Reprezentacja wiedzy: SPARQL, DBpedia, inżynieria wiedzy i integracja modeli danych

Wojciech Jaworski

Instytut Informatyki
Uniwersytet Warszawski

Spis treści

Basic Graph Pattern

- SPARQL - query language for getting information from RDF graphs. It provides facilities to: extract information in the form of URIs, blank nodes, plain and typed literals.
  - extract RDF subgraphs.
  - construct new RDF graphs based on information in the queried graphs
  - matching graph patterns
- variables – global scope; indicated by '?' or '$'
- terms delimited by '<>' are relative URI references
- Set of Triple Patterns
  - Triple Pattern – similar to an RDF Triple (subject, predicate, object), but any component can be a query variable; literal subjects are allowed

    ?book dc:title ?title
  - Matching a triple pattern to a graph: bindings between variables and RDF Terms
- Matching of Basic Graph Patterns
  - A Pattern Solution of Graph Pattern *GP* on graph *G* is any substitution *S* such that $S(GP)$ is a subgraph of *G*.

Basic Graph Pattern - Multiple Matches

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Johnny Lee Outlaw".
_:a foaf:mbox <mailto:jlow@example.com> .
_:b foaf:name "Peter Goodguy".
_:b foaf:mbox <mailto:peter@example.org> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name . ?x foaf:mbox ?mbox }
```

- Query Result

name	**mbox**
"Johnny Lee Outlaw"	<mailto:jlow@example.com>
"Peter Goodguy"	<mailto:peter@example.org>

Basic Graph Pattern - Blank Nodes

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
_:a foaf:name "Alice".
_:b foaf:name "Bob".
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?x ?name
WHERE { ?x foaf:name ?name }
```

- Query Result

x	**name**
_:c	“Alice“
_:d	“Bob”

Graph Patterns

- Basic Graph Pattern – set of Triple Patterns
- Group Pattern - a set of graph patterns must all match
- Value Constraints - restrict RDF terms in a solution
- Optional Graph Patterns .- additional patterns may extend the solution
- Alternative Graph Pattern – two or more possible patterns are tried
- Patterns on Named Graphs - patterns are matched against named graphs

Group Pattern

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { ?x foaf:name ?name . ?x foaf:mbox ?mbox }
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox
WHERE { {?x foaf:name ?name; foaf:mbox ?mbox } }
```

Value Constraints

- Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial".
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web".
:book2 ns:price 23 .
```

- Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x ns:price ?price . FILTER ?price < 30 . ?x dc:title ?title . }
```

- Query Result

title	**price**
"The Semantic Web"	23

Optional graph patterns

- Data

```
@prefix dc: <http://purl.org/dc/elements/1.1/> .
@prefix : <http://example.org/book/> .
@prefix ns: <http://example.org/ns#> .
:book1 dc:title "SPARQL Tutorial".
:book1 ns:price 42 .
:book2 dc:title "The Semantic Web".
:book2 ns:price 23 .
```

- Query

```
PREFIX dc: <http://purl.org/dc/elements/1.1/>
PREFIX ns: <http://example.org/ns#>
SELECT ?title ?price
WHERE { ?x dc:title ?title .
        OPTIONAL { ?x ns:price ?price . FILTER ?price < 30 }}
```

- Query Result

title	**price**
“SPARQL Tutorial“	
"The Semantic Web"	23

Multiple Optional Blocks

- Data

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
_:a foaf:name "Alice".
_:a foaf:homepage <http://work.example.org/alice/> .
_:b foaf:name "Bob".
_:b foaf:mbox <mailto:bob@work.example> .
```

- Query

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name ?mbox ?hpage
WHERE { ?x foaf:name ?name .
        OPTIONAL { ?x foaf:mbox ?mbox }.
        OPTIONAL { ?x foaf:homepage ?hpage } }
```

- Query Result

name	**mbox**	**hpage**
“Alice“		<http://work.example.org/al
“Bob“	<mailto:bob@example.com>	

Alternative Graph Patterns

- Data

```
@prefix dc10: <http://purl.org/dc/elements/1.0/> .
@prefix dc11: <http://purl.org/dc/elements/1.1/> .
_:a dc10:title "SPARQL Query Language Tutorial".
_:b dc11:title "SPARQL Protocol Tutorial".
_:c dc10:title "SPARQL".
_:c dc11:title "SPARQL (updated)".
```

- Query

