrolowane przez te bardziej ogólne schematy analizy słów. Ich parametry kontroli i przepływ wykonywania będą modulowane przez te bardziej ogólne schematy kontroli. Zbieżność hierarchii dziedziczenia i hierarchii kontroli oraz nakładanie się tej zbieżnej hierarchii na heterarchię asocjacyjno-podobieństwa jest sednem struktury „podwójnej sieci”. Nie jest ona zaprogramowana w Novamente, ale Novamente jest tak zaprojektowana, aby zachęcić ją do wyłonienia się. Konkretnie, pojawienie się tego rodzaju metawzorca dualnej sieci następuje dość naturalnie z harmonijnej interakcji:

- wnioskowanie o budowaniu podobieństw i dziedzicznych powiązań;
- aktualizacja ważności, kierowanie aktywacją atomów (a zatem stosowanie wbudowanych podstawowych procesów poznawczych do atomów) w oparciu o powiązania między nimi;
- uczenie się schematu, które rozszerza stosowalność schematu z jednego węzła na drugi w oparciu o istniejące powiązania między nimi (i w oparciu o obserwacje przeszłych sukcesów i porażek schematów, jak zostanie wyjaśnione później).

Struktura dualnej sieci jest statyczną reprezentacją dynamicznej współpracy tych procesów. Omówiliśmy ją tutaj na poziomie atomu, ale jej manifestacja na poziomie mapy jest w dużej mierze równoległa i równie ważna.

Jaźń

Równie ważna jak dualna sieć jest struktura umysłu, którą nazywamy „jaźnią”. Podkreślamy, że używamy roboczej definicji jaźni, ukierunkowanej na słuźenie jako użyteczna wytyczna dla inżynierii AGI. Celowo unikamy ontologicznych lub egzystencjalnych dyskusji na temat uniwersalnej natury jaźni i jej związku ze świadomością. „Surowiec” dla jaźni Novamente – podstawowe zmysły, w których Novamente może dokonać autorefleksji – składa się ze zbioru:

- wzorców, które system zaobserwował w sobie jako całości, czyli strukturalnych i dynamicznych wzorców w swojej wewnętrznej sieci dualnej;
- wzorców, które zaobserwował w swoich własnych zewnętrznych działaniach, czyli tej podsieci swojej sieci dualnej, która obejmuje śledzenie procedury i konsekwencji uruchamiania różnych schematów;
- wzorców, które system zaobserwował w swojej relacji z innymi inteligentnymi systemami. To, co nazywamy jaźnią, jest zbiorem wzorców rozpoznanych w tym zestawie.

Często rozpoznane wzorce są bardzo przybliżone, ponieważ zbiór zaangażowanych danych jest ogromny i różnorodny – nawet komputer nie ma zasobów, aby zapamiętać każdy szczegół każdej rzeczy, którą kiedykolwiek zrobił. Co więcej, konkretne elementy danych prowadzące do tworzenia wzorców psynetowych, które definiują self, będą często zapominane, tak że self jest słabo

ugruntowanym wzorcem (dostrojenie, jak słabo ugruntowane może być, a nadal być użyteczne, będzie subtelną i kluczową częścią nadania Novamente użytecznego, nietrywialnego poczucia self).

Na poziomie mapy możemy powiedzieć, że self składa się z:

- zestawu map obrazu self: map, które służą jako „wewnętrzne obrazy” istotnych aspektów struktury lub dynamiki systemu Novamente lub jego interakcji z innymi inteligentnymi systemami;
- większej mapy, która obejmuje różne mapy obrazu self wraz z innymi Atomami (to jest emergentne self).

Naprawdę interesującą rzeczą w self jest sprzężenie zwrotne między deklaratywną, zlokalizowaną wiedzą a rozproszoną, proceduralną wiedzą, którą ucieleśnia. W miarę jak zbiór wzorców wysokiego poziomu, które są self, staje się bardziej lub mniej aktywny, automatycznie przesuwa ogólnie zachowanie systemu we właściwych kierunkach. Oznacza to, że gdy system obserwuje i rozumie na podstawie swoich wzorców jaźni, może następnie dostosować swoje zachowanie, kontrolując różne procesy wewnętrzne w taki sposób, aby faworyzować wzorce, które, jak zaobserwowano, przyczyniają się do spójnego myślenia, „dobrych uczuć” i realizacji celów. Zauważamy kluczową rolę interakcji z ludźmi w rozwoju jaźni Novamente. Podczas gdy teoretycznie możliwe byłoby posiadanie jaźni bez społeczeństwa, społeczeństwo znacznie to ułatwia, dostarczając znacznie więcej danych do samokształtowania — a aby jaźń mogła wystarczająco funkcjonować w świecie, w którym istnieją inne jaźnie, społeczeństwo jest niezbędne. Z czasem może być interesujące stworzenie społeczności interaktywnych systemów sztucznej inteligencji Novamente. Początkowo Novamente będzie uczyć się o sobie poprzez interakcję z ludźmi. Gdy ludzie będą zadawać mu pytania i rozmawiać z nim, będzie zdobywał coraz więcej informacji nie tylko o ludziach, ale także o tym, czym jest samo Novamente, z punktu widzenia innych. Ukształtuje to jego przyszłe działania zarówno jawnie, jak i niejawnie.

Interakcja z ludźmi i magazynami danych

Wdrożenie Novamente do zarządzania wiedzą i analizowania jej wymaga zwrócenia uwagi na wiele kwestii wykraczających poza te występujące w odniesieniu do „Novamente AGI” w ogólności. Większość z tych kwestii mieści się w kategoriach źródeł danych i interakcji człowiek-komputer. Optymalny sposób radzenia sobie z takimi problemami zależy od domeny. W przypadku aplikacji bioinformatycznych przyjęliśmy podejście kierujące się szczególnymi potrzebami biologów analizujących zestawy danych generowane za pomocą wysokoprzepustowego sprzętu genomicznego. Ta sekcja zawiera krótki opis naszych planów w tych obszarach. Kluczowym punktem koncepcyjnym pojawiającym się tutaj jest związek między AI i IA (Intelligence Augmentation). Pomimo ambitnych długoterminowych celów AGI, jest jasne, że w średnim okresie Novamente nie przewyższy ludzkiej inteligencji pod każdym względem. Raczej należy je postrzegać jako uzupełnienie indywidualnej i zbiorowej ludzkiej inteligencji. Ludzie będą uczyć się z unikalnych spostrzeżeń Novamente, a Novamente będzie również uczyć się od ludzi. Konkretnie, Novamente wykorzystuje ludzką inteligencję poprzez:

- pobieranie danych zakodowanych przez ludzi w bazach danych;
- pobieranie wiedzy specjalnie zakodowanej przez ludzi do użytku Novamente;
- uczenie się z dialogów i interakcji z ludźmi;
- ludzkie konstruowanie zestawów treningowych do nadzorowanej kategoryzacji;
- uczenie się z ocen ludzi dotyczących własnych odpowiedzi i odpowiedzi innych ludzi na zapytania;

Projektowanie odpowiednich interfejsów użytkownika ucieleśniających te czynniki jest samo w sobie znaczącym przedsięwzięciem, którego nie będziemy rozwijać w tym rozdziale. Tutaj ograniczymy się do krótkiego omówienia kluczowych wymaganych funkcji i najbardziej istotnych problemów, które się z nimi wiążą.

