Politechnika Wrocławska

Ewolucyjna indukcja gramatyki bezkontekstowej

Olgierd UNOLD

olgierd.unold@pwr.wroc.pl

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Definicja

- Wnioskowanie gramatyczne (*grammar induction*, *automata induction*, *grammatical inference*, GI) (Gold 1967, Pinker 1979, Angluin i Smith 1983, Fu i Boot 1986) to uczenie gramatyk i języków na podstawie przykładowych danych*

*http://eurise.univ-st-etienne.fr/gi/

Zastosowania

- uczenie maszynowe (*machine learning*)
- syntaktyczne rozpoznawanie wzorców (*syntactic pattern recognition*)
- teoria automatów i języków formalnych (*automata and formal language theory*)
- lingwistyka obliczeniowa (*computational linguistics*)
- biologia obliczeniowa (*computational biology*)
- rozpoznawanie mowy (*speech recognition*)
- przetwarzanie języka naturalnego (*natural language processing*)
- drążenie danych (*data mining*)

Wyuczalność języków

- Identyfikacja w granicy - przyswajanie zbioru reguł gramatycznych na podstawie szczątkowych danych, przypadkowych elementów języka
- Gold (1970) pokazał, że wszystkie języki aż do języka kontekstowego mogą być identyfikowane w granicy na podstawie ich kompletnej prezentacji (*learning from informant*)
- Ale jeżeli zbiór uczący nie jest kompletny i zawiera tylko przykłady pozytywne (*learning from text*), to wtedy żaden zbiór języków składający się z języków skończonych i przynajmniej jednego języka nieskończonego nie może być identyfikowany w granicy

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Gramatyka bezkontekstowa

- Gramatyka bezkontekstowa (*context-free grammar*, CFG) *G = (N, T, P, S)* jest w postaci normalnej Chomsky'ego (PNC), jeśli każda produkcja ze zbioru *P* jest jednej z trzech postaci

$$P: S \to \varepsilon \mid A \to a \mid A \to BC$$

gdzie $A \in N, a \in T, B, C \in N \setminus \{S\}$

N – zbiór skończony zwany zbiorem symboli pomocniczych (nieterminalnych),

T – zbiór skończony zwanym zbiorem symboli końcowych (terminalnych),

P - jest relacją skończoną zwaną listą produkcji,

S - jest wyróżnionym symbolem pomocniczym zwanym symbolem początkowym.

Model indukcji

- Indukujemy $G = (N, T, P, S)$
- Mamy zbiór zdań uczących

$$R = R^+ \cup R^-,$$

$$R^+ = \{(r, +) \mid r \in N^*\},$$

$$R^- = \{(r, -) \mid r \notin N^*\},$$

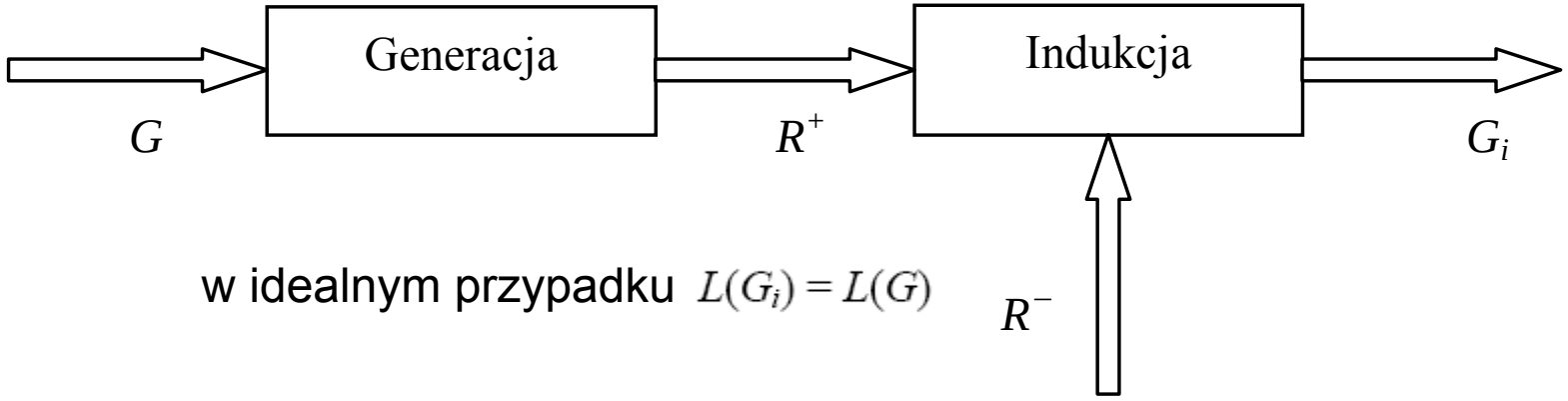
$$R^+ \cap R^- = \emptyset.$$

- Poszukujemy G_i

$$L(G_i) \cap R^+ = R^+,$$

$$L(G_i) \cap R^- = \emptyset,$$

$$L(G_i) \supseteq R^+.$$



w idealnym przypadku $L(G_i) = L(G)$

Problem przynależności

- Algorytm Cocke - Younger - Kasami (CYK) służy do sprawdzenia, czy słowo należy do języka generowanego przez *G*

- Niech

$$G = (\{A, B, C, S\}, \{a, b, c\}, \{S \to AB, S \to AC, C \to SB, C \to a, B \to BB, B \to b, A \to a\}, S)$$

		a	a	b	b
		$i \to$			
		1	2	3	4
	1	A, C	A, C	B	B
$j\downarrow$	2	S	S	B	
	3	C	C, S		
	4	C, S			

Wyuczalność

- Klasa jęz. bezkontekstowych jest identyfikowana w granicy na podstawie kompletnej prezentacji, ale nie są znane w tym ujęciu efektywne algorytmy wnioskowania
- Efektywne (wielomianowe) algorytmy uczące istnieją jedynie dla języków regularnych (de la Higuera 2000)

Podejścia do indukcji CFG

- Dodatkowa informacja strukturalna
- Alternatywne reprezentacje CFG
- Indukcja podklas CFG
- Indukcja gramatyk probabilistycznych
- Zastosowanie metod sztucznej inteligencji

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Definicja

- Uczący się system klasyfikujący (*learning classifier systems*, LCS) uczy się prostych syntaktycznie reguł (zwanych klasyfikatorami) w celu koordynacji swoich działań w dowolnym środowisku (Holland 1975)

  (nazywany również *systemem klasyfikującym* (Goldberg 1989, Michalewicz 1996, Cytowski 1996)

  lub *systemem klasyfikatorowym* (Cichosz 1997, Pawlak 1999)