```
PREFIX dc10: <http://purl.org/dc/elements/1.0/>
PREFIX dc11: <http://purl.org/dc/elements/1.1/>
SELECT ?x ?y
WHERE { { ?book dc10:title ?x } UNION { ?book dc11:title ?y } }
```

- Query Result

x	**y**
	"SPARQL (updated)"
	"SPARQL Protocol Tutorial"
"SPARQL"	
"SPARQL Query Language Tutorial"	

Query forms

- SELECT: returns all, or a subset of the variables bound in a query pattern match
- CONSTRUCT: returns an RDF graph constructed by substituting variables in a set of triple templates
- DESCRIBE: returns an RDF graph that describes the resources found.
- ASK: returns whether a query pattern matches or not.

Support for SPARQL

- SPARQL and Jena
  - module called ARQ that implements SPARQL; also parses queries expressed in RDQL or its own internal language.
  - not yet part of the standard Jena distribution; availbale from either Jena‘s CVS repository or as a self-contained download
- Twinkle
  - simple Java interface that wraps the ARQ SPARQL Processor library (the add-on to Jena).
- Redland
  - set of free software packages that provide support for RDF, including querying with RDQL and SPARQL using the Rasqal RDF Query Library. .

Spis treści

Wikipedia

- Wikipedia — electronic encyclopedia.
- 4.699.911 articles in English, 1.087.696 articles in Polish (25 November 2014)
- Number of articles is dynamically increasing
  - 4.137.668 articles in English, 942.309 articles in Polish on 6 January 2013,
  - 3.840.850 articles in English, 863.021 articles in Polish on 7 January 2012.
- Greatest existing set of concepts.
- Corresponding articles in different languages are linked.
- Concepts do not form the structure of the ontology, however, they are grouped into categories and related by links.
- Descriptions of concepts are expressed in natural language, and not as a logical formula.
- Wikipedia is constantly updated.

DBpedia

- DBpedia is a project aiming to extract structured content from the information created as part of the Wikipedia project.
- The whole DBpedia data set describes 4,580,000 entities, out of which 4,220,000 are classified in a consistent ontology, including 1,445,000 persons, 735,000 places, 123,000 music albums, 87,000 films, 19,000 video games, 241,000 organizations, 251,000 species and 6,000 diseases.
- The DBpedia project uses the Resource Description Framework (RDF) to represent the extracted information and consists of 3 billion RDF triples, 580 million extracted from the English edition of Wikipedia and 2.46 billion from other language editions.
- DBpedia is generating automatically by extraction of structured data from Wikipedia content.

Querying DBpedia

- There is a public SPARQL endpoint over the DBpedia data set at http://dbpedia.org/sparql.
- Query