Źródła danych

Omówiliśmy już kwestie koncepcyjne związane z zasilaniem baz danych Novamente w sekcji powyżej. Jak zauważono tam, Novamente ma współpracować z zewnętrznymi bazami danych, które zostały zintegrowane zgodnie z metodologią „integracji wiedzy”. Oznacza to, że translatory muszą być napisane, mapując schematy w DB na XML ustrukturyzowane zgodnie z XML DTD Novamente. To skutecznie mapuje informacje z bazy danych na węzły i łączy Novamente. W ten sposób można zbudować ujednolicony magazyn danych, zawierający różnorodną mieszankę danych i abstrakcyjnych informacji. Biorąc pod uwagę znaczną ilość wiedzy w nowoczesnych bazach danych jako tekstową, a nie ustrukturyzowaną, Novamente może przyswajać tekst przy użyciu uproszczonych metod statystycznych, a my eksperymentowaliśmy z tym w kontekście prac badawczych z zakresu biologii. Jednak prawdziwe zrozumienie języka naturalnego uzyskuje się tylko poprzez pozostawienie przetwarzania tekstu i umożliwienie Novamente tłumaczenia w obie strony między sekwencjami znaków językowych z jednej strony, a węzłami i linkami o znaczeniu semantycznym z drugiej. Wymaga to, aby przetwarzanie języka naturalnego było implementowane w bardzo głęboki sposób, jako część abstrakcyjnego poznania Novamente. Wierzmy, że projekt Novamente może przewyżżyć problemy doświadczane przez współczesne algorytmy NLP, dzięki swojemu zintegrowanemu podejściu, które obejmuje przeprowadzanie analizy składniowej poprzez logiczną unifikację, proces, który automatycznie włącza dostępną wiedzę semantyczną i pragmatyczną do swojego zachowania. Nie zaimplementowaliśmy jeszcze NLP w systemie Novamente, ale nasze doświadczenie z podobną implementacją w systemie Webmind daje nam tutaj uzasadnioną pewność. Wracamy do tej kwestii poniżej.

Kodowanie wiedzy

Czasami dane dostępne w istniejących bazach danych nie wystarczą, aby Novamente „nadążyło” za stosownym obszarem. Znaczna część ludzkiej wiedzy jest „milcząca” i nigdy nie jest nigdzie zapisywana, tekstowo ani w formie relacyjnej lub ilościowej. Ponadto w przypadku wiedzy wyrażonej tylko w trudnych do zrozumienia dokumentach tekstowych, zrozumienie Novamente może zostać poprawione poprzez dostarczenie jej części wiedzy w formie jawnej. Z tych powodów czasami warto, aby ludzie kodowali wiedzę formalnie, specjalnie do przyswojenia przez Novamente. Istnieją tutaj dwa różne podejścia:

- formalne kodowanie wiedzy w stylu „systemu eksperckiego”;
- wprowadzanie wiedzy za pośrednictwem interaktywnych formularzy internetowych.

Podejście formularzy internetowych zostało opracowane w Webmind Inc. i wydawało się rozsądnym sposobem dla osób z niewielkim przeszkoleniem na kodowanie dużych ilości stosunkowo prostych informacji. Do formalnego kodowania języka opracowaliśmy formalny język o nazwie NQL, który jest podobny do Cyc-L, ale ma znacznie prostszą składnię. Ostrzegamy, że nie proponujemy tutaj tradycyjnego podejścia „systemów eksperckich” ani tradycyjnego projektu kodowania wiedzy „zdrowego rozsądku” a la Cyc. Przewidujemy, że znacznie mniej niż 1% wiedzy w Novamente zostanie

tam umieszczone za pośrednictwem kodowania wiedzy ludzkiej. Naszym zdaniem rolą kodowania wiedzy powinno być wypełnianie luk, a nie dostarczanie podstawowej bazy wiedzy.

Zapytania

Omówiliśmy, w jaki sposób wiedza trafia do Novamente – ale jak się stamtąd wydostaje? W jaki sposób ludzie zadają pytania Novamente? W jaki sposób iterują z systemem, aby wspólnie znajdować i produkować wiedzę? Naszym zamiarem jest stworzenie prototypowego interfejsu użytkownika, który ma charakter integracyjny, obejmując szereg uzupełniających się mechanizmów. Praktyczne eksperymenty z tymi mechanizmami w kontekście analizy danych w świecie rzeczywistym nauczą nas, które z nich są uważane za najbardziej wartościowe przez użytkowników ludzkich w jakich kontekstach; i będzie to stanowić podstawę dalszego udoskonalania interfejsu użytkownika Novamente, a także wewnętrznych mechanizmów przetwarzania zapytań Novamente.

Zapytania w języku formalnym

Dla nieprzeszkolonych użytkowników zapytania w języku naturalnym i konwersacje w języku naturalnym są zdecydowanie najbardziej pożądanymi sposobami interakcji. Z drugiej strony, dla przeszkolonych użytkowników ekspertów, mogą istnieć znaczące korzyści z używania zapytań w języku formalnym lub zapytań łączących język formalny z naturalnym. Zapytania formalne umożliwiają poziom precyzji nieosiągalny przy użyciu języka naturalnego. Co więcej – i jest to punkt krytyczny – dzięki temu, że użytkownicy eksperci będą przysyłać te same zapytania zarówno w formacie języka naturalnego, jak i języka formalnego, Novamente uzyska pragmatyczną wiedzę na temat interpretacji zapytań. Jest to przykład tego, jak Novamente może uczyć się od ludzi, którzy przynajmniej początkowo będą o wiele mądrzejsi od niego w interpretowaniu złożonych zdań w języku ludzkim. Na przykład rozważmy zapytanie:

Chcę uzyskać informacje o wybitnych restauracjach sushi na Manhattanie

Jako formalne zapytanie językowe, wygląda to po prostu tak:

Znajdź X, Y tak, aby:

Dziedziczenie X „japońska restauracja”

lokalizacja X Manhattan

sprzedaje X Y

Dziedziczenie Y Sushi

jakość Y wybitne

Albo rozważ:

Chcę uzyskać informacje o dochodowych firmach ze Stanów Zjednoczonych, które sprzedają swoje usługi szkołom.

