Parsing i statystyka

Paweł Rychlikowski

Instytut Informatyki UWr

7 lutego 2019

Chunking. Przypomnienie

- Chunking to ekstremalnie płaski parsing.
- Dzielimy tekst na spójne kawałki, niektórym z nich przypisujemy etykiety (np. np, date, person)

Chunking i HMM

- Stanami ukrytymi są znaczniki:
  1. Znacznik **O** (poza frazą)
  2. Znaczniki $\mathbf{B}_c$ (gdzie c jest rodzajem frazy, na przykład date albo person czy name), mówiące o *początku* frazy typu c
  3. Znaczniki $\mathbf{I}_c$ mówiące o *wnętrzu bądź końcu* frazy typu c
- Przykład:

```
Jan Kowalski pojechał do Jeleniej Góry i Wałbrzycha .

B_p I_p      O        O  B_c      I_c  O B_c         O
```

Chunking i HMM

- **Pytanie**: dlaczego potrzebne są znaczniki *B* i *I*?
  (Jan Kowalski Janinę Nowacką kochał bez pamięci!)
- Inna możliwość: tagujemy przerwy za słowem (na tablicy)

Uwaga

Działa dokładnie ten sam kod, co w przypadku POS-taging!

Named Entity Recognition

Named Entities Recognition

Zadanie Named Entities Recognition jest zadaniem identyfikacji fraz w tekście, będących nazwami własnymi konkretnych obiektów, takich jak osoby, miejsca, instytucje.

- Czasem rozszerza się to do: dat, liczb z jednostkami, kwot pieniężnych.
- Listę można dość dowolnie rozszerzać (propozycje?)
  - Odnośniki do tekstów prawnych, na przykład:
    - art. 43 ust. 1 pkt 3 ustawy o podatku od towarów i usług, albo
    - art. 75 ust. 1 lit. c pkt i rozporządzenia Rady (WE) nr 1698/2005 z dnia 20 września 2005 r.
  - Pozycje GPS
  - Różne „byty informatyczne" (URL, e-mail, IP)

NER. Podsumowanie

- W pewnych sytuacjach jest to **dokładnie** to samo, co chunking.
- Dość naturalne jest pewne ograniczone zagnieżdżanie fraz
  - Dyrekcja III LO im. Adama Mickiewicza
    (2 frazy typu organizacja i jedna fraza typu osoba)

Jak rozwiązać problem z zagnieżdżaniem?

Można wprowadzić znaczniki opisujące strukturę, przykładowo dla Adama mielibyśmy $I_{\text{org}}\, I_{\text{org}}\, B_{\text{person}}$

Probabilistyczne gramatyki bezkontekstowe. Wprowadzenie

- Ze względu na dużą liczbę rozbiorów potrzebujemy narzędzi do rozróżniania między nimi.
- Można to robić przypisując każdemu rozbiorowi jego wartość i wybierać potem ten o największej wartości.
- Wartość może być związana z prawdopodobieństwem rozbioru.

Spróbujmy zastanowić się, jak najprościej dodać prawdopodobieństwo do gramatyk bezkontekstowych.

PCFG. Definicja

- Mamy symbole nieterminalne, terminalne, produkcje, drzewa wyprowadzeń, etc
- Każdej produkcji przypisujemy prawdopodobieństwo (liczbę od 0 do 1)
- Pilnujemy, żeby prawdopodobieństwa dla każdego nieterminala sumowały się do jedynki.

Prawdopodobieństwa odczytujemy jako

$$P(B \to \gamma | B)$$

czyli prawdopodobieństwo, że w drzewie dany symbol rozwinie się za pomocą odpowiedniej produkcji, pod warunkiem, że rozwija się właśnie ten symbol.

Prawdopodobieństwo rozbioru

- Najlepszy rozbiór dla zdania S to

$$\hat{T}(S) = \text{argmax}_{T,\ \text{plon}(T)=S} P(T|S)$$

- Mamy ponadto

$$P(T|S) = \frac{P(S|T)P(T)}{P(S)}$$

oraz $P(S|T) = 1$

- Czyli szukamy

$$\hat{T}(S) = \text{argmax}_{T,\ \text{plon}(T)=S} P(T)$$

Prawdopodobieństwo rozbioru (2)

Obliczanie P(T)

Prawdopodobieństwo $P(T)$ liczymy mnożąc wszystkie prawdopodobieńtwa z wszystkich produkcji.

