

CSE 842 Natural Language Processing

Lecture 3: NGram

1/21/2009

CSE842, Spring 2009, MSU

1

Announcement

Homework 1 is posted on the angel system.

- Due:
- Written part at the beginning of the class
- Programming part at 11:59pm on the due date via "handin"

1/21/2009

CSE842, Spring 2009, MSU

2

Next Word Prediction

From a NY Times story...

- Stocks plunged this
- Stocks plunged this morning, despite a cut in interest rates
- Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall ...
- Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall Street began

1/21/2009

CSE842, Spring 2009, MSU

3

- Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall Street began trading for the first time since last ...
- Stocks plunged this morning, despite a cut in interest rates by the Federal Reserve, as Wall Street began trading for the first time since last Tuesday's terrorist attacks.

1/21/2009

CSE842, Spring 2009, MSU

4

Human Word Prediction

- Clearly, at least some of us have the ability to predict future words in an utterance.
- How?

1/21/2009

CSE842, Spring 2009, MSU

5

Human Word Prediction

- Clearly, at least some of us have the ability to predict future words in an utterance.
- How?
 - Domain knowledge
 - Syntactic knowledge
 - Lexical knowledge

1/21/2009

CSE842, Spring 2009, MSU

6

Claim

- A useful part of the knowledge needed to allow Word Prediction (guessing the next word) can be captured using simple statistical techniques.
- In particular, we'll rely on the notion of the probability of a sequence (e.g., sentence) and the likelihood of words co-occurring

1/21/2009

CSE842, Spring 2009, MSU

7

Word Prediction

- We can formalize this task using what are called *N-gram* models.
- *N-grams* are token sequences of length *N*.
- Our earlier example contains the following 2-grams (aka bigrams)
 - (I notice), (notice three), (three guys), (guys standing), (standing on), (on the)
- Given knowledge of counts of *N-grams* such as these, we can guess likely next words in a sequence.

1/21/2009

CSE842, Spring 2009, MSU

8

Why is this useful?

Example applications that employ language models:

- Speech recognition
- Handwriting recognition
- Spelling correction
- Machine translation systems
- Optical character recognizers

1/21/2009

CSE842, Spring 2009, MSU

9

Real Word Spelling Errors

- They are leaving in about fifteen minuets to go to her horse.
- The study was conducted mainly be John Black.
- The design an construction of the system will take more than a year.
- Hopefully, all with continue smoothly in my absence.
- I need to notified the bank of....
- He is trying to fine out.

1/21/2009

CSE842, Spring 2009, MSU

10

Handwriting Recognition

- Assume a note is given to a bank teller, which the teller reads as *I have a gub*.
- NLP to the rescue
 - *gub* is not a word
 - *gun, gum, Gus,* and *gull* are words, but *gun* has a higher probability in the context of a bank

1/21/2009

CSE842, Spring 2009, MSU

11

Counting

- Simple counting lies at the core of any probabilistic approach. So let's first take a look at what we're counting.
 - *He stepped out into the hall, was delighted to encounter a water brother.*
 - 13 tokens, 15 if we include “,” and “.” as separate tokens.
 - Assuming we include the comma and period, how many bigrams are there?

1/21/2009

CSE842, Spring 2009, MSU

12

Counting

- Not always that simple
 - *I do uh main- mainly business data processing*
- Spoken language poses various challenges.
 - Should we count “uh” and other fillers as tokens?
 - What about the repetition of “mainly”? Should such do-overs count twice or just once?
 - The answers depend on the application.
 - If we’re focusing on something like ASR to support indexing for search, then “uh” isn’t helpful (it’s not likely to occur as a query).
 - But filled pauses are very useful in dialog management, so we might want them there.

1/21/2009

CSE842, Spring 2009, MSU

13

Counting: Types and Tokens

- How about
 - *They picnicked by the pool, then lay back on the grass and looked at the stars.*
 - 18 tokens (again counting punctuation)
- But we might also note that “*the*” is used 3 times, so there are only 16 unique types (as opposed to tokens).
- In going forward, we’ll have occasion to focus on counting both types and tokens of both words and *N* grams.

