Predictive Input Methods

- Anish Patil

RedHat i18n team

Agenda:

1. What/Why?
2. Language models
3. Projects
4. Demonstration

Need

- 1.21 Billion population (India)
  - 74% literate (read & write any language)
  - Still only 5-6% understand English
  - 51% youth in 1.21 Billion
- Diversity in India
  - 22 Officially recognized languages
  - 9 Major scripts

Rest of the world

- List of extinct language's
  - http://en.wikipedia.org/wiki/List_of_extinct_languages_of_Europe
  - http://www.unesco.org/culture/languages-atlas/en/atlasmap.html

Problems with Natural Languges

- Ambiguous
- Exceptions!
- Humans- Ignore!

How to predict next word?

- Statistical Techniques
- Probability

Probabilities

- Bayes' theorem
- p(A | B) =p(A) * p(B | A) / p(B)
- Max has two coins in his pocket, a fair coin (head on one side and tail on the other side) and a two-headed coin. He picks one at random from his pocket, tosses it and obtains head. What is the probability that he flipped the fair coin?

$$
\begin{aligned}
P(\text{head} \mid \text{fair coin}) &= \frac{1}{2} \\
P(\text{head} \mid \text{unfair coin}) &= 1 \\
P(\text{fair coin}) &= \frac{1}{2} \\
P(\text{unfair coin}) &= \frac{1}{2}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{head}) &= P(\text{head} \mid \text{fair coin})P(\text{fair coin}) + P(\text{head} \mid \text{unfair coin})P(\text{unfair coin}) \\
&= \frac{1}{2} \cdot \frac{1}{2} + 1 \cdot \frac{1}{2} \\
&= \frac{1}{4} + \frac{2}{4} = \frac{3}{4}
\end{aligned}
$$

$$
\begin{aligned}
P(\text{fair coin} \mid \text{head}) &= \frac{P(\text{head} \mid \text{fair coin})P(\text{fair coin})}{P(\text{head})} \\
&= \frac{\frac{1}{2} \cdot \frac{1}{2}}{\frac{3}{4}} \\
&= \frac{1}{4} \cdot \frac{4}{3} = \frac{1}{3}
\end{aligned}
$$

Language Model

- Lot of words in one language but what is the probability that one word follow another word?
- Simple model:- number of occurrence of word/Number of words in the language

Language Model

- A language model consists of a finite set V, and a function p(x1 , x2 , . . . xn) such that:

  1. For any x1 . . . xn ∈ V † , p(x1 , x2 , . . . xn) ≥ 0

  2. In addition,

  Sum(p(x1 , x2 , . . . xn))= 1

  x1 ...xn ∈V †

  Hence p(x1 , x2 , . . . xn) is a probability distribution over the sentences in V † .

Markov Models

- The probability of a word depends only on the probability of a limited history
- The probability of a word depends only on the probability of the n previous words
- Unigrams, Bigrams,Trigrams...

Markov Models cont..

- English words W = w1, w2, w3, ..., wn
- p(w1, w2, w3, ..., wn) = p(w1) p(w2|w1) p(w3|w1, w2)...p(wn|w1, w2, ...wn−1)
- Bigram model:- p(w1, w2, w3, ..., wn) = p(w1) p(w2|w1) p(w3|w2)...p(wn|wn−1)
- wn−1 is called the history
- For example, the dog barks STOP
- p(the dog barks STOP) = p(the|*, *)×p(dog|*, the)×p(barks|the, dog)×p(STOP|dog, barks)

Markov Models cont..

- maximum likelihood estimation
- p(w2|w1) = count(w1,w2) / count(w1)

Example

- Training Set:

  START ASIA IS AWESOME STOP

  START GNOME IS AWESOME STOP

  START GNOME ASIA IS AWESOME STOP

- v= {START, GNOME,.Asia,IS,AWESOME, END}
- Unigram Model:- p(ASIA) = 2/10 = 0.2

Example cont..

- Trigram Model:-

  P (GNOME/START,START) = P (2/3)

- P(START GNOME ASIA IS AWESOME STOP)= P (GNOME/START,START) * P(ASIA/GNOME,START) * P(IS/GNOME,ASIA) * P(AWESOME/ASIA,IS) * P(STOP/IS,AWESOME)

  =(2/3) * (2/3) * (2/1) * (3/1) * (3/2)

fedora

Training

- Data??
- Testing?

How to evaluate a L.M?

- Perplexity
- H(W) = 1/n log p (W)
- Lower is the perplexity higher is accuracy of your language model

Unseen Sentences

- GNOME SHELL is AWESOME
- Smoothing?
- Discounting?

Smoothing

- Zero probabilities of unigram costs zero probabilities of entire sentence
- For unigrams add 1 to every word and adjust the count and divide it by size vocabulary to normalize it
- Original $P(w) = c / N$
- New $P(w) = (c+1) / (V+N)$

Linear Interpolation

- $q_{ML}(w|u, v) = c(w,u,v) / c(u, v)$
- $q_{ML}(w|v) = c(v, w) / c(v)$
- $q_{ML}(w) = c(w) / c()$
- $q(w|u, v) = \lambda_1 \times q_{ML}(w|u, v) + \lambda_2 \times q_{ML}(w|v) + \lambda_3 \times q_{ML}(w)$
- $\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$ and $\lambda_1 + \lambda_2 + \lambda_3 = 1$

Libyokan and libyokan-data

- https://gitorious.org/libyokan
- https://gitorious.org/yokan-data-mr-in

Ibus-typing-booster

- https://fedorahosted.org/ibus-typing-booster/

Demo

Thank you!!

- Anish.developer@gmail.com
- apatil@redhat.com
- Irc: anish,__anish__