Tagowanie części mowy
Parsowanie składniowe

Agnieszka Ławrynowicz

Wydział Informatyki i Telekomunikacji Politechniki Poznańskiej

27 kwietnia 2020

Tagowanie części mowy

Tagowanie części mowy

Tagowanie części mowy (ang. *part-of-speech (POS) tagging*)

Obliczeniowe metody oznaczania wyrazów częściami mowy (rzeczownik, czasownik, ...).

Zestaw tagów Penn Treebank

Markus i inni (1993)

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	*and, but, or*	SYM	symbol	+,%, &
CD	cardinal number	*one, two*	TO	"to"	*to*
DT	determiner	*a, the*	UH	interjection	*ah, oops*
EX	existential 'there'	*there*	VB	verb base form	*eat*
FW	foreign word	*mea culpa*	VBD	verb past tense	*ate*
IN	preposition/sub-conj	*of, in, by*	VBG	verb gerund	*eating*
JJ	adjective	*yellow*	VBN	verb past participle	*eaten*
JJR	adj., comparative	*bigger*	VBP	verb non-3sg pres	*eat*
JJS	adj., superlative	*wildest*	VBZ	verb 3sg pres	*eats*
LS	list item marker	*1, 2, One*	WDT	wh-determiner	*which, that*
MD	modal	*can, should*	WP	wh-pronoun	*what, who*
NN	noun, sing. or mass	*llama*	WP$	possessive wh-	*whose*
NNS	noun, plural	*llamas*	WRB	wh-adverb	*how, where*
NNP	proper noun, sing.	*IBM*	$	dollar sign	*$*
NNPS	proper noun, plural	*Carolinas*	#	pound sign	*#*
PDT	predeterminer	*all, both*	"	left quote	' or "
POS	possessive ending	*'s*	"	right quote	' or "
PRP	personal pronoun	*I, you, he*	(	left parenthesis	[, (, {, <
PRP$	possessive pronoun	*your, one's*	)	right parenthesis	],), }, >
RB	adverb	*quickly, never*	,	comma	,
RBR	adverb, comparative	*faster*	.	sentence-final punc	. ! ?
RBS	adverb, superlative	*fastest*	:	mid-sentence punc	: ; ... – -
RP	particle	*up, off*			

źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

Tagowanie części mowy

- wyrazy często mają więcej niż jeden POS-tag
  - The **back** door - JJ
  - On my **back** - NN
  - Promised to **back** the bill - VB
- problem oznaczenia części mowy dotyczy jej oznaczenia dla danej instancji wyrazu

Tagowanie części mowy

Wejście:	Plays	well	with	others
Niejednoznaczność:	NNS/VBZ	UH/JJ/NN/RB	IN	NNS
Wyjście:	Plays/VBZ	well/RB	with/IN	others/NNS

Tagowanie części mowy: zastosowania

- text-to-speech (jak wymawiać 'lead'?)
- wyrażenia regularne do płytkiej analizy, np. (Det) Adj * N+
- jako wejście albo żeby przyspieszyć pełnowymiarowy parser

Parsowanie

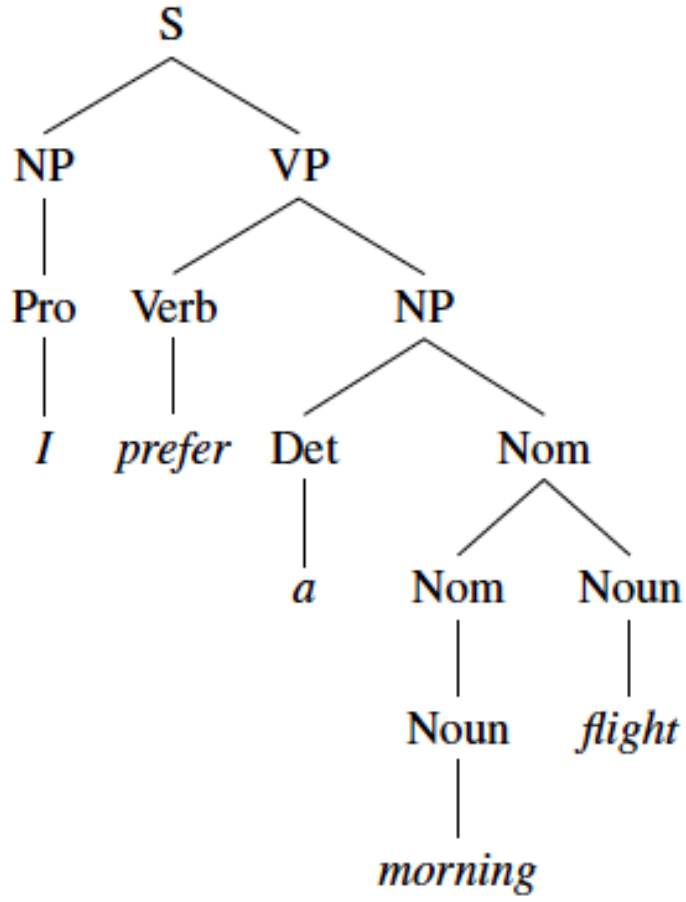
Parsowanie składnikowe

Parsowanie składnikowe (ang. *constituent parsing*)