Klasyfikator

- Każdy klasyfikator postaci

IF warunek THEN akcja

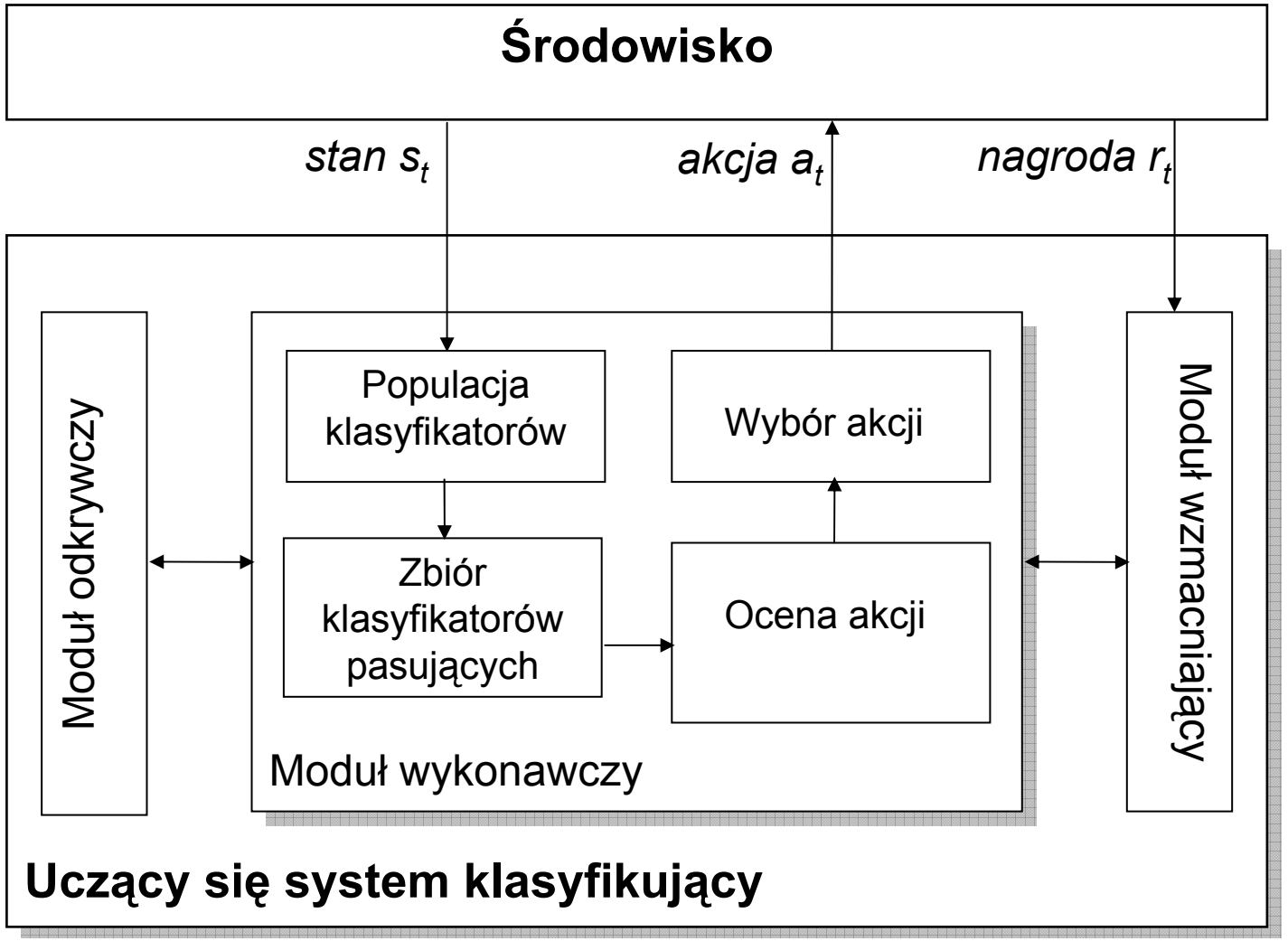
- przykład

0100	:	0010
11#0	:	0111

Klasyfikatory

- System produkcyjny jest uniwersalny obliczeniowo (Post 1943)
- Pojedynczy klasyfikator lub ich zbiór może w zwartej postaci reprezentować złożony zasób wiedzy
- LCS ogranicza postać klasyfikatorów, dzięki czemu można zastosować operatory genetyczne
- LCS uaktywnia reguły równolegle
- Klasyfikatory uczą się swojej wartości

Architektura



Kroki algorytmu

1. System otrzymuje na wejściu aktualny stan środowiska

State	if Condition	then	Action
1101	0100	:	0010
	11##	:	0111
	...		...

Kroki algorytmu

2. System buduje zbiór klasyfikatorów pasujących

State	if Condition	then	Action	Matched
1101	0100	:	0010	NO
	11##	:	0111	**YES**
	…		…	…

Kroki algorytmu

3. System ocenia użyteczność akcji w zbiorze klasyfikatorów pasujących; wybierana jest akcja i wysyłana do wykonania w środowisku

State	if Condition	then	Action	New action
1101	0100	:	0010	
	11##	:	0111	**0111**
	...		...	...

Kroki algorytmu

4. System otrzymuje nagrodę od środowiska; moduł wzmacniający dystrybuuje nagrodę pośród klasyfikatorów odpowiedzialnych za jej zdobycie

Kroki algorytmu

5. Moduł odkrywczy ewoluuje populację klasyfikatorów

Demo

- http://www.ai.tsi.lv/ga/lcs_maze_demo.html
- http://www.ai.tsi.lv/ga/lcs_nonmarkov_demo.html

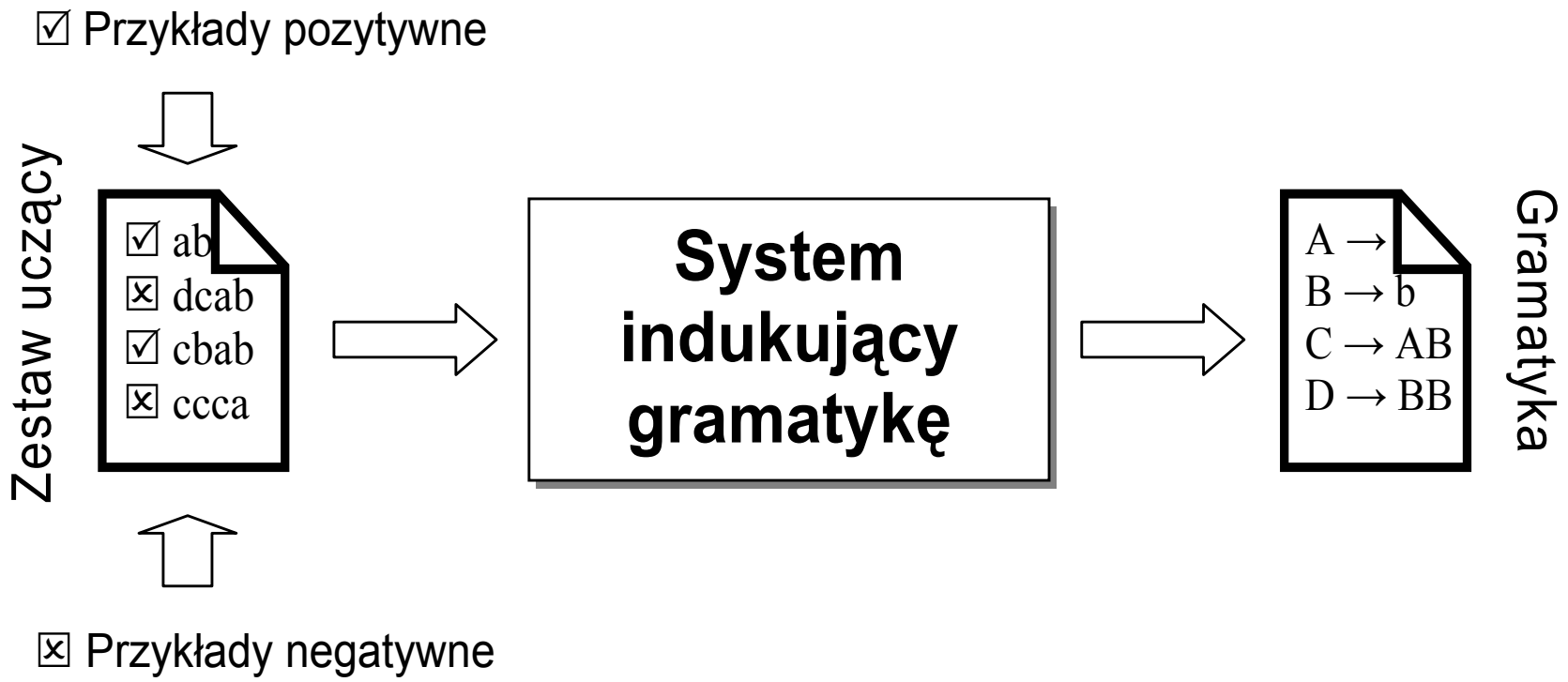
Wybrane modele

- XCS (*eXtended* CS)(Wilson 1995)
- EpiCS (CS *for Epidemiologic data*)(Holmes 1996)
- ACS (*Anticipatory* CS) (Stolzmann 1997)
- GCS (*Grammar-based* CS) (Unold 2005)

