

## Ontologia dla dużych zbiorów danych

W Sekcji wprowadzającej dowiedzieliśmy się, że duże zbiory danych przyczyniły się do szybkiego postępu w dziedzinie sztucznej inteligencji. Wynika to przede wszystkim z dostępności bardzo dużych zestawów danych z heterogenicznych źródeł i wykładniczego wzrostu mocy obliczeniowej z powodu przetwarzania rozproszonego. Niezwykle trudno jest czerpać wartość z dużych ilości danych, jeśli nie ma standaryzacji ani wspólnego języka do tłumaczenia danych na informacje i przekształcania informacji w wiedzę. Na przykład dwie osoby, które mówią dwoma różnymi językami i nie rozumiejąc się nawzajem, nie mogą wejść w rozmowę słowną, chyba że istnieje między nimi jakiś mechanizm tłumaczenia. Tłumaczenia i interpretacje są możliwe tylko wtedy, gdy słowo kluczowe ma znaczenie semantyczne i gdy reguły gramatyczne są stosowane jako spójniki. Jako przykład podajemy zdanie w języku polskim i hiszpańskim:

Jan je trzy banany każdego dnia

Jan come tres platanos todos los dias

Zasadniczo możemy rozbić zdanie w postaci przedmiotów, przedmiotów, czasowników i atrybutów. W tym przypadku Jan i banany są podmiotami. Łączy je aktywność, w tym przypadku jedzenie, a także atrybuty i dane kontekstowe w połączeniu z tematami i czynnościami. Tłumacze wiedzy można wdrożyć na dwa sposoby:

\* All-inclusive mapping: Utrzymanie mapowania między wszystkimi zdaniami w jednym języku a tłumaczeniami w innym języku. Jak możesz sobie wyobrazić, nie jest to możliwe do osiągnięcia, ponieważ istnieje niezliczona ilość sposobów na wyrażenie (obiekt, zdarzenie, atrybuty, kontekst) w języku.

\* Semantyczne spojrzenie na świat: jeśli skojarzymy znaczenie semantyczne z każdym bytem, który napotykamy w wyrażeniu językowym, znormalizowany semantyczny widok świata może działać jako scentralizowany słownik dla wszystkich języków.

Semantyczny i znormalizowany obraz świata jest niezbędny, jeśli chcemy wprowadzić sztuczną inteligencję, która zasadniczo czerpie wiedzę z danych i wykorzystuje wiedzę kontekstową do wglądu i znaczących działań w celu zwiększenia ludzkich możliwości. Ten semantyczny pogląd na świat jest wyrażony jako ontologie. W naszym kontekście Ontologia jest zdefiniowana jako: **zestaw pojęć i kategorii w obszarze tematycznym lub dziedzinie, pokazujący ich właściwości i relacje między nimi**. Tu przyjrzymy się następującym:

- \* Jak ludzki mózg łączy obiekty w swojej interpretacji świata
- \* Rola, jaką odgrywa Ontologia w świecie Big Data
- \* Cele i wyzwania związane z Ontologią w Big Data
- \* Struktura opisu zasobów
- \* Język Ontologii Sieciowej
- \* SPARQL, semantyczny język zapytań dla RDF
- \* Budowanie ontologii i używanie ontologii do budowania inteligentnych maszyn
- \* Nauka ontologii

## Ludzki mózg i ontologia

Chociaż postępy w naszym rozumieniu tego, jak funkcjonuje ludzki mózg, mechanizm jego przechowywania i przetwarzania jest daleki od pełnego zrozumienia, otrzymujemy setki i tysiące informacji sensorycznych w ciągu dnia, a jeśli przetwarzamy i przechowujemy każdą część tych informacji, ludzki mózg zostanie przytłoczony i nie będzie w stanie zrozumieć kontekstu i zareagować w znaczący sposób. Mózg ludzki stosuje filtry do odbieranych sensorycznych sygnałów wejściowych. Rozumie się, że istnieją trzy przedziały ludzkiej pamięci:

\* Pamięć sensoryczna: jest to pamięć pierwszego poziomu, a większość informacji jest usuwana w ciągu milisekund. Rozważmy na przykład, kiedy prowadzimy samochód. Po drodze napotykamy tysiące przedmiotów i dźwięków, a większość tych danych jest wykorzystywana do funkcji prowadzenia pojazdu. Poza ramami odniesienia w czasie większość danych jest zapomniana i nigdy nie jest przechowywana w pamięci.

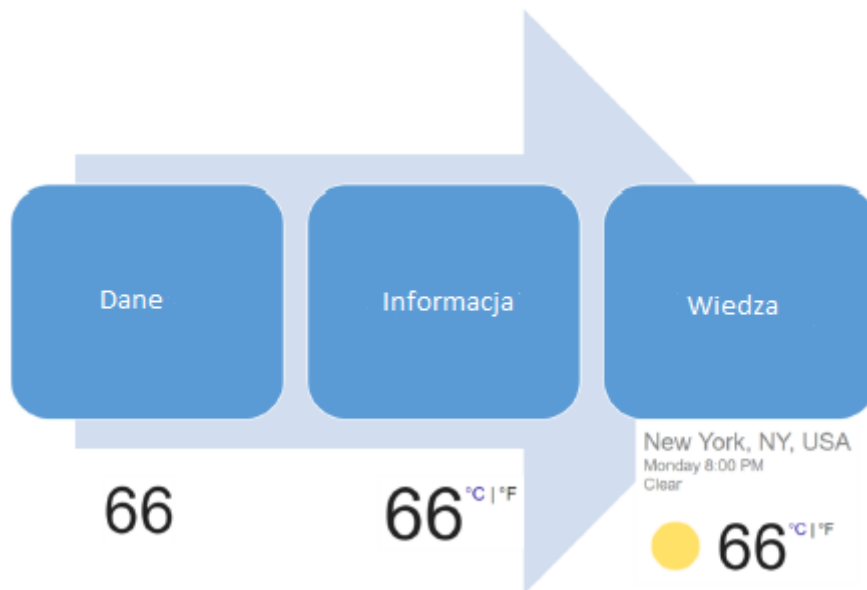
\* Pamięć krótkotrwała: jest wykorzystywana do informacji niezbędnych do realizacji tymczasowego celu. Rozważmy na przykład, że otrzymujesz telefon od współpracownika, aby przypomnieć ci o pilnym spotkaniu w pokoju D-1482. Kiedy zaczynasz chodzić od biurka do pokoju, liczba jest znacząca, a ludzki mózg przechowuje informacje w pamięci krótkotrwałej. Ta informacja może lub nie być przechowywana poza czasem kontekstu. Wspomnienia te mogą potencjalnie zostać przekształcone w pamięć długoterminową, jeśli zostaną napotkane w ekstremalnych sytuacjach.

