Natural Language Processing

Lecture 7

Today's Outline

Administration

- panel reminder
- homework

Review

Next Word Prediction

Probabilistic Models

The Probability of a Sequence

Training and Testing

Review

So far we've concentrated on individual words in isolation.

Now we're going to start looking at words in *context*.

We'll start with what looks like an artificial task: predicting the next word in a sequence.

Word Prediction Example

From Monday's NY Times...

Stocks plunged this ...

Word Prediction Example (cont.)

From Monday's NY Times...

Stocks plunged this morning, despite a cut in interest ...

Word Prediction Example (cont.)

From Monday's NY Times...

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

Word Prediction Example (cont.)

From Monday's NY Times...

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

Word Prediction Example (cont.)

From Monday's NY Times...

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 ...

Word Prediction Example (cont.)

From Monday's NY Times...

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.

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

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).

Why do this?

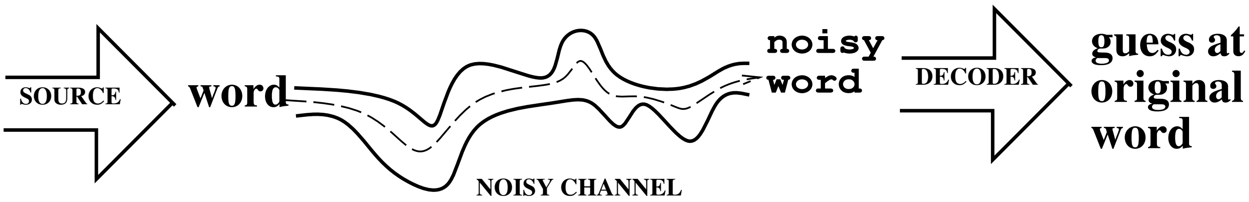
Why would anyone want to predict a word?

You don't really. But if you say you can predict the next word, it means you can rank the likelihood of sequences containing various alternative words.

- you can assess the likelihood/goodness of a sentence

Why do this? (continued)

The Noisy Channel probabilistic model:



Many NLP problems can be modeled as mapping from one string of symbols to another.

In statistical language applications, knowledge of the source (e.g, a statistical model of word sequences) is referred to as a Language Model or a Grammar.

Why do this? (continued)

Example applications that employ language models

- speech recognition
- handwriting recognition
- real word spelling correction
- augmentative communication

Handwriting Recognition Example

Assume a note is given to a bank teller, which the teller reads as *I have a gub.* (example courtesy of a Woody Allen film)

NLP to the rescue ...

- *gub* is not a word
- \- *gun* and *gull* are words, but *gun* has a higher probability in the context of a bank

Real Word Spelling Errors

They are leaving in about fifteen *minuets* to go to her house.
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.
Can they *lave* him my messages?
I need to *notified* the bank of [this problem.]
He is trying to *fine* out.

Real Word Spelling Correction Ex.

Collect a list of commonly substituted words

- piece/peace, whether/weather, their/there ...

Whenever you encounter one of these words in a sentence, construct the alternative sentence as well

Assess the goodness of each and choose the one (word) with the more likely sentence

Example

- blah blah blah the whether
- blah blah blah the weather

Basic Probability Background

Prior Probability (or unconditional probability)

- P(A)
- think of A as a proposition, as in some logic

Conditional Probability

- P(A|B)
- the probability of A given that we know B

Relating Conditionals and Priors

$$P(A \mid B) = \frac{P(A \land B)}{P(B)}$$

rearranging yields

$$P(A \land B) = P(A \mid B)P(B)$$

you could also say

$$P(A \land B) = P(B \mid A)P(A)$$

More Probability

We can view the probability of a word sequence as the probability of a conjunctive event.

For example, the probability of "the big cat" is

- P(the $\wedge$ big $\wedge$ cat)

Chain Rule

Recall the definition of conditional probability

$$P(A \mid B) = \frac{P(A \wedge B)}{P(B)}$$

Rewriting

$$P(A \wedge B) = P(B \mid A)P(A)$$

Or

$$P(the \wedge dog) = P(dog \mid the)P(the)$$

So, even if "the" is more frequent than "dog", conditional rather than individual relative word frequencies will prefer "the dog" rather than "the the"

$$P(dog \mid the) > P(dog)$$

Similarly, relative word frequencies are better than equal probabilities for all words.

Chain Rule (continued)

In general, the probability of a complete string of words

$$P(w_1 \dots w_n) =$$

$$P(w_1^n) =$$

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

$$\Pi_{k=1}^{n} P(w_k \mid w_1^{k-1})$$

Approximating Things

Unfortunately, that's really not helpful in general.