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowanie modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Cel



- Wejście: zestaw przykładowych zdań
- Wyjście: gramatyka bezkontekstowa w PNC

Szybki rzut oka

- Populację klasyfikatorów tworzą produkcje CFG w PNC (gramatyka = populacja)
- Każde zdanie ze zbioru uczącego jest parsowane algorytmem CYK
- Każda produkcja (klasyfikator) biorąca udział w parsowaniu otrzymuje miarę swojej użyteczności
- Model odkrywa nowe produkcje za pomocą algorytmu genetycznego oraz pokrycia
- Ocena ewoluowanej gramatyki (populacji) wykonywana jest po analize pełnego zestawu uczącego i uwzględnia liczbę zaakceptowanych zdań poprawnych i odrzuconych zdań niepoprawnych

Populacja klasyfikatorów

- Na populację klasyfikatorów składają się produkcje CFG w PNC:

  terminalne typu $A \rightarrow a$

  nieterminalne typu $B \rightarrow CD$

- Model ewoluuje jedną gramatykę zgodnie z podejściem Michigan (przedmiotem operacji genetycznych są pojedyncze klasyfikatory)

Klasyfikator

- $cl = \{P_L, P_P, f, u_p, u_n, p, d\}$

P_P – prawa strona produkcji, warunek (symbole typu *a* lub *AB)*

P_L - lewa strona produkcji, akcja (symbol typu *A*)

f – dopasowanie

u_p – liczba zastosowań produkcji przy parsowaniu zdań z R^+

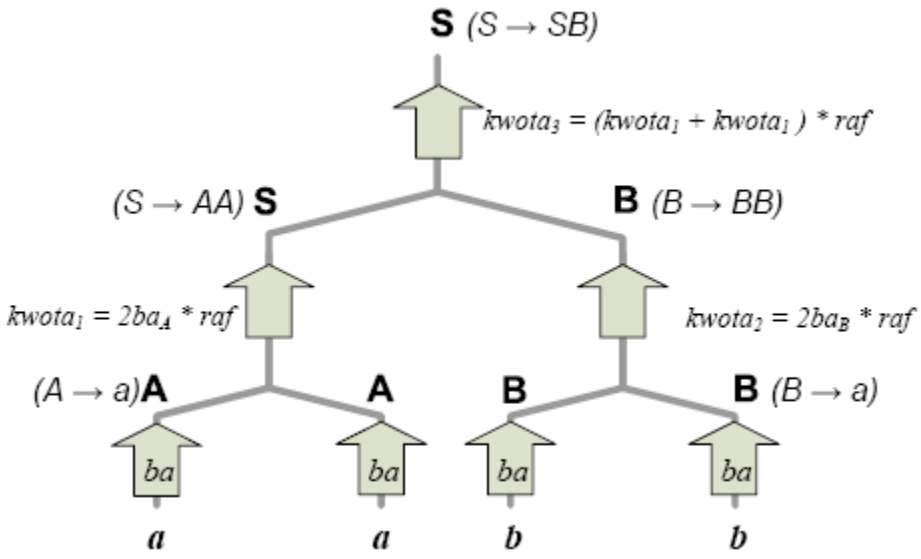
u_n – liczba zastosowań produkcji przy parsowaniu zdań z R^-

p – suma punktów zdobytych przy parsowaniu zdań z R^+

d - suma punktów zdobytych przy parsowaniu zdań z R^-

Płodność klasyfikatora

- W populacji występuje zjawisko silnej epistazy między klasyfikatorami (patrz: produkcja w ciągu wyprowadzeń, PNC)
- Ochrona produkcji stojących wyżej w ciągu wyprowadzeń, bardziej „płodnych”



Przystosowanie klasyfikatora

$$f = \frac{w_c f_c + w_f f_f}{w_c + w_f}$$

$$f_c = \begin{cases} \dfrac{w_p u_p}{w_n u_n + w_p u_p} & \text{dla} \quad u_n + u_p \neq 0 \\ f_0 & \text{dla} \quad u_n + u_p = 0 \end{cases}$$

$$f_f = \frac{p - d - f_{f\min}}{f_{f\max} - f_{f\min}}$$

$$f_{f\max} = \max_{cl \in G}(p - d)$$

$$f_{f\min} = \min_{cl \in G}(p - d)$$

f_0 – miara użyteczności klasyfikatora niebiorącego udziału w parsowaniu,
f_c – klasyczna funkcja przystosowania,
f_f – funkcja płodności klasyfikatora,
u_p – liczba zastosowań produkcji przy parsowaniu zdań poprawnych
u_n – liczba zastosowań produkcji przy parsowaniu zdań niepoprawnych
w_p – waga rozbioru zdania poprawnego
w_n – waga rozbioru zdania niepoprawnego
w_c – waga funkcji klasycznej,
w_f – waga funkcji płodności,
f_{fmax} – maksymalna liczba zdobytych punktów przez różnicę (p – d) w populacji klasyfikatorów,
f_{fmin} – minimalna liczba zdobytych punktów przez różnicę (p – d) w populacji klasyfikatorów.

Gramatyka

- $G_{GCS} = \{N, T, P_T, P_N, S, S_u\}$

N – zbiór symboli nieterminalnych

T – zbiór symboli terminalnych

P_T – produkcje terminalne

P_N – produkcje nieterminalne

S – wyróżniony symbol startowy

S_u – wyróżniony symbol uniwersalny

Przystosowanie gramatyki

$$f_G = \frac{U_p + NU_n}{|R|}$$

U_p – liczba sparsowanych zdań poprawnych

NU_n – liczba niesparsowanych zdań niepoprawnych

$|R|$ – liczba zdań w zestawie uczącym

Operatory pokrycia

- **Operator pokrycia terminalnego** - tworzenie klasyfikatorów przepisujących symbole terminalne. Tworzona jest produkcja typu $A \rightarrow a$ w sytuacji, gdy podczas rozbioru model napotyka nieznany wcześniej symbol terminalny
- **Operator pokrycia uniwersalnego** - tworzenie klasyfikatora pełniącego funkcję symbolu uniwersalnego (*don't care*). Dla każdego symbolu terminalnego *x* tworzona jest dodatkowa produkcja typu $S_u \rightarrow x$, gdzie *x* jest terminalem
- **Operator pokrycia startowego** - tworzenie symbolu startowego dla zdań o długości 1. Dla zdań pozytywnych $r \in R^+$ o długości 1 tworzona jest produkcja $S \rightarrow a$
- **Operator pokrycia pełnego** - dla zdań pozytywnych tworzona jest produkcja typu $S \rightarrow AB$ w sytuacji, gdy w populacji istnieją już wyprowadzenia symboli *A* i *B* oraz analizowana jest komórka [1, *n*] tablicy CYK.
- **Operator pokrycia agresywnego** - dla zdań pozytywnych tworzona jest produkcja typu $C \rightarrow AB$ w sytuacji, gdy w populacji istnieją już wyprowadzenia symboli *A* i *B*