1/21/2009

CSE842, Spring 2009, MSU

14

Counting: Wordforms

- Should “cats” and “cat” count as the same when we’re counting?
- How about “geese” and “goose”?
- Some terminology:
 - Lemma: a set of lexical forms having the same stem, major part of speech, and rough word sense
 - Wordform: fully inflected surface form
- Again, we’ll have occasion to count both lemmas and wordforms

1/21/2009

CSE842, Spring 2009, MSU

15

Counting: Corpora

- Corpora are (generally online) collections of text and speech
 - Linguistic Data Consortium (LDC)
 - Brown Corpus
 - Wall Street Journal and AP News corpora
 - ATIS, Broadcast News (speech)
 - TDT (text and speech)
 - Switchboard, Call Home (speech)
 - AQUAINT
- Brown et al (1992) large corpus of English text
 - 583 million wordform tokens
 - 293,181 wordform types
- Google
 - Crawl of 1,024,908,267,229 English tokens
 - 13,588,391 wordform types
 - That seems like a lot of types... After all, even large dictionaries of English have only around 500k types. Why so many here?
 - Numbers, misspellings, names, acronyms, etc.

1/21/2009

CSE842, Spring 2009, MSU

16

Summary of Terminology

- Sentence: unit of written language
- Utterance: unit of spoken language
- Word Form: the inflected form that appears in the corpus
- Lemma: lexical forms having the same stem, part of speech, and word sense
- Types: number of distinct words in a corpus (vocabulary size)
- Tokens: total number of words

1/21/2009

CSE842, Spring 2009, MSU

17

Language Modeling

- We can model the word prediction task as the ability to assess the conditional probability of a word given the previous words in the sequence
 - $P(w_n | w_1, w_2, \dots, w_{n-1})$
- We’ll call a statistical model that can assess this a *Language Model*

1/21/2009

CSE842, Spring 2009, MSU

18

Language Modeling

- How might we go about calculating such a conditional probability?

$$P(\text{the} \mid \text{its water is so transparent that}) = \frac{\text{Count}(\text{its water is so transparent that the})}{\text{Count}(\text{its water is so transparent that})}$$

1/21/2009

CSE842, Spring 2009, MSU

19

Language Modeling

- Unfortunately, for most sequences and for most text collections we won't get good estimates from this method.
 - What we're likely to get is 0. Or worse 0/0.
- Clearly, we'll have to be a little more clever.
 - Let's use the chain rule of probability
 - And a particularly useful independence assumption.

1/21/2009

CSE842, Spring 2009, MSU

20

Chain Rule

conditional probability: $P(A|B) = \frac{P(A \wedge B)}{P(B)}$

So: $P(A \wedge B) = P(B|A)P(A)$

“the dog”: $P(\text{The} \wedge \text{dog}) = P(\text{dog} \mid \text{the})P(\text{the})$

“the dog bites”:

$$P(\text{The} \wedge \text{dog} \wedge \text{bites}) = P(\text{The})P(\text{dog} \mid \text{The})P(\text{bites} \mid \text{The} \wedge \text{dog})$$

1/21/2009

CSE842, Spring 2009, MSU

21

Chain Rule

The probability of a word sequence is the probability of a conjunctive event.

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

Unfortunately, that's really not helpful in general. Why?

1/21/2009

CSE842, Spring 2009, MSU

22

Markov Assumption

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

- $P(w_n)$ can be approximated using only N-1 previous words of context
- This lets us collect statistics in practice
- Markov models are the class of probabilistic models that assume that we can predict the probability of some future unit without looking too far into the past
- Order of a Markov model: length of prior context

1/21/2009

CSE842, Spring 2009, MSU

23

Training and Testing

- Probabilities come from a **training corpus**, which is used to design the model.
 - overly narrow corpus: probabilities don't generalize
 - overly general corpus: probabilities don't reflect task or domain
- A separate **test corpus** is used to **evaluate** the model, typically using standard **metrics**
 - held out test set
 - cross validation
 - evaluation differences should be statistically significant

1/21/2009

CSE842, Spring 2009, MSU

24

Simple N-Grams

- An **N-gram model** uses the previous $N-1$ words to predict the next one:
 - $P(w_n | w_{n-1})$
 - We'll pretty much always be dealing with $P(\langle \text{word} \rangle | \langle \text{some prefix} \rangle)$
- unigrams: $P(\text{dog})$
- bigrams: $P(\text{dog} | \text{big})$
- trigrams: $P(\text{dog} | \text{the big})$
- quadrigrams: $P(\text{dog} | \text{the big dopey})$