\* Pamięć długoterminowa: jest to pamięć, która będzie trwać przez kilka dni lub całe życie. Na przykład pamiętamy nasze imię, datę urodzenia, krewnych, lokalizację domu i wiele innych rzeczy. Pamięć długoterminowa działa na podstawie wzorców i powiązań między obiektami. Umiejętności nie-przetrwania, których uczymy się i opanowujemy przez pewien czas, na przykład gra na instrumencie muzycznym, wymagają przechowywania wzorców połączeń i koordynacji odruchów w pamięci długoterminowej

Niezależnie od podziału pamięci informacje są przechowywane w postaci wzorców i łączone w ludzkim mózgu. W grze pamięciowej, która wymaga od graczy chwilowego spojrzenia na grupę 50 dziwnych obiektów i zapisania ich na papierze, gracz, który zapisuje najwięcej nazw obiektów, wygrywa. Jedną z sztuczek w tej grze jest ustanowienie powiązań między dwoma obiektami i stworzenie fabuły. Gracze, którzy próbują samodzielnie zapamiętać obiekty, nie mogą wygrać z graczami, którzy utworzą połączoną listę w swoim umyśle. Kiedy mózg otrzymuje dane z narządów zmysłów, a informacje muszą być przechowywane w pamięci długoterminowej, są one przechowywane w postaci wzorców i łączone z powiązаныmi obiektami lub istotami, tworząc mapy myśli. Gdy widzimy osobę naszymi oczami, mózg tworzy mapę obrazu i pobiera wszystkie informacje kontekstowe dotyczące tej osoby. To stanowi podstawę ontologii informatyki.

### **Ontologia informatyki**

Formalnie, ontologia nauk informatycznych jest zdefiniowana jako: **Formalne nazewnictwo i definicja typów, właściwości i wzajemnych powiązań bytów, które zasadniczo istnieją dla określonej dziedziny.** Istnieje zasadnicza różnica między ludźmi a komputerami, jeśli chodzi o obsługę informacji. W przypadku komputerów informacje są dostępne w postaci ciągów, natomiast w przypadku ludzi informacje są dostępne w postaci rzeczy. Rozumiem różnicę między ciągami znaków a rzeczami. Gdy dodamy metadane do łańcucha, staje się to rzeczą. Metadane to dane o danych (w tym przypadku ciąg) lub informacje kontekstowe o danych. Chodzi o przekształcenie danych w wiedzę. Poniższa ilustracja pokazuje nam, jak przekonwertować dane na wiedzę:



Tekst lub liczba 66 to Dane; samo w sobie 66 nie ma żadnego znaczenia. Kiedy mówimy, że 66° F, 66 staje się miarą temperatury i w tym momencie reprezentuje pewne Informacje. Kiedy mówimy 66° F w Nowym Jorku 3 października 2017 o 20:00, staje się Wiedzą. Gdy informacje kontekstowe są dodawane do danych i informacji, stają się wiedzą, umiejętnością.

W dążeniu do czerpania wiedzy z danych i informacji Ontologie odgrywają główną rolę w standaryzacji światopoglądu poprzez precyzyjnie zdefiniowane terminy, które mogą być przekazywane między ludźmi a aplikacjami. Umożliwiają wspólne rozumienie obiektów i ich relacji w ramach domen i między nimi. Zazwyczaj występują różnice schematyczne, strukturalne i semantyczne, a zatem powstaje konflikt między reprezentacjami wiedzy. Dobrze zdefiniowane i zarządzane Ontologie wypełniają luki między reprezentacjami.

### **Właściwości ontologiczne**

Na wysokim poziomie Ontologie powinny mieć następujące właściwości, aby stworzyć spójny obraz wszechświata zasobów danych, informacji i wiedzy:

- \* Ontologie powinny być kompletne, aby uwzględnić wszystkie aspekty bytów.
- \* Ontologie powinny być jednoznaczne, aby uniknąć błędnej interpretacji przez ludzi i aplikacje.
- \* Ontologie powinny być zgodne ze znajomością dziedziny, do której się odnoszą. Na przykład Ontologie dla nauk medycznych powinny być zgodne z formalnie ustalonymi terminologiami i relacjami w naukach medycznych.
- \* Ontologie powinny być ogólne, aby mogły być ponownie użyte w różnych kontekstach. Ontologie powinny być rozszerzalne w celu dodania nowych pojęć i ułatwienia przestrzegania nowych pojęć, które pojawiają się wraz z rosnącą wiedzą w tej dziedzinie.
- \* Ontologie powinny być odczytywalne maszynowo i interoperacyjne.

Najważniejszą zaletą reprezentacji ontologicznej dla pojęć i podmiotów w świecie rzeczywistym jest to, że ułatwia badanie pojęć niezależnie od języka programowania, platform i protokołów komunikacyjnych. Umożliwia to luźne sprzężenie, a jednocześnie ścisłą integrację między koncepcjami, co umożliwia procesowi tworzenia oprogramowania ponowne wykorzystanie oprogramowania i bazy wiedzy jako koncepcji modułowych.

## Zalety ontologii

- \* Podwyższona jakość analizy encji
- \* Zwiększone wykorzystanie, ponowne użycie i łatwość konserwacji systemów informatycznych
- \* Ułatwienie dzielenia się wiedzą domenową, ze wspólnym słownictwem w niezależnych aplikacjach

Ci, którzy są zaznajomieni z obiektowym paradygmatem programowania lub projektowaniem baz danych, mogą łatwo powiązać ontologiczną reprezentację jednostek domeny z klasami lub schematami baz danych. Klasy są ogólnymi reprezentacjami bytów, które zawierają właściwości i zachowania. Jedna klasa może odziedziczyć zachowanie i właściwości z innej klasy (jest to relacja). Na przykład kot jest zwierzęciem. W tym przypadku Animal to abstrakcyjna nadklasa Cat. Klasa Cat dziedziczy właściwości z klasy Animal i dodaje / zastępuje niektóre atrybuty i zachowania specyficzne dla kota.

Ten paradygmat ma zastosowanie w Ontologiach. Podobnie relacyjne bazy danych mają schematyczne reprezentacje jednostek domeny w organizacji. Istnieją pewne podstawowe różnice między bazami danych i ontologiami, takie jak:

- \* Ontologie są semantycznie bogatsze niż pojęcia reprezentowane przez bazy danych
- \* Reprezentacja informacji w ontologii oparta jest na częściowo ustrukturyzowanym, naturalnym języku tekstu i nie jest reprezentowana w formie tabelarycznej
- \* Podstawową przesłanką reprezentacji ontologicznej jest globalnie spójna terminologia stosowana do wymiany informacji między domenami i granicami organizacyjnymi
- \* Bardziej niż definiowanie ograniczonego kontenera danych, Ontologie skupiają się na ogólnej reprezentacji wiedzy w dziedzinie

## Składniki ontologii

Pojęcia: są to ogólne rzeczy lub podmioty podobne do klas w programowaniu obiektowym, na przykład osoba, pracownik i tak dalej.