Algorytm genetyczny

- Dokonaj selekcji dwóch produkcji nieterminalnych z populacji
- Stwórz kopie wyselekcjonowanych produkcji
  - Zastosuj *krzyżowanie* na kopiach produkcji
  - Zastosuj *mutację* na kopiach produkcji
  - Zastosuj *inwersję* na kopiach produkcji
- Dodaj *ze ściskiem* kopie produkcji do populacji, gwarantując przeżycie n_{elit} najlepszych osobników

Krzyżowanie

- Dwie produkcje nieterminalne wymieniają swoje lewe strony oraz symbole z prawej strony produkcji na losowo wybranej pozycji

$$A \rightarrow BC$$
$$D \rightarrow EF$$
$$\Downarrow$$
$$D \rightarrow EC$$
$$A \rightarrow BF$$

Mutacja

- Operator mutacji przechodzi przez kolejne pozycje produkcji typu $A \rightarrow BC$ i z prawdopodobieństwem p_a może zmienić symbol z danej pozycji produkcji na losowo wybrany ze zbioru nieterminali N

$$A \rightarrow BC$$
$$\Downarrow$$
$$G \rightarrow BD$$

Inwersja

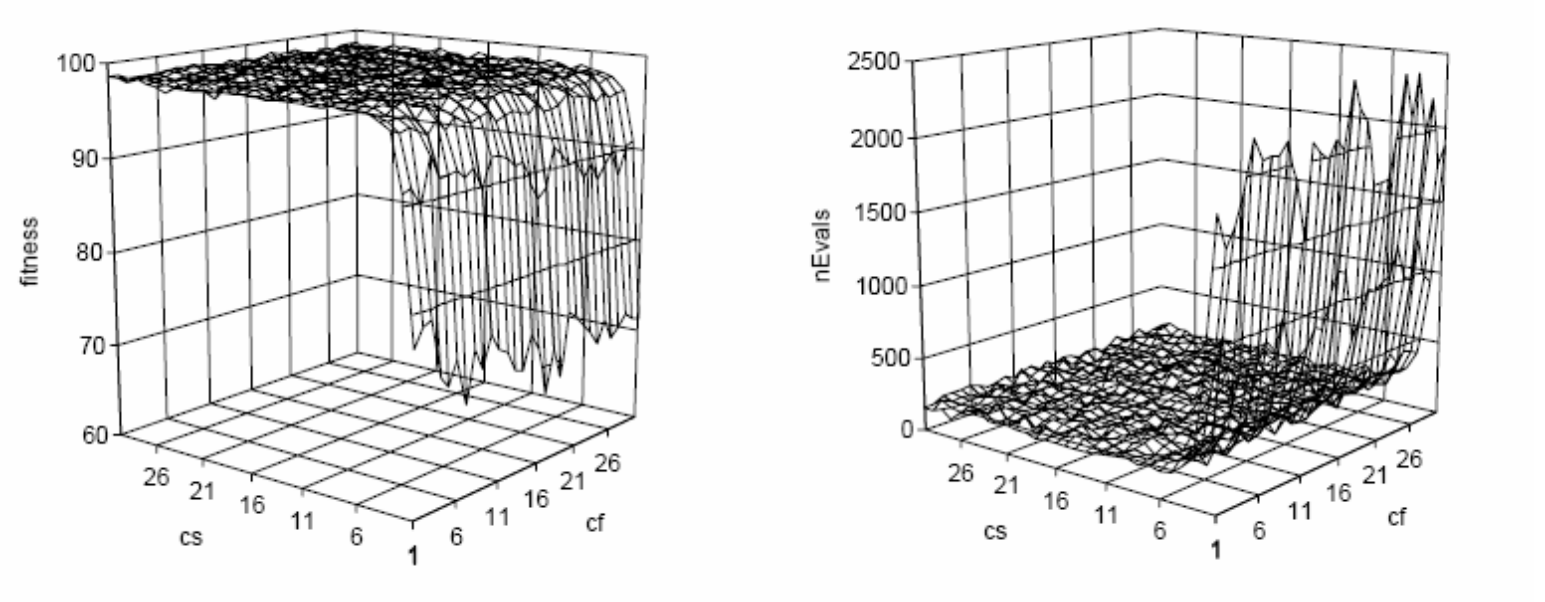
- Permutacji podlegają dwa symbole prawej strony produkcji