1/21/2009

CSE842, Spring 2009, MSU

25

Using N-Grams

- Recall that
 - $P(w_n | w_{1..n-1}) \approx P(w_n | w_{n-N+1..n-1})$
- For a bigram grammar
 - $P(\text{sentence})$ can be approximated by multiplying all the bigram probabilities in the sequence
 - $P(\text{I want to eat Chinese food}) = P(\text{I} | \langle \text{start} \rangle) P(\text{want} | \text{I}) P(\text{to} | \text{want}) P(\text{eat} | \text{to}) P(\text{Chinese} | \text{eat}) P(\text{food} | \text{Chinese})$

1/21/2009

CSE842, Spring 2009, MSU

26

Berkeley Restaurant Project Sentences

- *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*

1/21/2009

CSE842, Spring 2009, MSU

27

A Bigram Grammar Fragment from BERP

Eat on	.16	Eat Thai	.03
Eat some	.06	Eat breakfast	.03
Eat lunch	.06	Eat in	.02
Eat dinner	.05	Eat Chinese	.02
Eat at	.04	Eat Mexican	.02
Eat a	.04	Eat tomorrow	.01
Eat Indian	.04	Eat dessert	.007
Eat today	.03	Eat British	.001

1/21/2009

CSE842, Spring 2009, MSU

28

<start> I	.25	Want some	.04
<start> I'd	.06	Want Thai	.01
<start> Tell	.04	To eat	.26
<start> I'm	.02	To have	.14
I want	.32	To spend	.09
I would	.29	To be	.02
I don't	.08	British food	.60
I have	.04	British restaurant	.15
Want to	.65	British cuisine	.01
Want a	.05	British lunch	.01

1/21/2009

CSE842, Spring 2009, MSU

29

- $P(\text{I want to eat British food}) = P(\text{I} | \langle \text{start} \rangle) P(\text{want} | \text{I}) P(\text{to} | \text{want}) P(\text{eat} | \text{to}) P(\text{British} | \text{eat}) P(\text{food} | \text{British}) = .25 * .32 * .65 * .26 * .001 * .60 = .000080$
- vs. $P(\text{I want to eat Chinese food}) = .00015$
- Probabilities seem to capture ``syntactic'' facts, ``world knowledge''
 - eat is often followed by a NP
 - British food is not too popular

1/21/2009

CSE842, Spring 2009, MSU

30

How do we get the N-gram probabilities?

Use Maximum Likelihood Estimation

What is Maximum Likelihood Estimation?

1/21/2009

CSE842, Spring 2009, MSU

31

Some review

Toss a coin: head or tail.

Suppose we know: $P(\text{head}) = 0.5$

Let's toss the coin 9 times, what is the probability of having 4 heads and 5 tails?

1/21/2009

CSE842, Spring 2009, MSU

32

Some review

Toss a coin: head or tail.

Suppose we know: $P(\text{head}) = 0.5$

Let's toss the coin 9 times, what is the probability of having 4 heads?

$$\frac{9!}{4!(9-4)!} 0.5^4 \times (1-0.5)^{(9-4)} = 0.246$$

1/21/2009

CSE842, Spring 2009, MSU

33

Binomial Distribution

Toss a coin: head or tail.

Suppose we know: $P(\text{head}) = p$

Let's toss the coin n times, the probability of having h heads:

$$\frac{n!}{h!(n-h)!} p^h (1-p)^{n-h}$$

If p (i.e., the model parameter) is known, we can get the probability of a certain observation (i.e., $P(O|p)$)

1/21/2009

CSE842, Spring 2009, MSU

34

Maximum Likelihood Estimation

What if we only see the observation, we don't know p ,

Then we want to find: $\arg \max_p P(p|O)$

Because: $P(p|O) \propto P(O|p)$ (why?)