Takie zdanie stwarza problemy interpretacyjne dla obecnych systemów NLP. Mają one problem z określeniem, co jest poprzednikiem „ich”: „dochodowe firmy” czy „Stany Zjednoczone”. Dokonanie właściwego wyboru wymaga zrozumienia świata rzeczywistego lub rozległego dostosowywania systemu w określonej domenie. Z drugiej strony, dla doświadczonego użytkownika utworzenie odpowiedniego formalnego zapytania językowego, aby objąć ten przypadek, jest łatwe:

Dziedziczenie X „Stany Zjednoczone”

oparte na Y X

Dziedziczenie Y dochodowe

sprzedaje Y Z

kupuje Z W

Dziedziczenie W „szkoła”

Początkowy system NLP Novamente może czasami popełniać błędy przy rozwiązywaniu zdań, takich jak powyższe. Jeśli użytkownik prześle to zapytanie do Novamente zarówno w języku angielskim, jak i w języku formalnym, Novamente zaobserwuje poprawną interpretację zdania i odpowiednio dostosuje swoje schematy mapowania semantycznego (poprzez aktywność agentów umysłu uczących się schematów). Następnie następnym razem, gdy zobaczy podobne zdanie, będzie bardziej prawdopodobne, że wyda właściwy osąd. Kiedy ludzkie dziecko uczy się języka, koryguje swoje interpretacje poprzez obserwowanie interpretacji wypowiedzi innych osób w świecie rzeczywistym. Novamente będzie miał mniej okazji niż ludzie do dokonywania tego rodzaju korekt opartych na obserwacji, ale jako częściową rekompensatę ma zdolność porównywania zdań języka naturalnego z tworzonymi przez ekspertów formalnymi wersjami języka. A to „nauczanie języka” nie musi być wykonywane jako specjalny proces, może występować jako część zwykłego użytkownika systemu, ponieważ doświadczeni użytkownicy przesyłają formalne zapytania językowe i zapytania NL obok siebie.

Interakcja konwersacyjna

Modal interakcji „zapytanie/odpowieź” jest ważna i cenna, ale ma swoje ograniczenia. Często chce się mieć serię interakcji z trwałym kontekstem – tj. rozmowę. Novamente jest zaprojektowane tak, aby to obsługiwać, jak również bardziej konwencjonalne interakcje zapytanie/odpowieź. Obecnie prototypujemy konwersacje Novamente w kontekście środowiska zabawkowego ShapeWorld. Interakcja konwersacyjna dobrze współgra z ideą mieszanej komunikacji formalnej/naturalnej omówionej powyżej. Przykład konwersacji podany w tabeli 3 powyżej ilustruje tę koncepcję konkretnie. Ponieważ obecnie nie wdrożyliśmy żadnego NLP w Novamente, osiągnięcie tego rodzaju konwersacji z Novamente pozostaje przedsięwzięciem badawczo-rozwojowym ze zwykłymi towarzyszącymi ryzykami. Nasze obecne zastosowania Novamente są bardziej zgodne z analizą danych. Jednak wykonaliśmy prototyp interaktywnej konwersacji w powiązonym systemie oprogramowania Webmind, w ograniczonym zakresie, i dzięki temu doświadczeniu uzyskaliśmy dogłębne zrozumienie problemów związanych z podejściem do takiej funkcjonalności.

Generowanie raportów

Inną przydatną (i znacznie prostszą) funkcjonalnością interakcji człowieka jest generowanie raportów. System będzie mógł automatycznie generować raporty podsumowujące zawierające informacje istotne dla zapytań użytkowników lub po prostu podsumowujące interesujące wzorce znalezione przez własną spontaniczną aktywność. Raporty mogą zawierać:

- dane ilościowe;
- relacje wyrażone w języku formalnym (wyrażenia predykatów);
- język naturalny wytworzony przez algorytmy „generowania języka” i „podsumowania tekstu”.

Aktywne filtrowanie współpracy i modelowanie użytkowników

Na koniec Novamente zbierze informacje o preferencjach człowieka w ogóle i preferencjach poszczególnych użytkowników za pomocą technik udoskonalonych w społeczności „aktywnego filtrowania współpracy”. Zasadniczo oznacza to, że użytkownicy zostaną poproszeni o ocenę odpowiedzi Novamente w kilku skalach (np. przydatność, prawdziwość). Ponadto interfejs użytkownika Novamente zostanie skonfigurowany tak, aby zbierać „ukryte oceny” – informacje dotyczące tego, jak długo patrzą na dany element informacji, do czego go wykorzystują itd. Novamente włączy te informacje do swojego magazynu wiedzy, aby wykorzystać je jako przedmiot bieżącej analizy wzorców, co umożliwi dostosowanie zachowania w celu lepszej obsługi przyszłych użytkowników.

Przykładowe procesy AI Novamente

W tej sekcji krótko omówimy kilka najważniejszych procesów AI w systemie Novamente: wnioskowanie probabilistyczne, nieliniowe przydzielanie uwagi, uczenie się procedur, eksploracja wzorców, kategoryzacja i przetwarzanie języka naturalnego. Procesy te stanowią przyzwoity przekrój tego, co dzieje się w Novamente. Zilustrujemy każdy proces intuicyjnym przykładem tego, co proces ten wnosi do Novamente.

Wnioskowanie probabilistyczne

Wnioskowanie logiczne było głównym tematem badań nad AI od samego początku. Istnieje wiele różnych podejść, w tym:

- logika predykatów, np. Cyc , SNARK ;
- logika kombinatoryczna;
- logika niepewnych wyrazów, np. System rozumowania nieaksjomatycznego Pei Wanga (NARS);
- wnioskowanie probabilistyczne, np. sieci Bayesa [55], probabilistyczne programowanie logiczne;
- logika rozmyta ;
- logika parakonsystentna ;
- logika niemonotoniczna.

Podstawowym zadaniem obliczeniowej dedukcji logicznej jest rozwiązany problem, ale w obszarze AI i logiki nadal istnieje wiele otwartych problemów, na przykład:

- kontrola wnioskowania (jakie wnioski wyciągać i kiedy);
- reprezentacja i manipulacja niepewnością (rozmyta vs. probabilistyczna vs. wieloskładnikowa wartość logiczna itp.);
- optymalna logiczna reprezentacja określonych typów wiedzy, takich jak wiedza czasowa i proceduralna;
- wnioskowanie wykraczające poza dedukcję, takie jak indukcja, abdukcja i analogia;

W przypadku tych aspektów wnioskowania istnieje wiele podejść, w których nie ma konsensusu i niewiele jest ujednoczonych ram. Cyc jest prawdopodobnie najbardziej ambitną próbą zjednoczenia wszystkich różnych aspektów wnioskowania logicznego, ale jest słaby w zakresie wnioskowania

niededukcyjnego, a jego mechanizmy kontroli są wysoce specyficzne dla danej dziedziny i wyraźnie niewystarczające. Praktyczna potrzeba wnioskowania logicznego w kontekście bezpieczeństwa narodowego jest oczywista. Między innymi wnioskowanie może:

- syntetyzować informacje z wielu baz danych;
- pomagać w interpretacji języka naturalnego;
- pomagać w dopasowywaniu zapytań użytkowników do wiedzy systemowej;
- wyciągać złożone wnioski na podstawie integracji ogromnej liczby małych fragmentów informacji.

