**25. Uczenie indukcyjne: metody, zastosowania.**

Uczenie indukcyjne jest uczeniem się poprzez przykłady. Sprowadza się do znajdowania hipotez, które najlepiej opisują obserwowane fakty, znajdują wsród nich zależności. Można rozumieć uczenie się indukcyjne jako znajdowanie ogólnych zasad na podstwie pojedynczych przykładów. Znalezione w procesie uczenia hipotezy nie muszą sprawdzać się dla każdego przykładu, ale muszą być na tyle dokładne, by możliwe było na ich podstawie przewidywanie klasyfikacji dalszych przykładów.

Przykłady przedstawione w procesie uczenia to przykłady trenujące, które tworzą zbiór przykładów uczących. Czasami zdarzyć mogą się błędne przykłady uczące, które mają przekłamania jeśli chodzi o wartości atrybutów lub jeśli chodzi o etykietę kategorii. Różne algorytmy uczące się różnie reagują na błędy w zbiorze trenującym. O zbiorze uczącym zawierającym błędne przykłądy uczące mówimy, że jest *zaszumiony*.

**RODZAJE UCZENIA SIĘ**

Trzy najczęściej spotykane i mające najwięcej praktycznych zastosowań przykłady indukcyjnego uczenia się to: uczenie się pojęć, tworzenie pojęć oraz uczenie się aproksymacji funkcji.

**Uczenie się pojęć**

Pojęcie oznacza podział zbioru obiektów na dwa zbiory: obiektów, które określone pojęcie przedstawiają bądź nie. Te pierwsze są przykładami pozytywnymi, a te drugie negatywnymi dla danego pojęcia. Zbiór obiektów można także podzielić na więcej niż dwa rozłączne podzbiory, z któych każdy będzie reprezentował inne - rozłączne pojęcie. Uczenie się pojęć polega na umiejętności klasyfikacji przykładów do różnych klas na podstawie hipotezy wygenerowanej na zbiorze uczącym.

Przykłady uczące (trenujące) mają postać listy argumentów i ich wartości oraz oznaczenia kategorii do której należą, tzw. etykiety kategorii. W przypadku rozpatrywania jednej kategorii można mówić o atrybucie decyzyjnym - 'tak' lub 'nie' - decyzja przynależności do kategorii. Taka postać przykładu uczącego wskazuje na uczenie z nadzorem - uczeń ma znaleźć hipotezę opartą na przykładach uczących, która jak najlepiej (z jak najmniejszym błędem) potrafiłaby klasyfikować inne (nieznane przy uczeniu) przykłady.

Błąd w uczeniu się pojęć mierzony jest stosunkiem ilości przykładów błędnie zakwalifikowanych do ilości wszystkich przykładów przedstawionych uczniowi. Uczeń klasyfikuje przykłady spoza zbioru uczącego na podstawie znalezionej przez siebie na tym zbiorze hipotezy.

* Pojęcia służą do klasyfikacji obiektów na grupy (kategorie)
* Podstawowa wersja: podział na dwie grupy – obiekty należą do danego pojęcia (pozytywne przykłady) i nie należące do niego (negatywne przykłady)
* Pojęcia – funkcje przekształcające dziedzinę (obiekty) w zbiór kategorii: dwuelementowy dla pojęć pojedynczych i może mieć więcej pojęć w przypadku pojęć wielokrotnych (np. alfabet polski to jedno pojęcie o 33 kategoriach – literach)

**Tworzenie pojęć**

W tworzeniu pojęć przykłady uczące pozbawione są etykiety klasy, do której należą, uczeń ma za zadanie samodzielne znalezienie zależności (podobieństw) i pogrupowanie przykładów w klasy na podstawie zbioru uczącego. Grupowanie odbywa się na zasadzie minimalizowania różnić wsród przykładów wewnątrz grupy a maksymalizowania różnić pomięrzy grupami. Jest to przykład uczenia bez nadzoru. W przypadku tworzenia pojęć hipoteza określa zarówno tworzone kategorie jak i przynależność przykładów do nich.

W zasadzie także w tworzeniu pojęć zawiera się uczenie pojęć, gdyż po stworzniu pojęć i klasyfikacji przykładów algorytm ma za zadanie nauczyć się tych pojęć, by móc klasyfikować nowe przykłady. Dla danego przykładu uczeń jest w stanie podać nie tylko etykietę kategorii, lecz także jej opis.

* Nie zawsze kategorie pojęć są znane – uczeń obserwuje nieetykietowane przykłady (opisy obiektów)
* Uczeń, na podstawie obserwacji, grupuje obiekty w kategorie zgodnie z pewnymi kryteriami podobieństwa
* Tworzenie pojęć łączy dwa podzadania:
  + podział przykładów trenujących na grupy, które odpowiadają kategoriom
  + nauczenie się pojęć odpowiadających tworzonym kategoriom, aby było możliwe klasyfikowanie nowych przykładów

**Uczenie się aproksymacji funkcji**

Uczenie się aproksymacji funkcji jest uczeniem z nadzorem i podobne jest do uczenia się pojęć z tą różnicą, że liczba kategorii jest zbiorem liczb rzeczywistych, a nie określonym z góry skończonym zbiorem jak w przypadku uczenia się pojęć. Przykład uczący ma zatem postać sekwencji argumentów funkcji oraz wartości funkcji dla tych argumentów, która odpowiada etykiecie kategorii. Uczeń ma za zadanie aproksymować postać funkcji na podstawie przykładów na całą dziedzinę.

W przypadku uczenia się aproksymacji funkcji lepsze efekty uzyskiwane są wówczas, gdy uczniowi dostarczone są nie wartości funkcji dla podanych w przykładzie trenującym argumentów, a różnice (błędy) między szacowaniem przyjętą przez ucznia hipotezą a wartością rzeczywistą.

* Uczenie pojęć – uczenie się funkcji odwzorowującej przykłady na skończony i niewielki zbiór kategorii
* Uczenie się aproksymacji funkcji – zbiór wartości uczonej funkcji jest zbiór liczb rzeczywistych
* Przykłady – pary składające się z argumentu funkcji, reprezentowanego zazwyczaj przez wektor liczb rzeczywistych i jej wartości dla tego argumentu
* Uczeń (system) ma wygenerować funkcję dobrze przybliżającą funkcję docelową
* Wartość funkcji ma być obliczona z dużą dokładnością nie tylko dla przykładów, ale – indukcyjne uogólnienie – dla dowolnych innych argumentów z dziedziny

**TRYBY UCZENIA SIĘ**

Przez tryb uczenia się rozumie się sposób dostarczania przykładów trenujących uczniowu. W uczeniu indukcyjnym wyróżnia się także różne tryby uczenia się: wsadowy, inkrementacyjny, epokowy, korekcyjny.