Therefore we need to find:

$$\arg \max_p P(O|p)$$

Find the model parameter that makes the observation data most likely

1/21/2009

CSE842, Spring 2009, MSU

35

Simple example of MLE

If we toss a coin 100 times and observe 56 heads and 44 tails, what is $P(\text{head})$ - i.e., p ?

$$\arg \max_p P(p|O)$$

$$P(p = 0.5 | O) \propto P(O | p = 0.5) = \frac{100!}{56!44!} 0.5^{56} 0.5^{44} = 0.0389$$

$$P(p = 0.52 | O) \propto P(O | p = 0.52) = \frac{100!}{56!44!} 0.52^{56} 0.48^{44} = 0.058$$

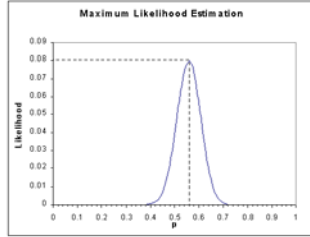
1/21/2009

CSE842, Spring 2009, MSU

36

Simple example of MLE

p	P(p O)
0.48	0.0222
0.50	0.0389
0.52	0.0581
0.54	0.0739
0.56	0.0801
0.58	0.0738
0.60	0.0576
0.62	0.0378



1/21/2009

CSE842, Spring 2009, MSU

37

MLE Cont.

$$P(p | x_1, \dots, x_n) \propto P(x_1, \dots, x_n | p)$$

$$= p^{x_1} (1-p)^{1-x_1} \dots p^{x_n} (1-p)^{1-x_n} = p^{\sum x_i} (1-p)^{\sum (1-x_i)}$$

$$= p^{\sum x_i} (1-p)^{n - \sum x_i}$$

$$x_i = 0 \text{ or } 1, \text{ and } i = 1, \dots, n$$

$$\ln P = \sum x_i \ln p + (n - \sum x_i) \ln(1-p)$$

$$\frac{d(\ln P)}{dp} = \frac{\sum x_i}{p} - \frac{n - \sum x_i}{1-p} = 0$$

$$\sum x_i - p \sum x_i = np - p \sum x_i$$

$$\hat{p} = \frac{\sum x_i}{n}$$

1/21/2009

CSE842, Spring 2009, MSU

38

MLE for N-gram

N gram models can be trained by **counting** and **normalization**

$$\text{Bigram: } P(w_n | w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

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

1/21/2009

CSE842, Spring 2009, MSU

39

BERP Bigram Counts

	I	Want	To	Eat	Chinese	Food	lunch
I	8	1087	0	13	0	0	0
Want	3	0	786	0	6	8	6
To	3	0	10	860	3	0	12
Eat	0	0	2	0	19	2	52
Chinese	2	0	0	0	0	120	1
Food	19	0	17	0	0	0	0
Lunch	4	0	0	0	0	1	0

1/21/2009

CSE842, Spring 2009, MSU

40

BERP Bigram Probabilities

- Normalization: divide each row's counts by appropriate unigram counts

I	Want	To	Eat	Chinese	Food	Lunch
3437	1215	3256	938	213	1506	459

- Computing P(II)
 - C(I,I)/C(all I)
 - $p = 8 / 3437 = .0023$
- A bigram grammar is an NxN matrix of probabilities, where N is the vocabulary size

1/21/2009

CSE842, Spring 2009, MSU

41

Kinds of Knowledge

- As crude as they are, N gram probabilities capture a range of interesting facts about language.

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

1/21/2009

CSE842, Spring 2009, MSU

42

- What about
 - $P(I | I) = .0023$ I I I want
 - $P(I | \text{want}) = .0025$ I want I want
 - $P(I | \text{food}) = .013$ the kind of food I want is ...

1/21/2009

CSE842, Spring 2009, MSU

43

Shannon's Method

- Assigning probabilities to sentences is all well and good, but it's not terribly illuminating. A more interesting task is to turn the model around and use it to generate random sentences that are *like* the sentences from which the model was derived.
- Generally attributed to Claude Shannon.



1/21/2009

CSE842, Spring 2009, MSU

44

Shannon's Method

- Sample a random bigram ($\langle s \rangle, w$) according to its probability
- Now sample a random bigram (w, x) according to its probability
 - Where the prefix w matches the suffix of the first.
- And so on until we randomly choose a $(y, \langle s \rangle)$
- Then string the words together
- $\langle s \rangle$ I
 I want
 want to
 to eat
 eat Chinese
 Chinese food
 food $\langle s \rangle$

1/21/2009

CSE842, Spring 2009, MSU

45

Approximating Shakespeare

- As we increase the value of N , the accuracy of the n -gram model increases
- Generating sentences with random unigrams...
 - Every enter now severally so, let
 - Hill he late speaks; or! a more to leg less first you enter
- With bigrams...
 - 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.