Aby wziąć przykład integracji baz danych, gdy baza danych 1 mówi:

„Pieniądze często płyną z banku XYZ do Luksemburga”.

A Baza Danych 2 mówi:

„M. Jones ma znaczne środki w banku XYZ”.

Następnie wnioskowanie abdukcyjne mówi:

„Może M. Jones wysyła pieniądze do Luksemburga”.

co jest spekulatywnym, ale potencjalnie interesującym wnioskiem. Komponent wnioskowania logicznego Novamente składa się z szeregu MindAgentów do tworzenia logicznych powiązań, zarówno z innych logicznych powiązań („wnioskowanie”), jak i z powiązań nielogicznych („bezpośrednia ocena”). Obejmuje on kilka różnych MindAgentów:

- LogicalLinkMiner MindAgent (buduje logiczne powiązania z powiązań nielogicznych)
- FirstOrderInference MindAgent
- HigherOrderInference MindAgent
- LogicalUnification MindAgent
- PredicateEvaluation MindAgent
- TemporalInference MindAgent

Tutaj omówimy tylko jeden z nich, FirstOrderInference (FOI) MindAgent. Ten agent realizuje trzy podstawowe reguły wnioskowania, dedukcję, inwersję i rewizję. Konwertuje również relacje podobieństwa na relacje dziedziczenia i odwrotnie. Każda z reguł wnioskowania ma charakter probabilistyczny, wykorzystując specjalny wzór do wzięcia prawdopodobnych wartości logicznych przesłanek i wyprowadzając prawdopodobną wartość logiczną dla wniosku. Wzory te są wyprowadzane przy użyciu nowatorskiej struktury matematycznej zwanej Probabilistic Term Logic (PTL). Wzór inwersji PTL jest zasadniczo regułą Bayesa; wzór dedukcyjny jest unikalny dla Novamente, chociaż można go po prostu wyprowadzić z elementarnej teorii prawdopodobieństwa. Rewizja jest ważoną regułą uśredniania, która łączy różne szacunki wartości logicznej tej samej relacji, pochodzące z różnych źródeł. Reguły dotyczą zarówno ciężaru dowodu, jak i siły, i mają warianty dotyczące wartości logicznych rozkładu. Połączenie dedukcji i inwersji daje dwie formy wnioskowania znane z literatury: indukcję i abdukcję. Indukcja i abdukcja to spekulatywne formy wnioskowania, z natury mniej pewne niż dedukcja, a odpowiadające im wzory to odzwierciedlają. Dynamika wartości prawdy Novamente w ramach PTL może być dość subtelna. W przeciwieństwie do systemu NARS i większości innych logicznych systemów wnioskowania (wyjątkiem są pętlowe sieci Bayesa), nie wykluczamy

wnioskowania kołowego; akceptujemy je. Wnioskowania kołowe występują niepoohamowanie, ostatecznie prowadząc do „stanu atraktora” wartości prawdy w całym systemie, w którym wszystkie wartości prawdy atomów są mniej więcej (choć niekoniecznie dokładnie) spójne ze sobą. Co ciekawe, chociaż PTL opiera się na logice formalnej, jego dynamika bardziej przypomina dynamikę sieci neuronowych atraktorowych. Specjalne dodatki do ram FOI dotyczą wiedzy hipotetycznej, subiektywnej i kontrfaktycznej, np. ze stwierdzeniami takimi jak

Joe wierzy, że Ziemia jest płaska.

Gdyby Teksas nie miał ropy, to...

Ważne jest, aby system był w stanie reprezentować te stwierdzenia

bez faktycznego uwierzenia, że „Ziemia jest płaska” lub „Teksas nie ma ropy”. Jest to możliwe dzięki konstrukcji HypotheticalLink i kilku prostym powiązanim regułom wnioskowania. Wnioskowanie wyższego rzędu dotyczy relacji takich jak:

$\text{ownerOf}(X, Y) \text{ IFF } \text{havingOf}(Y, X)$

W tym przykładzie użyliśmy tradycyjnej notacji logiki predykatów do przedstawienia antysymetrii relacji własności i posiadania, ale wewnątrz Novamente sprawy wyglądają nieco inaczej: nie ma żadnych zmiennych. Zamiast tego, podejście logiki kombinatorycznej jest używane do uzyskania reprezentacji złożonych relacji bez zmiennych, takich jak te, jako sieci PredicateNodes i SchemaNodes (w tym SchemaNodes ucieleśniających „elementarne kombinatory”). Używając na przykład kombinatory C, powyższa równoważność wygląda następująco:

$\text{EquivalenceLink ownerOf}(C \text{ possessionOf})$

Brak zmiennych oznacza, że reguły wnioskowania wyższego rzędu są zasadniczo takie same jak reguły wnioskowania pierwszego rzędu, ale istnieją pewne nowe zwroty akcji, takie jak logiczna unifikacja i reguły mieszania relacji pierwszego i wyższego rzędu. Szczegóły, dzięki którym te „zwroty akcji” są rozwiązywane, wynikają z integracyjnej natury systemu Novamente: na przykład logiczna unifikacja jest przeprowadzana za pomocą integracyjnego procesu uczenia się schematu/predykatu Novamente, obejmującego ewolucyjne i wzmacniające uczenie się, a także wnioskowanie. Logiczne wnioskowanie MindAgents działa za pośrednictwem wyboru opartego na ważności: to znaczy, gdy są aktywowane przez harmonogram, wybierają atomy do rozumowania z prawdopodobieństwem proporcjonalnym do ich ważności. Podstawowa kontrola wnioskowania jest zatem skutecznie delegowana do ImportanceUpdating MindAgent. Specjalna kontrola wnioskowania może być przeprowadzana za pomocą nauczonych lub zaprogramowanych schematów ucieleśnionych w SchemaInstanceNodes.

Nieliniowo-dynamiczna alokacja uwagi

Oprócz formalnej logiki, innym ważnym tematem w historii AI jest formalne modelowanie sieci neuronowych. Sieci neuronowe doskonale oddają holistyczne i dynamiczne aspekty inteligencji. Metody inspirowane sieciami neuronowymi są stosowane w Novamente w dwóch miejscach:

- w ImportanceUpdating MindAgent, który służy do kierowania uwagi systemu na różne atomy w sposób różnicowy;
- w HebbianLearning MindAgent, który modyfikuje wartości TruthValues logicznych powiązań zgodnie ze specjalnym mechanizmem kontroli wnioskowania, który luźno emuluje podstawową regułę uczenia się Hebba.