Sloty: są to właściwości lub atrybuty bytów, na przykład płeć, data urodzenia, lokalizacja i tak dalej.

Relacje: reprezentują interakcje między pojęciami lub relacje typu „a-a, has-a”, na przykład pracownik jest osobą.

Aksjomaty: są to stwierdzenia, które są zawsze prawdziwe w odniesieniu do pojęć, czasów i relacji, na przykład osoba jest pracownikiem, jeśli jest zatrudniona przez pracodawcę.

Instancje: są to obiekty klasy w kategoriach obiektowych. Na przykład John jest instancją klasy Employee. Jest to specyficzna reprezentacja koncepcji. Ontologia wraz z instancjami w pełni reprezentuje wiedzę.

Operacje: są to funkcje i reguły rządzące różnymi elementami Ontologii. W kontekście obiektowym reprezentują one metody klasy.

Rozwój Ontologii rozpoczyna się od zdefiniowania klas w Ontologii. Klasy te reprezentują byty świata rzeczywistego. Po jednoznacznym zidentyfikowaniu i zdefiniowaniu jednostek są one ułożone w hierarchii taksonomicznej. Po zdefiniowaniu hierarchii Sloty i Relacje są zdefiniowane. Podanie wartości dla miejsc i wystąpień kończy rozwój ontologii specyficznej dla domeny.

## Rola jaką Ontologia odgrywa w Big Data

Jak widzieliśmy w Sekcji wprowadzającej, wolumeny danych rosną w fenomenalnym tempie i aby uzyskać wartość z danych, niemożliwe jest modelowanie całych danych w tradycyjny sposób: Ekstrakcja, Transformacja i Ładowanie (ETL). Tradycyjnie źródła danych generują zestawy danych w formatach ustrukturyzowanych i nieustrukturyzowanych. Aby przechowywać te zasoby danych, musimy ręcznie modelować dane w oparciu o różne podmioty. Biorąc przykład osoby jako bytu w świecie relacyjnych baz danych, musimy utworzyć tabelę reprezentującą osobę. Ta tabela jest powiązana z różnymi podmiotami powiązanymi kluczem obcym. Jednak podmioty te są predefiniowane i mają stałą strukturę. Modelowanie jednostek wymaga ręcznego wysiłku i trudno je modyfikować. W świecie dużych zbiorów danych schemat jest definiowany w czasie odczytu zamiast w czasie zapisu. Daje nam to większą elastyczność dzięki strukturze encji i modelowaniu danych. Nawet przy elastyczności i rozszerzalnych możliwościach modelowania zarządzanie zasobami danych na skalę internetową jest bardzo trudne, jeśli podmioty nie są znormalizowane w różnych domenach. Aby ułatwić wyszukiwanie w sieci, Google wprowadził wykres wiedzy, który zmienił wyszukiwanie ze statystyk słów kluczowych opartych na reprezentacji na modelowanie wiedzy. Było to wprowadzenie paradygmatu wyszukiwania według rzeczy, a nie łańcuchów. Wykres wiedzy jest bardzo dużą ontologią, która formalnie opisuje obiekty w świecie rzeczywistym. Wraz ze wzrostem zasobów danych generowanych z heterogenicznych źródeł w przyspieszającym tempie, stale zmierzamy w kierunku większej złożoności. Paradygmat Big Data opisuje duże i złożone zestawy danych, których nie można zarządzać w tradycyjnych aplikacjach. Potrzebujemy przynajmniej sposobu na uniknięcie fałszywych interpretacji złożonych jednostek danych. Ramy integracji i przetwarzania danych można ewentualnie ulepszyć metodami z zakresu technologii semantycznej. Używając rzeczy zamiast tekstu, możemy poprawić systemy informacyjne i ich interoperacyjność, identyfikując kontekst, w którym one istnieją. Ontologie zapewniają semantyczne bogactwo wiedzy specyficznej dla dziedziny i jej reprezentacji. W przypadku zasobów dużych zbiorów danych konieczne jest ograniczenie ręcznego modelowania danych w informacje i wiedzę. Jest to możliwe, jeśli uda nam się stworzyć sposób na znalezienie korespondencji między surowymi jednostkami, wyprowadzenie ogólnego schematu z reprezentacją taksonomiczną i odwzorowanie pojęć na tematy w określonych domenach wiedzy z podobieństwami terminologicznymi i odwzorowaniami strukturalnymi. Wdrożenie to ułatwi automatyczne wsparcie zarządzania zasobami dużych zbiorów danych oraz integrację różnych źródeł danych, co spowoduje mniej błędów i przyspieszenie pozyskiwania wiedzy.

### **Dostosowanie ontologii**

Wyrównanie lub dopasowanie ontologii jest procesem określania mapowania jeden na jeden między jednostkami z heterogenicznych źródeł. Korzystając z tego odwzorowania, możemy wywnioskować typy jednostek i wyodrębnić znaczenie z surowych źródeł danych w spójny i semantyczny sposób

#### Cele ontologii w Big Data

- \* Dziel się powszechnym zrozumieniem struktur informacyjnych w różnych aplikacjach
- \* Spraw, aby ETL był szybszy, łatwiejszy i dokładniejszy
- \* Wyeliminuj potrzebę niestandardowych, specyficznych dla sytuacji potoków ETL
- \* Automatyczne włączanie nowych źródeł danych
- \* Ulepsz wydobywanie informacji z tekstu i przekształcaj je w zasoby wiedzy
- \* Wzbogacaj istniejące dane o informacje strukturalne i semantyczne
- \* Przełóż wiedzę biznesową na oprogramowanie do użytku maszynowego

\* Zbuduj raz, użyj wiele razy

### **Wyzwania związane z ontologią w Big Data**

\* Generowanie bytów (konwersja ciągów znaków na rzeczy)

\* Zarządzanie relacjami

\* Obsługa kontekstu

\* Wydajność zapytania

\* Jakość danych

### **RDF - uniwersalny format danych**

Na tle Ontologii i ich znaczenia w świecie dużych zbiorów danych przyjrzyjmy się uniwersalnemu formatowi danych, który definiuje schematyczne reprezentacje Ontologii. Jednym z najbardziej popularnych i popularnych frameworków jest Resource Description Framework (RDF). RDF jest rekomendacją W3C od 2004 roku. RDF zapewnia strukturę do opisywania zidentyfikowanych rzeczy, podmiotów lub koncepcji zaprojektowanych do czytania i interpretacji przez komputery. Istnieje krytyczna potrzeba jednoznacznej identyfikacji jednostki lub koncepcji w sposób uniwersalny. Jednym z najpopularniejszych sposobów w dziedzinie informatyki jest użycie uniwersalnych identyfikatorów zasobów (URI). Znane adresy witryn internetowych reprezentowane jako uniwersalne lokalizatory zasobów (URL). Mapują one na unikalny adres IP, a tym samym domenę internetową w Internecie. Identyfikator URI jest bardzo podobny do adresu URL, z tą różnicą, że identyfikatory URI mogą reprezentować rzeczywistą domenę internetową. Biorąc pod uwagę to rozróżnienie, identyfikatory URI reprezentujące rzeczywiste obiekty muszą być jednoznaczne. Każdy identyfikator URI powinien być wyłączny dla zasobu sieciowego lub obiektu świata rzeczywistego i nigdy nie powinien być używany do reprezentowania obu jednocześnie, aby uniknąć zamieszania i dwuznaczności.