$$A \rightarrow BC$$

$$\Downarrow$$

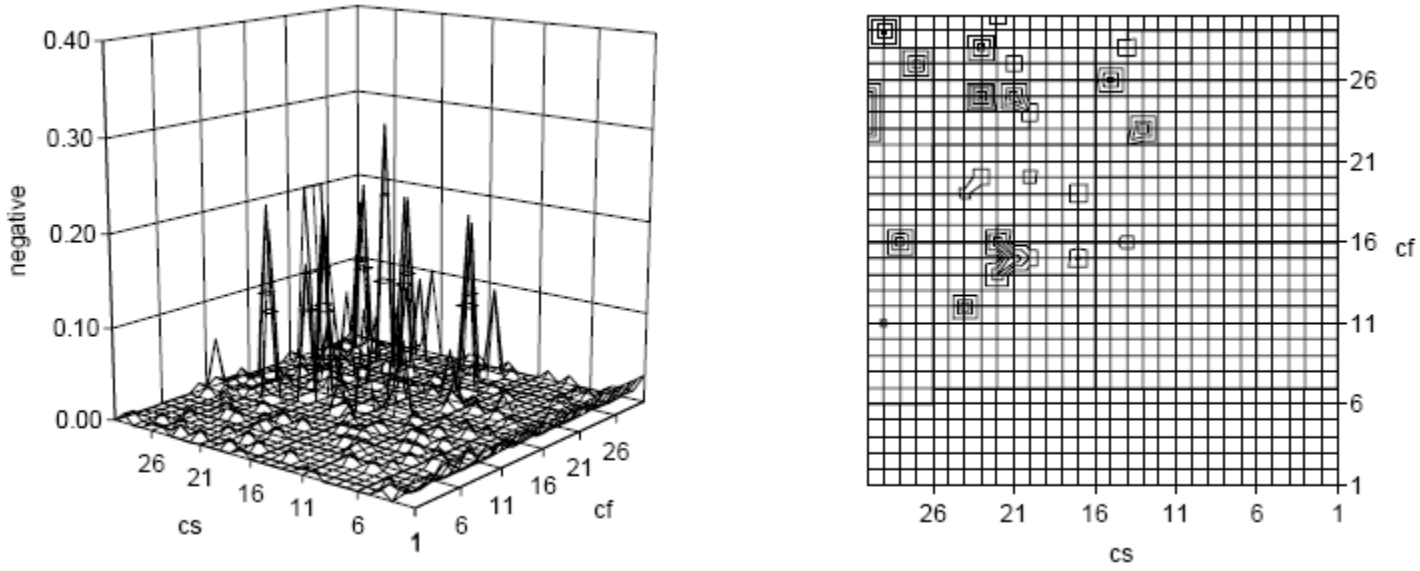
$$A \rightarrow CB$$

Wybrane badania symulacyjne



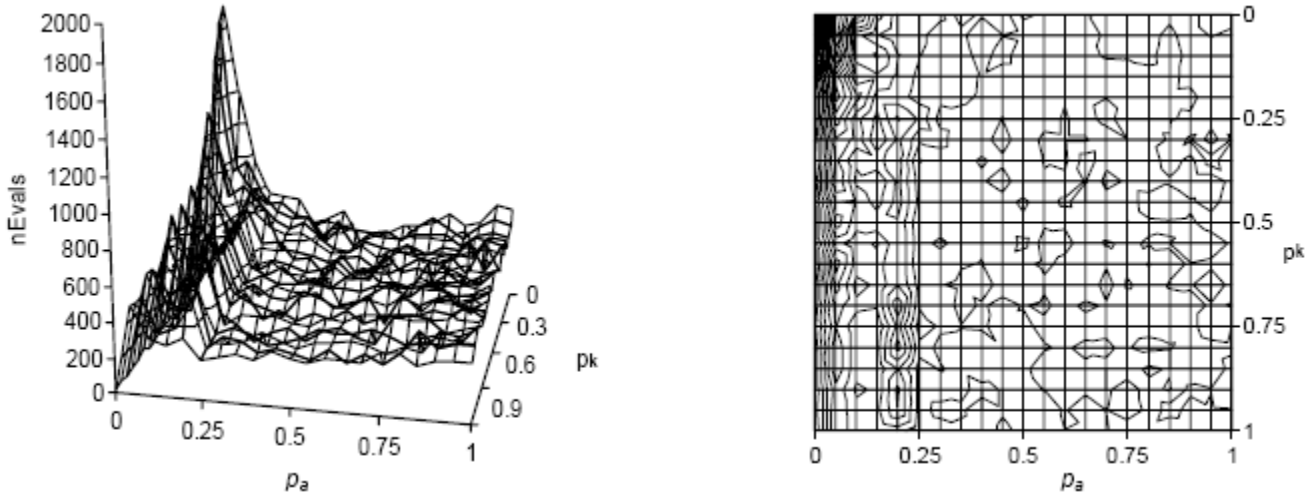
Wykres przestrzenny zmiany dokładności indukcji $fitness_{avg}$ w zależności od wartości współczynnika ścisku *cf* oraz wielkości podpopulacji *cs*

Wybrane badania symulacyjne



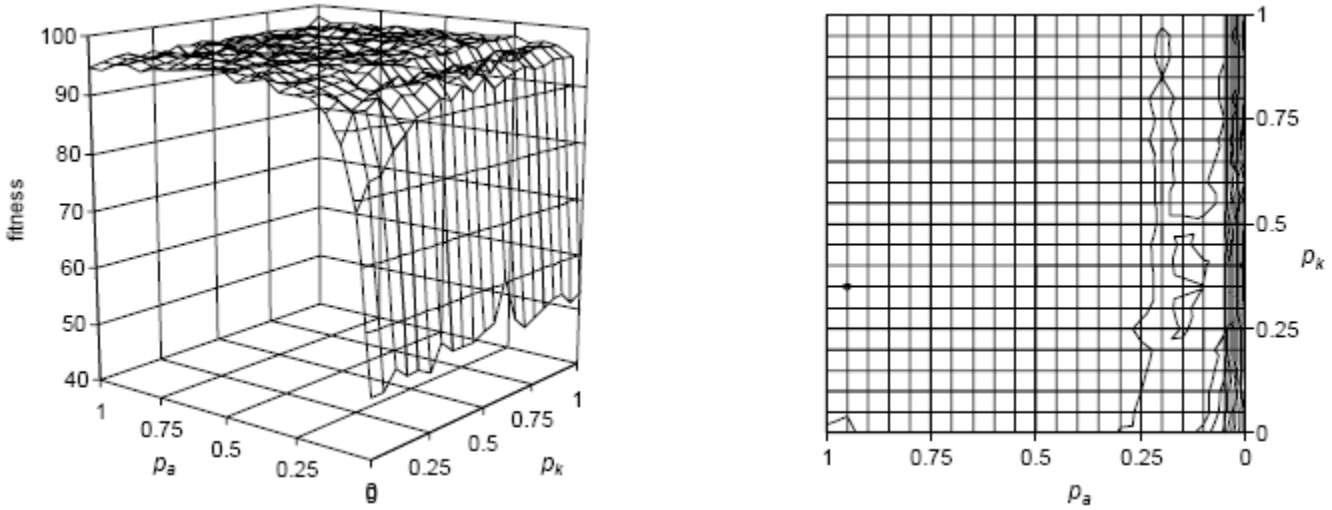
Zmiana kompetencji negatywnej $negative_{avg}$ w zależności od wartości współczynnika ścisku *cf* oraz wielkości podpopulacji *cs*: a) wykres przestrzenny, b) wykres poziomicowy (poziomica co 0,05%)

Wybrane badania symulacyjne



Zmiana kosztu indukcji *nEvals* w zależności od prawdopodobieństwa krzyżowania p_k oraz prawdopodobieństwa mutacji p_a: a) wykres przestrzenny, b) wykres poziomicowy (poziomica co 100 kroków)

Wybrane badania symulacyjne



Zmiana kompetencji ogólnej $fitness_{avg}$ w zależności od prawdopodobieństwa krzyżowania p_k oraz prawdopodobieństwa mutacji p_a: a) wykres przestrzenny, b) wykres poziomicowy (poziomica co 5%)

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Zastosowania modelu GCS

- Inżynieria lingwistyczna
- Genomika obliczeniowa
- Data mining

Inżynieria lingwistyczna

- Brak powszechnie uznanych, przyjętych i stosowanych zbiorów uczących i testowych
- Indukowano gramatyki formalne niewychodzące poza klasę CFG:
  - wyrażenia regularne, należące do tzw. zbioru Tomity,
  - wybrane języki bezkontekstowe,
  - gramatykę dziecięcą (*toy-grammar*)
- oraz korpusy językowe

Parametry modelu GCS

Zmienne decyzyjne:

f_{GA} = tak – zmienna zezwalająca na uruchomienie algorytmu genetycznego,
f_{GA}^{1} = ruletka – rodzaj zastosowanej selekcji podczas wyboru pierwszej produkcji,
f_{GA}^{2} = ruletka – rodzaj zastosowanej selekcji podczas wyboru drugiej produkcji,
f_{kor} = nie – zmienna zezwalająca na uruchomienie korekcji gramatyki,
f_{cs} = tak – zmienna zezwalająca na uruchomienie operatora pokrycia startowego,
f_{cp} = tak – zmienna zezwalająca na uruchomienie operatora pokrycia pełnego,
f_{cu} = nie – zmienna zezwalająca na na uruchomienie operatora pokrycia uniwersalnego.

Parametry ciągle

$n_{max} = 5000$ – maksymalna liczba kroków ewolucyjnych,
$n_{run} = 50$ – liczba iteracji,
$n_P = 40$ – rozmiar populacji,
$n_{start} = 30$ – liczba początkowych produkcji nieterminalnych,
$n_N = 19$ – liczba symboli nieterminalnych,
$n_T = 2$ – liczba symboli terminalnych,
$p_k = 0{,}2$ – prawdopodobieństwo krzyżowania dla algorytmu genetycznego,
$p_m = 0{,}8$ – prawdopodobieństwo mutacji dla algorytmu genetycznego,
$p_i = 0$ – prawdopodobieństwo inwersji dla algorytmu genetycznego,
$p_a = 0$ – prawdopodobieństwo zastosowania operatora pokrycia agresywnego,
$cf = 18$ – współczynnik ścisku,
$cs = 3$ – podpopulacja ścisku,
$ba = 1$ – kwota bazowa,
$raf = 0{,}5$ – współczynnik zmniejszania kwoty bazowej,
$n_{elit} = 0$ – wielkość elity,
$w_p = 1$ – waga rozbioru zdania poprawnego,
$w_n = 2$ – waga rozbioru zdania niepoprawnego,
$w_c = 1$ – waga funkcji klasycznej klasyfikatora,
$w_f = 0$ – waga funkcji płodności klasyfikatora,
$f_0 = 0{,}5$ – miara użyteczności klasyfikatora niebiorącego udziału w parsowaniu.