Jednak chociaż aktywność tych dwóch MindAgent jest luźno inspirowana sieciami neuronowymi, nie używamy algorytmów sieci neuronowych w Novamente. Jest to wybór dokonany ze względu na prostotę i wydajność. Zamiast sieci neuronowych używamy logiki terminów probabilistycznych w specjalnie kontrolowany sposób, który pozwala jej na przybliżone emulowanie interesującej dynamiki, którą można zaobserwować w sieciach neuronowych atraktorowych. Wierzymy, że prawdopodobnie możliwe byłoby osiągnięcie rodzaju precyzyjnego wnioskowania, jakie wykonuje PTL, przy użyciu metod opartych wyłącznie na sieciach neuronowych; i wykonaliśmy pewne wstępne prace w tym kierunku w 2002 r., opracowując eksperymentalne podejście do aktualizacji sieci neuronowych zwane „logiką Hebba”. Uważamy jednak, że byłoby to niedopuszczalnie nieefektywne podejście, biorąc pod uwagę realia implementacji komputera von Neumanna.

Aktualizacja ważności

Alokacja uwagi odnosi się do procesu, w którym system określa, ile czasu procesora powinien otrzymać każdy z jego atomów. Robi to ImportanceUpdating MindAgent, który dostosowuje AttentionValues węzłów i relacji, których dotyczy. Ważność jest określana przez specjalny wzór, funkcję Importance Updating, która łączy inne wielkości tworzące AttentionValue. Ta formuła opiera się na wnioskowaniu probabilistycznym i może być interpretowana jako specjalna „strategia kontroli wnioskowania”, która przeprowadza wnioskowanie w określonej kolejności dla każdego atomu w każdym cyklu. Wzór jest prosty, ale nieco subtelny i został uzyskany poprzez połączenie analizy matematycznej i praktycznych eksperymentów. Podstawowa idea wzoru wynika z następujących kryteriów:

1. W przypadku braku innych przyczyn ważność zanika.
2. LTI atomu jest obliczane jako wysokie, jeśli skumulowana niedawna użyteczność jest duża, i niskie w przeciwnym razie.
3. LTI jest interpretowane jako wartość spoczynkowa, dolna granica, przy której samo znaczenie ustala się w przypadku braku innych przyczyn. Zanika coraz wolniej, gdy zbliża się do tej wartości spoczynkowej.
4. Niedawny wzrost ważności blisko spokrewnionych atomów powoduje wzrost ważności. Niedawny spadek ważności blisko spokrewnionych atomów powoduje spadek ważności.
5. Ponadprzeciętna niedawna użyteczność powoduje wolniejszy spadek ważności.

Należy zauważyć, że między innymi te ramy obejmują model tego, co psychologowie nazywają pamięcią krótkotrwałą, pamięcią roboczą lub skupieniem uwagi. Skupienie uwagi (nasz preferowany termin) to zbiór bardzo ważnych atomów w danym momencie. Ważne atomy prawdopodobnie zostaną wybrane przez inne dynamiki do współpracy ze sobą, stąd istnieje tendencja do pozostawiania ważnymi poprzez budowanie połączeń między sobą i rozprzestrzenianie aktywacji między sobą wzdłuż tych połączeń. Jednak jeśli ważne atomy nie generują interesujących nowych relacji, ich niedawna użyteczność spadnie, a ich znaczenie zmniejszy się. Rezultatem netto tych dynamik jest wdrożenie „ruchomej bańki uwagi” stanowiącej zbiór atomów o wysokim znaczeniu. Aktualizację znaczenia można postrzegać, do pewnego stopnia, jako niepoznawczą część systemu, „tylko planowanie”. Jest to jednak bardzo wąski pogląd. Mapy formujące się w sieci za pośrednictwem nieliniowej dynamiki rozprzestrzeniania się aktywacji i aktualizacji ważności są bardzo ważne dla kierowania formowaniem węzłów, rozumowaniem, formowaniem skojarzeń i innymi procesami umysłowymi. Stanowią również główny węzeł przechowywania wiedzy.

Uczenie się schematów i predykatów

Być może najtrudniejszym aspektem AI Novamente jest to, co nazywamy „uczeniem się schematów i predykatów”. Dotyczy to tego, co powyżej nazwaliśmy „w pełni ogólną AI dla małych rozmiarów problemów”. Komponent procedury i uczenia się predykatów Novamente rozwiązuje problemy:

- biorąc pod uwagę opis pożądanego funkcjonalności, znajdź program komputerowy (schemat), który dostarcza tę funkcjonalność;
- biorąc pod uwagę zbiór informacji, znajdź w nim wzorce.

Rozwiązuje te problemy w sposób ogólny, ale istnieją znaczące problemy z wydajnością obliczeniową, co oznacza, że w praktyce metody te można stosować tylko na małą skalę. Większe problemy rozpoznawania wzorców i uczenia się schematów muszą zostać rozwiązane przy użyciu innych procesów poznawczych, aby dokonać podziału na zbiór mniejszych problemów, a następnie przy użyciu uczenia się schematów i predykatów w przypadku mniejszych problemów. Jest to oczywiście podejście heurystyczne znane z psychologii człowieka. Ludzie również nie rozwiązują dużych problemów na raz. Rozwiązują małe problemy i powoli je budują, stopniowo tworząc odpowiedzi na większe problemy. Ludzie rozwijają się, ucząc się „poznawczych bloków konstrukcyjnych”, przechowują je i stosują do różnych problemów na dużą skalę. Często używane bloki konstrukcyjne są przechowywane i często się do nich odwołują. Wartość ogólnego rozpoznawania wzorców w kontekście analizy danych jest oczywista i zostanie omówiona w późniejszych podsekcjach. Uczenie się predykatów, prawidłowo wdrożone, jest bardzo cennym narzędziem do analizy danych. Jest również, jak zobaczymy, istotną częścią rozumienia języka. Wartość uczenia się schematów może być mniej przejrzysta, ale nie mniej głęboka. Uczenie się schematów jest istotnym elementem filozofii „kontroli systemu” Novamente. Początkowo ogólne zachowanie Novamente będzie kierowane przez zakodowane przez człowieka heurystyczne schematy sterowania. Jednak system tak złożony jak Novamente nie może działać optymalnie na tej podstawie – wymagane jest adaptacyjne uczenie się schematów sterowania. W szczególności doświadczenie pokazuje, że złożone wnioski logiczne wymagają schematów sterowania dostosowanych do kontekstu. Literatura informatyczna zawiera szereg różnych podejść do uczenia schematów i predykatów, które można ogólnie podzielić na:

- algorytmy planowania oparte na logice;
- sieci neuronowe;
- programowanie ewolucyjne;
- uczenie przez wzmacnianie;
- podejścia hybrydowe;

Żadna z tych technik nie jest sama w sobie wystarczająca dla potrzeb Novamente, dlatego stworzyliśmy własne podejście algorytmiczne, integrując pomysły z kilku z tych istniejących podejść. Algorytmy planowania oparte na logice, takie jak GraphPlan i SATPlan, mają swoje mocne strony. Różne techniki planowania probabilistycznego z metodami Markowa okazały się stosunkowo skuteczne w robotyce. Ale ostatecznie są one skuteczne tylko w bardzo wąsko ograniczonych domenach – są „kruche”. Rekurencyjna propagacja wsteczna [62] przedstawia teoretycznie ogólne podejście oparte na sieciach neuronowych do uczenia się procedur, ale jego problemy z wydajnością są poważne. Bardziej wyspecjalizowane podejścia sieci neuronowych, takie jak sprytne algorytmy planowania oparte na sieciach neuronowych opracowane przez Jamesa Albusa i współpracowników do wykorzystania w ich zintegrowanej architekturze robotyki [3], wykazują większą wydajność, ale mają zastosowanie tylko w wąskich domenach zastosowań. Uczenie się przez wzmacnianie to podejście do uczenia się procedur oparte na „nienadzorowanym” uczeniu się Hebba w mózgu. Najbardziej znane jest to, że zostało

ucieleśnione w systemach klasyfikatorów Johna Hollanda . Chociaż jest to interesujące podejście koncepcyjne, uczenie się przez wzmacnianie ma poważne problemy z dostrajaniem parametrów i skalowalnością i rzadko było skutecznie stosowane w praktyce. Wreszcie programowanie ewolucyjne emuluje proces selekcji naturalnej w celu „ewolucji” procedur spełniających zadane kryteria. Jest to bardzo obiecujące podejście, ale podobnie jak uczenie się za pomocą sieci neuronowych ma problemy ze skalowalnością – uczenie się może być bardzo, bardzo powolne w przypadku dużych problemów. Podejście, które przyjęliśmy w Novamente, jest syntezą podejścia logicznego, uczenia się przez wzmacnianie i ewolucyjnego. Używamy uczenia się przez wzmacnianie (uczenie się Hebba za pośrednictwem HebbianLinks) i planowania logicznego (wnioskowanie wyższego rzędu PTL), ale traktujemy je jako techniki pomocnicze, a nie jako główne źródła mocy uczenia się schematów i predykatów. Nasze główne podejście do uczenia się schematów i predykatów polega na naprawianiu problemów skalowania programowania ewolucyjnego przy użyciu szeregu sztuczek zorientowanych na inteligencję integracyjną. Jedną z technik niedawno wprowadzonych w celu przyspieszenia programowania ewolucyjnego jest algorytm optymalizacji bayesowskiej. Wykorzystujemy ulepszoną wersję BOA, połączoną w unikalny sposób z naszą kombinatoryczną reprezentacją logiczną predykatów i schematów, i zmodyfikowaną w celu wykorzystania PTL jako algorytmu modelowania bayesowskiego (zapewniającego większą inteligencję niż algorytm modelowania oparty na drzewie decyzyjnym używany w standardowym BOA) i jako algorytmu alokacji uwagi w celu sterowania kierunkiem ewolucji. Krótko mówiąc, nasze podejście jest następujące:

- Przedstaw schematy/predykaty za pomocą skierowanych grafów acyklicznych, których węzły i łącza są typizowane, a węzły zawierają „elementarne funkcje” pochodzące z: arytmetyki, logiki boolowskiej i prob. oraz logiki kombinatorycznej. Zastosowanie funkcji logiki kombinatorycznej pozwala nam uzyskać pętle i rekurencję bez użycia zmiennych lub grafów cyklicznych.
- Zakoduj te programy jako „genotypy” za pomocą specjalnej metody kodowania.
- Przeszukaj przestrzeń programu za pomocą specjalnej odmiany algorytmu optymalizacji bayesowskiej (BOA), działającego na genotypach i używającego PTL (jako schematu modelowania i jako sterownika alokacji uwagi) do kierowania jego operacjami.

W chwili pisania tego tekstu jest to tylko częściowo zaimplementowane, a przy obecnym tempie postępu może minąć nawet rok, zanim zostanie w pełni zaimplementowane i przetestowane (choć ograniczone wersje zostały już przetestowane na różnych problemach matematycznych i będą szeroko testowane na rzeczywistych problemach rozpoznawania wzorców w ciągu najbliższych kilku miesięcy). Wierzmy, że będzie to praktyczne podejście, jeśli chodzi o zapewnienie dobrej funkcjonalności przeciętnego przypadku dla ogólnego problemu uczenia się schematów i predykatów o małym rozmiarze. Jak poradzimy sobie z uczeniem się większych schematów i predykatów? Tutaj zamierzamy powrócić do powszechnej strategii stosowanej przez mózgi i inne złożone systemy: hierarchicznego rozbicia. Novamente nie będzie w stanie nauczyć się ogólnych dużych schematów i predykatów, ale będzie w stanie nauczyć się dużych schematów i predykatów, które składają się z małych schematów lub predykatów, których wewnętrzne węzły odnoszą się do małych schematów lub predykatów, których wewnętrzne węzły odnoszą się do małych schematów lub predykatów itd. Zmodyfikowaliśmy algorytm BOA specjalnie, aby działał dobrze w przypadku hierarchicznych schematów lub predykatów. Choć może się wydawać, że nie jest to tak ogólne podejście, jakiego oczekuje się od „sztucznej inteligencji ogólnej”, uważamy, że tego rodzaju heurystyka hierarchicznego rozbicia jest sposobem działania ludzkiego umysłu/mózgu i jest zasadniczo nieunikniona w każdym praktycznym inteligentnym systemie ze względu na oczywistą nierozwiązywalność ogólnego schematu i problemu uczenia się predykatów.

Eksploracja wzorców

Teraz przejdziemy krótko do niektórych praktycznych, komercyjnych zastosowań obecnego, częściowo ukończonego systemu Novamente. Żaden proces nie leży bliżej sedna problemu „analizy ogromnych magazynów danych” niż eksploracja wzorców. Jest to zatem aspekt Novamente, nad którym zastanawialiśmy się szczególnie uważnie, w kontekście zarówno bioinformatyki, jak i zastosowań bezpieczeństwa narodowego. Nasza obecna praca bioinformatyczna z Novamente była znacząco sukcesywna w tym obszarze, znajdując nigdy wcześniej nie wykryte wzorce wzajemnej regulacji między genami poprzez analizę danych ekspresji genów w kontekście wiedzy o tle biologicznym, jak pokazano bardziej szczegółowo w Załączniku do tego rozdziału. Konwencjonalne programy do nienadzorowanej eksploracji wzorców obejmują:

- Analiza głównych składowych (PCA);
- algorytmy klastrowania;
- wykorzystanie algorytmów optymalizacji do przeszukiwania „przestrzeni wzorców”;
- algorytmy eksploracji reguł asocjacyjnych dla ogromnych zestawów danych (takich jak Apriori);
- rozpoznawanie wzorców oparte na sieciach neuronowych (SOMs [48], propagacja wsteczna).