1/21/2009

CSE842, Spring 2009, MSU

46

- Trigrams
 - Sweet prince, Falstaff shall die.
 - This shall forbid it should be branded, if renown made it empty.
- Quadrigrams
 - What! I will go seek the traitor Gloucester.
 - Will you not tell me who I am?

1/21/2009

CSE842, Spring 2009, MSU

47

- There are 884,647 tokens, with 29,066 word form types, in about a one million word Shakespeare corpus
- Shakespeare produced 300,000 bigram types out of 844 million possible bigrams: so, 99.96% of the possible bigrams were never seen (have zero entries in the table).
- Quadrigrams worse: What's coming out looks like Shakespeare because it *is* Shakespeare.

1/21/2009

CSE842, Spring 2009, MSU

48

N-Gram Training Sensitivity

- If we repeated the Shakespeare experiment but trained on a Wall Street Journal corpus, there would be little overlap in the output
- This has major implications for corpus selection or design

Unigram

Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives

Bigram

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her

Trigram

They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

1/21/2009

CSE842, Spring 2009, MSU

49

Evaluation

- How do we know if our models are any good?
 - And in particular, how do we know if one model is better than another.
- Well Shannon's game gives us an intuition.
 - The generated texts from the higher order models sure look better. That is, they sound more like the text the model was obtained from.
 - But what does that mean? How do we quantify that?

1/21/2009

CSE842, Spring 2009, MSU

50

Evaluation

- Standard method
 - Train parameters of our model on a **training set**.
 - Look at the models performance on some new data
 - This is exactly what happens in the real world; we want to know how our model performs on data we haven't seen
 - So use a **test set**. A dataset which is different than our training set, but is drawn from the same source
 - Then we need an **evaluation metric** to tell us how well our model is doing on the test set.
 - One such metric is **perplexity**

1/21/2009

CSE842, Spring 2009, MSU

51

Unknown Words

- But once we start looking at test data, we'll run into words that we haven't seen before (pretty much regardless of how much training data you have).
- With an *Open Vocabulary* task
 - Create an unknown word token <UNK>
 - Training of <UNK> probabilities
 - Create a fixed lexicon L, of size V
 - From a dictionary or
 - A subset of terms from the training set
 - At text normalization phase, any training word not in L changed to <UNK>
 - Now we count that like a normal word
 - At test time
 - Use UNK counts for any word not in training

1/21/2009

CSE842, Spring 2009, MSU

52

Evaluating a Language Model

- Given two language models, how to measure which one is better?
- What could be an intuitive but expensive way?
 - Extrinsic evaluation: speech recognition
 - Expensive
- Intrinsic evaluation: Perplexity

1/21/2009

CSE842, Spring 2009, MSU

53

Perplexity

- Perplexity is the probability of the test set (assigned by the language model), normalized by the number of words:

$$PP(W) = P(w_1, w_2, \dots, w_N)^{\frac{1}{N}}$$

$$= \sqrt[N]{\frac{1}{P(w_1, w_2, \dots, w_N)}}$$

- For bigrams: $PP(W) = \sqrt[N]{\prod_{t=1}^N \frac{1}{P(w_t | w_{t-1})}}$

Minimizing perplexity is the same as maximizing probability

- **The best language model is one that best predicts an unseen test set**

1/21/2009

CSE842, Spring 2009, MSU

54

Evaluating a Language Model

- Given test data (T), which has n sentences: S1, S2, ... Sn
- We could look at the probability under our model P(T) = $\prod_{i=1}^n P(S_i)$, or more conveniently, the log probability

$$\log \prod_{i=1}^n P(S_i) = \sum_{i=1}^n \log P(S_i)$$

- The usual measure is **perplexity**

$$\text{Perplexity} = 2^{-x} \quad \text{where} \quad x = \frac{1}{W} \sum_{i=1}^n \log P(S_i)$$

W is the total number of words in the test data

1/21/2009

CSE842, Spring 2009, MSU

55

Lower perplexity means a better model

- Training 38 million words, test 1.5 million words, WSJ

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

1/21/2009

CSE842, Spring 2009, MSU

56