Polega na identyfikacji fraz ('składników') w zdaniu oraz ich rekurencyjnej, zagnieżdżonej struktury.

Drzewo wyprowadzenia

Drzewo wyprowadzenia (ang. *parse tree*)



frazy ('składniki') oznaczone przez: NP (noun phrase), VP (verb phrase) itd.

źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

Składniki

Skąd wiemy co jest składnikiem?

- Rozkład: składnik 'zachowuje się' jak jednostka, która może wystąpić w różnych miejscach
  - Mary talked [to the children] [about drugs]
  - Mary talked [about drugs] [to the children]
  - *Mary talked drugs to the children about
- zastąpienia:
  - I sat [right on the box/right on top of the box/there]

Gramatyki bezkontekstowe

Struktura frazy organizuje wyrazy w (zagnieżdżone) składniki.

a dog

the cat

a large cat

a barking dog

a large cat on the table

Gramatyka bezkontekstowa

- N - zbiór nieterminalnych symboli (zmiennych)
- $\sum$ - zbiór terminalnych symboli (rozłączny z N)
- R - zbiór reguł (produkcji), w formie $A \to \beta$, gdzie A jest symbolem nieterminalnym, β jest łańcuchem symboli z nieskończonego zbioru $\sum \cup N*$
- S - symbol startu, należący do N

Prosta gramatyka

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, August 28, 2017, `https://web.stanford.edu/~jurafsky/slp3/`

Niejednoznaczności łączenia

Scientists count whales from space

Kluczową decyzją w parsowaniu jest jak łączymy różne składniki

Bank drzew

Bank drzew wyprowadzenia (ang. treebank)

Korpus, w którym każde zdanie ma odpowiadające mu drzewo wyprowadzenia.

Parsowanie zależnościowe

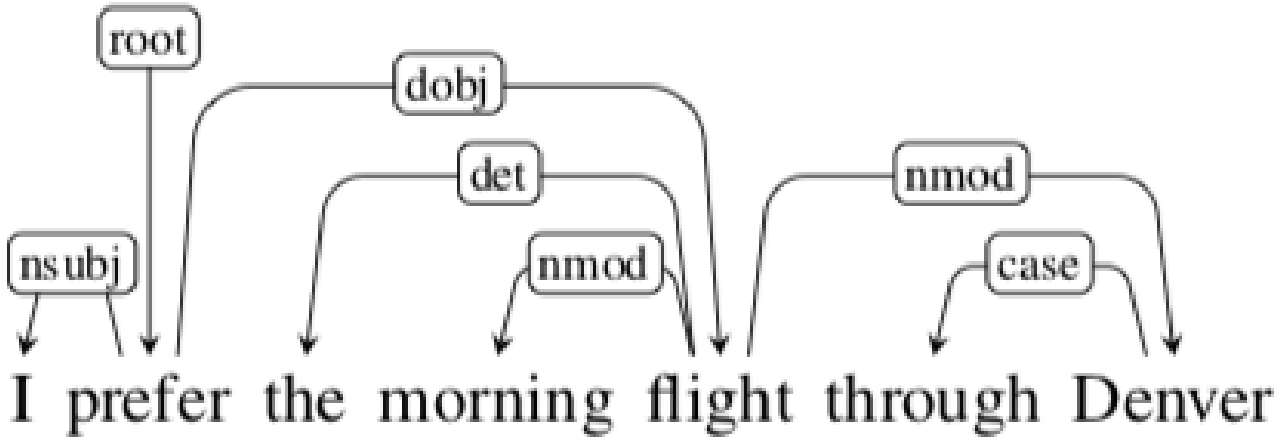
Parsowanie zależnościowe

Parsowanie zależnościowe (ang. *dependency parsing*)

Polega na wyznaczeniu struktury składniowej zdania (struktury zależnościowej) w postaci skierowanych, binarnych relacji pomiędzy wyrazami w zdaniu.

Pokazuje, które wyrazy *zależą* od (modyfikują lub są argumentami) innych wyrazów.