Apriori to prosty, ale potężny algorytm, którego zamierzamy użyć lub jego pewnej odmiany w ramach procesu Fishera w ramach DINI, aby wychwycić potencjalnie użyteczne wzorce z całego magazynu danych i przedstawić je Novamente do dalszych badań. Apriori nie ma żadnego poczucia semantyki, po prostu znajduje kombinacje elementów danych, które występują razem z zaskakującą częstotliwością, stosując zachłanne podejście „wspinaczki po wzgórzach”; doskonale sprawdza się w roli początkowego filtra w magazynie danych, który jest zbyt duży, aby załadować go do Novamente na raz. Algorytmy klastrowania są powszechnie stosowane w wielu domenach; dzielą się na kilka różnych kategorii, w tym:

- algorytmy aglomeracyjne, ogólnie przydatne do budowania hierarchicznych systemów kategorii;
- algorytmy partycjonowania, dobre do dzielenia danych na kilka dużych kategorii (k-means, Bioclust [10] i inne);
- algorytmy statystyczne: maksymalizacja oczekiwania.

Klastrowanie ma tendencję do bycia problematyczną technologią, ponieważ algorytmy są trudne do dostrojenia i walidacji. Każdy algorytm wydaje się być najlepszy – jeśli spojrzysz na odpowiedni rodzaj zestawu danych. Novamente wykonuje aglomeracyjne klastrowanie niejawnie poprzez iteracyjne działanie ConceptFormation MindAgent. Istnieje również Clustering MindAgent, który wykonuje jawne klastrowanie oparte na partycjonowaniu, postrzegając atomy w systemie jako symetryczny ważony graf złożony z SimilarityLinks i partycjonując ten graf za pomocą wariantu algorytmu Bioclust. Bardziej wydajne, ale droższe niż inne techniki, uczenie się predykatów może być używane do przeszukiwania przestrzeni wzorców w danych pod kątem wzorców „dopasowania”. Tutaj „dopasowanie” jest definiowane jako kombinacja:

- zwartości lub „minimalnej długości opisu” ; i
- częstotliwości i przejrzystości występowania w obserwowanych danych.

Uczenie się predykatów oparte na programowaniu ewolucyjnym może znajdować wzorce znacznie subtelniejsze niż te wykrywalne za pomocą prostych metod, takich jak Apriori lub klastrowanie.

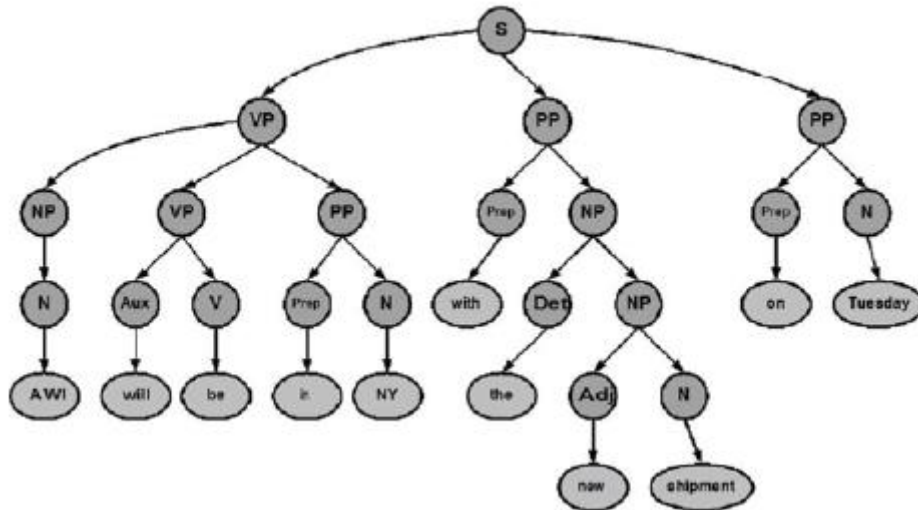
Logiczne wnioskowanie odgrywa tutaj również rolę, w tym sensie, że po znalezieniu wzorców za pomocą którejkolwiek z wyżej wymienionych metod, wnioskowanie jest w stanie dokonać wiarygodnych szacunków co do faktycznej ważności wzorców. Nie tylko obserwuje, jak widoczny jest wzorec w obserwowanych danych, ale wykorzystuje wnioskowanie do zbadania podobieństwa między wzorcem a innymi obserwowanymi wzorcami w innych zestawach danych, dokonując w ten sposób integracyjnego osądu ważności. Przykładowy wzorec znaleziony przez Novamente poprzez eksplorację danych ekspresji genów jest pokazany w Załączniku. W późniejszych wersjach Novamente wzorce te mogły być prezentowane użytkownikowi w formie werbalnej. Obecnie wzorce są prezentowane w języku formalnym, który następnie musi zostać zwizualizowany przez interfejs użytkownika lub werbalizowany przez doświadczonego użytkownika.

