N-gramowy model języka (2). Zastosowania modelu językowego.

Paweł Rychlikowski

Instytut Informatyki UWr

30 października 2018

Wygładzanie. Przypomnienie

- Konieczne jest modyfikowanie liczników (choćby po to, by uniknąć wartości zerowych)
- Można o nim myśleć jako o modyfikowaniu zaobserwowanej częstości występowania słów

Wygładzanie Gooda Turinga

- Koncentrujemy się tym razem na słowach z dużego korpusu (NKJP).
- Przykładowe słowa, które wystąpiły k razy w korpusie, dla
  - $k = 1$: zaleśnym, kłołzy, piotrosze, praded, kapitałowcy, operkami, deniecki, tyrkę, doregulowany, atrakcjnego
  - $k = 2$: resultado, wypasającego, spakowaliby, bacuje, burczącym, gralewo, jedździ, etnografizmu
  - $k = 10$: politurowanym, machabejskie, hadleya, łowieccy, glazurników, ogór, kilkusekundowych, impresjonistami
  - $k = 100$: nagłaśniane, zaborski, semantycznych, prześladowanym, pozbywamy, browser, niezrealizowanych, urojonych
  - $k = 1234$: sprawiało, składy, przewidział, odsunął, stanowiącej, tomu

Częstość częstości

- Definiujemy N_c jako liczbę tych (różnych) słów, które w korpusie wystąpiły dokładnie c razy.
- Zauważmy, że niezerowe N_{1234} sugeruje, że ciąg $\{N_i\}$ jest dość gęsty.

Popatrzmy, jak wyglądają wybrane wartości N_i.

Uwaga

Będziemy szacować liczby wystąpień słów w grupy i na podstawie liczby występień słów grupy $i + 1$.

Czyli szacujemy prawdopodobieństwo tego, co się nie zdarzyło na podstawie tego, co zdarzyło się raz.

Wygładzanie Gooda Turinga. Wzór

- Zmieniamy częstości obserwowane na częstości „wygładzone”, aby przerzucić część masy prawdopodobieństwa na słowa, których nie widzieliśmy.
- Wzór na zmodyfikowaną liczność słowa:

$$c^* = (c+1)\frac{N_{c+1}}{N_c}$$

- Co to jest N_0?

$$P(\text{unseen}) = \frac{N_1}{N}$$

Obliczone zmodyfikowane częstości

W korpusie mamy **298 745 366** słów. Ponadto:

- $N_1 = 911213$
- $N_2 = 255221$
- $N_3 = 127195$
- $N_{10} = 21099$
- $N_{11} = 18181$

Czyli słowo wiewióreczkami „wystąpiło” 0.003 raza, słowo kapitałowcy wystąpiło 0.56 raza, słowo bacuje wystąpiło 1.5 raza, słowo występujące 10 razy osłabiamy do 9.47.

Uwaga

W którymś momencie szacowania N_i stają się bardziej przypadkowe, dlatego często dla większych i zostawiamy oryginalne wartości.

Absolute discounting

- Podstawowa idea:

  Jak coś wystąpiło k razy w korpusie, to ile razy ***wystąpiło naprawdę***

- Metoda: dzielimy korpus na K_1 (do liczenia rzeczy) i K_2
- Zadajemy pytanie:

  Ile średnio razy występują w K_2 rzeczy, które w K_1 wystąpiły 7 razy (to w przypadku, gdy $|K_1| = |K_2|$)

Przykładowe wyniki

Church, Gale, 1991

Bigram count in training set	Bigram count in heldout set
0	0.0000270
1	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21
9	8.26

Źródło: Jurafsky, Martin, Speech and Language Processing

- Rzeczy niewystępujące wyraźnie się wyróżniają.
- Jak coś jest 1 raz, to „tak jakby" powinno być 0.5
- W pozostałych przypadkach dobre przybliżenie to odjąć 0.75

Absolute discounting

Wzór

$$P_{AD}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{C(w_{i-1})} + \lambda(w_{i-1})P(w_i)$$

$d = 0.75$ dla wszystkich, ewentualnie dla pojedynczych bierzemy $d = 0.5$.