Struktura zależnościowa



źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, September, 2018, `https://web.stanford.edu/~jurafsky/slp3/`

Struktura zależnościowa

- drzewo rozpinające: każdy wierzchołek ma jedną krawędź wchodzącą (z wyjątkiem `root`)
- wierzchołki odpowiadają tokenom w zdaniu
  - korzeń `root` nie ma krawędzi wchodzących i ma jedną krawędź wychodzącą
- krawędzie reprezentują relacje:
  - krawędź skierowana od tokenu nadrzędnika do tokenu podrzędnika
  - etykiety krawędzi reprezentują funkcję gramatyczną podrzędnika
  - etykiety krawędzi pochodzą z ustalonego zbioru relacji gramatycznych

Czy dla języka polskiego lepsza będzie gramatyka (parsery) zależnościowa czy składnikowa? Dlaczego?

Relacje zależnościowe

Relacje dot. argumentów zdaniowych	Opis
NSUBJ	podmiot nominalny
DOBJ	dopełnienie bezpośrednie
IOBJ	dopełnienie niebezpośrednie
CCOMP	podrzędne frazy zdaniowe dołączane do predykatu; pełnią funkcje analogiczne do dopełnienia i przymiotnika; kontrolują własny podmiot
XCOMP	podrzędne frazy zdaniowe bez własnego podmiotu; odwołują się do podmiotu głównego predykatu

więcej w:
de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology.. In LREC, Vol. 14, pp. 4585–92.

Relacje zależnościowe c.d

Relacje dot. modyfikatorów rzeczowników	Opis
NMOD	frazy nominalne modyfikujące rzeczowniki
AMOD	modyfikator przymiotnikowy, to przymiotnik lub fraza przymiotnikowa modyfikująca znaczenie rzeczownika
NUMMOD	modyfikator liczbowy, czyli liczba lub fraza liczbowa modyfikująca rzeczownik pod względem ilości
APPOS	apozycja, która jest elementem nominalnym, definiującym, nazywającym lub opisującym rzeczownik
DET	relacja pomiędzy elementem determinującym a nadrzędnym elementem nominalnym
CASE	relacja przypadku jest używana w przypadku kiedy osobne słowo wyznacza przypadek gramatyczny powiązanego słowa
Inne	**Opis**
CONJ	relacja koordynacji, spójniki
CC	spójnik koordynacji, odpowiada za koordynowanie dwóch fraz

więcej w:
de Marneffe, M.-C., Dozat, T., Silveira, N., Haverinen, K., Ginter, F., Nivre, J., and Manning, C. D. (2014). Universal stanford dependencies: A cross-linguistic typology.. In LREC, Vol. 14, pp. 4585–92.

Universal Dependencies

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajic, J., Manning, C. D., McDonald, R. T., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R., and Zeman, D. (2016). Universal Dependencies v1: A multilingual treebank collection. In LREC-16.

Projekt Universal Dependencies `http://universaldependencies.org/` zawiera spis relacji zależnościowych

Universal Dependencies

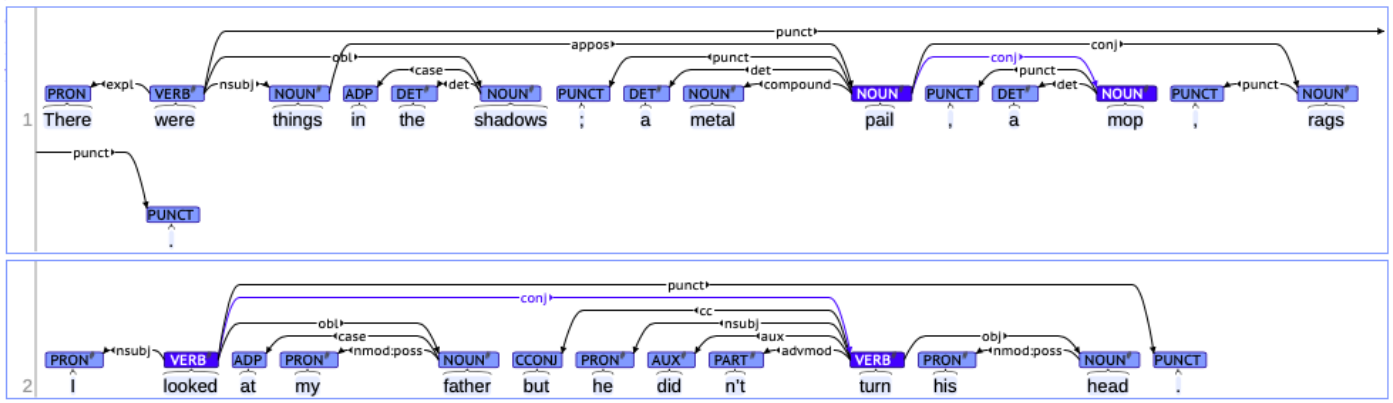
Treebank Statistics: UD_English-GUM: Relations: `conj`

This relation is universal.