**Tryb wsadowy**

Polega na dostarczeniu wszystkich przykładów uczących uczniowu przed rozpoczęciem procesu uczenia. Hipoteza generowana jest na podstawie dostarczonego zbioru trenującego, nie ma możliwości 'douczania' - każdy nowo dodany przykład to konieczność uruchomienia procesu uczenia od nowa ze zbiorem uczącym wzbogaconym o ten przykład. Nie ma także możliwości 'wglądu' do roboczych postaci hipotezy.

Praktycznie każdy algorytm może pracować w trybie wsadowym, tryb ten nie ma dodatkowych wymagań.

**Tryb inkrementacyjny**

Tryb ten, zwany także trybem 'na bieżąco', polega na przedstawianiu kolejno pojedynczych przykładów uczących i każdorazowym ulepszaniu hipotezy na podstawie tego przykładu. W dowolnym momencie jest możliwość wglądu w bieżącą hipotezę, można również zaprzestać prezentacji przykładów uczących uznająć hipotezę za obowiązującą. Ważną cechą tego trybu jest to, że zbiór uczący nie ma określonego rozmiaru, istnieje możliwość 'douczenia' poprzez prezentację kolejnych przykładów.

**Tryb epokowy**

Tryb ten jest wypadkową dwóch poprzednich - wsadowego i inkrementacyjnego. Polega na dzieleniu zbioru uczącego na podzbiory zwane epokami i prezentowaniu epok uczniowi w sposób wsadowy. Jednak po każdej epoce, jak w trybie inkrementacyjnym, istnieje możliwość wglądu do hipotezy i podjęcia decyzji o dalszym uczeniu lub zaprzestaniu podawania wzorców i przyjęcia aktualnej hipotezy jako obowiązującej.

**Tryb korekcyjny**

Jest podobny do trybu inkrementatyjnego i stosowany wyłącznie do uczenia nadzorowanego. Różnica polega na tym, że kolejny przykład uczący podawany jest uczniowi najpierw bez etykiety kategorii, uczeń przetwarza przykład według swojej bieżącej hipotezy uzyskując okreśłony wynik, po czym prezentowana jest informacja rzeczywista - wynik prawidłowy lub wartość błędu (różnica między wynikiem ucznia a wartością rzeczywistą). Na podstawie tej informacji dokonywana jest modyfikacja hipotezy.

### Rodzaje indukcyjnego uczenia się

Wyróżniamy trzy rodzaje indukcyjnego uczenia się:

1. **uczenie się pojęć** - uczenie się sposobu klasyfikowania obiektów do klas
2. **tworzenie pojęć** - uczenie się tworzenia klas i przydzielania do nich obiektów
3. **uczenie się aproksymacji** - uczenie się odwzorowywanie obiektów na liczby rzeczywiste
4. **uczenie się pojęć - uczenie się sposobu klasyfikowania obiektów do klas**  
     
   Termin "pojęcie" jest ważnym słowem przy opisie świata i wiedzy o nim. Poszczególne pojęcia pozwalają na odróżnianie rzeczy, zjawisk, uczuć, cech itd. Znając pojęcie np. kota możemy w zoo odróżnić zwierzęta, które są kotami od tych które nimi nie są. Oczywiste jest że różne koty nie wyglądają identycznie, a mimo to potrafimy odróżnić je od innych zwierząt. To samo tyczy się psów oraz wielu innych terminów, jak np. krzesło, ołówek, kubek, samochód, litera "a". Możemy więc przyjąc, że pojęcia służą do klasyfikowania obiektów na **kategorie** (czyli grupy).   
     
   Możemy rozpatrywać sytuacje, gdy dziedzina podzielona jest na dwie kategorie: obiektów należących i nienależących do pojęcia. Mówimy wówczas, że przykład jest pozytywny lub negatywny. Możemy jednak rozpatrywać **pojęcia wielokrotne** - wówczas dziedzina podzielona jest na kilka kategorii. Np. cyfra będzie pojęciem wielokrotnym złożonym z 10 kategorii. W związku z tym nie ma przeszkód, by stwierdzić, że pojęcie jest funkcją przekształcającą dziedzinę w zbiór kategorii (który jest conajmniej dwuelementowy). Można także powiedzieć, że pojęcia przypisuje obiektowi **etykietę** kategorii, do której należy.   
     
   Przykłady trenujące składają się z par (opis obiektu, etykieta kategorii do której należy obiekt). Jest to zatem uczenie nadzorowane.   
     
   Otrzymana hipoteza może być czytelna dla człowieka ale nie musi - zależy to od przyjętej metody reprezentacji wiedzy.   
     
   Oznaczenia:  
   Oznaczmy przez ***C*** skończony zbiór kategorii pojęć pewnej klasy pojęć zdefiniowanej na dziedzinie.  
   Pojęcie oznaczamy przez ***c∈C***. Jest ono funkcją ***c:X→C***. Dla pojęć pojedynczych przyjmuje się ***C={0,1}***, dla pojęć wielokrotnych liczność zbioru ***C*** będzie większa od 2 (***|C|>2***).  
   ***H*** oznacza przestrzeń możliwych hipotez - zawiera ona wszystkie hipotezy, jakie może skonstruować uczeń. ***h∈H*** oznacza pojedynczą hipotezę: ***h: X→C***. Należy zwrócić uwagę, że aby dokładnie nauczyć się każdego pojęcia musi być spełniony warunek ***C⊆H*** - wówczas jest pewność, że przestrzeń hipotez zawiera hipotezę identyczną z pojęciem docelowym. Nie zawsze jednak ten warunek jest spełniony w praktyce i wówczas nie można mieć pewności, że pojęcie docelowe zostanie nauczone dokładnie.  
     
   Termin **pokrywanie** używa się w odniesieniu do pojęć pojedynczych i mówi się, że hipoteza pokrywa przykład jeśli klasyfikuje go jako pozytywny. Mówimy również o pokrywaniu zbioru przykładów przez hipotezę, jeżeli każdy przykład z tego zbioru jest pokrywany.
5. **tworzenie pojęć - uczenie się tworzenia klas i przydzielania do nich obiektów**  
     
   Ten rodzaj indukcyjnego uczenia się wykorzystuje się wtedy, gdy nie znane są kategorie przykładów trenujących, czyli nie znane są pojęcia. Zadaniem jest właśnie na podstawie tych przyładów stworzenie pojęć i wówczas określenie przynależności do nich przykładów. Jest to typowa sytuacja uczenia się bez nadzoru - uczeń obserwuje jedynie opisy obiektów i sam wyciąga wnioski.
6. **uczenie się aproksymacji - uczenie się odwzorowywanie obiektów na liczby rzeczywiste**  
     
   Koncepcyjnie jest to bardzo podobny rodzaj do "uczenia się pojęć", z tą różnicą, że funkcja odwzorowuje przykłady na zbiór liczb rzeczywistych. Zadanie polega więc na odgadnięciu przebiegu funkcji na podstawie kilku danych wartości aproksymowanej funkcji.

