Algorytmy parsingu zależnościowego, metryki i sieci neuronowe

Paweł Rychlikowski

Instytut Informatyki UWr

7 lutego 2019

Universal Deppendencies

- Dane o rozbiorach (i POS-tagach) w ujednoliconym formacie, dla olbrzymiej liczby języków.
- Prosty, czytelny format tekstowy typu *tsv* (CONLL**U**)
- Dla każdego języka osobne zbiory *train* i *test*
- Informacje o charakterze każdego korpusu: (zob. strona www)
- Spróbujmy odcyfrować fagment jednego rozbioru

Universal Deppendencies. Przykład

1	Zwierzę	zwierzę	NOUN	subst:sg:nom:n		2	nsubj	-	-
2	polubiło	polubić	VERB	praet:sg:n:perf		0	root	-	-
3	piwko	piwko	NOUN	subst:sg:acc:n		2	obj	-	-
4	i	i	CCONJ	conj	-	8	cc	-	-
5	bez	bez	ADP	prep:gen:nwok		6	case	-	-
6	kufla	kufel	NOUN	subst:sg:gen:m3		8	obl	-	-
7	nie	nie	PART	qub	-	8	advmod	-	-
8	daje	dawać	VERB	fin:sg:ter:imperf		2	conj	-	-
9	się	się	PRON	qub		8	expl:pv	-	-
10	wydoić	wydoić	VERB	inf:perf		8	xcomp	-	x
11	.	.	PUNCT	interp	-	2	punct	-	-

Usunięta kolumna

Nie zmieściła się kolumna, w której znajdują się umiędzynarodowione parametry gramatyczne słowa:

- **zwierzę** – Case=Nom|Gender=Neut|Number=Sing
- **piwko** – Case=Acc|Gender=Neut|Number=Sing
- **bez** – AdpType=Prep|Case=Gen|Variant=Short
- **się** – PronType=Prs|Reflex=Yes

W ostatniej kolumnie zamiast **x** powinno być SpaceAfter=No
Przedostatnia kolumna to zależności *siostrzane*

Relacje

Clausal Argument Relations	Description
NSUBJ	Nominal subject
DOBJ	Direct object
IOBJ	Indirect object
CCOMP	Clausal complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numeric modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions, postpositions and other case markers
Other Notable Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction

Figure 13.2 Selected dependency relations from the Universal Dependency set. (de Marneffe et al., 2014)

Źródło: Speech and Language Processing (3rd ed. draft), Jurafsky, Martin

Problem z „i”

Problem z 'Jasiem i Małgosią' (jak wyglądać ma rozbiór zdania Jaś i Małgosia zjedli pierniki). Opcje:

1. zjedli(i(jaś,małgosia), pierniki)
2. zjedli(jaś, małgosia, pierniki)
3. zjedli(jaś(i(małgosia)), pierniki)
4. zjedli(jaś(i, małgosia)), pierniki)

W pierwszym przypadku mamy niezbyt ciekawe połączenie zjedli(i), w drugim – pomijamy i, a ponadto mamy połączenia: zjedli(jaś) oraz zjedli(małgosia), w trzecim i czwartym pozostaje problen niezgodności liczby, do tego Jaś i Małgosia nie są równo traktowani.

Problem z „i”

Problem z 'Jasiem i Małgosią' (jak wyglądać ma rozbiór zdania Jaś i Małgosia zjedli pierniki). Opcje:

1. zjedli(i(jaś,małgosia), pierniki) (stara Składnica)
2. zjedli(jaś, małgosia, pierniki)
3. zjedli(jaś(i(małgosia)), pierniki)
4. zjedli(jaś(i, małgosia)), pierniki) (UD)

W pierwszym przypadku mamy niezbyt ciekawe połączenie zjedli(i), w drugim – pomijamy i, a ponadto mamy połączenia: zjedli(jaś) oraz zjedli(małgosia), w trzecim i czwartym pozostaje problen niezgodności liczby, do tego Jaś i Małgosia nie są równo traktowani.

