Processamento Estatístico da Linguagem Natural

Aula 8

Professora Bianca
(Sala 302 – Bloco E)
bianca@ic.uff.br
http://www.ic.uff.br/~bianca/peln/

Aula de Hoje

- Python
  - Expressões regulares
  - Números aleatórios
  - Natural Language Toolkit
- Cap. 4 – Jurafsky & Martin
  - Seções 4.1, 4.2 e 4.3

Módulo útil: re

- Expressões regulares

```
import re

r = re.compile(r'\bdis(\w+)\b')
s = 'Then he just disappeared.'
match = r.search(s)
if match:
  print "Found the regex in the string!"
  print "The prefix was", match.group(1)
```


Módulo útil: random

```
>>> import random

>>> random.uniform(0,1)
0.16236

>>> list = ['first', 'second', 'third', 'fourth']
>>> random.choice(list)
'third'
>>> random.choice(list)
'first'
```


NLTK: Python Natural Language Toolkit

- NLTK é um conjunto de módulos que podem ser importados.
  - Exemplo: `from nltk_lite.utilities import re_show`
- NLTK vem com vários *corpora* (conjuntos de textos).
  - Exemplos:
    - gutenberg (obras de literatura do projeto Gutenberg)
    - treebank (textos com árvores de análise sintática to projeto Penn treebank)
    - Brown (1961 milhões de palavras com etiquetas de POS = *part-of-speech*)
- Para utilizar um *corpus*:

```
>>> from nltk_lite.corpora import gutenberg
>>> print gutenberg.items
['autsen-emma', 'austen-persuasion',...]
```


Operações simples com *corpus*

- Processamento simples de corpus inclui tokenização (quebrar o texto em palavras), normalização, *tagging* e *parsing*.
- Exemplo:

```
from nltk_lite.corpora import gutenberg
nwords = 0
for word in gutenberg.raw('shakespeare-macbeth'):
    nwords += 1
print nwords
```

Operações mais complexas

- O corpus Gutenberg contém apenas uma seqüência de palavras, sem estrutura.
- O corpus de Brown tem frases anotadas e é guardado como uma lista de frases, onde cada frase é uma lista de palavras.
- Podemos usar a função **`extract`** para obter frases individuais.

```
from nltk_lite.corpora import brown
from nltk_lite.corpora import extract
firstSentence = extract(0, brown.raw('a'))
# ['The', 'Fulton', 'County', 'Grand', 'jury'...]
```

- Texto com POS também pode ser extraído.

```
taggedFirstSentence = extract(0, brown.tagged('a'))
# [('The', 'at'), ('Fulton', 'np-tl'),
   ('County', 'nn-tl')...
```


Texto com *parsing*

```
>>> from nltk_lite.corpora import treebank
>>> parsedSent = extract(0, treebank.parsed())
>>> print parsedSent

(S:
  (NP-SBJ:
      (NP: (NNP: 'Pierre') (NNP: 'Vinken'))
      (,: ',')
      (ADJP: (NP: (CD: '61') (NNS: 'years')) (JJ: 'old'))
      (,: ','))
  (VP:
      (MD: 'will')
      (VP: (VB: 'join') (NP: (DT: 'the') (NN: 'board'))
  (PP-CLR: (IN: 'as') (NP: (DT: 'a') (JJ: 'nonexecutive')
  (NN: 'director'))) (NP-TMP: (NNP: 'Nov.') (CD: '29'))))
  (.: '.'))
```


Mais informações

- Introdução ao NTLK
  - http://nltk.sourceforge.net/doc/en/ch01.html

Aula de Hoje

- Python
  - Expressões regulares
  - Números aleatórios
  - Natural Language Toolkit
- Cap. 4 – Jurafsky & Martin

N-gramas

- Qual palavra seria mais provável depois de:
  "Por favor, entregue seu trabalho..."
- O capítulo formaliza essa idéia usando modelos chamados de "**N-gramas**".
  - Prevêem a próxima palavra a partir das N-1 palavras anteriores.
  - Podem ser usados para estimar a probabilidade de uma determinada seqüência de palavras.
- Também são chamados de modelos de linguagem ou LMs.