**WYBRAĆ SOBIE ALGORYTM DO OMÓWIENIA**

### Find-S

Algorytm **Find-S** jest algorytmem uczenia się pojęć (ang. *[concept-learning](http://www.ci.pwr.wroc.pl/~kwasnick/tekstystudenckie/apw/ind_rodzaje.htm)*):

1. Zainicjuj *h* na najbardziej specyficzną hipotezę w *H*
2. Dla każdego pozytywnego przypadku x

Dla każdego atrybutu ograniczeń *ai* w *h*

* + jeśli ograniczenie *ai* w *h* jest spełnione w x - nie rób nic
  + w przeciwnym wypadku zamień *ai* w *h* przez następne ogólne ograniczenie, które jest spełnione w x

1. Otrzymana hipoteza *h* jest wynikiem algorytmu

**Cechy algorytmu:**

* Generuje jedną hipotezę *h*
* Ignoruje każdy negatywny przykład
* Nie musi pamiętać poprzednich przykładów, a jednak każda kolejna postać *h* pokrywa wszystkie przykłady pokrywane przez poprzednie postaci rozwiązania

**Wady algorytmu:**

* Nie stwierdza, czy nauczył się pojęcia
* Nie wykrywa błędnych (sprzecznych) danych wejściowych (np. dwa przykłady o identycznym wartościowaniu atrybutów, ale z różną oceną)
* Generuje jedną najbardziej specyficzną hipotezę *h*
* Znaleziona hipoteza *h* jest spójna (zgodna) także z niektórymi przykładami negatywnymi (czyli inaczej mówiąc znaleziona hipoteza może pokrywać także niektóre przykłady negatywne)
* Znalezione rozwiązanie nie jest unikalne
* Nie informuje, że język H nie wystarcza, by się nauczyć pojęcia
* Brak odporności na dane zaszumione

### Algorytm eliminacji kandydatów (Candidate Elimination - CAE)

**Pojęcia wstępne**

Najbardziej ogólna (ang. *general*) hipoteza:   h = < ?,?,?,...,? > - hipoteza, która pokrywa wszystkie przykłady  
Najbardziej specyficzna (ang. *specific*) hipoteza:   h = < Ø,Ø,Ø,...,Ø > - hipoteza, która nie pokrywa żadnego przykładu  
Generalizacja - przekształcenie hipotezy poszerzające ilość przykładów przez nią pokrywanych  
Specjalizacja - przekształcenie hipotezy zawężające ilość przykładów przez nią pokrywanych

**Treść algorytmu Eliminacji Kandydatów**Inicjuj G i S    (G = < ?,?,?,...,? >, S = < Ø,Ø,Ø,...,Ø >)Dla każdego przykładu d wykonaj

Jeśli d jest pozytywnym przykładem

Usuń z G każdą hipotezę niepokrywającą d  
Dla każdej hipotezy s z S, która nie pokrywa d

* Usuń s z S
* Dodaj do S wszystkie minimalne uogólnienia h hipotezy s, takie że h pokrywa d i choć jedna z hipotez z G jest bardziej ogólna niż h
* Usuń z S każdą hipotezę, która jest bardziej ogólna od innych z S

Jeśli d jest negatywnym przykładem

Usuń z S każdą hipotezę pokrywającą d  
Dla każdej hipotezy g z G, która pokrywa d

* Usuń g z G
* Dodaj do G wszystkie minimalne specjalizacje h hipotezy g, takie że h nie pokrywa d i choć jedna z hipotez z S jest bardziej specyficzna niż h
* Usuń z G każdą hipotezę, która jest mniej ogólna od innych z G

Znalezione zbiory S i G określają [przestrzeń wersji](http://www.ci.pwr.wroc.pl/~kwasnick/tekstystudenckie/apw/slownik.htm) znalezioną przez algorytm: G stanowi granicę ogólną, a S granicę specyficzną. Każda hipoteza ze znalezionej przestrzeni wersji jest bardziej szczegółowa niż pewna hipoteza z G i jest bardziej ogólna niż pewna hipoteza z S. Formalnie opisuje to twierdzenie reprezentacji przestrzeni wersji:

**Podsumowanie**

Algorytm jest zbieżny do hipotez prawidłowo opisujących pojęcie, jeśli

* nie ma błędów w ciągu uczącym,
* H zawiera hipotezy prawidłowo opisujące docelowe pojęcie

Wady algorytmu:

* nowy przykład (spoza przykładów trenujących) jest klasyfikowany tylko wtedy gdy jest on zgodny z wszystkimi hipotezami z przestrzeni wersji
* istnieje obciążenie, by docelowe pojęcie zawarte było w przestrzeni hipotez
* mała odporność na zaszumione dane

**26. Uczenie ze wzmocnieniem.**

Uczenie się ze wzmocnieniem (reinforcement learning, RL) jest pod wieloma względami odmienne od innych form maszynowego uczenia się. W przeciwieństwie do klasyfikacji i regresji, jego celem nie jest aproksymowanie pewnego nieznanego odwzorowania przez generalizację na podstawie zbioru przykładów trenujących, chociaż wewnątrz systemów uczących się ze wzmocnieniem możemy łatwo odkryć wykorzystanie aproksymatorów. Jednak systemowi uczącemu się ze wzmocnieniem nie są dostarczane żadne przykłady trenujące, a jedynie wartościująca informacja trenująca, oceniające jego dotychczasową skuteczność.

U podstaw uczenia się ze wzmocnieniem leżą dynamiczne interakcje ucznia ze środowiskiem, w którym działa, realizując swoje zadanie. Interakcje te odbywają się dyskretnych (na ogół) krokach czasu i polegają na obserwowaniu przez ucznia kolejnych *stanów* środowiska oraz wykonywaniu wybranych zgodnie z jego obecną *strategią*decyzyjną *akcji*. Po wykonaniu akcji uczeń otrzymuje rzeczywistoliczbowe wartości *wzmocnienia* lub *nagrody*, które stanowią pewną miarę oceny jakości jego działania. Wykonanie akcji może również powodować zmianę stanu środowiska.

W każdym kroku czasu :

1. obserwuj aktualny stan ;
2. wybierz akcję  do wykonania w stanie ;
3. wykonaj akcję ;
4. obserwuj wzmocnienie  i następny stan ;
5. ucz się na podstawie doświadczenia .

Środowisko pod wpływem wykonywanych przez ucznia akcji może zmieniać stany oraz dostarczać nagrody, stanowiące ocenę skuteczności działania ucznia. W uczeniu się ze wzmocnieniem dopuszcza się *niepewność* środowiska i zakłada się jego*nieznajomość* przez ucznia. Pierwsze oznacza, że generowane pod wpływem wykonywanych akcji wzmocnienia i zmiany stanów mogą być stochastyczne. Drugie oznacza, że leżące u podstaw tych stochastycznych mechanizmów rozkłady prawdopodobieństwa nie są znane uczniowi. Ponadto środowisko jest *niekontrolowalne*: uczeń nie ma na te rozkłady prawdopodobieństwa żadnego wpływu. To ostatnie założenie ma decydujące znaczenie na wytyczenie granicy między uczniem a środowiskiem: uczeń ma wpływ na swoje własne mechanizmy działania, parametry itp., lecz nie ma wpływu na środowisko.