Podczas definiowania RDF należy wziąć pod uwagę następujące kwestie:

\* Zdefiniuj prosty model danych

\* Zdefiniuj formalną semantykę

\* Używaj rozszerzalnego słownictwa opartego na URI

\* Najlepiej użyć składni opartej na XML

Podstawowym elementem składowym RDF jest trójka, która składa się z podmiotu, predykatu i obiektu.

Spójrzmy na przykład bazy danych książek i przedstawmy ją za pomocą RDF XML:

**Nazwa książki : Autor: Firma : Rok**

Hit Refresh : Satya Nadella : Microsoft : 2017

Shoe Dog : Phil Knight : Nike : 2016

```

<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:book="http://www.artificial-intelligence.big-data/book#">
  <rdf:Description
rdf:about="http://www.artificial-intelligence.big-data/book/Hit-Refresh">
    <book:author>Satya Nadella</book:author>
    <book:company>Microsoft</book:company>
    <book:year>2017</book:year>
  </rdf:Description>
  <rdf:Description
rdf:about="http://www.artificial-intelligence.big-data/book/Shoe-Dog">
    <book:author>Phil Knight</book:author>
    <book:company>Nike</book:company>
    <book:year>2016</book:year>
  </rdf:Description>
  .
  .
  .
</rdf:RDF>

```

Pierwszy wiersz dokumentu RDF to deklaracja XML. Po deklaracji XML następuje element główny dokumentów RDF, <rdf: RDF>.

Przestrzeń nazw xmlns: rdf określa, że elementy z przedrostkiem rdf pochodzą z przestrzeni nazw <http://www.w3.org/1999/02/22-rdf-syntax-ns#>. Przestrzenie nazw XML służą do udostępniania elementów i atrybutów o unikatowych nazwach w dokumencie XML.

Przestrzeń nazw xmlns: book określa, że elementy z prefiksem book pochodzą z przestrzeni nazw - <http://www.artificial-intelligence.big-data/book#>. Element <rdf: Description> zawiera opis zasobu identyfikowanego przez atrybut rdf: about. Elementy <book: author>, <book: company>, <book: year> itd. Są właściwościami zasobu. W3C zapewnia usługę walidacji online (<https://www.w3.org/RDF/Validator/>), która sprawdza poprawność RDF pod względem jego składni i generuje tabelaryczne i graficzne widoki

## Dokument RDF

### Pojemniki RDF

Kontenery RDF służą do opisywania grup rzeczy. Oto przykład:

```

<rdf:Description
rdf:about="http://www.artificial-intelligence.big-data/book/Hit-Refresh">
  <book:author>Satya Nadella</book:author>
  <book:company>Microsoft</book:company>
  <book:year>2017</book:year>
  <book:chapters>
    <rdf:Bag>
      <rdf:li>1. From Hyderabad to Redmond</rdf:li>
      <rdf:li>2. Learning to Lead</rdf:li>
      <rdf:li>3. New Mission, New Momentum</rdf:li>
      ..
    </rdf:Bag>
  </book:chapters>
</rdf:Description>

```

Element <rdf: Bag> służy do opisanego listy wartości, które nie muszą znajdować się w określonej kolejności.

<rdf: Seq> jest podobny do <rdf: Bag>. Jednak elementy reprezentują uporządkowaną listę.

<rdf: Alt> służy do przedstawienia listy alternatywnych wartości dla elementu.

## Klasy RDF

Klasy RDF są wymienione na następujących obrazach:

Element	Class of	Subclass of
rdfs:Class	All classes	
rdfs:Datatype	Data types	Class
rdfs:Resource	All resources	Class
rdfs:Container	Containers	Resource
rdfs:Literal	Literal values (text and numbers)	Resource
rdf:List	Lists	Resource
rdf:Property	Properties	Resource
rdf:Statement	Statements	Resource
rdf:Alt	Containers of alternatives	Container
rdf:Bag	Unordered containers	Container
rdf:Seq	Ordered containers	Container
rdfs:ContainerMembershipProperty	Container membership properties	Property
rdf:XMLLiteral	XML literal values	Literal

## Właściwości RDF

Właściwości RDF są wymienione w następujący sposób:

Element	Domain	Range	Description
rdfs:domain	Property	Class	The domain of the resource
rdfs:range	Property	Class	The range of the resource
rdfs:subPropertyOf	Property	Property	The property is a sub property of a property
rdfs:subClassOf	Class	Class	The resource is a subclass of a class
rdfs:comment	Resource	Literal	The human readable description of the resource
rdfs:label	Resource	Literal	The human readable label (name) of the resource
rdfs:isDefinedBy	Resource	Resource	The definition of the resource
rdfs:seeAlso	Resource	Resource	The additional information about the resource
rdfs:member	Resource	Resource	The member of the resource
rdf:first	List	Resource	
rdf:rest	List	List	
rdf:subject	Statement	Resource	The subject of the resource in an RDF Statement
rdf:predicate	Statement	Resource	The predicate of the resource in an RDF Statement
rdf:object	Statement	Resource	The object of the resource in an RDF Statement
rdf:value	Resource	Resource	The property used for values
rdf:type	Resource	Class	The resource is an instance of a class

## Atrybuty RDF

Różne atrybuty RDF są wymienione w następujący sposób:

Attribute	Description
rdf:about	Defines the resource being described
rdf:Description	Container for the description of a resource
rdf:resource	Defines a resource to identify a property
rdf:datatype	Defines the data type of an element
rdf:ID	Defines the ID of an element
rdf:li	Defines a list
rdf:_n	Defines a node
rdf:nodeID	Defines the ID of an element node
rdf:parseType	Defines how an element should be parsed
rdf:RDF	The root of an RDF document
xml:base	Defines the XML base
xml:lang	Defines the language of the element content

