Parser składniowo-semantyczny dla języka polskiego

część II: integracja z zasobami leksykalnymi, parsowanie

Wojciech Jaworski

Instytut Informatyki Uniwersytetu Warszawskiego

Instytut Podstaw Informatyki Polskiej Akademii Nauk

25 stycznia 2016

Spis treści

Preprocessing — integracja zasobów

- Tekst reprezentowany jest jako graf.
- Krawędzie etykietowane są tokenami.
- Podczas preprocesingu dodawane są nowe krawędzie i wierzchołki
- Przetwarzaną jednostką jest akapit.
- Etapy preprocesingu:
  - identyfikacja wielkich i małych liter
  - podział na tokeny
  - rozpoznawanie liczebników zapisanych za pomocą cyfr, dat itp.
  - rozpoznawanie odmienionych akronimów i wyrazów obcych
  - rozpoznawanie form wielotokenowych
  - usunięcie białych znaków
  - lematyzacja
  - rozwijanie skrótów
  - rozpoznawanie wyrażeń wielosłownych
  - wykrywanie nazw własnych
  - określanie sensów słów
  - zwijanie interpretacji
  - określanie walencji

Znaki

- Podział tekstu na znaki.
- Klasyfikacja znaków na
  - cyfry,
  - symbole,
  - małe litery,
  - wielkie litery,
  - inne
- Przyporządkowywanie wielkim literom ich małych odpowiedników.
- Analizowane bloki Unicode:
  - Basic Latin
  - Latin-1 Supplement
  - Latin Extended-A
  - Latin Extended-B
  - Latin Extended Additional
  - General Punctuation
  - Currency Symbols

Tokeny

- Sekwencje cyfr wraz z typami np: liczby naturalne, sekwencje 3-cyfrowe, numery miesięcy
- Liczby rzymskie wraz z tłumaczeniem na arabskie
- Sekwencje liter z podziałem na sekwencje małych liter, sekwencje wielkich, sekwencje małych poprzedzone wielką i inne.
- Wielkie litery są interpretowane jako użycie CapsLock lub początek zdania.
- Rozpoznawane są sufiksy: -em, -m, -eś, -ś, -eśmy, -śmy, -eście, -ście, -by.
- Generowane są wszystkie możliwe interpretacje.
- Przetwarzanie symboli (NKJP1M):
  - interpretacja cudzysłowów zapisanych jako: ” „ ” “ „ ” “ « » << >>
  - interpretacja myślników i dywizów zapisanych jako: --- -- - – — –
  - interpretacja apostrofów zapisanych jako: ‘ ‘ ’ ’
  - interpretacje przecinka, kropki i wielokropka uwzględniające ich różne znaczenia i haplologię
  - rozpoznawanie sekwencji znaków interpunkcyjnych
  - rozpoznawanie emotikonów.

Rozpoznawanie złożonych tokenów

- Liczby (NKJP1M):

$$\text{pref3dig . 3dig . 3dig . 3dig} \rightarrow \text{natnum}$$
$$\text{day . month . year} \rightarrow \text{date}$$
$$\text{- natnum , dig} \rightarrow \text{realnum}$$

- Odmienione akronimy i nazwiska obce (SGJP i Polimorf):

$$\text{* - ów} \rightarrow \text{* subst:pl:acc:m1|subst:pl:gen:m1.m2.m3.n2}$$
$$\text{* - cie} \rightarrow \text{*T subst:sg:loc.voc:m3}$$
$$\text{* - cie} \rightarrow \text{*TA subst:sg:dat.loc:f}$$
$$\text{* - owscy} \rightarrow \text{*-owski adj:pl:nom.voc:m1.p1:pos}$$
$$\text{* ’ ego} \rightarrow \text{* subst:sg:gen.acc:m1}$$

- Leksemy wielotokenowe (SGJP i Polimorf):

$$\text{ping - ponga} \rightarrow \text{ping-pong subst:sg:gen.acc:m2}$$
$$\text{rock ’ n ’ rollem} \rightarrow \text{rock’n’roll subst:sg:inst:m2}$$

- Wyszukiwane są najdłuższe dopasowania.
- Znalezione dopasowania zastępują sekwencje tokenów.

Lematyzacja

- Na podstawie SGJP-20151020 oraz Polimorf-20151020.
- W pierwszym kroku na podstawie końcówki słowa odnajdywane są możliwe lematy i interpretacje.
- W drugim kroku następuje wybór znanych lematów spośród odnalezionych
- Jeśli żaden znany lemat nie jest wystąpił wśród odnalezionych zwracane są wszystkie odnalezione lematy i interpretacje.
- Tokeny będące sekwencjami małych liter poprzedzonych wielką literą są lematyzowane w swojej pierwotnej postaci, a w razie niepowodzenia w wersji ze zmniejszoną pierwszą literą.