Probabilistyczne CYK. Postać normalna Chomskiego

- Gramatyka PCFG może być również w postaci normalnej (np. Chomskiego)
- Procedura sprowadzania nie różni się zbytnio, należy pamiętać, żeby dodawać jedynki do sztucznie wprowadzonych symboli nieterminalnych

Przykład

Produkcję $A \rightarrow BCDE$ $[p]$ zamieniamy na 2 produkcje:

- $A \rightarrow BX_{CDE}$ $[p]$
- $X_{CDE} \rightarrow CDE$ [1.0]

Algorytm Cocke–Younger–Kasami. Wersja PCFG

- W oryginale dla każdego i,j obliczaliśmy zbiór symboli nieterminalnych, takich że:

$$U(i,j) = \{X \mid X \Rightarrow^* s[i:j]\}$$

- A w wersji PCFG?

Odpowiedź

Dla każdego symbolu z $U(i,j)$ musimy pamiętać jego najmniejszy koszt, czyli największe prawdopodobieństwo.

Algorytm CYK-PCFG (2)

- Dla prostoty zapisu założymy, że pamiętamy wszystkie pary (N, p) mówiące o tym, że jakiś nieterminal jest osiągalny z p-stwem p.
- Jeżeli $j = i + 1$, wówczas U[i,i+1] jest równe:

$$\{(X, p) | (X \to s[i] : p) \in P\}$$

gdzie P jest zbiorem produkcji

- Dla $j - i > 1$ zakładamy, że mamy policzone U dla mniejszych rozpiętości i

$$U(i,j) = \bigcup_{k=1,\ldots,j} \{ \quad | \ (X \to AB : p) \in P \wedge (A, p_a) \in U[i,k] \wedge (B, p_b) \in U[k,j] \}$$

Uwaga

Oczywiście w prawdziwym algorytmie nie będziemy pamiętać zbioru par (N,p), tylko słownik, który N przypisuje największe p.

Algorytm CYK-PCFG (2)

- Dla prostoty zapisu założymy, że pamiętamy wszystkie pary (N, p) mówiące o tym, że jakiś nieterminal jest osiągalny z p-stwem p.
- Jeżeli $j = i + 1$, wówczas U[i,i+1] jest równe:

$$\{(X, p) | (X \to s[i] : p) \in P\}$$

gdzie P jest zbiorem produkcji

- Dla $j - i > 1$ zakładamy, że mamy policzone U dla mniejszych rozpiętości i

$$U(i,j) = \bigcup_{k=1,\ldots,j} \{(X, p \cdot p_a \cdot p_b) \mid (X \to AB : p) \in P \wedge (A, p_a) \in U[i,k] \wedge (B, p_b) \in U[k,j]\}$$

Uwaga

Oczywiście w prawdziwym algorytmie nie będziemy pamiętać zbioru par (N,p), tylko słownik, który N przypisuje największe p.

Wyznaczanie prawdopodobieństw

- Zakładamy, że mamy bank drzew
- Wówczas możemy łatwo policzyć wszystkie prawdopodobieństwa:

$$P(A \to \beta | A) = \frac{C(A \to \beta)}{C(A)}$$

Uwaga

Nie potrzebujemy gramatyki innej niż ta, którą dedukujemy z Treebanku. Dlaczego?

(Bo inne produkcje i tak miałyby prawdopodobieństwo równe 0)

Problemy z PCFG

PCFG są fajne (bo proste), ale w wielu istotnych miejscach nie dają rady dobrze opisać języka. Dlaczego?

Problem 1

P-stwo tego, do czego rozwinie się NP nie zależy od tego, gdzie to NP było wprowadzone do zdania. Przykładowo (j.ang):

	Pronoun	Non-Pronoun
Subject	91%	9%
Object	34%	66%

Problemy z PCFG (cd)

Problem 2

Brak zależności leksykalnych. Rozważmy następującą gramatykę (j.ang.), do użycia w rozbiorze zdań:

He examined a man with a sthetoscope.
He examined a man with a broken leg.

```
VP -> VBD NP PP
VP -> VBD NP
NP -> NP PP
S  -> 'he' VP
```

Po dodaniu prawdopodobieństw będzie ona **zawsze** preferowała wiązanie PP do czasownika (lub zawsze do rzeczownika).