3188 nodes (4%) are attached to their parents as `conj`.

3188 instances of `conj` (100%) are left-to-right (parent precedes child). Average distance between parent and child is 6.64930991217064.

The following 71 pairs of parts of speech are connected with `conj`: NOUN-NOUN (1102; 35% instances), VERB-VERB (868; 27% instances), PROPN-PROPN (396; 12% instances), ADJ-ADJ (187; 6% instances), ADJ-VERB (71; 2% instances), NOUN-VERB (67; 2% instances), PROPN-NOUN (50; 2% instances), VERB-ADJ (45; 1% instances), NOUN-PROPN (38; 1% instances), VERB-NOUN (33; 1% instances), NUM-NUM (31; 1% instances), ADJ-NOUN (30; 1% instances), NOUN-ADJ (30; 1% instances), ADV-ADV (29; 1% instances), PRON-PRON (16; 1% instances), ADP-ADP (13; 0% instances), NOUN-NUM (12; 0% instances), NOUN-X (11; 0% instances), NOUN-PRON (10; 0% instances), PRON-NOUN (10; 0% instances), PROPN-X (10; 0% instances), ADJ-ADV (6; 0% instances), NOUN-ADV (6; 0% instances), VERB-PRON (6; 0% instances), ADV-NOUN (5; 0% instances), NUM-ADJ (5; 0% instances), PROPN-VERB (5; 0% instances), VERB-ADV (5; 0% instances), X-NOUN (5; 0% instances), X-X (5; 0% instances), PROPN-PRON (4; 0% instances), SCONJ-NOUN (4; 0% instances), VERB-ADP (4; 0% instances), VERB-PART (4; 0% instances), VERB-SCONJ (4; 0% instances), ADV-VERB (3; 0% instances), NOUN-SYM (3; 0% instances), PRON-PROPN (3; 0% instances), PRON-VERB (3; 0% instances), PROPN-ADJ (3; 0% instances), PROPN-ADV (3; 0% instances), ADJ-PROPN (2; 0% instances), AUX-AUX (2; 0% instances), CCONJ-NOUN (2; 0% instances), CCONJ-PRON (2; 0% instances), CCONJ-X (2; 0% instances), NOUN-ADP (2; 0% instances), NUM-NOUN (2; 0% instances), PRON-ADJ (2; 0% instances), SCONJ-VERB (2; 0% instances), SYM-NOUN (2; 0% instances), SYM-SYM (2; 0% instances), VERB-AUX (2; 0% instances), VERB-NUM (2; 0% instances), ADJ-PART (1; 0% instances), ADJ-PRON (1; 0% instances), ADP-ADV (1; 0% instances), ADP-PART (1; 0% instances), ADV-ADJ (1; 0% instances), ADV-ADP (1; 0% instances), ADV-PRON (1; 0% instances), CCONJ-NUM (1; 0% instances), CCONJ-PROPN (1; 0% instances), INTJ-INTJ (1; 0% instances), INTJ-VERB (1; 0% instances), NUM-ADV (1; 0% instances), NUM-VERB (1; 0% instances), PART-VERB (1; 0% instances), SCONJ-SCONJ (1; 0% instances), VERB-PROPN (1; 0% instances), VERB-X (1; 0% instances).



Universal Dependencies - przykłady

Relacja	Przykłady z **nadrzędnikiem** i podrzędnikiem
NSUBJ	Po bieganiu coś się wam od życia **należy**.
DOBJ	We **booked** her the last flight to Boston.
IOBJ	Żołnierze zamierzali **wywieźć** uchodźców ciężarówkami.
NMOD	We took the evening **flight**.
AMOD	Dorota z trudem przepchnęła słowa przez zaciśnięte **gardło**.
NUMMOD	Urząd nie będzie porównywał obu **PIT-ów**.
APPOS	Projekt został podpisany przez **cesarza** Wilhelma II.
DET	- Nie wiem, jak zrodziła się ta **inicjatywa**.
CONJ	Zmiany wart są coraz **częstsze** i szybsze.
CC	Zmiany wart są coraz częstsze i **szybsze**.
CASE	Book the flight through **Warsaw**.

Składnica

Składnica (zależnościowa)
http://zil.ipipan.waw.pl/Składnica

Składnica — a treebank of Polish

All listed resources have been made available under the GPLv3 license.