  Kotem $\rightarrow$ kotem $\rightarrow$ kot subst:sg:inst:m2 $\rightarrow$ Kot subst:sg:inst:m2

Nazwy własne

- Na podstawie SGJP-20151020 oraz Polimorf-20151020.
- Uzupełnione o inicjały, nazwy dni tygodnia i miesięcy.
- Nazwom przyporządkowywane są typy nazw np.: toponim, nazwisko
- oraz typy obiektów wskazywanych przez nazwy np.: obszar, osoba.
- Aktualnie tylko rzeczowniki mogą być nazwami własnymi.
- Rzeczowniki napisane wielką literą, ale nie znalezione w słowniku nazw własnych traktowane są jako nazwy własne nieznanego typu.

Sensy słów

- Na podstawie Słowosieci 2.1.0.
- Lematom przypisywane są listy sensów.
- Sensy reprezentowane są jako lematy zaopatrzone w numery wariantów.
- Sensy uzupełnione są o listy hiperonimów.
- Hiperonimy to synsety reprezentowane przez kanonicznie wybrane sensy.
- Aktualnie nie korzystam z liczbowych identyfikatorów ze Słowosieci.
- Nazwom własnym przypisywane są sensy wynikające z ich typów.

zamek 1: budowla: rezultat 1, wytwór 1, konstrukcja 1, budowla 1, budynek 1, dom 1, rezydencja 1, zamek 1
zamek 2: urządzenie do zamykania: obiekt 2, rzecz 4, przedmiot 1, zamknięcie 12, zamek 2
zamek 6: suwak: obiekt 2, rzecz 4, przedmiot 1, zamknięcie 12, zapięcie 2, zamek błyskawiczny 1

Walencja (część semantyczna)

- Słownik walencyjny *Walenty* zawiera
  - schematy opisujące cechy składniowe podrzędników wybranych czasowników, rzeczowników, przymiotników i przysłówków,
  - ramy opisujące cechy semantyczne (role tematyczne i preferencje selekcyjne) podrzędników,
  - powiązania pomiędzy schematami i ramami.
- Na potrzeby parsowania łączę ramy ze schematami.
- Predefiniowane preferencje selekcyjne dodałem do relacji hiperonimii w *Słowocieci*.
- Oprócz tego dodaję do preferencji selekcyjnych oraz do hiperonimów każdego sensu symbol T.
- Ma on na celu umożliwienie parsowania zdań nie spełniających preferencji selekcyjnych.
- Relacyjne preferencje selekcyjne na razie nie są zaimplementowane.
- Przed parsowaniem następuje dezambiguacja (selekcja) hiperonimów i preferencji selekcyjnych.

Testy

- Test wykonany na 19957 zdaniach ze *Składnicy*.
- Czas wykonania bez kosztu komunikacji między procesami: 373.535s (0.018s na zdanie).
- Czas wykonania wraz z kosztem komunikacji między procesami: 393.431s (0.019s na zdanie).
- Poprzednia wersja preprocesingu miała szybkość 0,41-0,43s na zdanie.
- Czas uruchamiania preprocesingu: ok. 4 minuty.
- Pokrycie bez niepoświadczonych lematów i niepoświadczonych nazw własnych:

	liczba tokenów	pokrycie *Słowosieci*	pokrycie *Walentego*
noun	127851	0.855	0.226
verb	59842	0.917	0.941
adj	43121	0.894	0.353

Spis treści

Gramatyka kategorialna (Type Logical Categorial Grammar)

- parsowanie jest dowodzeniem w niekomutatywnej intuicjonistycznej logice liniowej
  - można implementować fragmenty systemu dowodowego uzyskując szybkie parsery i mając gwarancję poprawności
- gramatyka jest w pełni zleksykalizowana (reguły gramatyczne to uniwersalne reguły dowodzenia w logice)
  - ułatwia to integrację z zasobami słownikowymi np. Walentym
- izomorfizm Curriego-Howarda wiąże reguły wnioskowania z termami liniowego rachunku lambda
  - zadaje sposób konstruowania formuł języka reprezentacji znaczenia podczas parsowania
- zamiast występującej w innych formalizmach gramatycznych unifikacji, dopasowywanie termu do wzorca
  - redukcja złożoności obliczeniowej

Spójniki LCG

Wyrażają podstawowe zjawiska występujące w języku.