Cel działania ucznia jest pośrednio określony przez wartości wzmocnienia. W najbardziej ogólnym przypadku możemy powiedzieć, że od ucznia oczekuje się nauczenia się strategii (czyli odwzorowania stanów na akcje do wykonania w tych stanach), która maksymalizuje pewne kryterium jakości zdefiniowane za pomocą otrzymywanych przez niego nagród. Rodzaj tego kryterium decyduje o konkretnym typie uczenia się ze wzmocnieniem.

Najciekawszy i najczęściej rozważany jest przypadek, kiedy uczeń ma maksymalizować swoje nagrody *długoterminowo*: dobra strategia niekoniecznie przynosi natychmiast wysokie nagrody, lecz jest opłacalna w dłuższym horyzoncie czasowym. Ten typ uczenia się ze wzmocnieniem wymaga uwzględnienia przez ucznia opóźnionych skutków wykonywanych przez niego akcji i określany jest mianem*uczenia się z opóźnionym wzmocnieniem* lub *uczenia się na podstawie opóźnionych nagród*. Stosowane wówczas algorytmy uczenia się rozwiązują tzw. problem *temporalnego przypisania zasługi*(*temporal credit assignment*), polegający na przypisaniu zasługi (bądź winy) za długoterminowe dochody ucznia jego poszczególnym akcjom, być może wykonanym wiele kroków przed faktycznym uzyskaniem tych dochodów.

Jako dokładne kryterium jakości działania ucznia często przyjmuje się oczekiwaną zdyskontowaną sumy otrzymanych nagród. Uczeń rozpoczynający działalność w czasie  ma za zadanie maksymalizowanie sumy:

|  |  |
| --- | --- |
|  | *(1)* |

gdzie *współczynnik dyskontowania*  reguluje względną ważność krótko- i długoterminowych nagród.

## Procesy decyzyjne Markowa

Modelem matematycznym problemu uczenia się ze wzmocnieniem (a właściwie modelem środowiska) jest proces decyzyjny Markowa (*Markov decision process*, MDP), który definiujemy jako czwórkę:

|  |  |
| --- | --- |
|  | *(4)* |

gdzie

* jest skończonym zbiorem stanów,
* jest skończonym zbiorem akcji,
* jest funkcją nagrody (wzmocnienia),
* jest funkcją przejścia stanów.

Dla każdej pary  wartość  jest zmienną losową oznaczającą nagrodę otrzymywaną po wykonaniu akcji  w stanie , a wartość  jest zmienną losową oznaczającą następny stan po wykonaniu akcji  w stanie . Oznacza to, wobec wprowadzonej wcześniej notacji, że  jest realizacją zmiennej losowej  oraz  jest realizacją zmiennej losowej .

Dla ułatwienia posługiwania się funkcjami przejścia i wzmocnienia wygodnie jest zdefiniować dodatkowe oznaczenia:

|  |  |  |
| --- | --- | --- |
|  |  | *(5)* |
|  |  | *(6)* |

### Własność Markowa

Kluczowe znaczenie ma tzw. *własność Markowa*:  i  nie zależą od historii. W każdym kroku nagroda i następny stan zależą (probabilistycznie) tylko od aktualnego stanu i akcji. Własność ta jest teoretycznie warunkiem stosowalności wszystkich algorytmów, o których będziemy dalej mówić, chociaż w praktyce często można spotkać środowiska niemarkowowskie. Jeśli własność Markowa nie zachodzi, uczeń nie ma dostępu do informacji umożliwiającej wybór najlepszej akcji - mówi się wówczas o *ukrytym stanie* albo wręcz nazywa obserwacje ucznia *sytuacjami*, rezerwując termin stan dla*prawdziwych* (nieznanych) stanów środowiska, które zapewniałyby własność Markowa.

### Programowanie dynamiczne a uczenie się ze wzmocnieniem

Uczenie się ze wzmocnieniem można porównać z programowaniem dynamicznym w następujący sposób:

* programowanie dynamiczne wymaga znajomości  dla wszystkich  i  oraz  dla wszystkich  i , podczas gdy uczenie się ze wzmocnieniem nie zakłada, że wielkości te są znane, i wykorzystuje faktycznie zaobserwowane nagrody i przejścia stanów,
* programowanie dynamiczne opiera się na wyczerpującym przeglądaniu całej przestrzeni stanów i akcji, podczas uczenie się ze wzmocnieniem wykorzystuje faktyczne trajektorie,
* programowanie dynamiczne prowadzi do obliczenia pełnej strategii optymalnej, podczas gdy uczenie się ze wzmocnieniem ma w gruncie rzeczy na celu działanie (w przybliżeniu) optymalne, które może być oparte na częściowej strategii (nie jest konieczne nauczenie się optymalnej strategii dla stanów, które nie występują w trakcie faktycznego działania ucznia).

## Stosowanie uczenia się ze wzmocnieniem

Nieco upraszczając sytuację, można sformułować następujące równanie:

które pokazuje, że konstruktor systemu uczącego się ze wzmocnieniem musi przede wszystkim określić odpowiednią reprezentację stanów i zbiór akcji oraz zaprojektować funkcję wzmocnienia, która dobrze określa stawiany projektowanemu systemowi cel. Chociaż istotnych decyzji do podjęcia jest znacznie więcej, te są najbardziej podstawowe. Reprezentacja stanów powinna zapewniać dostarczanie systemowi informacji potrzebnych do podejmowania optymalnych decyzji (zachowanie własności Markowa). Jeśli nie jest to możliwe, można rozważyć różne algorytmy uczenia się ze wzmocnieniem w środowiskach niemarkowowskich, których tu nie omawiamy (są dość nowe i tylko częściowo satysfakcjonujące). Akcje powinny być określane na odpowiednim poziomie abstrakcji: na tyle niskim, aby możliwe było ich bezpośrednie wykonywanie przez system, i na tyle wysokim, aby czas potrzebny do uzyskania za ich pomocą pożądanych celów nie był zbyt długi. Funkcja wzmocnienia oraz współczynnik dyskontowania muszą być tak dobrane, aby maksymalizacja zdyskontowanej sumy nagród była osiągana przez strategie realizujące cel, dla którego jest konstruowany system.

Paradygmat uczenia się ze wzmocnieniem jest prosty i abstrakcyjny na tyle, by być potencjalnie bardzo szeroko stosowalnym. Zakres tych zastosowań ograniczają wprawdzie, oprócz ludzkiej pomysłowości, ograniczenia obecnie znanych metod (związane zwłaszcza z niezadowalającą szybkością uczenia się), te jednak będą prawdopodobnie z czasem pokonywane. Dziedziny, w których możliwości zastosowań uczenia się ze wzmocnieniem wydają się w świetle dotychczasowych prac najbardziej obiecujące, to przede wszystkim:

* inteligentne sterowanie optymalne,
* uczące się roboty,
* gry planszowe,
* optymalizacja kombinatoryczna i szeregowanie.