Metoda Knesera-Neya

Korpus

Wyobrażamy sobie zbiór tekstów, stanowiący dokumenty z Kaliforni Północnej, ze szczególnym uwzględnieniem miasta i hrabstwa San **Francisco**.

Pytanie

Co możemy powiedzieć na temat słowa **Francisco**:

a) Jest w korpusie stosunkowo często.

b) W zasadzie jedynym kontekstem, w którym występuje jest poprzedzające słowo **San**.

Jeszcze o losowaniu

- Wyobraźmy sobie losowanie 2-gramowe, w którym gdy nie ma bigramu, przełączamy się na unigramy.
- Z jakim p-stwem powinniśmy losować unigramy?
- Intuicja: słowo Francisco powinno być losowane bardzo rzadko (mimo swojego dużego prawdopodobieństwa), bo pasuje tylko do San (a ten przypadek obsługują bigramy)

Uwaga

Kluczowa jest nie częstość słowa, lecz liczba kontekstów, w których dane słowo występuje.

Metoda Knessera-Neya (2)

- Prawdopodobieństwo kontynuacji (jak często słowo jest kontynuacją):

$$P_{\text{CONT}}(w) = \frac{|\{v : C(vw) > 0\}|}{\sum_{w'} |\{v : C(vw') > 0\}|}$$

- Wzór KN:

$$P_{\text{KN}}(w_{i-1}|w_i) = \frac{\max(C(w_{i-1}w_i) - d, 0)}{C(w_{i-1})} + \lambda(w_{i-1})P_{\text{CONT}}(w_i)$$

- Współczynnik λ (dokładna analiza na ćwiczeniach):

$$\lambda(w_{i-1}) = \frac{d}{C(w_{i-1})}|\{w|C(w_{i-1}w) > 0\}|$$

Interpolacja. Przypomnienie

Prawdopodobieństwo (np) trigramowe szacujemy wykorzystując również prawdopodobieństwo bigramów i unigramów.

$$P^*(w_3|w_1\,w_2) = \lambda_1 * P(w_3) + \lambda_2 * P(w_3|w_2) + \lambda_3 * P(w_3|w_1\,w_2)$$

, gdzie $\lambda_1 + \lambda_2 + \lambda_3 = 1$ oraz $\lambda_i > 0$.

- Wyznaczaliśmy wartości λ za pomocą algorytmu **deleted interpolation**
- Możliwe uogólnienia tego algorytmu, w celu uwzględnienia na przykład N-gramów gramatycznych:

$$P_T(w_3|w_1\,w_2) = P(t_3|t_1\,t_2) * P(w_3|t_3)$$

które możemy zmieszać z normalnymi N-gramami

Katz back-off

Alternatywa

Wykorzytujmy zawsze najlepszą (najdokładniejszą) informację, jaką mamy.

Czyli nie patrzymy na 4-gramy, jak mamy informacje o wystąpieniach 5-gramu!

Katz backoff – szczegóły

- Dla trigramów wzór wygląda tak:

$$P_{katz}(z|xy) = \begin{cases} P^*(z|xy), & \text{jeśli } C(xyz) > 0 \\ \alpha(x,y)P_{katz}(z|y), & \text{jeśli } C(xy) > 0 \\ P^*(z), & \text{w przeciwnym przypadku} \end{cases}$$

- Dla bigramów natomiast:

$$P_{katz}(z|y) = \begin{cases} P^*(z|y), & \text{jeśli } C(yz) > 0 \\ \alpha(y)P^*(z), & \text{w przeciwnym przypadku} \end{cases}$$

Katz backoff – szczegóły (2)

- $P^*(\dots)$ to prawdopodobieństwo zmniejszone (discounted), bo inaczej wyszlibyśmy poza 1.
- Oczywiście to stosuje się również do N-gramów wyższych rzędów.

Katz backoff – obliczenia α

Obliczamy $\alpha(x, y)$

- Definiujemy $\beta(x, y)$ jako część masy prawdopodobieństwa, którą mamy podzielić.

$$\beta(x, y) = 1 - \sum_{z:C(xyz)>0} P^*(z|xy)$$

- Obliczoną masę dzielimy proporcjonalnie:

$$\alpha(x, y) = \frac{\beta(x, y)}{\sum_{z:C(x,y,z)=0} P_{katz}(z|y)}$$

Stupid back-off

A teraz coś dla zmęczonych tymi wszystkimi normalizacjami, etc.