- $\bullet$ konkatenacja (tworzenie wektorów cech)
- $\&$ niejednoznaczność
- $/$, $\backslash$, $|$ wymaganie argumentu
- $\oplus$ polimorficzne argumenty, sumowanie typów
- 1 opcjonalny argument
- ? wielokrotny argument
- $\&$ parametryzowana niejednoznaczność
- $\star$ tworzenie listy

Umożliwiają bezpośrednie wyrażenie informacji dostarczanej przez lematyzację i zawartej w Walentym.

zielony adj:sg:nom.voc:m1.m2.m3:pos|adj:sg:acc:m3:pos

$$(adj \bullet sg \bullet (nom \,\&\, voc) \bullet (m1 \,\&\, m2 \,\&\, m3) \bullet pos) \,\&\, (adj \bullet sg \bullet acc \bullet m3 \bullet pos)$$

Generowanie leksykonu z uwzględnieniem preferencji selekcyjnych

- Rozpatrzmy zdanie *Jan aranżował*.
- Załóżmy, że dla *aranżował* mamy jeden schemat mówiący, że czasownik ten bierze argument subj{np(str)}, czyli w tym wypadku podmiot w mianowniku.
- oraz jedną ramę związaną z tym schematem mówiącą, że argument ten powinien należeć do kategorii LUDZIE:

$$\bigotimes_{gender \in \{m1,m2,m3\}} ip \bullet T \bullet sg \bullet gender \bullet ter$$

$$\mid 1 \oplus np \bullet T \bullet sg \bullet nom \bullet gender \bullet ter \oplus np \bullet LUDZIE \bullet sg \bullet nom \bullet gender \bullet ter$$

Generowanie leksykonu z uwzględnieniem preferencji selekcyjnych c.d.

- Rzeczownik *Jan*, został rozpoznany jako nazwa własna typu *imię*.
- Skutkuje to przypisaniem mu znaczeń *imię 4* i *osoba 1*.
- W znaczeniu *imię 4* nie ma on hiperonimów zgodnych z preferencjami selekcyjnymi, więc będzie miał typ:

$$\mathrm{np} \bullet \mathrm{T} \bullet \mathrm{sg} \bullet \mathrm{nom} \bullet \mathrm{m1} \bullet \mathrm{ter}$$

- W znaczeniu *osoba 1* ma on hiperonim LUDZIE zgodny z preferencjami selekcyjnymi:

$$\bigwedge_{sense \in \{\mathrm{T},\mathrm{LUDZIE}\}} \mathrm{np} \bullet sense \bullet \mathrm{sg} \bullet \mathrm{nom} \bullet \mathrm{m1} \bullet \mathrm{ter}$$

Spis treści

Aplikacja

- Podstawowa reguła — aplikacja argumentu do funktora

$$\frac{\Gamma \vdash \psi \,/\, \varphi \qquad \Delta \vdash \varphi}{\Gamma, \Delta \vdash \psi}\,[/\,E] \qquad \frac{\Delta \vdash \varphi \qquad \Gamma \vdash \psi \setminus \varphi}{\Delta, \Gamma \vdash \psi}\,[\setminus E]$$

- Przykładowy wywód gramatyczny:
- Będziemy korzystać z leksykonu

$$\mathrm{Jan}_1 \vdash \mathrm{np},\ \mathrm{widzi}_2 \vdash (\mathrm{ip} \setminus \mathrm{np}) \,/\, \mathrm{np},\ \mathrm{stół}_3 \vdash \mathrm{np},\ .\!_4 \vdash \mathrm{s} \setminus \mathrm{ip}$$

- Otrzymamy następujące drzewo wywodu:

$$\dfrac{\dfrac{\mathrm{Jan}_1 \vdash \mathrm{np} \qquad \dfrac{\mathrm{widzi}_2 \vdash (\mathrm{ip} \setminus \mathrm{np}) \,/\, \mathrm{np} \qquad \mathrm{stół}_3 \vdash \mathrm{np}}{\mathrm{widzi}_2, \mathrm{stół}_3 \vdash \mathrm{ip} \setminus \mathrm{np}}}{\mathrm{Jan}_1, \mathrm{widzi}_2, \mathrm{stół}_3 \vdash \mathrm{ip}} \qquad .\!_4 \vdash \mathrm{s} \setminus \mathrm{ip}}{\mathrm{Jan}_1, \mathrm{widzi}_2, \mathrm{stół}_3, .\!_4 \vdash \mathrm{s}}$$