## Korzystanie z OWL, języka ontologii sieciowej



Podczas gdy RDF i odpowiednie definicje schematów (RDFS) zapewniają strukturę semantycznego widoku zasobów informacyjnych, istnieją pewne ograniczenia dotyczące RDFS. RDFS nie może opisać jednostek wystarczająco szczegółowo. Nie ma możliwości zdefiniowania zlokalizowanych zakresów dla atrybutów encji, a ograniczenia specyficzne dla domeny nie mogą być wyraźnie wyrażone. Istnienie lub brak istnienia powiązanej jednostki, wraz z ograniczeniami liczebności (jeden do jednego, jeden do wielu itd.), Nie mogą być reprezentowane przez RDFS. Trudno jest przedstawić relacje przechodnie, odwrotne i symetryczne. Jednym z ważnych aspektów relacji między bytami w świecie rzeczywistym jest logiczne rozumowanie i wnioskowanie, bez wyraźnej wzmianki o związku. RDFS nie może zapewnić obsługi wnioskowania dla powiązanych podmiotów. Język ontologii sieciowej (OWL) rozszerza się i opiera na RDF / RDFS. OWL to rodzina języków reprezentacji wiedzy do tworzenia ontologii. W rzeczywistości OWL nie jest prawdziwym akronimem. Język zaczął się jako WOL. Jednak grupie roboczej nie podobał się skrót WOL. Opierając się na rozmowach w grupie roboczej, OWL miał tylko jedną oczywistą wymowę, łatwą do ucha, i otworzył wielkie możliwości dla logo - sowy kojarzą się z mądrością!

Aby zbudować inteligentne systemy, które mogą komunikować się w różnych domenach, należy pokonać ograniczenia RDFS i wyposażyć maszyny w dostęp do uporządkowanych zbiorów zasobów wiedzy i zestawów reguł wnioskowania, które można wykorzystać do automatycznego wnioskowania. OWL zapewnia formalną semantykę reprezentacji wiedzy i próbuje precyzyjnie opisać znaczenie bytów oraz ich relacji i rozumowania. Istnieją trzy gatunki OWL:

\* OWL DL: Służy do obsługi logiki opisu. Zapewnia to maksymalną ekspresję i możliwości logicznego wnioskowania. Charakteryzuje się to:

- Dobrze zdefiniowana semantyka
- Dobrze zrozumiane właściwości formalne podmiotów
- Łatwość implementacji znanych algorytmów wnioskowania

\* OWL Full: Opiera się na semantyce kompatybilnej z RDFS. Uzupelnia predefiniowane słownictwo RDF i OWL. Jednak dzięki OWL Full oprogramowanie nie może całkowicie rozumować i wnioskować.

\* OWL Lite: Służy do wyrażania taksonomii i prostych ograniczeń, takich jak liczebność od zera do jednego.

OWL reprezentuje byty jako klasy. Na przykład, zdefiniujmy obiekt PlayGround za pomocą OWL:

```
<owl: Class rdf: ID = "PlayGround">
```

Teraz zdefiniuj FootballGround i stwierz, że FootballGround jest rodzajem PlayGround:

```
<owl: Class rdf: ID = "FootballGround">
```

```
<rdf: subclassOf rdf: resource = "# PlayGround" />
```

```
</ owl: Class>
```

OWL zapewnia kilka innych mechanizmów definiowania klas:

- \* ekwiwalent Klasy: Reprezentuje, że dwie klasy (w Ontologiach i domenach) są synonimami.
- \* disjointWith: reprezentuje, że instancja klasy nie może być instancją innej klasy. Na przykład FootballGround i HockyGround są określone jako klasy rozłączone.
- \* Kombinacje boolowskie:

- unionOf: reprezentuje, że klasa zawiera rzeczy, które pochodzą z więcej niż jednej klasy
- intersectionOf: reprezentuje, że klasa zawiera rzeczy, które są zarówno w jednym, jak i drugim
- complementOf: reprezentuje, że klasa zawiera rzeczy, które nie są innymi rzeczami

## Język zapytań SPARQL

Dzięki ogólnemu zrozumieniu ontologii, RDF i OWL jesteśmy w stanie zasadniczo zrozumieć, w jaki sposób inteligentne systemy mogą płynnie komunikować się ze sobą z semantycznym widokiem świata. Dzięki semantycznemu światopoglądowi istoty ożywają, przekształcając zasoby danych w informacje i zasoby informacyjne w wiedzę. Konieczne jest, aby istniał wspólny język do wykorzystywania semantycznego światopoglądu, aby heterogeniczne systemy mogły się ze sobą komunikować. SPARQL to standard W3C, który stara się być globalnym językiem zapytań, którego głównym celem jest interoperacyjność. SPARQL to powtarzający się akronim oznaczający protokół SPARQL i język zapytań RDF. Jak sama nazwa wskazuje, jest to język zapytań do zapytania o wiedzę (jako potrójne) przechowywany w formacie RDF. Tradycyjnie przechowujemy informacje w relacyjnych bazach danych w formacie tabelarycznym. Widok relacyjnej bazy danych jednostek można łatwo przedstawić w postaci potrójnej. Na przykład raz jeszcze rozważmy tabelę BOOK:

### Book\_ID : Tytuł : Autor : Firma : Rok

1 : Hit Refresh : Satya Nadella : Microsoft : 2017

2 : Shoe Dog : Phil Knight : Nike : 2016

W tym przypadku identyfikator wiersza (Book\_ID i tytuł) jest tematem, nazwa kolumny to predykat, a wartość kolumny to obiekt. Na przykład:

Potrójny:

{1: Hit Refresh} {Autor} {Satya Nadella}

Temat (nazwa jednostki) Predykat (nazwa atrybutu) Obiekt (wartość atrybutu)

Podmioty i predykaty są reprezentowane za pomocą identyfikatorów URI, które uniwersalnie identyfikują określone podmioty i predykaty jako zasoby:

<http://www.artificial-intelligence.big-data/book#>

<http://www.artificial-intelligence.big-data/book#author> „Satya Nadella”

Składnia Turtle pozwala na pełne napisanie wykresu RDF w zwartej i naturalnej formie tekstu. Zapewnia skróty dla typowych wzorców użytkownika i typów danych. Ten format jest zgodny ze składnią potrójnego wzorca SPARQL.

Użyjmy składni żółwia do przedstawienia tabeli książek w formacie RDF:

@prefix book: <<http://www.artificial-intelligence.big-data/book#>>

book:1 book:Title "Hit Refresh"

book:1 book:Author "Satya Nadella"

book:1 book:Company "Microsoft"

book:1 book:Year "2017"

book:2 book:Title "Shoe Dog"

book:2 book:Author "Phil Knight"

book:2 book:Company "Nike"

book:2 book:Year "2016"

Użyjemy prostego zapytania SPARQL, aby uzyskać listę książek opublikowanych w roku 2017:

```
PREFIX book: <http://www.artificial-intelligence.big-data/book#>
```

```
SELECT ?books
```

```
WHERE
```

```
{
```

```
?books book:year "2017" .
```

```
}
```

Mamy następujący wynik:

```
?books
```

```
book:1
```

Oto kolejne zapytanie SELECT, które pobiera więcej elementów danych ze zbioru danych:

```
PREFIX book: <http://www.artificial-intelligence.big-data/book#>
```

```
SELECT ?books ?bookName ?company
```

```
WHERE
```

```
{
```

```
?books book:year "2017" .
```

```
?books book:title ?bookName .
```

```
?books book:company ?company .
```

```
}
```

Wynik jest następujący:

```
?books ?bookName ?company
```

```
book:1 Hit Refresh Microsoft
```

Podczas gdy omawiamy rolę Ontologii w kontekście sztucznej inteligencji dla dużych zbiorów danych, pełne odniesienie do OWL i SPARQL jest poza zakresem tego tekstu. W poniższych podsekcjach przedstawimy ogólne odniesienie do języka SPARQL, które pomoże nam wykorzystać Ontologie do budowy sztucznej inteligencji.