Zbiór Tomity

L1: a^*

L2: $(ab)^*$

L3: $(b|aa)^*(a^*|(abb(bb|a)^*))$

dowolne zdanie nad {*a*, *b*} bez nieparzystej liczby symboli *b* po nieparzystej liczbie *a*

L4: $a^*((b|bb)aa^*)^*(b|bb|a^*)$

dowolne zdanie nad {*a*, *b*} niezawierające podciągu trzech lub więcej symboli *b*

L5: $((aa|bb)^*((ba|ab)(bb|aa)^*(ba|ab)(bb|aa)^*)^*(aa|bb)^*$

dowolne zdanie nad {*a*, *b*} zawierające parzystą liczbę symboli *a* i parzystą liczbę symboli *b*

L6: $((b(ba)^*(a|bb))|(a(ab)^*(b|aa)))^*$

dowolne zdanie nad {*a*, *b*} takie, że różnica liczby symboli *a* i liczby symboli *b* jest wielokrotnością liczby 3

L7: $b^*a\ ^*b^*a^*$

Zbiór Tomity - statystyki

Język	$\|U\|$	$\|U^+\|$	$\|U^-\|$	$\|T\|$	$\|T^+\|$	$\|T^-\|$
L1	16	8	8	65 534	15	65 519
L2	15	5	10	65 534	7	65 527
L3	24	12	12	65 534	9447	56 087
L4	19	10	9	65 534	23 247	42 287
L5	21	9	12	65 534	10 922	54 612
L6	21	9	12	65 534	21 844	43 690
L7	20	12	8	65 534	2515	63 019

$|U|$ - moc zbioru uczącego
$|U^+|$ - moc zbioru zdań uczących pozytywnych
$|U^-|$ - moc zbioru zdań uczących negatywnych
$|T|$ - moc zbioru testowego
$|T^+|$ - moc zbioru zdań testowych pozytywnych
$|T^-|$ - moc zbioru zdań testowych negatywnych

Zbiór Tomity - wyniki

- Zarówno uczenie, jak i testy generalizacji przebiegały na identycznych zbiorach przykładów, jak te zastosowane w (Luke i in. 1999) oraz (Lucas i Reynolds 2005)
- Obie prace prezentują – podobnie jak model GCS – ewolucyjne podejście do indukcji automatów, a opublikowane w nich wyniki należą do najlepszych ze znanych w literaturze i to nie tylko wśród metod ewolucyjnych

Zbiór Tomity - (Luke i in. 1999)

- W metodzie zaproponowanej przez Luke'a i in. (1999), zwanej dalej metodą GP („*Genetic*" *Programming*), indukowany automat skończony jest reprezentowany przez genom składający się z nieograniczonej liczby genów
- Każdy gen reprezentuje stan automatu
- Pozostałe atrybuty każdego genu są używane do sterowania przejściami pomiędzy stanami
- Do każdego genu przypisana jest zmienna logiczna wskazująca, czy skojarzony z genem stan jest stanem akceptującym

Zbiór Tomity - (Lucas i Reynolds 2005)

- Ewolucji podlega jedynie macierz przejść automatu, a „sprytny” (*smart*) algorytm etykietowania stanów indukowanego automatu uzupełnia jego opis. Zastosowanym mechanizmem ewolucyjnym jest prosta strategia ewolucyjna (1 + 1)
- Lucas i Reynolds zbadali trzy różne mutacje własnej metody: metodę, w której ewolucji podlegała zarówno macierz przejść automatu, jak i wektor etykiet stanów (tzw. metoda *Plain*), metodę,w której ewoluowano jedynie macierz przejść automatu, a liczba stanów automatu była stała i równa 10 (*Smart*) i wreszcie metodę, w której liczba stanów automatu była równa minimalnej liczbie stanów automatu reprezentującego badany język (*nSmart*)

Zbiór Tomity - wyniki

Język	*nSuccess*		*nEvals*		*nGen*	
	GP	GCS	GP	GCS	GP	GCS
L1	31/50	50/50	30	2	88,4	100
L2	7/50	50/50	1010	2	84	100
L3	1/50	1/50	12 450	666	66,3	100
L4	3/50	24/50	7870	2455	65,3	100
L5	0/50	50/50	13 670	201	68,7	92,4
L6	47/50	49/50	2580	1471	95,9	96,9
L7	1/50	11/50	11 320	2902	67,7	92

nSuccess - iloraz liczby iteracji zakończonych sukcesem do liczby wszystkich iteracji

nEvals - średnia liczba kroków ewolucyjnych, podczas których algorytm znajduje 100% przystosowaną gramatykę

nGen - *wartość* estymatora dokładności generalizacji

Zbiór Tomity - wyniki

Język	Plain	Smart	nSmart	GP	GCS
L1	107	25	15	30	2
L2	186	37	40	1010	2
L3	1809	237	833	12 450	666
L4	1453	177	654	7870	2455
L5	1059	195	734	13 670	201
L6	734	93	82	2580	1471
L7	1243	188	1377	11 320	2902

nEvals - średnia liczba kroków ewolucyjnych, podczas których algorytm znajduje 100% przystosowaną gramatykę

Zbiór Tomity - wyniki

Język	Smart	nSmart	EDSM	GP	GCS
L1	81,8	100	52,4	88,4	100
L2	88,8	95,5	91,8	84	100
L3	71,8	90,8	86,1	66,3	100
L4	61,1	100	100	65,3	100
L5	65,9	100	100	68,7	92,4
L6	61,9	100	100	95,9	96,9
L7	62,6	82,9	71,9	67,7	92

nGen - wartość estymatora dokładności generalizacji

- EDSM (*Evidence Driven State Merging*) - jest jedną z najlepszych metod indukcji DFA. Metoda jest przykładem heurystycznego algorytmu, który iteracyjnie kompresuje początkowo duży automat, za każdym razem dbając o to, by zachować prawidłową klasyfikację zbioru uczącego przed i po kompresji

Wybrane języki bezkontekstowe

AB: dowolne zdanie nad $\{a, b\}$
zawierające taką samą liczbę symboli a i b,

AnBn: $a^n b^n$

BRA1: język zbilansowanych nawiasów

BRA3: język zbilansowanych nawiasów trzech typów

PAL2: palindromy nad $\{a, b\}$ o parzystej długości

TOY: gramatyka dziecięca

Gramatyka dziecięca

S → *np vp*
np → *det n* | *np pp*
pp → *prep n*
vp → *v np* | *vp pp*

gdzie:
np oznacza grupę rzeczownikową (*nominal phrase*)
vp – grupę czasownikową (*verbal phrase*)
det – rodzajnik (*determiner*)
n – rzeczownik (*noun*)
pp – grupę przyimkową (*prepositional phrase*)
prep – przyimek (*preposition*)
v – czasownik (*verb*)

Wybrane języki bezkontekstowe - statystyki

Język	$\|U\|$	$\|U^+\|$	$\|U^-\|$	$\|T\|$	$\|T^+\|$	$\|T^-\|$
AB	200	101	99	65 534	4706	60 828
AnBn	200	15	185	65 534	7	65 527
BRA1	200	101	99	65 534	625	64 909
BRA3	200	100	100	65 534*	22274	43 260
PAL2	200	101	99	65 534	224	65 310
TOY	200	99	101	65 534*	105	65 429

Wybrane języki bezkontekstowe - wyniki

- (Bianchi 1996, LCS)- system oparty o klasyczną architekturę systemu klasyfikującego z uproszczoną wersją algorytmu kubełkowego, a rozbiór zdań wykonywany jest z zastosowaniem algorytmu CYK