›Treebank search engine

Development version of Składnica

This version is the result of development in the project NEKST and in two stages of CLARIN-PL (CLARIN-PL, CLARIN-PL-2).

- Constituency forests as XML files, version 2018.07.23 (final result of CLARIN-PL-2, using Walenty notation for phrase types; 13035 full trees) Składnica-frazowa-180723.tar.gz

Składnica v.½ (2011)

The following page presents the results of the research project N N104 224735 *Construction of a treebank for Polish using machine parsing*, financed by the Ministry of Science and Higher Education in 2008-2011.

Składnica frazowa — constituency treebank

The primary resource presented is the constituency treebank (*Składnica frazowa*), version 0.5. The treebank is a result of parsing 20,000 Polish sentences with the syntactic parser Świgra. For every sentence, the parser generates all possible syntactic parse trees predicted by the rules of its grammar. Within the Dendrarium system, a single correct parse tree has been selected for each sentence by linguists (termed "dendrologists"). Dendrologists have established parse trees for 8,227 sentences to be correct. Other sentences under consideration have undergone classification on the basis of their (un)grammaticality and reasons for their rejection by the parser. The largest class among the rejected sentences consists of utterances with no finite verb. Their analysis of which will be a subject of separate research.

- Constituency forests as XML files: Składnica-frazowa-0.5.tar.bz2
  The files contain all trees generated by the parser, the interpretation selected by dendrologists is marked through attributes.
- XML schema for the constituency treebank files: Składnica-frazowa.xsd
- Trees in the Tiger XML format: Składnica-frazowa-0.5-TigerXML.xml.gz
  The format represents parse trees selected by dendrologists only (one interpretation per sentence).

Składnica zależnościowa — dependency treebank

The dependency treebank (*Składnica zależnościowa*), version 0.5, is a result of an automatic conversion of manually disambiguated constituency trees into dependency structures.

Składnica zależnościowa

(Wróblewska, 2012, 2014)

- struktury zależnościowe automatycznie przekonwertowane z drzew składnikowych (Woliński i in., 2011)
- krawędziom w drzewach przypisane etykiety bazujące na zbiorze zdefiniowanych reguł
- typy relacji zależnościowych: `http://zil.ipipan.waw.pl/FunkcjeZaleznosciowe`
- ponad 8 tys drzew

Zalety banków drzew

Tworzenie banku drzew może wydawać się o wiele wolniejsze i mniej pożyteczne niż zbudowanie gramatyki, ale ma to wiele zalet:

- re-używalność: wiele parserów, tagerów części mowy itp. może być zbudowane w oparciu o nie
- szerokie pokrycie, nie tylko kilka intuicji
- informacje o częstości i rozkładzie
- sposób ewaluacji systemów ('złoty standard')

Jak możemy zdecydować, które słowa zależą od których?

Źródła informacji dla parserów zależnościowych

- powinowactwa leksykalne ([dyskusja → zagadnień])
- odległość pomiędzy zależnymi wyrazami
- treść pomiędzy (zależności rzadko rozciągają się poprzez czasowniki i znaki przestankowe)
- walencja nadrzędników (walencja = liczba argumentów, z jakimi zazwyczaj 'łączy się' dany wyraz, przede wszystkim czasownik)