Do najbardziej spektakularnych przykładów należy użycie uczenia się ze wzmocnieniem w połączeniu z reprezentacją funkcji wartości za pomocą sieci neuronowej do gry w *trik-traka* (*backgammon*): uzyskany w ten sposób program na podstawie własnej gry (ze sobą) doszedł do mistrzostwa (należy do kilku najlepszych graczy na świecie).

**31. Podstawy formalnego opisu języka naturalnego: założenia i stosowane metody.**

* Lingwistyka informatyczna — Computational Linguistics (CL) (lingwistyka komputerowa)
  + nauka o języku naturalnym: struktura, znaczenia i jego użycie w procesie komunikacji
  + stosowanie metod formalnych do opisu języka
  + formułowanie teorii, hipotez i ich obiektywna weryfikacja
    - konstrukcja teorii implementowalnych
  + ostatnio: budowa programów analizujących język w oparciu o sformalizowaną wiedzę lingwistyczną
  + coraz mniej tej wiedzy jest wymagane (→ lingwistyka formalna)
* Metody formalne w obrębie lingwistyki
  + gramatyki formalne w opisie składni
  + semantyka formalna języka naturalnego
* Poziomy opisu języka naturalnego
  + Fonologiczny (fonetyka)
    - segmentacja sygnału mowy
    - relacja pomiędzy formą graficzną a dźwiękową
  + Morfologiczny / morfo-syntaktyczny
    - segmentacja wyrazów ortograficznych
    - relacja pomiędzy formą a związkami składniowymi
    - relacja pomiędzy formą z znaczeniem
    - słowotwórstwo (derywacje)
  + Syntaktyczny : struktury wyrażenia językowego; abstrahujący od znaczenia
  + Semantyczny, w zakresie leksykalnym , zdań , struktury dyskursu / tekstu
  + Pragmatyczny : użycie i kontekst interpretacji , pozajęzykowe relacje i skutki
* Formalizacja w opisie języka naturalnego
  + morfologia: gramatyki formalne poziomu znaków i morfemów
  + składnia: gramatyki formalne
  + semantyka:
    - opis znaczeń leksykalnych (słów) za pomocą grafu relacji
    - logika jako podstawa reprezentacji znaczenia
  + pragmatyka
    - modele oparte na logice i formalnej reprezentacji wiedzy
    - modele wieloagentowe
* Metody analizy głębokiej - schemat
  + formalizm gramatyki
  + opracowanie gramatyki
  + konstrukcja parsera
    - kwestia algorytmu parsowania
    - problem wyboru właściwej analizy
    - problem efektywności
* Gramatyki formalne
  + inspirowane wczesnymi pracami Noama Chomskyego, np. gramatyka generatywno-transformacyjna Chomsky’ego
  + teoria Lexical-Functional Grammar (LFG) Joan Bresnan
  + Head-driven Phrase Structure Grammar (HPSG) Carl Pollard i Ivan Sag
  + gramatyka DCG (metamorficzna):
    - oparta na Prologu,
    - dla języka polskiego gramatyki Szpakowicza (1978) oraz Świdzińskiego (1992)

Tradycyjne gramatyki opisowe z wielu powodów nie spełniają warunku rozstrzygalności postulowanego powyżej. Po pierwsze, gramatyki takie zwykle opisują tylko pewne konstrukcje danego języka, nie opisując warstwy leksykalnej tego języka; nie zawierają one słownika. Po drugie, gramatyki takie zwykle są bardzo nieścisłe, często tylko ilustrują pewne zjawiska lingwistyczne zamiast podawać warunki występowania tych zjawisk. Po trzecie, i to jest problem znacznie poważniejszy od dwóch poprzednich, gramatyki opisowe, opisujące język naturalny, same sformułowane są w języku naturalnym, w którym nie tylko poszczególne wyrazy są wieloznaczne, ale też poszczególne konstrukcje językowe mogą być interpretowane na wiele sposobów. Aby zatem gramatyka mogła posiadać cechę rozstrzygalności, musi ona być sformułowana w sposób precyzyjny i jednoznaczny, w pewnym jednoznacznie interpretowanym języku formalnym, na przykład w języku logiki predykatów. Taką gramatykę nazywać będziemy **gramatyką formalną**.

Język naturalny cechuje duża swoboda konstruowania zdań (brak ścisłych reguł gramatycznych) oraz duża liczba wyjątków. Języki formalne natomiast charakteryzują się ścisłym i jednoznacznym opisem konstrukcji. **Gramatyki naturalne** pozwalają określić zbiory reguł

budowy zdań, natomiast **gramatyki formalne** zajmują się pojęciami abstrakcyjnymi, powstającymi

drogą uogólnienia grupy wyrazów i stanowią sposób badania i opisu zbioru reguł. W ramach języków formalnych istotne są następujące definicje:

* Alfabet V = {v0 , ... , vn } – dowolny skończony i niepusty zbiór symboli (liter), z których będą zestawiane słowa języka
* Słowo nad alfabetem V – wszystkie skończone, uporządkowane ciągi złożone z elementów alfabetu.
* Słowo puste ε – słowo nie mające żadnej litery
* Język uniwersalny V\*– zbiór wszystkich słów nad alfabetem V
* Język – dowolny podzbiór języka uniwersalnego V\* nad V
* Syntaktyka (składnia) języka – reguły budowy zdań w języku
* Semantyka języka – interpretacja tych reguł, zasady stosowania składni

Wyróżniamy następujące typy gramatyk formalnych:

* Gramatyka rozpoznająca – dla dowolnego rozpatrywanego słowa potrafi rozstrzygnąć, czy słowo to jest poprawne czy nie, a w przypadku odpowiedzi pozytywnej potrafi podać wskazówki dotyczące budowy tego słowa
* Gramatyka generacyjna – potrafi zbudować dowolne słowo poprawne, podając przy tym wskazówki dotyczące jego budowy oraz nie tworzy ani jednego słowa niepoprawnego
* Gramatyka przetwarzająca – dla dowolnie poprawnie zbudowanego słowa potrafi ona zbudować jego odwzorowania również w postaci słowa poprawnego, określając przy tym wskazówki dotyczące kolejności stosowania odwzorowań

Gramatyka Noama Chomsky'ego :

* Formalnie gramatykę określamy jako: G = < V, T, P, S > , gdzie:
  + V – zbiór symboli terminalnych – skończony, niepusty zbiór symboli końcowych, z których budowane są słowa generowane przez gramatykę (zwany czasem alfabetem końcowym)
  + T – zbiór symboli nieterminalnych – skończony, niepusty zbiór symboli pomocniczych
  + P – lista produkcji – reguły gramatyki
  + S – symbol startowy (aksjomat) – jest to wyróżniony symbol pomocniczy, z niego wyprowadzane są wszystkie generowane przez gramatyk G napisy.
* Produkcja – możliwość zastąpienia symbolu pomocniczego przez słowo, w skład którego mogą
* wchodzić zarówno symbole pomocnicze, jak i końcowe.
* Proces wyprowadzania słów języka zdefiniowanego przez gramatykę odbywa się następująco:
  + zaczynamy od symbolu S
  + do poszczególnych symboli pomocniczych
  + stosujemy którąś z produkcji ze zbioru P tak długo, aż wszystkie symbole pomocnicze znikną.