- Lankhorst (1995) indukował niedeterministyczny automat ze stosem, używając dwóch metod reprezentacji zadania: binarnej (*Lan1*) oraz całkowitoliczbowej (*Lan2*). Funkcja oceny uwzględniała poprawność klasyfikacji zdań uczących, częściowo poprawne analizowane podciągi oraz stopień wypełnienia stosu

Wybrane języki bezkontekstowe - wyniki

Język	LCS	GA	GCS
BRA1	53	500	40
BRA3	319	9500	47
TOY	3000	–	210

nEvals - średnia liczba kroków ewolucyjnych, podczas których algorytm znajduje 100% przystosowaną gramatykę

Język	*nEvals*			*nGen*		
	Lan1	Lan2	GCS	Lan1	Lan2	GCS
AB	398	629	288	96,50	96,50	100
BRA1	18	19	40	97	90,50	100
PAL2	727	704	418	79	82	100
TOY	328	235	210	98,50	98,50	100

Wybrane języki bezkontekstowe - wyniki

Język	IO	KL	GCS
AB	82	95	100
AnBn	94	100	100
BRA1	92	100	100
PAL2	42	85	100

nSuccess - iloraz liczby iteracji zakończonych sukcesem do liczby wszystkich iteracji

- KL –algorytm genetyczny w indukcji stochastycznych gramatyk bezkontekstowych reprezentowanych w PNC (Keller i Lutz 1997, 2005). W ich podejściu ewolucji podlegała nie tyle cała gramatyka, co jedynie zbiór parametrów opisujących poszczególne produkcje, w tym prawdopodobieństwa produkcji
- IO – podejście, w którym zbiór parametrów gramatyki stochastycznej jest optymalizowany za pomocą algorytmu *inside-outside*

Korpusy językowe

- Proces uczenia zasilany jest zbiorem uczącym składającym się ze zdań poprawnych i niepoprawnych
- Korpus językowy wymaga oznakowania morfosyntaktycznego (*part-of-speech tags*, POS), czyli przejścia z tekstu języka naturalnego na zapis symboliczny, składający się z sekwencji tagów
- W indukcji bez nadzoru stosuje się stosunkowo często repozytoria językowe (*treebanks*), które oprócz zbioru zdań języka naturalnego zawierają tzw. metadane, jak: oznaczenia końców zdań, akapitów, oznaczenia morfosyntaktyczne słów, informacje o strukturze syntaktycznej zdań, informacje semantyczne (np. podział korpusu na części tematyczne)

Korpusy językowe

- Rezultatem działania modelu nie jest PCFG lub jakiś rodzaj gramatyki kategorialnej, lecz nieprobabilistyczna gramatyka bezkontekstowa
- Proces uczenia skupia się na rozwiązywaniu problemu przynależności zdania do języka, a nie budowania jego struktury syntaktycznej (chociaż w procesie indukcji znajdowane są wszystkie możliwe drzewa rozbioru)
- Wnioskowanie wymaga etykietowanego zbioru uczącego
- Jedną z nielicznych w literaturze przedmiotu prac spełniających postawione założenia jest (Aycinena i in. 2003) - indukowana gramatyka algorytmem genetycznym jest reprezentowana przez PNC, a rozbiór dokonywany jest przez parser tablicowy CYK

Korpusy językowe - zbiory uczące

- **Korpus children**

  Korpus tworzą wybrane teksty z literatury dziecięcej, dostępnej pod adresem http://www.magickeys.com/books
- **Korpus wizard**

  Na korpus składają się obszerne fragmenty z książki *Czarownik z krainy Oz* (*The Wizard of Oz*) L. Franka Bauma, dostępne pod adresem http://www.ucalgary.ca/dkbrown/storclas.html
- **Korpus alice**

  W skład korpusu wchodzą obszerne fragmenty z książki *Alicja w krainie czarów* (*Alice in Wonderland*) L. Carrolla, dostępne pod adresem http://www.ucalgary.ca/dkbrown/storclas.html
- **Korpus tom**

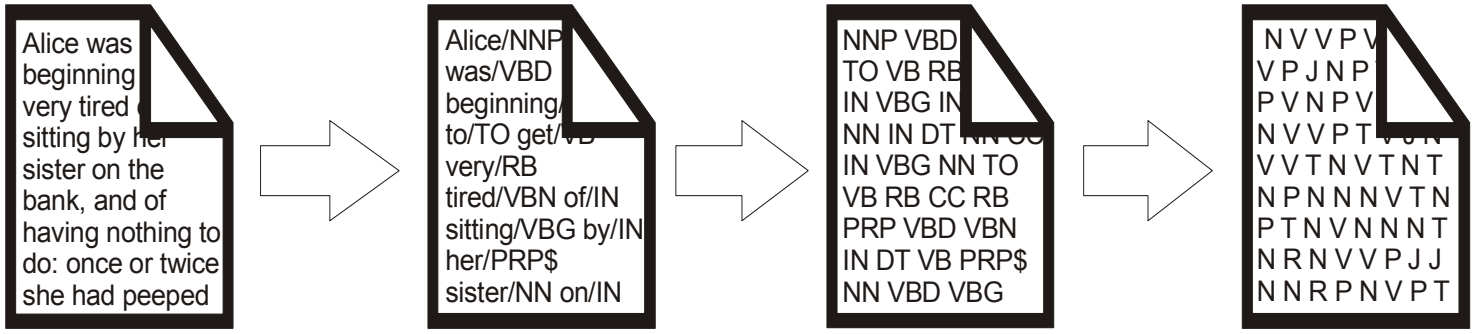
  Korpus składa się z obszernych fragmentów książki *Tomek Sawyer* (*Tom Sawyer*) M. Twaina, dostępnej pod adresem http://www.infomotions.com/alex/authors.html
- **Korpus brown**

  Korpus tworzy pięć nieoznakowanych fragmentów z repozytorium Browna, oznaczanych brown_a do brown_e

  (afs/ir.stanford.edu/data/linguistic-data/Brown/ICAME-Brown1)

Korpusy językowe - przygotowanie

- Aby przygotować odpowiedni zestaw uczący, słowa w korpusie zostały najpierw oznaczone symbolami morfosyntaktycznymi przy użyciu *taggera* Brilla (1993)
- Następnie usunięte zostały słowa języka angielskiego, a pozostałe ciągi tagów zredukowano do 7 nieterminali
- Zredukowany ciąg tagów sformatowany został do używanego przez program gcs formatu *abbadingo*
- Zbiór uczący został uzupełniony o przykłady negatywne, reprezentowne przez losowo wygenerowane ciągi zredukowanych tagów o długości od 5 do 15 słów zgodnie z rozkładem normalnym



Korpusy językowe - statystyki

Korpus	$\|U\|$	$\|U^+\|$	$\|U^-\|$	$\|T\|$	$\|T^+\|$	$\|T^-\|$
children	1972	986	986	986	493	493
wizard	3080	1540	1540	1542	771	771
alice	2024	1012	1012	1014	507	507
tom	7202	3601	3601	3602	1801	1801
brown_a	5578	2789	2789	2790	1395	1395
brown_b	3560	1780	1780	1782	891	891
brown_c	2198	1099	1099	1100	550	550
brown_d	2124	1062	1062	1064	532	532
brown_e	5022	2511	2511	2512	1256	1256