Koniunkcje i przyimki w UD

1	A	CCONJ	conj	3	cc
2	co	PRON	subst:sg:nom:n	3	nsubj
3	ma	VERB	fin:sg:ter:imperf	0	root
4	być	AUX	inf:imperf	6	cop
5	jej	PRON	ppron3:sg:gen:f:ter:akc:npraep	6	nmod:p
6	spoiwem	NOUN	subst:sg:inst:n	3	xcomp
7	i	CCONJ	conj	8	cc
8	siłą	NOUN	subst:sg:inst:f	6	conj
9	napędową	ADJ	adj:sg:inst:f:pos	8	amod
10	?	PUNCT	interp	3	punct

Narysujmy fragment z koniunkcją.

Koniunkcje i przyimki w UD

1	-	PUNCT	interp	5	punct
2	A	CCONJ	conj	5	cc
3	jeszcze	PART	qub	4	advmod
4	niedawno	ADV	adv:pos	5	advmod
5	mówił	VERB	praet:sg:m1:imperf	0	root
6	eś	AUX	aglt:sg:sec:imperf:wok	5	aux:aglt
7	o	ADP	prep:loc	8	case
8	pani	NOUN	subst:sg:loc:f	5	obl
9	Gronkiewicz	PROPN	subst:sg:loc:f	8	flat
10	tak	ADV	adv	11	advmod
11	ciepło	ADV	adv:pos	5	advmod
12	.	PUNCT	interp	5	punct

Narysujmy fragment z przyimkiem (mówiłeś o pani Gronkiewicz)

Korzyści z wielojęzykowości

Korpus słowacki

Czy rozumiemy zdanie:

Radikálnou inováciou je hláskoslovie a tvaroslovie.

A czy umiemy je sparsować?

Transfer learning

Są możliwości łączenia korpusów – możemy czegoś się dowiedzieć o parsingu po polsku patrząc na słowacki (szczególnie istotne, gdy mamy język z małym korpusem (polski) podobny do innego z dużym (czeski).

Można pokazać, że uczenie wielojęzykowe parserów daje rzeczywistą, mierzalną poprawę!

Generowanie przy użyciu DG. Kilka idei

- Wyrazy będące liśćmi (z klas otwartych) można zamieniać bez obawy o poprawność gramatyczną zdania.
- Wyrazy mają typy (popatrzmy na plik z typami).
- Wyrazy o tym samym typie można zamieniać ze sobą (ewentualnie pilnując, aby się zbytnio nie zmieniła semantyka).

Uwaga

Losujemy dzieci dla zadanego rodzica (zarówno rodzic, jak i dzieci mają tagi). Losowanie przypomina losowanie bigramowe.

Deppendency parsing. Algorytm dynamiczny, wersja projekcyjna

- W przypadku parsingu projekcyjnego też tworzą się frazy, odpowiadające poddrzewom rozbioru.
- To są specyficzne poddrzewa, w których możemy pominąć pewne „zewnętrzne" gałęzie na najwyższym poziomie. Przykładowo dla

  *miała(babuleńka, *, koziołki(dwa,rogate,*))*

  poddrzewami są

  *miała(babuleńka, *)*
  miała(, koziołki(dwa,rogate,*))*
  koziołki(rogate,)*

- Parsing polega na łączeniu takich fraz (poddrzew) ze sobą.

Zadanie parsingu

Mamy znaleźć graf, taki że:

a) Jest drzewem, z korzeniem w root, z węzłami w wyrazach w zdaniu.

b) Strzałki mają określone etykiety.

c) Minimalizujemy jakąś funkcję kosztu:
   - Wariant 0/1 – pewne połączenia niemożliwe (1), pewne dozwolone (0)
   - Można też: dozwolone == obserwowane w korpusie
   - Można też szacować wartość prawdopodobieństwa połączenia (koszt = log. prawdopodobieństwa)
   - Preferujemy bliskie połączenia (składowa odległościowa w prawdopodobieństwie), być może zależna od typu słowa (przyimki łączą się blisko, czasownik ma odległe dzieci, itd).

d) Być może dodatkowe właściwości: jest projekcyjny, albo spełnia dodatkowe więzy, typu czasownik ma jeden podmiot, albo: dwa rzeczowniki połączone **conj** implikują obecność spójnika.