Język generowany przez gramatykę:

* To, co zostanie, czyli słowo złożone z samych symboli końcowych, nazywamy słowem wyprowadzalnym z gramatyki.
* Zbiór wszystkich słów wyprowadzalnych z danej gramatyki nazywa się językiem generowanym przez gramatykę.
* Język generowany przez gramatykę G jest to zbiór wszystkich możliwych słów powstałych na bazie listy produkcji i wyprowadzonych z symbolu startowego języka

Automaty skończone (FSA):

* Automat skończony (ang. Finite-State Automaton) to abstrakcja matematyczna, określa język formalny
* Składa się z skończonej ilości stanów i funkcji przejścia, która określa sposób przechodzenia od jednego stanu w inny
* Automat wczytuje pojedyncze symbole z taśmy wejściowej i w zależności od aktualnego stanu i wczytanego symbolu przechodzi w inny stan
* Automat jako akceptor – akceptuje wczytywany ciąg symboli, gdy należy do języka określonego przez dany automat, odrzuci w przypadku przeciwnym

Wady i zalety modeli skończenie stanowych:

* Wady:
  + Braki pod względem ekspresywności w porównaniu z językami wyższych klas
  + Uważa się, że języki naturalne nie są regularne (ale pewne podzbiory są!)
  + Modele skończenie stanowe nie wnoszą wiele do wiedzy o językach naturalnych
* Zalety:
  + Matematycznie dogłębnie zbadane
  + Kompaktowa reprezentacja (Minimalizacja)
  + Dobre własności obliczeniowe (Determinizacja)
  + Prostota konstrukcji ze względu na własności zamknięcia

Maszyny skończone w przetwarzaniu języków naturalnych:

* Modelowanie słowników
* Przetwarzanie mowy
* Analiza morfologiczna
* Analiza składniowa wybranych podzbiorów języków naturalnych
* Modelowanie gramatyk skończenie stanowych przybliżonych do gramatyk o większych możliwościach generatywnych

Automaty skończone jako słowniki:

* Wspólne przedrostki dla każdego hasła (drzewo deterministyczne, drzewo Trie)
* Dodatkowo wspólne przyrostki po minimalizacji drzewa Trie
* Jeżeli informacje dołączone do haseł zostają umieszczone w stanach końcowych, minimalizacja nie jest pełna – wspólne przyrostki przy wspólnych informacjach

Rozszerzenie koncepcji: Transduktory (FST)

* Transduktor to automat skończony z wyjściem (odwzorowanie ciągów symboli)
* Transduktor dla każdego wczytanego symbolu z taśmy wejściowego dopisuje odpowiedni symbol do taśmy wyjściowej
* Interpretacja jako FSA z parami symboli jako etykiety
* Interpretacja jako podwójny FSA (FSA wejściowy, FSA wyjściowy)
* Transduktor jako akceptor i analizator ciągów symboli - sprawdza przynależność słowa do języka AS wejściowego i odwzorowuje ja na słowo języka AS wyjściowego

**32. Współczesna technologia językowa: narzędzia, zasoby językowe i ich zastosowania.**

* technologia językowa – Language Technology (LT) / Human Language Technology (HLT)
  + budowa zasobów i narzędzi językowych
* Schemat przetwarzania: zasoby i narzędzia językowe
  + Zasoby językowe
    - uporządkowane zasoby danych językowych, najczęściej anotowanych (opatrzonych metaopisem) lub bazy wiedzy opisujące język naturalny
    - prowadzone są prace nad standaryzacją zasobów językowych
    - przykłady anotowanych danych językowych
      * korpus tekstu opisanego morfologicznie i ujednoznacznionego
      * korpus tekstu z oznaczeniem sensów poszczególnych wystąpień wyrazów
    - przykłady baz wiedzy:
      * gramatyka formalna opisująca podzbiór języka
      * leksykon semantyczny zawierający sformalizowane opisy znaczeń słów
      * model językowy opisujący prawdopodobieństwo występowania ciągów wyrazów
  + Narzędzia językowe
    - specjalistyczne programy do przetwarzania danych językowych
    - przykłady:
      * analizatory morfologiczne, tagery, parsery
      * konwertery, programy tłumaczące
    - Infrastruktury językowe: lokalne i rozległe
* Narzędzia:
  + TaKIPI — Tager Korpusu IPI PAN
    - wersja rozwojowa TIPI (Tager IPI PAN)
    - dezambiguator morfo-syntaktyczny pracujący na zbiorze znaczników korpusu IPI PAN
    - dokładność ponad 93% (ponad 85% dla niejednoznacznych)
    - ujednoznacznia wyjście analizatora morfologicznego Morfeusz SIAT (Woliński, 2006)
    - zadanie: dla każdego wyrazu wybór jednej, właściwej analizy
    - podział zadania na warstwy : 1. klasy gramatyczne , 2. rodzaj i liczba , 3. przypadek
  + Disaster
    - rozpoznawane typy całostek : frazy rzeczownikowe , czasownikowe
    - System uczący się na podstawie ręcznie anotowanych korpusów
    - Zastosowania
      * Wzbogacenie opisu tekstu o jednostki wielosłowowe i wyrażenia
      * Wydobywanie informacji
      * Automatyczne odpowiadanie na pytania (Question Answering)
      * Automatyczne tworzenie zasobów lingwistycznych
  + Słowosieć 2.0 — znaczenie i zastosowania
    - Leksykalna sieć semantyczna
    - Standard de facto w reprezentacji znaczeń leksykalnych na potrzeby analizy znaczenia
    - Definiuje zbiór znaczeń — punkt odniesienia (podstawa do ujednoznaczniania znaczenia słów)
    - Wyszukiwanie informacji
    - Wydobywanie informacji
    - Systemy odpowiadające na pytania otwarte
    - Wydobywanie wiedzy z tekstu
    - Rozumienie języka naturalnego
    - Podstawa do budowy specjalizowanych zasobów
  + Lexicographer-Controlled Semi-automatic word sense Disambiguation (LexCSD)
    - Cel (Broda & Piasecki, 2009) : Metoda WSD, która wymaga minimalnego nadzoru, ale nie powiela wad znanych z podejść nienadzorowanych
    - Podejście zainspirowane przez metodę stosowaną przez leksykografów podczas ich pracy nad opracowywaniem haseł słownikowych
      * Key Word In Context (KWIC) – zbieranie sugestii z korpusu
      * Podział danych na grupy – wstępne wyznaczenie znaczeń
      * Analiza grup w poszukiwaniu elementów wspólnych dla zebranych przykładów – cechy opisowe i różnicujące
      * Sformułowanie haseł słownikowych
    - schemat :
      * Zbieranie przykładów
      * Grupowanie - grupowanie krótkich fragmentów tekstu (text snippets) zawierających

niejednoznaczne słowa , utworzenie zbioru próbek odpowiadającego określonemu znaczeniu leksykalnemu