Inne przykłady na problemy w języku polskim

- Przymiotnik przed czy po rzeczowniku? (głupi miś vs miś polarny)
- Przypisanie ram walencyjnych słowom z określonym prawdopodobieństwem.
  - Prosty przykład: czasowniki upaść i przeczytać mają różne prawdopodobieństwo pozostania bez dopełnienia.

Pomysł 1: Podział nieterminali

- Problem pierwszy można rozwiązać dzieląc nieterminale, czyli rozważać osobno NP_{subj} oraz osobno NP_{obj}.
- Można to zrobić automatycznie, definiując generalnie nieterminale z „anotacją rodzicielską”, czyli gramatyka zrobiłaby się automatycznie czymś takim

```
S -> NP^S VP^S
NP^S -> ... #dla podmiotu
VP^S -> V NP^VP
NP^VP -> ... #dla dopełnienia
```

- Istnieją algorytmy, które dobierają dla gramatyki i treebanku odpowiedni stopień szczegółowości podziału (bo nie wszystko warto dzielić).

Pomysł 2: leksykalizowana gramatyka PCFG

- Wchodzi do gry parametr HEAD.
- Prawdopodobieństwa zależą od głów poszczególnych fraz.
- Popatrzmy jak to by mogło wyglądać na przykładzie typów z Walentego.

Walenty – PCFG

```
czytać: subj{np(str)}+{np(dat)}+{cp(że)}
czytać: subj{np(str)}+{prepnp(o,loc)}+{cp(że)}
zakrywać: subj{np(str)}+{np(inst)}+{np(acc)}
              +{prepnp(przed,inst)}+{refl}
```

Przykładowe zdania:

Stefan czytał Judycie, że w Kielcach pączków nigdy nie robi się z budyniem.
Judyta czytała o pączkach, że nie robi się ich z budyniem.
Stefan zakrył sobie ścierką kolano przed spojrzeniem Judyty.

Zwykła gramatyka PCFG

Stefan czytał Judycie, że w Kielcach pączków nigdy nie robi się z budyniem.
Judyta czytała o pączkach, że nie robi się ich z budyniem.
Stefan zakrył sobie ścierką kolano przed spojrzeniem Judyty.

```
S -> NP(nom) V NP(dat) , że S [0.002]
S -> NP(nom) V o NP(loc) , że S [0.00123]
S -> NP(nom) V sobie NP(inst) NP(acc)
                    przed NP(inst) [0.00068]
```

Problemy gramatyki z poprzedniego slajdu

Taka gramatyka:

1. Nie zawierałaby informacji o tym, że poprawność (i prawdopodobieństwo) powyższych zdań zależy od doboru czasownika
2. Nie zawierałaby informacji o innych dopuszczalnych kolejnościach argumentów
3. Nie zawierałaby informacji o pomijalności argumentów
4. Nie zawierałaby informacji o tym, że w zależności od roli inaczej powinny rozkładać się prawdopodobieństwa poszczególnych rzeczowników (NP(inst) w dwóch rolach: czym zakrywamy i przed czym zakrywamy)

Próba poprawy (1). Leksykalizacja typów

- Dodajemy parametr Head (dla zdania):

  ```
  S[czytać] -> NP(nom) V[czytać] NP(dat) że S[_]
  ```

  (zdanie złożone może mieć dowolny czasownik)
- Dla każdego podzbioru (akceptowalnego) typu z Walentego dodajemy prawdopodobieństwo i osobną produkcję, żeby można było pisać zdania:

  Judyta zakryła sobie twarz przed słońcem. Stefan zakrył się szerokim krawatem. Wiktor zakrył fotel przed deszczem.

  Ale również:

  Sobie twarz przed słońcem zakryła Judyta. Się Stefan zakrył szerokim krawatem. Fotel przed deszczem zakrył Wiktor.

Próba poprawy (2). Leksykalizacja typów

- Nie wszystkim można zakryć wszystko. Trudniej zakryć hustkę twarzą niż twarz hustką.
- Pełna leksykalizacja mogłaby wyglądać tak:

```
S[H1] -> NP(nom, H2) V[H3] NP(dat, H4) że S[H5]
```

- A prawdopodobieństwo byłoby takie:

```
P(S[czytać] -> NP(nom, kobieta) V[czytać]
      NP(dat, koledze) że S[podrożeć] | S[czytać])
```

- Tego się nie da niestety sensownie oszacować!