Rodzaje parserów zależnościowych

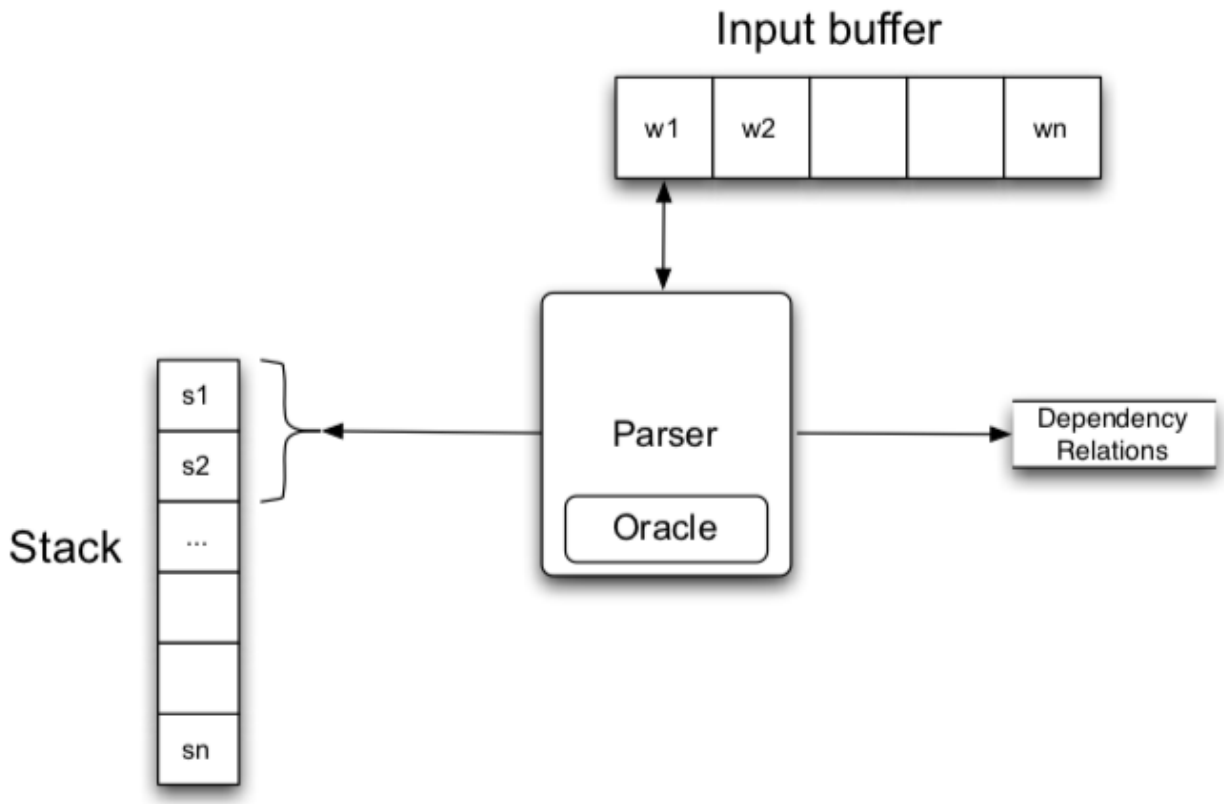
- oparte na programowaniu dynamicznym
- oparte na algorytmach spełnienia ograniczeń (ang. *constraint satisfaction*)
- grafowe
- oparte na przejściach

Parsery zależnościowe trenowane z nadzorem

:

- grafowe: mając na wejściu zdanie, parser definiuje zbiór możliwych drzew zależnościowych jako kandydatów, którzy są następnie oceniani na podstawie wytrenowanego modelu i wybierane jest nawyżej ocenione drzewo
- oparte na przejściach: budują optymalną sekwencję przejść (wybór elementów do dołączenia) na podstawie wytrenowanego modelu uczenia maszynowego

Podstawowy parser zależnościowy oparty na przejściach



źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, September, 2018, `https://web.stanford.edu/~jurafsky/slp3/`

Konfiguracja

- Konfiguracja składa się ze:
  - stosu,
  - bufora wejściowego wyrazów lub tokenów,
  - zestawu relacji reprezentujących drzewo zależności.
- Proces analizy składa się z sekwencji przejść w przestrzeni możliwych konfiguracji
- Celem tego procesu jest znalezienie ostatecznej konfiguracji, w której uwzględniono wszystkie wyrazy i zsyntetyzowano odpowiednie drzewo zależności

Parser zależnościowy: podejście standardowe

(Covington 2001)
Operatory (akcje) używane do wyprodukowania nowych konfiguracji, poprzez analiozowanie wyrazów w pojedynczym przejściu wejścia, od lewej do prawej:

- przypisz bieżący wyraz jako nadrzędnik wcześniej widzianego wyrazu,
- przypisz jakiś wcześniej widziany wyraz jako nadrzędnik bieżącego wyrazu,
- lub odrocz zrobienie czegokolwiek z bieżącym wyrazem, dodaj go do sstosu w celu późniejszego przetworzenia.

Parser zależnościowy: podejście standardowe, łuki (*arc-standard transition-based parser*)

(Nivre 2003)
W celu bardziej precyzyjnych akcji, definiujemy trzy operatory przejścia, operujące na dwóch najwyższych elementach stosu:

- LEFTARC: przypisz relację zależną od nadrzędnika pomiędzy wyrazem na szczycie stosu i wyrazem bezpośrednio pod nim; usuń niżej leżący wyraz ze stosu,
- RIGHTARC: przypisz relację zależną od nadrzędnika pomiędzy drugim wyrazem od szczytu stosu i wyrazem na szczycie stosu; usuń wyraz ze szczytu stosu,
- SHIFT: usuń wyraz z początku bufora wejściowego i odłóż go na stos.