```
PREFIX dbprop: <http://dbpedia.org/property/>
PREFIX db: <http://dbpedia.org/resource/>
SELECT ?who, ?work, ?genre WHERE {
db:Tokyo_Mew_Mew dbprop:author ?who .
?work dbprop:author ?who .
OPTIONAL { ?work dbprop:genre ?genre } . }
```

- DBpedia ontology is available in OWL format at http://wiki.dbpedia.org/Downloads2014

Spis treści

Inżynieria wiedzy

- Przejście od słownego opisu do formalnej specyfikacji według tradycyjnej metodologii następuje w sposób nieanalityczny: projektant systemu na podstawie opisu tworzy w umyśle model i wyraża go w sposób formalny jako diagram encji, UML, czy hierarchię klas.
- Inżynieria wiedzy ma na celu wykonanie translacji nieformalnej specyfikacji słownej do wykonywalnego programu w sposób systematyczny.

Inżynieria wiedzy

- Inżynieria wiedzy jest zastosowaniem logiki i ontologii do konstruowania obliczalnych modeli dla zadanych dziedzin w jakimś określonym celu.
- Specyfikację traktujemy tu jako dane w języku naturalnym, które tłumaczymy do postaci mającej jawną strukturę i ustaloną semantykę.
- Obliczalność, dziedzina zastosowań oraz istnienie celu różni ją od czystej matematyki, która:
  - nie potrzebuje dziedziny, w której będzie stosowana.
  - modele mogą być nieobliczalne, a nawet nieskończone.
  - nie ma celu innego niż estetyczna satysfakcja i kontemplowanie eleganckich abstrakcji.

Nieformalne specyfikacje

There is a traffic light that automatically turns red or green, but it also has an option for manual control under special circumstances.

- Ekspert pisze w sposób nieformalny korzystając z języka, którego inżynier wiedzy nie zna.
- Każdy oczywisty dla eksperta termin techniczny musi zostać zanalizowany przez inżyniera wiedzy i wyjaśniony.
- Problem w automatycznym tłumaczeniu nieformalnej specyfikacji do formalnych algorytmów stanowi olbrzymia ilość wiedzy, którą przywołuje każde słowo, a nie składnia języka naturalnego.
- Tłumacząc specyfikację do obliczalnej postaci trzeba czynić założenia i dodawać detale.

Interpretacja specyfikacji

The color of the traffic light X may be either red or green. X has an automatic control switch, which may be either on or off.
If the automatic control switch is on,
then X behaves according to the following two rules:
When the color of X becomes green,
it remains green for g seconds;
then it changes red.
When the color of X becomes red,
it remains red for r seconds;
then it changes green.

- Interpretacja zawiera dodatkowe założenia, na przykład światła pozostają zielone lub czerwone przez stałą ilość sekund.
- Założenia opierają się na źródłach wiedzy niezależnych od specyfikacji takich jak
  - wiedza ogólna inżyniera wiedzy
  - dodatkowe źródła informacji
  - dyskusje z ekspertem

Zasady reprezentacji wiedzy

- Obiekty których nie można bezpośrednio przechować w pamięci komputera reprezentujemy za pomocą surogatów. Manipulując surogatami program komputerowy może symulować zewnętrzny system i wnioskować o nim.
- Modelując dziedzinę podejmujemy decyzje odnośnie pojęć w ontologii i relacji między nimi. Ustalamy jakie rodzaje bytów istnieją w analizowanej dziedzinie.
- Tworzymy częściową teorię opisującą dziedzinę.