N-gramas

- N-gramas são úteis para muitas tarefas em que temos que identificar palavras.
  - Reconhecimento de voz
  - Reconhecimento de escrita
  - Tradução automática
  - Correção ortográfica
  - ...

Contagem de Palavras

- Probabilidades são baseadas em contagens.
  - As contagens em PELN são feitas com base em um **corpus** (plural **corpora**) = coleção em formato eletrônico de textos ou gravações de voz.
  - Temos que decidir o que conta como uma palavra.
    - Pontuação conta?
    - Para voz, pausas "preenchidas" (como "ahn") ou fragmentos de palavras contam?
    - Palavras com letras maiúsculas ou minúsculas são diferentes ou iguais?
    - Singular e plural contam como a mesma palavra?

Contagem de Palavras

- Tipos = palavras distintas em um corpus
- Tokens = cada palavra em um corpus
- "they picnicked by the pool then lay back on the grass and looked at the stars"
  - 16 tokens
  - 14 tipos
- Brown et al (1992) – corpus grande
  - 583 milhões de tokens
  - 293,181 de tipos
- Seja N = número de tokens, V = vocabulário = número de tipos
  - V > O(sqrt(N))

Contagem de Palavras

- Terminologia
  - **Lema ("lemma")**: um conjunto de formas léxicas tendo o mesmo radical, mesma classe gramatical, e aproximadamente o mesmo significado.
    - "Cat" e "cats" = mesmo lema
  - **Palavra ("wordform")**: a forma de superfície, flexionada.
    - "Cat" e "cats" = palavras diferentes.
- Em inglês, normalmente se contam as palavras e não os lemas nos N-gramas.

Contagem direta

- Nosso objetivo é calcular a probabilidade de uma palavra *w*, dada alguma história *h*, ou $P(w|h)$.
- Exemplo:
  P(the | its water is so transparent that)
- Essa probabilidade poderia ser calculada como
  $$\frac{\text{Count(its water is so transparent that the)}}{\text{Count(its water is so transparent that)}}$$
- Este método não funciona em geral porque a linguagem é criativa, ou seja, frases novas são geradas continuamente.

A Regra da Cadeia

- Lembrando a definição de probabilidade condicional:
  $$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$
- Reescrevendo:
  $$P(A \wedge B) = P(A \mid B)P(B)$$
- Forma geral:
  $$P(x_1,x_2,x_3,\ldots x_n) = P(x_1)P(x_2|x_1)P(x_3|x_1,x_2)\ldots P(x_n|x_1\ldots x_{n-1})$$

Aplicação da regra da cadeia a seqüências de palavras

$$P(w_1^n) = P(w_1)P(w_2|w_1)P(w_3|w_1^2)\ldots P(w_n|w_1^{n-1})$$
$$= \prod_{k=1}^{n} P(w_k|w_1^{k-1})$$

A notação w_i^j significa a seqüência que começa na palavra ***i*** e termina na palavra ***j***.

- P("the big red dog was")=
  P(the)*P(big|the)*P(red|the big)*P(dog|the big red)*P(was|the big red dog)

Hipótese de Markov

- A palavra seguinte só depende da palavra anterior
  - P(lizard|the,other,day,I,was,walking,along,and ,saw,a) = P(lizard|a)
- Ou talvez das duas palavras anteriores
  - P(lizard|the,other,day,I,was,walking,along,and ,saw,a) = P(lizard|saw,a)

Hipótese de Markov

- Substituir cada componente na regra da cadeia com a aproximação (supondo um prefixo de N)

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1})$$

- Versão bigrama

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1})$$

Estimando as probabilidades

- A estimativa de máxima verossimilhança (MLE) é

$$P(w_i \mid w_{i-1}) = \frac{count(w_{i-1}, w_i)}{count(w_{i-1})}$$

Exemplo

- <s> I am Sam </s>
- <s> Sam I am </s>
- <s> I do not like green eggs and ham </s>