Przetwarzanie języka naturalnego

Novamente to ogólna architektura AGI, nieograniczona do żadnego konkretnego typu danych wejściowych. Omówiliśmy tutaj, w jaki sposób może ona wykorzystywać dane typu symbolicznego ładowane z baz danych RDB, ale w naszej obecnej pracy z systemem wykorzystujemy w równym stopniu jego zdolność do przetwarzania złożonych ilościowych zestawów danych (np. danych ekspresji genów pochodzących z mikromacierzy). Co najważniejsze dla praktycznej funkcjonalności analizy/zarządzania danymi, system jest również zdolny do przetwarzania informacji językowych. Niektóre informacje językowe mogą trafiać do systemu jako dźwięk, rejestrowany za pomocą magnetofonów, mikrofonów komputerowych i tym podobnych. Obsługa tego rodzaju danych wymaga pewnego specjalistycznego wstępnego przetwarzania danych ilościowych przy użyciu narzędzi, takich jak analiza falkowa. Nie będziemy tutaj omawiać tych kwestii, chociaż poświęcono wiele uwagi dostosowaniu Novamente do takich celów. Zamiast tego skupimy się na lingwistyce, zakładając, że system odbiera język w formie tekstowej, albo z plików komputerowych, albo z wyjścia programu oprogramowania do zamiany głosu na tekst. Istnieje wiele technologii przetwarzania języka naturalnego (NLP), ale faktem jest, że żadna z nich nie działa zbyt dobrze. Nowoczesna technologia NLP działa całkiem nieźle, jeśli spełnione jest jedno z dwóch kryteriów: zdania są bardzo proste lub wszystkie zdania dotyczą jednej, bardzo wąskiej dziedziny. Nie radzi sobie jeszcze z realistycznie złożonymi wypowiedziami na temat świata w ogólności. Lexiquest8, Connexor9 i inne firmy oferują narzędzia NLP, które zapewniają pragmatycznie użyteczną funkcjonalność w wąskich dziedzinach, ale nie ma niczego, co mogłoby na przykład wziąć e-mail pobrany losowo z Internetu i poprawnie przeanalizować 90% zdań w nim zawartych. Wierzymy, że architektura Novamente ma potencjał, aby zapewnić prawdziwy przełom w NLP. Wynika to z jej integracyjnej natury. Największym problemem w dzisiejszym NLP jest integracja wiedzy semantycznej i pragmatycznej z procesem analizy składniowej [53]. Novamente, które automatycznie integruje wszystkie dostępne formy informacji, jest idealnie przygotowane do tego celu. Konkretnie, jednym z wiodących podejść do analizy składniowej języka naturalnego jest „analiza struktury cech unifikacyjnych”. W tym podejściu analiza składniowa odbywa się poprzez proces unifikacji struktur danych reprezentujących zdania. Podejście to idealnie pasuje do LogicalUnification MindAgent firmy Novamente, którego projekt został specjalnie dostrojony do zastosowań językowych. Typowe parsery unifikacyjne działają poprzez wyspecjalizowane procesy „unifikacji językowej”, ale teoretycznie pokazaliśmy, że logiczna unifikacja firmy Novamente może przeprowadzać ten sam proces bardziej wydajnie. Oprócz unifikacji gramatyki struktur cech zazwyczaj obejmują niewielką liczbę specjalnych transformacji językowych (takich jak transformacje przekształcające stwierdzenia w pytania); można je wyrazić kompaktowo jako SchemaNodes firmy Novamente i wygodnie uczyć się i manipulować nimi w tej formie. Rozważmy na przykład zdanie

„AWI będzie w Nowym Jorku z nową dostawą we wtorek”.

Skuteczny parser przeanalizuje części mowy i relacje składniowe w tym zdaniu i wygeneruje drzewo składniowe. Drzewo składniowe takie jak to można znaleźć w Novamencie poprzez logiczną unifikację, wspomaganą przez inne MindAgents. Reprezentacja drzewa składniowego w kategoriach węzłów i łączy Novamencie jest prosta, chociaż obejmuje znacznie więcej węzłów i łączy niż proste wykonanie „drzewa składniowego” pokazane na rysunku



Po zakończeniu parsowania następuje naprawdę znaczący krok, którym jest „mapowanie semantyczne” – wykorzystanie zrozumienia składniowego do przeprowadzenia interpretacji semantycznej. Schematy mapowania semantycznego – niektóre dostarczone przez programistów Novamencie, większość poznana dzięki doświadczeniu w systemie – są używane do mapowania atomów reprezentujących drzewo składniowe na atomy reprezentujące znaczenie systemu, takie jak:

atTime(Location(Z, Nowy Jork) AND with(Z, Y), 11/02/2003)

representative_of(Z,X)

name_of(X, "AWI")

name_of(X, "Associated Widgets Incorporated")

EIN(X, "987654321")

InheritanceLink(Y, shipping)

ContextLink(InheritanceLink (Y, new), 12/02/2003)

Proces generowania języka, używany przez Novamencie do generowania języka reprezentującego wzorce, które znalazł lub pomysły, które miał, jest odwrotnością tego: zaczyna się od zbioru atomów reprezentujących znaczenia semantyczne, używa specjalnych schematów do wytworzenia zbioru atomów reprezentujących drzewo składniowe zdania, a następnie generuje zdanie. Obecnie pracujemy nad prototypem tego rodzaju procesu parsowania, który działa poza rdzeniem Novamencie; a nasze eksperymenty były bardzo udane, w tym sensie, że jesteśmy w stanie pomyślnie tworzyć dokładne semantyczne struktury węzłów i łączy ze wszystkich stosunkowo prostych zdań angielskich i pewnego procenta zdań złożonych. Jest to bardzo zachęcające pod względem wyniku, który będzie

możliwy do osiągnięcia, gdy pomysły w prototypie zostaną zintegrowane z rdzeniem Novamente, zgodnie z projektem systemu Novamente

Wnioski

Obecnie istnieje niewiele projektów AI mających na celu ogólną inteligencję, a większość z nich jest przedstawiona w tym tomie. Uważamy, że spośród tego wybranego zestawu projekt Novamente jest tym, który najbardziej rozwinął się w pragmatycznych aspektach projektowania i wdrażania oprogramowania. Projekt Novamente posiada konkretny i szczegółowy projekt matematyczny, koncepcyjny i programistyczny, który zapewnia ujednolicone traktowanie wszystkich głównych aspektów inteligencji, jak szczegółowo opisano w naukach kognitywnych i informatyce. Nasze wcześniejsze doświadczenie z inżynierią AI zorientowaną na działalność komercyjną i badania i rozwój nauczyło nas niezliczonych lekcji, z których jedną jest to, aby nigdy nie lekceważyć złożoności i trudności, jakie stwarzają fazy „tylko inżynierii” projektu. Inna ważna lekcja dotyczy potrzeby uczenia się przez doświadczenie jako sposobu na szkolenie i nauczanie proto-systemu AGI. Chociaż Novamente jest obecnie częściowo inżynierowany, wyniki, które uzyskaliśmy do tej pory z systemem (patrz np. Dodatek do tego rozdziału), są bardziej zgodne z linią eksploracji danych niż ambitnej AGI. Jednakże te praktyczne zastosowania dostarczają nam bezcennego wglądu w praktyczne kwestie otaczające zastosowania AGI i proto-AGI systemów do rzeczywistych problemów. Biorąc pod uwagę ponurą historię wielkich wizji AI, zdajemy sobie sprawę z potrzeby ostrożności w formułowaniu wielkich roszczeń dotyczących jeszcze niezademonstrowanych zdolności naszego systemu. Jednak nie uważamy historii za dobry powód do konserwatyzmu co do przyszłości. Wierzymy, że czerpiąc z najlepszych spostrzeżeń istniejących paradygmatów AI i integrując je w ramach nowych syntetycznych ram opartych na samoorganizacji i doświadczalnym interaktywnym uczeniu się, mamy poważną szansę na przeniesienie AI na nowy poziom i zbliżenie się do prawdziwej Sztucznej Inteligencji Ogólnej. I mamy precyzyjny projekt i plan działania w celu przetestowania naszych hipotez w tym zakresie.