### **Ogólna struktura zapytania SPARQL**

Ogólna struktura SPARQL jest następująca:

\* PREFIX: Podobnie jak deklaracja przestrzeni nazw w kontekście XML i pakiet w kontekście Java lub innych podobnych języków programowania, PREFIX jest odpowiednikiem SPARQL, co zapewnia wyjątkowość między reprezentacjami jednostek i eliminuje potrzebę wpisywania długich wzorców URI w kodzie SPARQL.

\* SELECT / ASK / DESCRIBE / CONSTRUCT:

-SELECT: jest to odpowiednik klauzuli SELECT języka SQL. Definiuje atrybuty, które należy pobrać z trójek RDF, które spełniają kryteria wyboru.

-ASK: Zwraca wartość logiczną true lub false w zależności od dostępności potrójnych funduszy RDF i na podstawie kryteriów wyboru w bazie wiedzy RDF.

-DESCRIBE: Ta konstrukcja zapytania zwraca wykres zawierający wszystkie dostępne trzykrotki z bazy wiedzy RDF, które pasują do kryterium wyboru.

-CONSTRUCT: Jest to bardzo przydatne podczas tworzenia nowego wykresu RDF z istniejącego RDF na podstawie kryteriów wyboru i warunków filtrowania. Jest to odpowiednik XSLT w kontekście XML. XSLT przekształca XML w zamierzonym formacie.

\* FROM: Definiuje źródło danych punktu końcowego RDF, dla którego zostanie uruchomione zapytanie. Jest to równoważnik SQL klauzuli FROM <TABLE\_NAME>. Punktem końcowym może być zasób w Internecie lub lokalny magazyn danych dostępny dla silnika zapytań.

\* WHERE: Definiuje interesującą nas część wykresu RDF. Jest to odpowiednik klauzuli SQL WHERE, która określa warunki filtrowania w celu pobrania ok

### **Dodatkowe funkcje SPARQL**

Dodatkowe funkcje SPARQL są następujące:

Opcjonalne dopasowanie: W przeciwieństwie do tradycyjnych relacyjnych magazynów danych, gdzie schematy bazy danych i ograniczenia są wstępnie zdefiniowane dla strukturyzowanej reprezentacji danych, w słowie big data mamy do czynienia z nieustrukturyzowanymi zestawami danych. Atrybuty dwóch zasobów tego samego typu mogą być różne. Opcjonalne dopasowanie przydaje się przy obsłudze heterogenicznych reprezentacji encji. Blok OPCJONALNY służy do wyboru elementów danych, jeśli istnieją.

Alternatywne dopasowanie: Ponownie, biorąc pod uwagę nieustrukturyzowany charakter zasobów wiedzy, naprzemienne dopasowanie zapewnia mechanizm zwracania dowolnych dostępnych właściwości.

UNION: Jest to sprzeczne ze wzorem OPTIONAL. W przypadku UNION co najmniej jeden zestaw danych musi znaleźć dopasowanie, biorąc pod uwagę kryteria zapytania.

DISTINCT: Jest to odpowiednik klauzuli SQL DISTINCT, która wyklucza wielokrotne występowanie tego samego potrójnego wyniku.

ORDER BY: instruuje zapytanie, aby uporządkować wyniki według określonej zmiennej w porządku rosnącym lub malejącym. Jest to również równoważne klauzuli ORDER BY w SQL.

FILTERS i wyrażenia regularne: SPARQL zapewnia funkcje ograniczające potrójne zestawy wyników za pomocą wyrażeń. Oprócz wyrażeń matematycznych i logicznych SPARQL umożliwia stosowanie wyrażeń regularnych do stosowania filtrów w zestawach danych na podstawie wzorców tekstowych.

GROUP BY: Umożliwia grupowanie wyników potrójnych RDF w oparciu o jedną lub więcej zmiennych.

HAVING: Ułatwia to wybór wyników zapytania na poziomie grupy.

SUM, COUNT, AVG, MIN, MAX i tak dalej są funkcjami dostępnymi do zastosowania na poziomie grupy.

### **Budowanie inteligentnych maszyn za pomocą ontologii**

Przyjrzelśmy się roli Ontologii w zarządzaniu zasobami dużych zbiorów danych jako repozytoriów wiedzy i zrozumieliśmy, że systemy obliczeniowe powinny postrzegać dane jako rzeczy zamiast ciągów. Chociaż niektóre duże systemy i wyszukiwarki internetowe używają semantycznego poglądu na świat, przyjęcie ontologii jako podstawy dla systemów jest powolne. Opiekunowie zasobów danych (rządy i wszyscy inni) muszą modelować wiedzę i zasoby w spójny i znormalizowany sposób, abyśmy mogli przekształcić obecne systemy obliczeniowe w inteligentne.

Rozważmy przypadek użycia, który wykorzystuje wykresy wiedzy oparte na ontologii, aby uprościć proces wejścia na pokład. Wszyscy doświadczyliśmy niezwykle ręcznego i czasochłonnego procesu podczas wsiadania do samolotu. Od momentu wejścia na lotnisko do czasu wejścia na pokład przechodzimy szereg kontroli bezpieczeństwa i weryfikujemy dokumenty. W połączonym świecie, w którym wszystkie zasoby wiedzy są znormalizowane i zdefiniowane jako Ontologie specyficzne dla domeny, możliwe jest opracowanie inteligentnych agentów, aby proces wejścia na pokład samolotu był bezproblemowy i bezproblemowy.

Nieznaczone rozwinięcie cech jest następujące:

**Cele:** Każdy inteligentny system powinien mieć dobrze zdefiniowany zestaw celów. Cele te regulują racjonalne decyzje podejmowane przez inteligentny system i kierują działaniami, a tym samym wynikami. Na przykład w przypadku inteligentnego agenta odpowiedzialnego za proces wejścia na pokład lotu jednym z celów jest ograniczenie dostępu do każdego, kto nie przejdzie wszystkich kontroli bezpieczeństwa, nawet jeśli dana osoba ma ważny bilet. Przy określaniu celów dla inteligentnych agentów jednym z głównych czynników powinno być to, że agent lub systemy AI powinny uzupełniać i zwiększać ludzkie możliwości.