$P(\texttt{I} \mid \texttt{<s>}) = \frac{2}{3} = .66$ $P(\texttt{Sam} \mid \texttt{<s>}) = \frac{1}{3} = .33$ $P(\texttt{am} \mid \texttt{I}) = \frac{2}{2} = 1.0$
$P(\texttt{<\s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5$ $P(\texttt{<s>} \mid \texttt{Sam}) = \frac{1}{2} = 0.5$ $P(\texttt{Sam} \mid \texttt{am}) = \frac{1}{2} = .5$
$P(\texttt{do} \mid \texttt{I}) = \frac{1}{1} = 1.0$

$$P(w_n|w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})}$$

- Essa é estimativa de máxima verossimilhança, porque ela maximiza P(corpus de treinamento|modelo).

Mais exemplos: frases do Berkeley Restaurant Project

- can you tell me about any good cantonese restaurants close by
- mid priced thai food is what i'm looking for
- tell me about chez panisse
- can you give me a listing of the kinds of food that are available
- i'm looking for a good place to eat breakfast
- when is caffe venezia open during the day

Contagens de bigramas

- Out of 9222 sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	0	1	0	0
spend	1	0	1	0	0	0	0	0

Probabilidades de bigramas

- Normalizar por unigramas

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

- Resultado:

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Estimativas das probabilidades de frases

- P(<s> I want english food </s>) =
  p(I|<s>)p(want|I)p(english|want)
  p(food|english)p(</s>|food)
  =.000031

Quais as razões para essas probabilidades?

- P(english|want) = .0011
- P(chinese|want) = .0065
- P(to|want) = .66
- P(eat | to) = .28
- P(food | to) = 0
- P(want | spend) = 0
- P (i | <s>) = .25

Razões podem ser sintáticas ou semânticas (culturais)

Conjuntos de Treinamento e Teste

- As probabilidades do modelo N-grama vêm do corpus em que ele é treinado.
  - Para saber se o modelo funciona bem, temos que testá-lo em dados novos.
- Então o primeiro passo ao construir um modelo é dividir o corpus em duas partes:
  - Corpus ou conjunto de treinamento (geralmente 80%)
    - Usado para estimar as probabilidades
  - Corpus ou conjunto de teste
    - Usado para avaliar o modelo
- Às vezes são necessários outros conjuntos.
  - Hold-out para ajustar parâmetros
  - Dev-set para testar durante o desenvolvimento

O método de visualização de Shannon

- Para gerar sentenças aleatórias:
- Escolha um bigrama aleatório <s>, w de acordo com a sua probabilidade.
- Escolha um bigrama (w, x) de acordo com a sua probabilidade.
- Até que </s> seja escolhido.
- <s> I
  I want
  want to
  to eat
  eat Chinese
  Chinese food
  food </s>

Unigram	• To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have • Every enter now severally so, let • Hill he late speaks; or! a more to leg less first you enter • Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like
Bigram	• What means, sir. I confess she? then all sorts, he is trim, captain. •Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. •What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman? •Enter Menenius, if it so many good direction found'st thou art a strong upon command of fear not a liberal largess given away, Falstaff! Exeunt
Trigram	• Sweet prince, Falstaff shall die. Harry of Monmouth's grave. • This shall forbid it should be branded, if renown made it empty. • Indeed the duke; and had a very good friend. • Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done.
Quadrigram	• King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in; • Will you not tell me who I am? • It cannot be but so. • Indeed the short and the long. Marry, 'tis a noble Lepidus.

Shakespeare como corpus

- N=884,647 tokens, V=29,066
- Shakespeare produziu 300,000 tipos de bigramas dos V^2= 844 milhões de possíveis bigramsa: então, 99.96% dos bigramas possíveisnunca foram vistos. (tem entrada zero na tabela)
- Quadrigramas: o que sai parece com Shakespeare porque é Shakespeare.