- W pewnych sytuacjach możemy się wyluzować i nie przejmować rozkładami prawdopodobieństwa, tylko tworzyć takie jakby trochę prawdopodobieństwo
- a wyniki będą i tak zadowalające
- Dla modelu trigramowego (jak mamy trigram):

$$S(x, y, z) = \frac{C(x, y, z)}{C(x, y)}$$

w przeciwnym przypadku (czego się spodziewamy?):

$$S(x, y, z) = \alpha S(y, z)$$

- Dobre wyniki dla $\alpha = 0.4$, oczywiście działa też dla wyższych rzędów (≥ 3).

Format ARPA

```
\data\
ngram 1=7
ngram 2=7

\1-grams:
-1.0000 <unk> -0.2553
-98.9366 <s>  -0.3064
-1.0000 </s>  0.0000
-0.6990 wood  -0.2553
-0.6990 cindy -0.2553
-0.6990 pittsburgh -0.2553
-0.6990 jean  -0.1973

\2-grams:
-0.2553 <unk> wood
-0.2553 <s> <unk>
-0.2553 wood pittsburgh
-0.2553 cindy jean
-0.2553 pittsburgh cindy
```

Format ARPA (2)

W pliku tekstowym mamy zapamiętane:

- Informacje o wszystkich *ciekawych* N-gramach (dla różnych N, z unigramami łącznie)
- Dodatkowe informacje dotyczące procedury back-off (wartości α, po prawej stronie)
- Zakładamy, że wszystkie częstości zostały wcześniej poddane *discountingowi* (przez twórcę pliku)

Każdy wiersz z danymi (oprócz N-gramów najwyższego rzędu) zawiera (przykład dla trigramów):

$$\log P^*(w_i|w_{i-2}w_{i-1}) \quad w_{i-2}w_{i-1}w_i \quad \log \alpha(w_{i-2}w_{i-1}w_i)$$

Format ARPA (3)

Uwaga 1

Ten model format ciągle jest używany, na przykład w rozpoznawaniu mowy

Uwaga 2

Istnieją programy, które z tekstu (z korpusu) robią plik ARPA, sterowane różnymi parametrami (na przykład metodą wygładzania, informacją, jakie N-gramy uwzględniamy, itd).

Narzędzia do modelowania języka

Lektura

Kenneth Heafield, *KenLM: Faster and Smaller Language Model Queries*

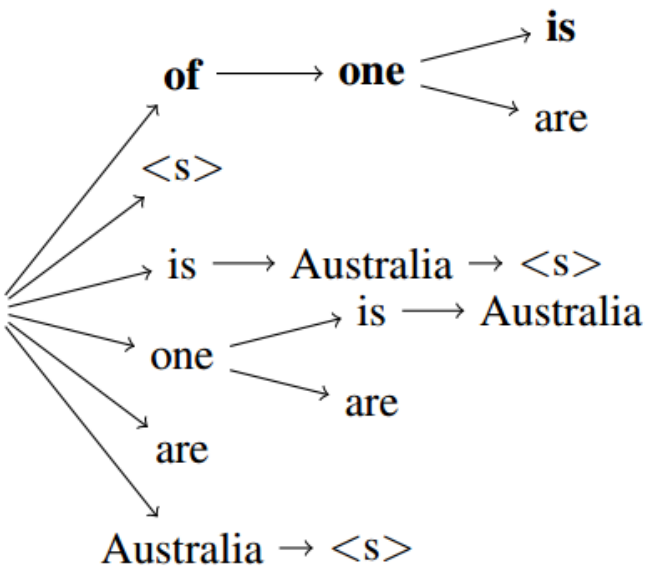
Kilka uwag odnośnie implementacji:

- W pracy opisane są sposoby haszowania, efektywnie wyszukujące iformacje w tablicy
- Będziemy o tym trochę mówić na kolejnych ćwiczeniach, teraz powiemy o jednej strukturze:
- mianowicie o drzewie Trie (dla N-gramów)

Trie dla słów

- Użycie drzewa **trie** pozwala na kilkukrotne zmniejszenie pamięci
- Warto pamiętać o tym sposobie zawsze, gdy mamy do zapamiętania pewną (dużą) liczbę fraz.