**Środowisko:** inteligentny agent powinien działać w kontekście środowiska. Jego decyzje i działania nie mogą być niezależne od kontekstu. W naszym przykładzie użycia środowisko to lotnisko, bramy dla pasażerów, rozkłady lotów i tak dalej. Agenci postrzegają środowisko za pomocą różnych czujników, na przykład kamer wideo.

**Zasoby danych:** inteligentny agent potrzebuje dostępu do danych historycznych pod względem domeny i kontekstu, w którym działa. Zasoby danych mogą być dostępne lokalnie i globalnie (punkty końcowe w Internecie). Te zasoby danych idealnie powinny być zdefiniowane jako struktury schematu RDF ze znormalizowanymi reprezentacjami i protokołami. Te zasoby danych powinny mieć możliwość zapytania za pomocą standardowych języków i protokołów (SPARQL), aby zapewnić maksymalną interoperacyjność.

**Model:** w tym miejscu dostępna jest prawdziwa inteligencja agenta w postaci algorytmów i systemów uczenia się. Modele te ewoluują w sposób ciągły w oparciu o kontekst, decyzje historyczne, działania i wyniki. Zasadniczo model powinien działać lepiej (dokładniej) w czasie dla podobnych danych kontekstowych

**Efekty:** Są to namacalne aspekty agenta, które ułatwiają działania. W przykładzie agenta linii lotniczych na pokładzie efektem może być zautomatyzowany system otwierania bramy, który otwiera bramę,

gdy wszyscy pasażerowie zostaną w pełni zatwierdzeni (posiadając ważny bilet, tożsamość i kontrolę bezpieczeństwa). Świat zewnętrzny postrzega inteligentnego agenta poprzez efektory.

Działania i wyniki: W oparciu o kontekst środowiskowy, zasoby danych i wyszkolone modele inteligentny agent podejmuje decyzje, które wywołują działania poprzez efektory. Działania te zapewniają wyniki oparte na racjonalności decyzji i dokładności wyuczonego modelu. Wyniki są ponownie uwzględniane w szkoleniach modelowych w celu poprawy dokładności w pewnym okresie czasu. Na wysokim poziomie metoda inteligentnego agenta, która ułatwia wejście na pokład samolotu można przedstawić następująco:

1. Gdy pasażer wchodzi na lotnisko, kamera wideo odczytuje obraz i dopasowuje go do zasobów danych dostępnych dla agenta. Te zasoby danych są obiektami Ontologii, które są luźno powiązane i mają elastyczność struktury i atrybutów. Niektóre wnioski są dokonywane na pierwszym poziomie dopasowania, aby poprawnie zidentyfikować osobę, która weszła na lotnisko.
2. Jeżeli nie można zidentyfikować osoby za pomocą strumienia wideo, pierwsza brama lotniska nie otwiera się automatycznie i wymaga od pasażera skanowania linii papilarnych. Skanowanie odcisków palców jest sprawdzane względem zestawu danych, który jest po raz kolejny reprezentacją obiektu ontologicznego podmiotu osoby. Jeśli osoba nie zostanie zidentyfikowana na tym etapie, zostanie oznaczona do dalszych ręcznych procedur bezpieczeństwa.
3. Gdy osoba zostanie poprawnie zidentyfikowana, agent skanuje globalny katalog aktywnych biletów, aby upewnić się, że osoba ma ważny bilet na lot, który odlatuje z lotniska w rozsądnym terminie. Globalny katalog biletów i baza danych lotów są również dostępne jako obiekty Ontologii, do których agent może się odwoływać w czasie rzeczywistym.
4. Po zagwarantowaniu ważności biletu, karta pokładowa jest generowana i dostarczana do smartfona pasażera, po raz kolejny poprzez odwołanie się do osoby Ontologii w celu uzyskania danych osobowych w bezpieczny sposób. Instrukcje w czasie rzeczywistym dotyczące dojazdu do bramy są również wysyłane do urządzenia.

Agent może bezproblemowo poprowadzić pasażera do odpowiedniej bramki wejściowej. System można łatwo zbudować, gdy wszystkie heterogeniczne źródła danych zostaną znormalizowane i będą miały reprezentację ontologiczną, co ułatwia maksymalną interoperacyjność i eliminuje potrzebę kodowania różnorodnych reprezentacji wiedzy. Powoduje to ogólne zmniejszenie złożoności oprogramowania agenta i wzrost wydajności.

### **Nauka ontologii**

Z podstawowymi pojęciami dotyczącymi ontologii, a także ich znaczeniem w budowaniu inteligentnych systemów, konieczne jest, aby dla płynnie połączonego świata zasoby wiedzy były konsekwentnie reprezentowane jako Ontologie domenowe. Jednak proces ręcznego tworzenia ontologii dla konkretnej domeny wymaga dużego wysiłku, weryfikacji i zatwierdzenia. Uczenie się ontologii jest próbą zautomatyzowania procesu generowania ontologii przy użyciu algorytmicznego podejścia do tekstu w języku naturalnym, który jest dostępny w skali internetowej. Istnieją różne podejścia do nauki ontologii, takie jak:

\* Uczenie się ontologii z tekstu: W tym podejściu dane tekstowe są pobierane z różnych źródeł w sposób zautomatyzowany, a słowa kluczowe są wydobywane i klasyfikowane na podstawie ich występowania, sekwencji słów i wzorców.

\* Eksploracja danych połączonych: w tych procesach łączy się identyfikowane na opublikowanych wykresach RDF w celu uzyskania ontologii opartych na domyślnym rozumowaniu.

\* Uczenie się koncepcji od OWL: W tym podejściu wykorzystuje się istniejące Ontologie specyficzne dla domen w celu rozszerzenia nowych domen przy użyciu podejścia algorytmicznego.

\* Crowdsourcing: Podejście to łączy automatyczne ekstrakcję i odkrywanie ontologii w oparciu o analizę tekstową i współpracę z ekspertami w dziedzinie w celu zdefiniowania nowych ontologii. To podejście działa świetnie, ponieważ łączy moc obliczeniową i algorytmiczne podejście maszyn oraz wiedzę specjalistyczną ludzi w dziedzinie. Powoduje to poprawę prędkości i dokładności.

### **Oto niektóre z wyzwań związanych z nauką ontologii**

Radzenie sobie z heterogenicznymi źródłami danych: źródła danych w Internecie oraz w sklepach z aplikacjami różnią się formą i reprezentacją. Nauka ontologii stoi przed wyzwaniem ekstrakcji wiedzy i konsekwentnego wydobywania znaczeń z powodu heterogeniczny charakter źródeł danych.

Niepewność i brak dokładności: Ze względu na niespójne źródła danych, kiedy uczenie się ontologii próbuje zdefiniować struktury ontologii, istnieje poziom niepewności w odniesieniu do intencji oraz reprezentacji bytów i atrybutów. Powoduje to niższy poziom dokładności i wymaga interwencji człowieka od ekspertów w dziedzinie w celu dostosowania.