* + - * Nadanie grupom etykiet - Wyrocznia (Oracle) nadaje etykiety najbardziej

reprezentatywnym przykładom dla każdej z grup , indukowane znaczenia mają etykiety zrozumiałe dla ludzi

* + - * Klasyfikacja
      * Przykłady użycia dla poszczególnych znaczeń
  + Korpus
    - Definicja (Przepiórkowski) : zbiór tekstów elektronicznych reprezentatywnych dla danego języka (dziedziny, żargonu, dialektu)
    - oznaczony (anotowany):
      * strukturalnie: określona segmentacja
      * lingwistycznie:
        + przynajmniej morfosyntaktycznie
        + niekiedy strukturalnie (składnia) — tzw. ‘banki drzew’, np. Penn Tree Bank,
        + rzadko semantycznie, np. typy argumentów, ich role
      * zawierający dodatkową informację morfo-syntaktyczną o słowach, strukturach składniowych, znaczeniu
    - znane korpusy: British National Corpus (BNC), Brown Corpus, CollinsWordbanksOnline, Czeski Korpus Narodowy, Korpus IPI PAN (pierwszy duży polski)
    - Rola korpusu w ujednoznacznianiu POS
      * źródło wzorców uczących
      * wzorzec testowy
  + analizatory morfologiczne (Morfeusz), konwertery, programy tłumaczące

**33. Proces przetwarzania języka naturalnego: typowe etapy, cele, stosowane metody.**

* Przetwarzanie języka naturalnego — Natural Language Processing (NLP)
  + dziedzina sztucznej inteligencji
  + konstrukcja programów analizujących/przetwarzających informację zapisaną w języku naturalnym
  + podejście inżynierskie, niekoniecznie uwzględniające wiedzę lingwistyczną o języku
* Typowy schemat procesu przetwarzania języka naturalnego
  + Rozpoznanie struktury dokumentu i wydobycie tekstu
  + Segmentacja (na tokeny, zdania oraz jednostki bardziej złożone )
    - podział tekstu na elementy składowe: jednostki / segmenty / tokeny
    - Typowe segmenty:
      * akapity (kwestia granic , akapit jako element struktury dyskursu — semantyczne rozumienie )w WWW częste ‘graficzne’ użycie akapitu
      * zdania (kwestia granic , luźna interpunkcja i ‘plakatowość’ stron WWW , haplologia kropki - kropka kończy jednocześnie skrót i zdanie , specjalne konwencje np. w bibliografii )
      * słowa — problem definicji
  + Analiza morfologiczna
    - Steming
      * określenie symbolu reprezentującego różne formy wyrazowe różniące się wartościami kategorii gramatycznych (np. przypadek, liczba, rodzaj itd.) ale też czasami klas gramatycznych, np. bezokolicznik, imiesłów itd.
      * często realizowany bez uwzględnienia wiedzy lingwistycznej
    - Lematyzacja
      * określenie formy wyrazowej reprezentującej zbiór form wyrazowych różniące się wartościami kategorii gramatycznych dla wyrazu
    - Specyfikacja
      * rzutowanie: wyraz → 〈wyraz, znacznik〉
      * rzutowanie a nie funkcja, tzn. mamy:
      * wyraz → 〈wyraz, znacznik1〉, ..., 〈wyraz, znacznikn〉
    - Algorytmy
      * oparte na paradygmatach odmiany
        + słownik tablic odmiany definiujących rdzenie (osnowy) i końcówki
        + rozpoznawanie wyrazu po końcówce i wiązanie z tablicami
      * oparte na transduktorze
        + indeks a tergo
        + formy wyrazowe zapamiętane w ścieżkach transduktora
        + opis i forma podstawowa zakodowane w języku wyjściowym transduktora
        + wada: podejście wybitnie pamięciowe – całość słownika w transduktorze
        + zalety: działanie w dwie strony – zarówno analiza jak i synteza
  + Ujednoznacznianie morfo-syntaktyczne (tagowanie)
    - ustalenie dla każdego wyrazu:
      * klasy gramatycznej (lub części mowy)
      * wartości kategorii gramatycznych
    - rozwiązuje problem homografów
    - nie rozwiązuje problemu polisemii
    - Kategorie gramatyczne
      * np. liczba, rodzaj, osoba, przypadek, aspekt, stopień, itd.
      * np. wartości kategorii rodzaj w Korpusie IPI PAN: ()ęski osobowy m1, męski zwierzęcy m2, męski rzeczowy m3, żeński f, nijaki , zbiorowy n1, nijaki zwykły n2, przymnogi osobowy p1, przymnogi zwykły p2, przymnogi opisowy p3 )
    - Znaczniki — ‘tagi’ (ang. tags)
      * symbole reprezentujące informację o części mowy i wartościach kategorii gramatycznych , przypisywane do wyrazów w tekście
    - Cel — różne punkty widzenia
      * klasyfikacja wyrazów tekście wg klas przypisanych do znaczników
        + wewnętrzna struktura klas w przypadku znaczników strukturalnych
      * konwersja wyników analizy morfologicznej do postaci funkcji
      * określenie jednoznacznego dla każdego wyrazu w tekście po analizie morfologicznej jednoznacznego przypisania znacznika
    - Podział algorytmów ujednoznaczniania (Dębowski, Gaweł)
      * metody statystyczne:
        + oparte na ukrytych modelach Markowa - wyznaczenie znacznika dla segmentu na podstawie kontekstowego prawdopodobieństwa wystąpienia danego znacznika dla danego segmentu
        + oparte na modelowaniu maksimum entropii - tworzony jest model nieznanego modelu probabilistycznego znaczników w tekście , następnie model jest przybliżany przez rozkłady prawdopodobieństwa generowane wg określonego schematu
      * metody regułowe (nieoparte jawnie na modelach probabilistycznych):
        + oparte na pamięci,
        + oparte na transformacjach
      * metody lingwistyczne (regułowe) - tworzone ręcznie na podstawie wiedzy eksperckiej
  + Ujednoznacznianie sensu słów (znaczeń leksykalnych)
    - Leksykalna sieć semantyczna
      * opisująca leksykalne relacje znaczeniowe pomiędzy poszczególnymi jednostkami leksykalnymi
        + jednostka leksykalna — jedno lub wielowyrazowy leksem, reprezentowany jako para 〈lemat, sens〉, np. zamek 1,
      * zalety: efektywność budowania, generalizacja
      * wady: opis nie wprost, ograniczona siła ekspresji
      * Opis częściowy
        + zidentyfikowane są jedynie relacje znaczeniowe
        + relacje wyrażają ograniczenia na znaczenie jednostek leksykalnych
      * Klasy relacji (Hirst, 2003)
        + identyczność (np. synonimia)
        + zawieranie się znaczeniowe (np. hiponimia)
        + część — całość (np. meronimia)
        + opozycja znaczeniowa (np. antonimia)
        + powiązanie
      * Struktura
        + synsety (synset - zbiór jednostek leksykalnych – prawie synonimów )
        + jednostki leksykalne
        + relacje pomiędzy synsetami
        + relacje pomiędzy jednostkami
      * Ontologia a sieć leksykalnych relacji semantycznych
        + Podobieństwo : znaczenie leksykalne jako wyznacznik kategorii , leksykalne relacje semantyczne jako analogii relacji między kategoriami
        + Natury

ontologia jest bytem formalnym, sieć wynika z obserwacji języka

ontologia: precyzyjna definicja relacji , sieć: niedookreśloność relacji, ciągła przestrzeń relacji znaczeniowych, np. rozmyte granice i podtypy meronimii