Ogólny, zachłanny algorytm parsera zależnościowego opartego na przejściach

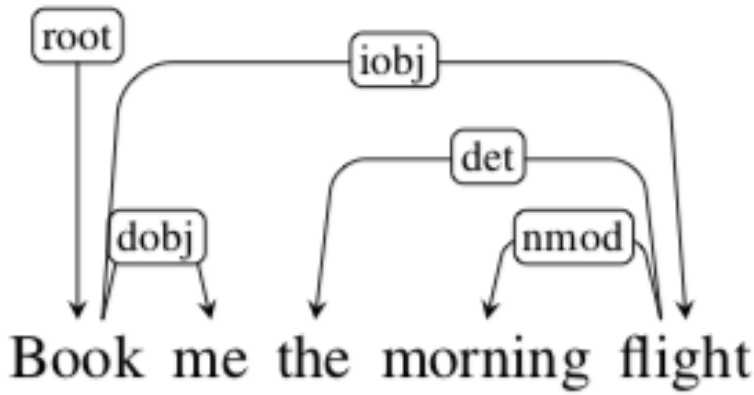
function dependency_parse(wyrazy) **returns** drzewo zależnościowe

1: stan $\leftarrow$ {[`root`], [wyrazy], []} ; *początkowa konfiguracja dla stanu, który nie jest finalny*
2: $t \leftarrow$ ORACLE(stan) ; *wybór operatora przejścia*
3: stan $\leftarrow$ APPLY(t, stan) ; *aplikacja operatora, utworzenie nowego stanu*
4: **return** stan

Ogólny, zachłanny algorytm parsera zależnościowego opartego na przejściach c.d.

Złożoność tego algorytmu jest liniowa względem długości zdania (pojedyncze przejście od lewej do prawej, każdy wyraz jest najpierw odkładany na stos, a później redukowany).

Przykład



źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, September, 2018, `https://web.stanford.edu/~jurafsky/slp3/`

Przykład c.d.

Step	Stack	Word List	Action	Relation Added
0	[root]	[book, me, the, morning, flight]	SHIFT	
1	[root, book]	[me, the, morning, flight]	SHIFT	
2	[root, book, me]	[the, morning, flight]	RIGHTARC	(book → me)
3	[root, book]	[the, morning, flight]	SHIFT	
4	[root, book, the]	[morning, flight]	SHIFT	
5	[root, book, the, morning]	[flight]	SHIFT	
6	[root, book, the, morning, flight]	[]	LEFTARC	(morning ← flight)
7	[root, book, the, flight]	[]	LEFTARC	(the ← flight)
8	[root, book, flight]	[]	RIGHTARC	(book → flight)
9	[root, book]	[]	RIGHTARC	(root → book)
10	[root]	[]	Done	

źródło: "*Speech and Language Processing (3rd ed. draft)*", Dan Jurafsky and James H. Martin. Draft chapters in progress, September, 2018, `https://web.stanford.edu/~jurafsky/slp3/`

Konstruowanie wyroczni (ORACLE)

- najnowocześniejsze systemy oparte na przejściach wykorzystują nadzorowane metody uczenia maszynowego do uczenia klasyfikatorów, które pełnią rolę wyroczni
- mając dane odpowiednie dane trenujące, metody te uczą się funkcji odwzorowującej konfiguracje na operatory przejścia
- każda akcja jest przewidywana przez klasyfikator:
  - maksymalnie 3 ogólne wybory; maksymalnie $|R| \times 2 + 1$, gdy weźmiemy pod uwagę typy relacji (LEFTARC jako `OBJ` itd., około 80 klas)
  - cechy (kategoryczne): wyraz na szczycie stosu, POS; pierwszy wyraz w buforze, POS, itd.
- z początku uzywane były 'konwencjonalne' klasyfikatory: regresja logistyczna, SVM,
- brak przeszukiwania (ale można zastosować przeszukiwanie wiązkowe)

Wady podejść opartych na 'konwencjonalnych' klasyfikatorach

- Problem 1: bardzo rzadkie cechy (np. rzędu 15 mln cech)
- Problem 2: niekompletność (wiele konfiguracji nie pojawiło się wcześniej, na etapie trenowania, brak dla nich cech)
- Problem 3: kosztowne obliczenia (więcej niż 95% czasu parsowania jest poświęconych na liczenie cech)

Wady podejść opartych na 'konwencjonalnych' klasyfikatorach

Alternatywne podejście: wyuczenie gęstej i kompaktowej reprezentacji cech (około 1000 cech)

Neuronowy parser zależnościowy

(Chen & Manning 2014)