Fragment drzewa trie (pytanie: dlaczego pamiętamy frazę **is one of** w odwrotnej kolejności.



Źródło: KenLM: Faster and Smaller Language Model Queries

Ocena modeli językowych

Są generalnie dwa sposoby oceniania modeli językowych (i, tak naprawdę, wszystkiego innego też):

1. Wewnętrzna (instristic) – mamy jakąś mniej lub bardziej naturalną miarę jakości modelu
2. Zewnętrzna (exstrinsic) – sprawdzamy, jak model poradzi sobie z pewnym zadaniem (które jest naszym celem, i w którym mamy naturalną miarę jakości)

Za chwilę zastosujemy sobie modele językowe, ale najpierw standardowa miara wewnętrzna.

Perplexity (miara nieokreśloności)

Intuicje

1. To co się zdaża, powinno mieć wysokie prawdopodobieństwo.
2. Gdy dobrze przewidujemy kolejne słowo (na podstawie pełnego prefiksu), to jesteśmy w stanie dobrze kompresować tekst (dlaczego?)
3. Oczywiście powinniśmy dzielić korpus (na część przeszłą (zdarzyła się) i przyszłą (zdarzy się, chcemy jej dać spore prawdopodobieństwo, ale jej nie znamy)

Perplexity (2)

Wzór

$$PP(w_1 \dots w_N) = P(w_1 \dots w_N)^{-\frac{1}{N}}$$

gdzie N jest wielkością części testowej korpusu

Pytanie

Jakie jest perplexity całkiem losowego ciągu cyfr?(odpowiedź: **10**)

Można rozumieć perplexity jako średni ważony **współczynnik rozgałęzienia** języka.

Wall Street Journal

Trening na 38M słów, walidacja na 1.5M słów

- 1-gram: 962
- 2-gram: 170
- 3-gram: 109

Uwaga

$\log_2(109) = 6.7414669864011465$ (bitów)

Perplexity dla języka polskiego

PolEval 2018

Poz. 1: **ULMFiT-SP-PL**, **117.67**, FastAI (neuronowy)

Poz. 2: **AGHUJ**, 146.7, Klasyczny model N-gramowy (SRILM)

Poz. 3: **PocoLM Order 6**, 208.62

Uwaga

Można porównywać wartości perplexity **tylko dla tego samego korpusu!** (Dlaczego?)

... bo perplexity mówi dwie rzeczy: jak dobry jest model i jak trudny jest korpus.

Zastosowania modelu językowego: zadania rekonstrukcji tekstu

Zagadka

Rozszyfruj tekst (po angielsku):

m gd smbd hs stln ll m vwls!

- Celem naszym będzie rekonstrukcja samogłoskowa tekstu, czyli poprawne wprowadzenie samogłosek do tekstu, który został ich pozbawiony.
- Wydaje się, że to nie ma większego sensu, ale... ... podobno niektóre fragmenty Biblii byay napisane w ten właśnie sposób.

Postać normalna słowa

- Pewna liczba zadań ma dość podobną strukturę do powyższego.
- Nazwiemy je zadaniami z postacią normalną słowa.
- W naszym zadaniu postacią normalną będzie „wyciąg spółgłoskowy”, czyli słowo powstałe po usunięciu samogłosek.

Postać normalna słowa. Przykłady

- Dla słowa przetwarzanie, postacią normalną jest prztwrzn.
- Dla słowa idea postacią normalną jest d
- Dla słowa uaaaa postacią normalną jest _ (wygodniej, niż napis pusty)

Podejście 1. Najprostszy model językowy

Najprostszym modelem językowym jest: model unigramowy, w którym interesuje nas jedynie, jak często występują słowa w korpusie.

Jak wyglądałby algorytm i jakie struktury danych są nam potrzebne?

- Dla każdej postaci normalnej pamiętamy słowa, które ją mają.
- Korekty dokonujemy słowo po słowie.
- Dla korygowanego słowa bierzemy najczęstszą formę, która ma tę samą postać normalną.