- Reprezentacja musi działać wydajnie na dostępnym sprzęcie.
- Reprezentacja powinna umożliwiać komunikację pomiędzy inżynierem wiedzy a ekspertem.

Implementacja proceduralna

- Każdy istotny byt taki jak aktualny czas czy kolor świateł ma przypisaną sobie zmienną, która stanowi symboliczną reprezentację zewnętrznego bytu.
- Wartości tych zmiennych mogą być zmieniane by symulować działanie systemu.
- W podejściu proceduralnym każde zdarzenie jest operacją, która zmienia stan modelu, czyli procedurą.
- Tworzymy mechanizm kontrolujący, który wywołuje procedury, gdy ich warunki początkowe są spełnione.
- Uruchamiamy program i parzymy jak ewoluuje w czasie.

Implementacja deklaratywna

- Podejście deklaratywne jest oparte na więzach, czy też aksjomatach, które definiują warunki początkowe i końcowe oraz zmiany które zachodzą w modelu podczas każdej akcji.
- Zamiast wykonywać programy korzystamy z systemów dowodzenia twierdzeń, żeby wywnioskować konsekwencje wykonania akcji.
- System dowodzenia twierdzeń jest interpreterem, który wykonuje aksjomaty.
- Efekt proceduralnej symulacji uzyskujemy za pomocą dedukcji.

Metapoziom

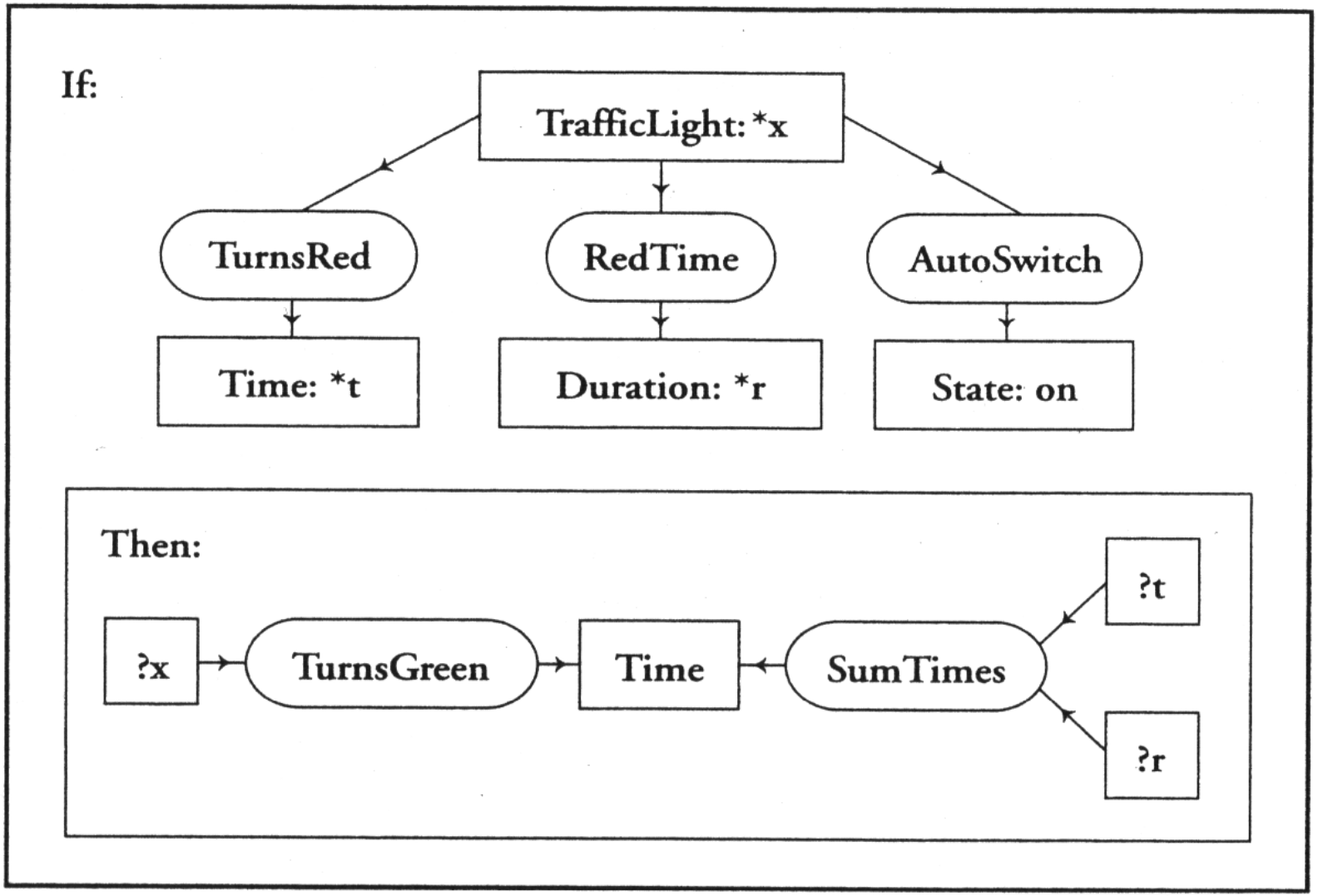
- Podejście proceduralne nie pozwala wyjaśnić dlaczego system działa tak jak działa.
- Nie przechowuje też historii zmian.
- Nie można powiedzieć, dlaczego światło zmienia kolor.
- Nie odpowiada na pytanie typu: przez jak długi średni czas światła były obsługiwane manualnie.
- Gdybyśmy potrzebowali tych informacji musielibyśmy dodać instrukcje z metapoziomu umożliwiające wnioskowanie o światłach.
- Opis tych instrukcji może być bardziej skomplikowany niż pierwotny program.
- W podejściu deklaratywnym symulujemy światła drogowe dowodząc twierdzenia o ich stanie.
- Mając interpretację logiczną możemy zadać dowolne pytanie wyrażalne za pomocą naszej ontologii i logiki uzyskując na nie odpowiedź, pod warunkiem, że da się tą odpowiedź obliczyć.

Komunikacja z ekspertem

- Kiedy inżynier wiedzy zinterpretuje specyfikację inaczej niż jej autor pojawiają się błędy.
- Po uszczegółowieniu i sformalizowaniu trzeba ponownie skonsultować specyfikację z ekspertem, co wymaga posiadania zrozumiałego formalizmu.
- Jest to szczególnie istotne, gdy częścią specyfikacji są akty prawne, regulaminy itp.
- Stylizowany angielski:
  If a traffic light x turns red at time t,
  has a red time of a duration r,
  and has autoswitch in the state on,
  then x turns green at a time, which is the sum of t and r.
- Eksperci często wolą diagramy stosowane w ich dziedzinach niż opis.

Komunikacja z ekspertem: grafy pojęć

Precyzyjne (formalna semantyka), czytelne, dają się wyjaśnić ekspertowi.



Spis treści

Integracja modeli danych (podejście tradycyjne)

- Zbiór źródłowych schematów (schematów istniejących baz danych).
- Schemat globalny (perspektywa integrująca bazy).
- Odwzorowanie tłumaczące zapytania.
- Global as View (GAV):
  - Schemat globalny jest zbiorem perspektyw stworzonych na podstawie schematów źródłowych.
  - Zapytania korzystają z tych perspektyw, nie wymagają więc dodatkowego przetwarzania.