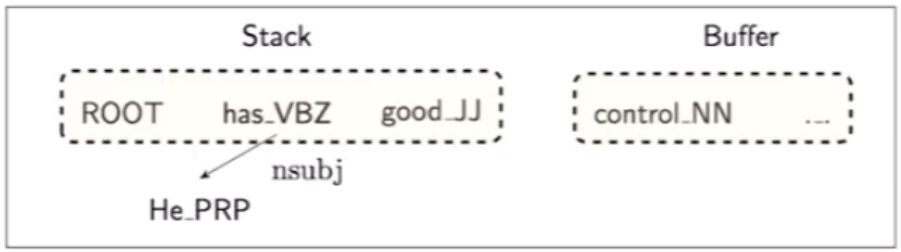
- rozproszone reprezentacje wektorowe
- ekstrakcja tokenów i następnie tworzenie reprezentacji wektorowych z konfiguracji

Rozproszone reprezentacje wektorowe

- każdy wyraz reprezentowany jako d-wymiarowy gęsty wektor (tj. *word embedding*)
  - podobne wyrazy będą miały zbliżone wektory
- części mowy (part-of-speech, POS) i etykiety zależnościowe także reprezentowane jako d-wymiarowe wektory
  - mniejsze, dyskretne zbiory także wykazują wiele semantycznych podobieństw, np. NNS (plural noun) jest bliskie NN (singular noun)

Ekstrakcja tokenów i następnie tworzenie reprezentacji wektorowych z konfiguracji

- ekstrakcja tokenów, bazując na pozycjach w stosie / buforze
- konwertowanie ich na *embeddingi* i ich konkatenacja
- jakakolwiek konfiguracja parsera jest reprezentowana jako wektor z około 1000 wymiarów



	word	POS	dep.
s_1	good	JJ	∅
s_2	has	VBZ	∅
b_1	control	NN	∅
$lc(s_1)$	∅	∅	∅
$rc(s_1)$	∅	∅	∅
$lc(s_2)$	He	PRP	nsubj
$rc(s_2)$	∅	∅	∅

Architektura modelu

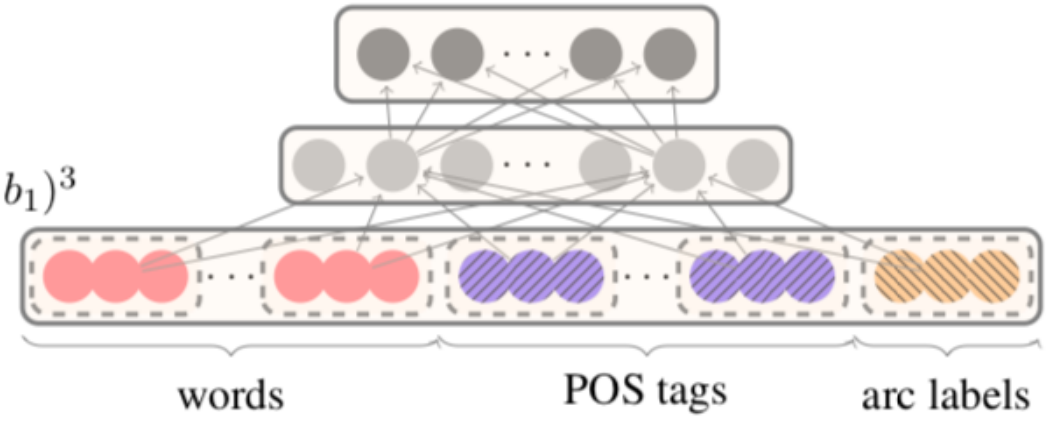
Softmax layer:

$$p = \texttt{softmax}(W_2 h)$$

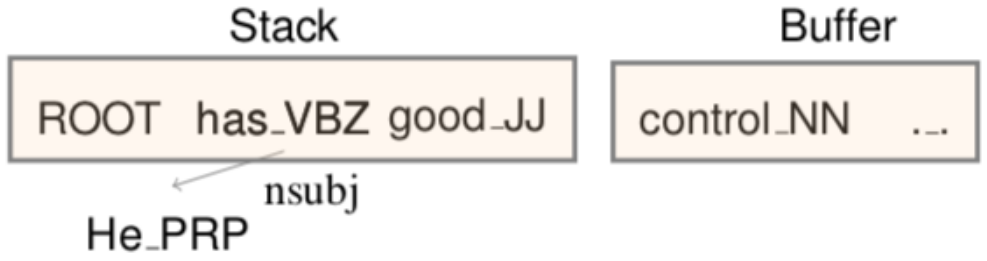
Hidden layer:

$$h = (W_1^w x^w + W_1^t x^t + W_1^l x^l + b_1)^3$$

Input layer: $[x^w, x^t, x^l]$



Configuration



Dziękuję za uwagę!