* + - * + Struktura

|  |  |
| --- | --- |
| Ontologia | Sieć |
| wymóg rozłącznych kategorii | nakładanie się znaczeń jednostek o wspólnym hiperonimie |
| precyzja definiowania | język potoczny a język naukowy np. lew jako synonim lwa afrykańskiego |
| kompletność opisu | struktura zależna od leksykalizacji |
| hiperonimia jako drzewo | luki na poziomie pojęć ogólnych |
| intencjonalny opis rzeczywistości | warunkowana użyciem języka |

* + - wprowadzenie
      * Nazwa angielska: Word Sense Disambiguation, w skrócie WSD
      * Cel: automatyczne przypisanie (najlepiej jednoznaczne) znaczenia do wystąpienia lematu w tekście , ujednoznacznianie znaczeń leksykalnych – wybór znaczenia właściwego dla określonego kontekstu
      * Założenia
        + Słowosieć jako zbiór znaczeń
        + znaczenia są zdefiniowane przez synsety
      * Problemy: definicja zbioru znaczeń , wiedza o konkretnym użyciu (kontekst i treść) , metoda klasyfikacji
      * Główne podejścia
        + nadzorowane – oparte na wzorcowym korpusie oznaczonym znaczeniami leksykalnymi (`sensami’)

Repozytorium znaczeń leksykalnych (sense inventory)

tezaurus lub słownik w formacie elektronicznym lub zasobu językowego

wordnet

Metody klasyfikacji

dowolny algorytm klasyfikacji ze znanych

problem w definicji i wydobyciu cech

* + - * + nienadzorowane – rożne realizacje
      * Problemy : niska dokładność , grupy lematów nie są jednoznaczne i proste do interpretacji
  + Płytki parsing (opcjonalnie )
    - metody powierzchniowego przetwarzania tekstów ukierunkowane na znajdowanie struktury częściowej, w które reprezentowane są jedynie wybrane zależności
    - Cel:
      * rozpoznanie wybranych zależności strukturalnych i semantycznych
      * relatywnie duża efektywności przetwarzania
      * dokładność wystarczająca do praktycznych zastosowań
    - Cechy (Przepiórkowski, 2008):
      * duża precyzja działania – rozpoznawane są tylko te struktury, które mogą być identyfikowane w miarę pewny sposób
      * najpierw są rozpoznawana konstrukcje, które można zidentyfikować z wysokim
      * prawdopodobieństwem , później konstrukcje trochę bardziej wątpliwe
      * niepodejmowanie trudnych decyzji
    - Zastosowania
      * Wzbogacenie opisu tekstu o jednostki wielosłowowe i wyrażenia
      * Wydobywanie informacji
      * Automatyczne odpowiadanie na pytania (Question Answering)
      * Automatyczne tworzenie zasobów lingwistycznych
  + Rozpoznawanie wyrażeń wielowyrazowych (w tym jednostek identyfikujących, np. nazw własnych )
  + Rozpoznawanie związków w tekście (np. anafory, koreferencji, relacji semantycznych, sytuacji )
  + Głęboki parsing
    - metody oparte na formalnych gramatykach pozwalających na znalezienie pełnej struktury składniowe
    - Metody analizy głębokiej - schemat
      * formalizm gramatyki
      * opracowanie gramatyki
      * konstrukcja parsera
        + kwestia algorytmu parsowania
        + problem wyboru właściwej analizy
        + problem efektywności
  + Głęboka analiza semantyczna
  + Analiza pragmatyczna (w tym struktury dyskursu)
* Realizacje procesu w zastosowaniach
  + Wydobywanie informacji (Information Extraction)
    - Analiza treści zadania
    - Wydobycie tekstu
    - Tokenizacja i segmentacja na zdania
    - Tagowanie morfo-syntaktyczne: analiza i ujednoznacznianie
    - Rozpoznawanie jednostek identyfikacyjnych (`bytów nazwanych’)
    - Płytki parsing
    - Rozpoznawanie powiązań koreferencyjnych
    - Rozpoznawanie relacji
    - Łączenie instancji relacji w opisy sytuacji (merging)
    - Konwersja opisów sytuacji do rekordów odpowiedzi
    - Uzupełnianie i przygotowanie formy wynikowej
    - Przekształcenie sformalizowanej informacji w wiedzę oraz ewentualne wnioskowanie (Text Mining)
  + Systemy odpowiedzi na pytania (Open Domain Question Answering)
    - Analiza pytania: klasyfikacja i rozpoznanie struktury
    - Wydobycie kluczowych elementów pytania i przygotowanie zapytania wyszukiwawczego
    - Wyszukanie dokumentów zawierających odpowiedzi
    - Identyfikacja kluczowych fragmentów tekstu
    - Analiza kluczowych fragmentów:
      * analiza morfo-syntaktyczna i płytki parsing
      * rozpoznanie jednostek identyfikacyjnych i całostek (chunks)
    - Określenie podobieństwa kluczowych fragmentów do kluczowych elementów pytania i założonego schematu odpowiedzi
    - Wydobycie elementów odpowiedzi i dopasowanie ich do schematu odpowiedzi
    - Prezentacja wyników wraz ze wskazaniem fragmentów dokumentów
  + Automatyczne tłumaczenie (Machine Translation)
    - Wydobycie tekstu połączone z zapamiętaniem struktury prezentacji tekstu
    - Segmentacja: tokenizacja oraz segmentacja na zdania i (opcjonalnie) pseudo-akapity
    - Analiza morfo-syntaktyczna
    - (Rzadko) Ujednoznacznianie sensu słów
    - Parsing – zwykle głęboki
    - Rozpoznawanie wyrażeń wielowyrazowych
    - Ograniczone rozpoznawanie anafory
    - Transfer struktur leksykalno-syntaktycznych do języka docelowego
    - Ujednoznacznienie tłumaczenia na poziomie leksykalnym
    - Porządkowanie struktur po stronie języka docelowego
    - Synteza form wyrazowych po stronie języka docelowego
    - Formatowanie tekstu wyjściowego zgodnie z zapamiętaną strukturą prezentacji