Skalowalność: Jednym z podstawowych źródeł uczenia się ontologii jest internet, który jest stale rosnącym repozytorium wiedzy. Internet jest również w większości nieustrukturyzowanym źródłem danych, co utrudnia skalowanie procesu uczenia się ontologii w celu uwzględnienia szerokości domeny z dużych fragmentów tekstu. Jednym ze sposobów rozwiązania problemu skalowalności jest wykorzystanie nowych, rozproszonych platform obliczeniowych typu open source (takich jak Hadoop).

Potrzeba przetwarzania końcowego: Podczas gdy nauka ontologii ma być procesem automatycznym, w celu przezwyciężenia problemów z jakością wymagamy pewnego poziomu przetwarzania końcowego. Proces ten musi być szczegółowo zaplanowany i zarządzany, aby zoptymalizować szybkość i dokładność nowych definicji Ontologii.

### **Proces uczenia się ontologii**

Proces uczenia się ontologii składa się z sześciu R. Są one wyjaśnione w następujący sposób:

Pobieranie (**Retrieve**) : zasoby wiedzy są pobierane ze źródeł internetowych i aplikacji ze sklepów specyficznych dla domeny przy użyciu przeszukiwania sieci i dostępu do aplikacji opartego na protokole. Terminy i aksjomaty specyficzne dla dziedziny są wyodrębniane za pomocą obliczeń wartości TF / IDF i przez zastosowanie metod wartości C / wartości NC. Wykorzystywane są powszechnie stosowane techniki klastrowania, a mierniki podobieństwa statystycznego są stosowane do wyodrębnionych tekstowych reprezentacji zasobów wiedzy.

Udoskonalenie (**Refine**): zasoby są czyszczone i przycinane, aby poprawić stosunek sygnału do szumu. W tym przypadku zastosowano podejście algorytmiczne w celu udoskonalenia. Na etapie udoskonalania terminy są grupowane zgodnie z pojęciami w ramach zasobów wiedzy.

Reprezentuj (**Represent**): na tym etapie system uczenia się ontologii organizuje koncepcje w strukturę hierarchiczną przy użyciu metody klastrowania bez nadzoru (w tym momencie rozumiem to jako podejście uczenia maszynowego do segmentacji danych)

Ponowne wyrównanie (**Re-align**): jest to rodzaj etapu przetwarzania końcowego, który wymaga współpracy z ekspertami w dziedzinie. W tym momencie hierarchie są wyrównywane pod kątem

dokładności. Ontologie są dostosowane do instancji pojęć i odpowiadających im atrybutów wraz z ograniczeniami liczności (jeden do jednego, jeden do wielu itd.). W tym kroku zdefiniowane są reguły definiowania struktury składniowej.

Ponowne użycie (**Reuse**): na tym etapie ponownie wykorzystywane są podobne ontologie specyficzne dla domeny z punktami końcowymi połączeń, a synonimy są definiowane w celu uniknięcia równoległych reprezentacji tych samych pojęć, które są finalizowane w innych definicjach ontologii.

Wydanie (**Release**): Na tym etapie Ontologie są wydawane do ogólnego użytku i dalszej ewolucji.

### **Często Zadawane Pytania**

P: Czym są ontologie i jakie mają znaczenie w inteligentnych systemach?

O.: Ontologia jako termin ogólny oznacza znajomość wszystkiego, co istnieje w tym wszechświecie. W odniesieniu do systemów informatycznych Ontologie reprezentują semantyczny i znormalizowany obraz światowych zasobów wiedzy. Są to specyficzne dla domeny reprezentacje wiedzy i modele związane z reprezentacjami bytów rzeczywistych. Inteligentne systemy łączące heterogeniczne dziedziny wiedzy muszą mieć dostęp do spójnych reprezentacji wiedzy w celu współdziałania i zrozumienia zdarzeń kontekstowych w celu wyciągania wniosków i podejmowania decyzji, które uruchamiają działania, a tym samym wyniki, w celu uzupełnienia ludzkich możliwości.

P: Jakie są ogólne właściwości Ontologii?

O: Ontologie powinny być kompletne, jednoznaczne, specyficzne dla domeny, ogólne i rozszerzalne.

P: Jakie są różne elementy Ontologii?

O: Różne komponenty ontologii to Pojęcia, Automaty, Relacje, Aksjomaty, Instancje i Operacje.

P: Jakie znaczenie ma uniwersalny format danych w systemach zarządzania wiedzą?

O: Format opisu zasobów (RDF) ma być uniwersalnym formatem reprezentacji wiedzy, umożliwiając heterogenicznym systemom interakcję i integrację w spójny i niezawodny sposób. To stanowi podstawę semantycznego spojrzenia na świat.

P: Jak można modelować światopogląd za pomocą Ontologii? Czy można zautomatyzować proces definiowania ontologii, biorąc pod uwagę ogromne i stale rosnące zasoby wiedzy we wszechświecie?

O: Zasoby wiedzy rosną wykładniczo z czasem. Aby stworzyć ontologiczną reprezentację tych zasobów, potrzebujemy zautomatyzowanego podejścia, bez którego trudno będzie nadrobić zaległości. Uczenie się ontologii przyjmuje podejście algorytmiczne poprzez wykorzystanie rozproszonych ram obliczeniowych do stworzenia podstawowego modelu światopoglądu. Proces uczenia się ontologii pobiera tekstowe, nieuporządkowane dane ze źródeł heterogenicznych, udoskonala je i reprezentuje w sposób hierarchiczny. Zostało to dostosowane do przetwarzania końcowego poprzez ponowne wykorzystanie istniejących zasobów wiedzy specyficznych dla domeny i ostatecznie udostępnione do ogólnego użytku przez inteligentnych agentów.

### **Podsumowanie**

Zbadaliśmy potrzebę znormalizowanego i spójnego przedstawienia wiedzy światowej na temat ewolucji inteligentnych systemów oraz sposobu modelowania tych systemów w stosunku do ludzkiego mózgu. Ontologie stosowane w systemach informatycznych to standard W3C, który określa ogólne



zasady reprezentacji wiedzy. Przedstawiono podstawowe pojęcia dotyczące RDF, OWL i języka zapytań w celu wyodrębnienia reprezentacji wiedzy w instancjach Ontologii za pomocą SPARQL. W tym rozdziale zbadaliśmy, jak używać ontologii do budowania inteligentnych agentów, patrząc na ich ogólne cechy. Na koniec dowiedzieliśmy się, w jaki sposób uczenie się ontologii ułatwia szybkie przyjęcie ontologii do światopoglądu przy spójnych zasobach wiedzy i reprezentacjach. W następnym rozdziale zapoznamy się z podstawowymi koncepcjami uczenia maszynowego i tego, w jaki sposób Big Data ułatwia proces uczenia się.