Próba poprawy (3). Realistyczna leksykalizacja typów

Zdarzenie z poprzedniego slajdu można rozłożyć jako jednoczesne występowanie następujących rzeczy

- Ktoś czyta komuś że coś się dzieje (rama walencyjna)
- Czyta kobieta
- Ktoś czyta koledze
- Ktoś czyta że podrożeje

To samo, tylko trochę precyzyjniej

- Prawdopodobieństwa powinny być warunkowe, pod warunkiem przyjętej ramy.
- Czyli interesuje nas prawdopodobieństwo tego, że:
  - Czyta kobieta, pod warunkiem, że ktoś komuś że coś czyta
  - Ktoś czyta koledze, pod warunkiem, że ktoś komuś że coś czyta
  - czyta że podrożeje, pod warunkiem, że ktoś komuś że coś czyta

Uwaga

Oczywiście wszystkie prawdopodobieństwa wygładzamy, np. tak, żeby prawdopodobieństwo tego, że kobieta czyta, wynikało trochę z tego, że kobieta robi różne rzeczy (występuje często jako podmiot).

Te prawdopodobieństwa można już szacować z dużych treebanków

Ocena parserów

- Chcemy mieć precyzyjną miarę, mówiącą, że 1 parser jest lepszy od drugiego.
- Możemy testować parsery w ten sposób, że mamy dobre i złe zdania i oczekujemy, że dobry parser parsuje te dobre, a nie parsuje złych.
- Ale to nie jest najlepszy sposób – dlaczego?

Taka ocena nie uwzględnia tego, że nas na ogół interesuje coś więcej niż to, że fraza jest poprawna. Ponadto bardzo trudno tworzyć zbiory 'nie-zdań'.

Gramatyka, która wszystko sparsuje...

i zarazem jest sensowna:

```
S -> ...
S -> Unparsed [0.0000001]
Unparsed -> Phrase Unparsed [0.1] | Phrase [0.9]
Phrase -> NP [0.2]
Phrase -> AdjP [0.2]
Phrase -> PP [0.2]
(...)
NP -> Unparsed [0.0000001]
VP -> Unparsed [0.0000001]
(...)
```

PARSEVAL

- PARSEVAL to powszechnie używana metryka oceniająca drzewa rozbioru.
- Nawiązuje do pojęć z Information Retrieval, precision (dokładność, precyzja) i recall (pokrycie, komplentość)

Uwaga

Na parsing możemy patrzeć jak na zadanie wyszukiwania fraz.

Wyszukiwanie fraz

W zdaniu

> *Stefan w styczniu często myślał, że noworoczne postanowienia są zdecydowanie przeceniane.*

wyszukujemy fraz (pomijamy jednowyrazowe):

> *pp(w styczniu), s(noworoczne postanowienia są zdecydowanie przeceniane), np(noworoczne postanowienia), ap(zdecydowanie przekłamane), s(...całe zdanie...)*

Algorytm parsingu powinien być karany za nieznalezienie którejś z powyższych, za niezgadnięcie typu frazy i za znalezienie czegoś poza tą listą.

Precision/Recall

- Precyzja:

$$P = \frac{\text{Liczba poprawnie odgadniętych fraz w zwróconym rozbiorze}}{\text{Liczba fraz w zwróconym rozbiorze}}$$

- Kompletność:

$$R = \frac{\text{Liczba poprawnie odgadniętych fraz w zwróconym rozbiorze}}{\text{Liczba fraz we wzorcowym rozbiorze}}$$

F_1-score

Gdy chcemy scharakteryzować rozbiór jedną liczbą, używamy średniej harnonicznej P i R,

$$F_1 = (\frac{\frac{1}{P} + \frac{1}{R}}{2})^{-1} = \frac{2PR}{P+R}$$

Inne sposoby oceny

Sposób 1

Czasem rozważa się dwa warianty oceny – z uwzględnieniem nazw fraz oraz bez uzwględniania.

Sposób 2

Cross-brackets – liczba fraz w których wzorcowy parsing i niewzorcowy mają niezgodne nawiasowania.

State of the art

- Najlepsze parsery frazowe dla języka angielskiego osiągają obecnie ponad 93%
  - *Grammar as a Foreign Language*, Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, Geoffrey Hinton
- Dla języka polskiego – to trudniejsze pytanie. Istnieje praca (Woliński, Rogozińska), która podaje 94.1% skuteczności.
- Wynik niestety nie jest do końca miarodajny, bowiem korzysta ze Świgry i z anotacji w treebanku.