  - Dodanie nowego źródła danych wymaga modyfikacji kodu istniejących perspektyw.
- Local as View (LAV):
  - Schematy źródłowe są zbiorami perspektyw stworzonych na podstawie schematu globalnego.
  - Aby wykonać zapytanie trzeba odwrócić perspektywy co wymaga przeszukania przestrzeni możliwych zapytań.
  - Dodanie nowego źródła danych jest niezależne od innych źródeł.

Integracja modeli danych (z użyciem ontologii)

- Kolekcja modeli danych.
- Ontologia opracowana na potrzeby zadania integracji, wyrażona w OWL.
- Struktura bazy relacyjnej odwzorowywana jest następująco:
  - Tabelom odpowiadają pojęcia, kolumnom atrybuty (rdf:property)
  - Tabele reprezentujące relacje wiele do wielu tłumaczymy na atrybuty.
  - Klucz wiersza w tabeli stanowi identyfikator indywiduum.
  - Wartość w kolumnie to wartość atrybutu.
- Struktura XML odwzorowywana jest następująco:
  - Atrybutom XML odpowiadają atrybuty w RDF
  - Elementy o wartościach należących do prostych typów tłumaczymy na atrybuty.
  - Pozostałym elementom odpowiadają pojęcia.
  - Trzeba utworzyć identyfikatory dla indywiduów i scalić powtarzające się dane.
- Zapytania do bazy danych zadawane są w SPARQL i tłumaczone na SQL (dla baz relacyjnych) lub XQuery (dla baz XML), co wymaga zakodowania odwzorowania.

Integracja modeli danych (z użyciem bazy wiedzy)

- Kolekcja modeli danych.
- Uniwersalna baza wiedzy.
- Odwzorowanie pojęć i relacji z poszczególnych modeli danych do pojęć z bazy wiedzy.
- Zanurzamy identycznościowo modele danych w bazie wiedzy.
- Definiujemy więzy (w postaci aksjomatów) pomiędzy pojęciami z modeli danych a pojęciami z bazy wiedzy.
- Tłumaczenie zapytań realizuje system wnioskujący.
- Baza wiedzy służy za uniwersalny interfejs umożliwiający za pomocą języka reprezentacji wiedzy wyszukiwanie informacji, modyfikację danych oraz wnioskowanie.
- Deklaratywne odwzorowanie modelu uważane jest za lepsze od proceduralnego z uwagi na zdolność systemu do określania swoich kompetencji (wnioskowanie o przekonaniach).
- W Cyc'u zrealizowane pod nazwą Semantic Knowledge Source Integration (SKSI), jest to zasadniczo to samo co Semantic Integration Bus.

Jak odwzorować pojęcia?

- UMBEL (Upper Mapping and Binding Exchange Layer)
- Zawierająca 28.000 pojęć ontologia
- UMBEL Vocabulary służy do rozpoznawania potencjalnych związków pomiędzy pojęciami występującymi w modelach danych.
- Z założenia odwzorowanie to jest przybliżone.
- Drugim celem UMBEL jest zapewnienie ustalonego zestawu pojęć, za pomocą których ustalone w poprzednim kroku odwzorowania mogą być uporządkowane.
- Ten zbiór pojęć nie ma na celu modelowania świata, tylko zapewnienie ustalonego zbioru punktów referencyjnych, które pozwolą uporządkować treść.
- UMBEL jest podzbiorem pojęć wybranych z OpenCyc'a.

Spis treści

Wyszukiwarki semantyczne

- Information Retrieval: odnajdywanie dokumentów na podstawie zadanego zapytania.
- Information Extraction: odnajdywanie faktów.
- Zapytanie jest wzorcem semantycznym np.
  - zbiorem pojęć,
  - formułą logiczną, czyli zbiorem pojęć powiązanych relacjami.
- Wzorce semantyczne trzeba zwykle wydobyć z ciągu słów zadanego przez użytkownika.