Algorytm dynamiczny, wersja projekcyjna

- Odpowiedni struktury U (dla CYK) pamięta teraz dla każdego przedziału informację o najlepszych drzewach zakorzenionych we wszystkich słowach.
- Jak łączymy dwie takie struktury, to musimy rozważyć połączenie każdego słowa z lewej z każdym z prawej (w obie strony) i wybrać takie, które maksymalizuje

$$v(w_1) + v(w_2) + \max\{v(w_1 \rightarrow w_2), v(w_1 \leftarrow w_2), \}$$

gdzie w_1 jest z lewej części, a w_2 – z prawej.

Złożoność

Niestety złożoność jesst dość duża: (jaka?) algorytm pracuje w czasie $O(N^5)$.
Da się to poprawić: Algorytmem Elsnera (ale o nim nie powiemy).

Algorytm Chu-Liu Edmonds: idea

CLU działa wg następującego schematu:

1. Każdy wybiera ulubionego tatusia
2. Jak nie ma cyklu – ok.
3. Jak jest cykl, to zamieniamy cykl na jeden sztuczny węzeł, rekurencyjnie rozwiązujemy mniejsze zadanie, potem rozbijamy cykl.

Złożoność

Algorytm działa w czasie $O(N^2)$

Algorytm Chu-Liu Edmonds współcześnie

- Algorytm CLE na początku wybiera optymalnego tatusia.
- Moglibyśmy na tym poprzestać, ale czasami będą cykle.
- Ale z drugiej strony, taki graf z cyklami możemy ocenić (sprawdzamy poprawność strzałek)

Uwaga

Współczesne algorytmy parsujące często nie przejmują się cyklami, bo w niektórych przypadkach usuwanie cykli psuje (wyjaśnienie na tablicy).

Projekcyjne vs nieprojekcyjne

- Szukanie drzew nieprojekcyjnych jest szybsze.
- Wiemy, że algorytm projekcyjny w wielu sytuacjach nie znajdzie poprawnego rozbioru.
- Z drugiej strony wydaje się, że warto preferować rozbiory projekcyjne

Uwaga

Częściowo rozwiązuje ten problem preferowanie połączeń niezbyt odległych.

Deterministyczny parsing

- Motywacja: nasz wewnętrzny parser (raczej) nie wykonuje żadnego dynamicznego algorytmu $O(N^5)$, a mimo to rozumiemy, co do nas mówią (na ogół)
- Wydaje się, że o rozbiorze podejmujemu dezycję lokalnie, czasem – raczej rzadko – wykonując coś w rodzaju płytkiego nawrotu.

Uwaga

Istnieje dużo algorytmów tego typu, przeglądających wejście od lewej do prawej i tworzących przyrastająco rozbiór. Oczywiście są one szybkie i (co ciekawe) wcale niekoniecznie gorsze od wolniejszych optymalizacyjnych.

Garden path sentences

Można „zhakować” nasz wewnętrzny parser za pomocą specjalnych zdań (tzw. garden patsh sentences)

The old man the boat.

Rozbiorem jest:

man(old(the),boat(the))

ale problem z rozbiorem jest taki, że bardzo nas sugeruje pojawienie się ciągu wyrazów the old man

Transition-based Deppendency Parsing

- Zarządzamy następującymi strukturami:
  1. Listą nieprzetworzonych słów (sufiksem zdania)
  2. Zbiorem strzałek, czyli znalezionych relacji (słowo $\rightarrow^{lab}$ słowo)
  3. Stosem słów, na który odkładamy słowa jeszcze nieprzetworzone
- Dysponujemy wyrocznią, która podejmuje decyzję, jaką akcję wybrać (S[i] – wierzchołek stosu, S[i-1] – przedostatni):
  1. LeftArc – dodaj strzałkę S[i-1] $\leftarrow$ S[i], usuń S[i-1]
  2. RightArc – dodaj strzałkę S[i-1] $\rightarrow$ S[i], usuń S[i]
  3. Shift – wrzuć na stos kolejne słowo z wejścia
- Popatrzmy na tablicy na zdanie: młoda kobieta płynie starą łódką wesoło (zwróćmy uwagę na projekcyjność i ją zaburzmy)