Why?

Approximating Things (continued)

So we make a Markov Assumption...

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

That is, the probability of a word given all previous words is approximated using N previous words.

In other words, we revert to a technique that allows us to 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.

Counting Words in Corpora

Probabilities are based on counting things, so . . .

- what to count?
  - words (this chapter), word classes, word senses, speech acts . . .
  - what is a word? (e.g., are *cat* and *cats* the same word?)
- where to find the things to count?
  - Corpora (online collections of text and speech)

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

Simple N-Grams

An N-gram model uses the previous N-1 words to predict the next one:

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

We'll pretty much always be dealing with $P(word \mid some\ prefix)$

- unigrams: $P(dog)$
- bigrams: $P(dog \mid big)$
- trigrams: $P(dog \mid the\ big)$
- quadrigrams: $P(dog \mid chasing\ the\ big)$

Using N-Grams

Recall

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

For a bigram grammar

$$P(w_1^n) \approx \Pi_{k=1}^{n} P(w_k \mid w_{k-1})$$

Thus, P(sentence) can be computed by multiplying bigram probabilities

- P(I want to each Chinese food) = P(I | "sentence start")P(want | I)P(to | want)P(eat | to)P(Chinese | eat)P(food | Chinese)

BERP: Berkely Restaurant Project

Example bigram grammar fragment:

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

Probabilities seem to capture "syntactic" facts, "world knowledge"

- "eat" is often followed by a NP
- British food is not too popular

N-gram models can be trained by <u>Counting</u> and <u>Normalization</u>

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

Note all the zeroes even with a coherent word set.

BERP Bigram Probabilities

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

	I	want	to	eat	Chinese	food	lunch
I	.0023	.32	0	.0038	0	0	0
want	.0025	0	.65	0	.0049	.0066	.0049
to	.00092	0	.0031	.26	.00092	0	.0037
eat	0	0	.0021	0	.020	.0021	.055
Chinese	.0094	0	0	0	0	.56	.0047
food	.013	0	.011	0	0	0	0
lunch	.0087	0	0	0	0	.0022	0

Computing the probability of "I I"

- "I" unigram count $= 8 + 1087 + 13 \ldots = 3437$
- $p = 8 / 3437 = .0023$

A bigram grammar can be represented as an N-by N matrix of probabilities, where N is the vocabulary size.

Observations on these Probabilities

What's being captured with ...

- P(want | I) = .32
- P(to | want) = .65
- P(eat | to) = .26
- P(food | Chinese) = .56
- P(lunch | eat) = .055

What about...

- P(I | I) = .0023
- P(I | want) = .0025
- P(I | food) = .013

Observations (continued)

P(I | I) = .0023

- I I I I want ...

P(I | want) = .0025

- I want I want ...

P(I | food) = .013

- the kind of food I want is ...

Approximating Shakespeare

As we increase the value of N, the accuracy of the n-gram model increases.

Unigrams

- *Every enter now severally so, let*
- *Hill he late speaks; or! a more to leg less first you enter*

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.*

Approximating Shakespeare (cont.)

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?*

Approximating Shakespeare (cont.)

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. Thus, 99.96 % of the possible bigrams were never seen (have zero entries in the table).

The quadrigrams are worse. What's coming out looks like Shakespeare because it is Shakespeare.

All those zeroes are causing problems.

Some Useful Empirical Observations

A small number of events occur with high frequency.

A large number of events occur with low frequency.

You can quickly collect statistics on the high frequency events.

You might have to wait an arbitrarily long time to get valid statistics on low frequency events.

Some of the zeroes in the table are really zeroes. Some are simply low frequency events you haven't seen yet.

Another Example

Jim Martin did the following...

```
% deroff -w /usr/man/man1/csh.1 | sort |
uniq -c | sort -r
```

And got 1626 unique types out of about 11,000 tokens.

The top seven in terms of frequency:

- 709 the
- 304 is
- 275 to
- 250 of
- 192 and
- 188 command
- 184 csh

But 740 of the 1626 only occur once!

Example (continued)

Even if we add in more and more man pages...

- very few/no additional new high frequency types. But lots of additions to the counts for the existing high frequency types.
- more and more new single instance types

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.

- implications for corpus design

Methodology: Training and Testing

Probabilities come from a Training portion of a 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 portion of the corpus is used to *evaluate* the model, typically using standard *metrics*.

- held out test set
- cross validation
- evaluation differences should be statistically significant

For Next Time

Finish Chapter 6 (6.3 on)

Acknowledgements

Some of this material was taken from the course web pages of James Martin.