Semantyka

- Język reprezentacji znaczenia (MRL), np. logika pierwszego rzędu
- Język konstruowania formuł MRL: liniowy rachunek lambda.
- Reguły zaopatrzone są w λ-termy opisujące wpływ aplikacji reguły na semantykę:

$$\frac{\Gamma \vdash \psi \,/\, \varphi : M \qquad \Delta \vdash \varphi : N}{\Gamma, \Delta \vdash \psi : MN}\,[/E] \qquad \frac{\Delta \vdash \varphi : N \qquad \Gamma \vdash \psi \setminus \varphi : M}{\Delta, \Gamma \vdash \psi : MN}\,[\setminus E]$$

- Leksykon uzupełniony o semantykę (na potrzeby prezentacji maksymalnie uproszczoną):

$$\mathrm{Jan}_1 \vdash \mathrm{np} : j, \quad \mathrm{widzi}_2 \vdash (\mathrm{ip} \setminus \mathrm{np}) \,/\, \mathrm{np} : \lambda x \lambda y.w(y,x),$$

$$\mathrm{stół}_3 \vdash \mathrm{np} : s, \quad .\!_4 \vdash \mathrm{s} \setminus \mathrm{ip} : \lambda x.x$$

- Drzewo wywodu uzupełnione o semantykę:

$$\dfrac{\mathrm{Jan}_1 \vdash \mathrm{np} : j \qquad \dfrac{\mathrm{widzi}_2 \vdash (\mathrm{ip} \setminus \mathrm{np}) \,/\, \mathrm{np} : \lambda x \lambda y.w(y,x) \qquad \mathrm{stół}_3 \vdash \mathrm{np} : s}{\mathrm{widzi}_2, \mathrm{stół}_3 \vdash \mathrm{ip} \setminus \mathrm{np} : \lambda y.w(y,s)}}{\mathrm{Jan}_1, \mathrm{widzi}_2, \mathrm{stół}_3 \vdash \mathrm{ip} : w(j,s)}$$

Parser (bez kompresji niejednoznaczności)

$$\frac{\Gamma \vdash \psi \,/\, \varphi \qquad \Delta \vdash \varphi}{\Gamma, \Delta \vdash \psi}\,[/\,E] \qquad \frac{\Delta \vdash \varphi \qquad \Gamma \vdash \psi \setminus \varphi}{\Delta, \Gamma \vdash \psi}\,[\setminus E]$$

- Bazuje na algorytmie CYK.
- Wypełnia tablicę biorąc po dwa tokeny.
- Z pierwszego z nich próbuje wywnioskować $\psi \,/\, \varphi : M$ a z drugiego $\varphi : N$.
- Jeśli mu się uda dodaje $\psi : MN$ do tablicy, wykonując przy tym redukcję MN.
- Analogicznie z drugiego próbuje wywnioskować $\psi \setminus \varphi : M$ a z pierwszego $\varphi : N$.
- Wnioskowania są przeprowadzane za pomocą ograniczonego systemu dowodowego logiki liniowej,
- który parser w bezpośredni sposób implementuje.

Kompresja niejednoznaczności

- Stosując aplikację w przód po wywnioskowaniu $\psi \,/\, \varphi : M$, bierze wszystkie tokeny znajdujące się na drugim polu.
- Z każdego z nich próbuje wywnioskować φ otrzymując listę możliwych semantyk $N_1, \ldots, N_k$.
- Jeśli lista ma jeden element dodaje do tablicy $\psi \,/\, \varphi : MN_1$.
- Jeśli lista ma więcej niż jeden element:
  - tworzy nową etykietę e,
  - dodaje etykietowany wariant $\psi \,/\, \varphi : M\langle e_1 : N_1, \ldots, e_2 : N_2\rangle$ do tablicy.
- Aplikacja w tył działa analogicznie.

Niejednoznaczność sensów słów

- Sensy słów wprowadzają olbrzymią niejednoznaczność, która tylko w niewielkim stopniu redukowana jest przez preferencje selekcyjne
- Wynika to m.in. z tego, że poszczególne sensy danego leksemu są do siebie na tyle podobne, że wpadają w te same preferencje selekcyjne.
- Np. w zdaniu *Człowiek aranżuje*
  - czasownik ma pięć ram/schematów (skojarzonych z 3 sensami),
  - w których podmiot ma preferencje LUDZIE, bądź PODMIOTY.
  - a rzeczownik ma 5 znaczeń,
  - z czego znaczenia 2, 4 i 5 mają jako hiperonim znaczenie 1.