Korpusy językowe - wyniki

Korpus	*fitness*$_{max}$		*positive*		*negative*		*Evals*	
	GCS	AKM	GCS	AKM	GCS	AKM	GCS	AKM
children	93,2	93,1	98,8	91,8	12,5	5,7	9	200 000
wizard	94,6	90,2	99,3	89,5	10,2	9,2	32	200 000
alice	89,5	92,1	96,8	92,5	17,9	8,4	81	200 000
tom	86,3	92,1	98,4	92,7	25,9	8,6	3	200 000
brown_a	93,8	94,0	98,3	94,1	11,6	6,1	45	48 500
brown_b	94,6	94,0	99,3	94,7	10,2	6,7	506	200 000
brown_c	92,5	87,9	96,7	80,5	11,7	4,7	592	15 500
brown_d	91,6	91,3	97,1	88,2	13,8	5,6	18	45 000
brown_e	89,5	94	93,4	93,9	14,5	5,9	38	122 000

fitness$_{max}$ – wartość najlepszego procentu prawidłowo sparsowanych zdań ze zbioru uczącego; *positive* - wartość procentu sparsowanych zdań poprawnych dla *fitness*$_{max}$; *negative* - wartość procentu sparsowanych zdań niepoprawnych dla *fitness*$_{max}$; *Evals* - liczba kroków potrzebnych do wyuczenia się gramatyki o *fitness*$_{max}$

Genomika obliczeniowa

- Jeżeli przyjąć, że sekwencję kwasów nukleinowych można opisać formalnym językiem (Searls 1993, 2002), to sekwencjonowanie biosekwencji za pomocą metod lingwistycznych można wykorzystać do:
  - edycji sekwencji,
  - poszukiwania miejsc restrykcyjnych,
  - projektowania starterów i sond oligonukletydowych,
  - analizowania składu nukleotydowego i aminokwasowego,
  - wyznaczania punktu izoelektrycznego białka,
  - poszukiwania rejonów kodujących, granic intron-ekson,
  - translacji, poszukiwania otwartej ramki odczytu,
  - analizie charakteru poszczególnych części sekwencji białka,
  - przewidywania struktury dwu- i trzeciorzędowej białek i kwasów nukleinowych lub ich fragmentów,
  - tworzenia interfejsów z biologicznymi bazami danych.

Predykcja regionów promotorowych

- Klasycznym zadaniem przewidywania regionów kodujących i niekodujących w sekwencjach DNA jest predykcja regionów promotorowych u prokariotycznej bakterii *E. coli* (pałeczki okrężnicy) (Towell i in. 1990, Handley 1995, Chen i in. 2002)

Predykcja regionów promotorowych - wyniki

Method	Specificity	Sensitivity	Accuracy
KBANN	97	16	56
WANN	82	69	75
GCS	94	61	78

Specificity - (True Negatives/(False Positives + True Negatives)) x 100

Sensitivity - (True Positives/(True Positives + False Negatives)) x 100

Accuracy - ((True Positives + True Negatives/Total) x 100

KBANN –hybrydowe podejście wykorzystujące sieć neuronową i zestaw reguł (Towell i in. 1993)

WANN –kombinacja KBANN i gramatyki z ważonymi matrycami (Leung 2001)

Data mining

Zbiór danych	XCSL	XCS	ACS	C4.5	GCS
Monk's 1	100,0%	100,0%	99,1%	83,9%	76,0%
Monk's 2	99,8%	100,0%	74,5%	65,0%	96,0%
Monk's 3	97,0%	95,0%	96,5%	93,4%	76%
Voting-record	99,4%	–	98,1%	96,3%	68%
WBC	100,0%	95,5%	97,5%	96,0%	83%

nGen – wartość dopasowania gramatyki dla zbioru testowego

Zbiory Monk1-3, Voting-record, WBC – zbiory z repozytorium UCI

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Rozszerzenia modelu GCS

- Rezygnacja z klasyfikatorów w PNC
- Stochastyczny GCS (sGCS)
- Reczywistoliczbowy GCS (rGCS)

Real-valued GCS

- „Zdaniem” uczącym jest oznakowany ciąg liczb rzeczywistych (wektor)
- Reguły terminalne typu $A \rightarrow$ *liczba_rzeczywista*
- Reguły nieterminalne jak w klasycznym GCS
- Dopasowanie reguły terminalnej do kolejnych elementów wektora uczącego na podstawie odległości
- Parsowanie algorytmem CYK
- Po zakończeniu analizy zbioru uczącego następuje:
  - ocena reguł nieterminalnych,
  - modyfikacja reguł terminalnych uwzględniająca liczbę wykonanych cykli uczenia oraz odległość reguły od elementu wektora uczącego.

Plan

- Wnioskowanie gramatyczne
- Indukcja gramatyk bezkontekstowych
- Uczące się systemy klasyfikujące
- Model GCS
- Zastosowania modelu GCS
- Rozszerzenia modelu GCS
- Podsumowanie
- Literatura dotycząca modelu GCS

Podsumowanie

- Opracowano parametryczny, ewolucyjny model indukcji gramatyki bezkontekstowej
- Możliwe szerokie spektrum zastosowań, od lingwistyki, poprzez biologię obliczeniową, do … klasyfikacji obrazów spektroskopowych i kartograficznych
- Model jest podatny na modyfikację i rozszerzenia

Literatura dotycząca modelu GCS

- **Unold O.** (2007), Grammatical Inference with Grammar-based Classifier System, Applied Soft Computing *(artykuł zgłoszony)*.
- Cielecki L., **Unold O.** (2007), GCS with Real-Valued Input, [w:] Mira J., Alvarez J.R. (red.) IWINAC 2007, Part I, LNCS 4527, 488-497.
- **Unold O.** (2007), Learning classifier system approach to natural language grammar induction, [w:] Shi Y. i in. (red.) ICCS 2007, Part II, **LNCS** 4488, 1210-1213.
- **Unold O.** (2007), Grammar-based classifier system for recognition of promoter regions, [w:] Beliczynski B. i in. (red.) ICANNGA07, Part I, LNCS 4431, 798-805.
- **Unold O.**, Cielecki Ł. (2007), Learning Context-Free Grammars from Partially Structured Examples: Juxtaposition of GCS with TBL, [w:] 7th International Conference on Hybrid Intelligent Systems, Germany, IEEE Computer Society Press, 348–351.
- Cielecki L., **Unold O.** (2007), Real-valued GCS classifier system, Int. J. Appl. Math. Comput. Sci., vol.17, no.4.
- **Unold O.** (2006), Ewolucyjne wnioskowanie gramatyczne, Monografia habilitacyjna, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław.
- **Unold O.** (2005), Playing a toy-grammar with GCS, [w:] Mira J., Álvarez J.R. (red.) IWINAC 2005, LNCS 3562, 300-309.
- **Unold O.** (2005), Context-free grammar induction with grammar-based classifier system, Archives of Control Science, vol. 15 (LI), 4, 681–690.
- **Unold O.** (2005), Analiza wpływu wybranych parametrów na efektywność systemu GCS, Informatyka Teoretyczna i Stosowana 8(5), Częstochowa, 177–190.
- **Unold O.**, Cielecki L. (2005), Grammar-based Classifier System, [w:] Hryniewicz O. i in. (red.) Issues in Intelligent Systems: Paradigms, EXIT, Warszawa, 273–286.
- **Unold O.** (2005), GCS – nowy model uczącego się systemu klasyfikującego, [w:] Zamojski W. (red.) Inżynieria Komputerowa, WKŁ, Warszawa, 60–72.
- **Unold O.**, Cielecki Ł. (2005), How to use crowding selection in Grammar-based Classifier System, [w:] Kwaśnicka H., Paprzycki M. (red.) Proc. of 5th International Conference on Intelligent Systems Design and Applications, Los Alamitos, IEEE Computer Society Press, 126–129.
- **Unold O.** (2005), Learning context-free language using Grammar-based Classifier System, [w:] Vetulani Z. (red.) Human language technologies as a challenge for computer science and linguistics, 2nd Language & Technology Conference, Poznań, Wydaw. Poznańskie, 423–426.