- Poszukiwane dokumenty nie muszą zawierać pojęć z zapytania mogą jedynie dotyczyć zadanego tematu.
- Wyszukiwarka może próbować odgadnąć (zapytać się) intencje użytkownika i kontekst zapytania.

Wyszukiwarki semantyczne

- Dane częściowo ustrukturalizowane:
  - pola tekstowe różnej długości,
  - etykiety pól,
  - tagi wewnątrz tekstów.
- Metadane: dane mówiące o danych.
- Inne dane (obraz, dźwięk, animacja, dane numeryczne) sprowadzamy do postaci tekstu ze strukturą.
- Modele języka:
  - worek słów,
  - ciąg słów,
  - fraza (parsowanie powierzchniowe),
  - zdanie (parsowanie głębokie).

Worki słów

- Pojemność semantyczna ograniczona do pojęć wyrażonych przez pojedyncze słowa.
- Relacje między pojęciami są tracone.
- Analiza lingwistyczna ogranicza się do sprowadzania słów do form podstawowych.
- Listy imion, nazwisk, miejsc itp. oraz Wordnet wystarczają jako zasoby semantyczne.
- Reprezentacja worka słów jako wektora częstości pozwala traktować tekst jako punkt w przestrzeni liniowej.
- Tradycyjny model stosowany w wyszukiwarkach.

Ciągi słów

- Pokrywa wszystkie pojęcia wyrażone przez spójne ciągi słów, (dlatego lepiej sprawdza się dla angielskiego niż polskiego)
- Problemem jest ustalenie granic reprezentacji poszczególnych pojęć i określenie dopuszczalnych permutacji słów w opisie pojęcia.
- Nazwy pojęć i ich synonimy możemy wydobyć z Wikipedii.
- Wikipedię możemy wykorzystać również do rozstrzygania niejednoznaczności.
- O relacjach między pojęciami możemy wnioskować na podstawie ich bliskości w tekście.
- Dobre efekty daje wstępny podział na zdania.
- Model nadaje się do przybliżonego wydobywania faktów rozumianych tutaj jako zdania dotyczące zapytania (Textpresso).
- W tekstach o ograniczonej tematyce (ontologii) zbiór pojęć wyznacza jednoznacznie relacje, które łączą te pojęcia.

Parsowanie powierzchniowe

- Obejmuje relacje między znajdującymi się blisko siebie pojęciami.
- Problemem jest odwzorowanie między związkami syntaktycznymi a relacjami semantycznymi.
- Odwzorowanie to określa baza wiedzy,
- albo gramatyka semantyczna, czyli zbiór reguł mówiących jak fakty dotyczące pojęć z ontologii wyrażane są w języku.
- Wyrażenia regularne
- Parsowanie powierzchniowe ma charakter rozpoznawania znanych wzorców w morzu informacji.

Parsowanie głębokie

- Kompletne odwzorowanie tekstu na formułę języka reprezentacji wiedzy.
- Aktualnie wykonalne jedynie dla zbiorów tekstów o ograniczonej tematyce.
- Etykiety pól mogą wskazywać temat dla zadanego pola.

Informacja niesiona przez etykiety pól

- Etykieta może wskazywać pojęcie, a wartość pola indywiduum, np. autor, data.
- Jak wykorzystać informację, że dane pole zawiera tytuł, abstrakt, albo tytuł sekcji?
- Jak wykorzystać informację, że dane pola zawierają opis objawów, diagnozę, czy opis kuracji?
- (Zakładamy, że użytkownik nie pyta wprost o abstrakt, czy diagnozę)
- Wykorzystanie informacji niesionej przez etykiety pól wymaga integracji struktury zadanej przez etykiety pól z bazą wiedzy.

Acknowledgements

Part of material has been taken from the lecture

SPARQL Query Language for RDF
Cristina Feier
http://www.wsmo.org/wsml/papers/presentations/sparql.ppt