Language Models

Language Models

- A language model
  - an abstract representation of a (natural) language phenomenon.
  - an approximation to real language

- Statistical models
  - predictive
  - explicative

Claim

- A useful part of the knowledge needed to allow letter/word predictions can be captured using simple statistical techniques.
- Compute:
  - probability of a sequence
  - likelihood of letters/words co-occurring
- Why would we want to do this?
  - Rank the likelihood of sequences containing various alternative hypotheses
  - Assess the likelihood of a hypothesis

Why is This Useful?

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

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

Real Word Spelling Errors

- They are leaving in about fifteen *minuets* to go to her house.
- The study was conducted mainly *be* John Black.
- Hopefully, all *with* continue smoothly in my absence.
- Can they *lave* him my messages?
- I need to *notified* the bank of….
- He is trying to *fine* out.

For Spell Checkers

- Collect list of commonly substituted words
  - piece/peace, whether/weather, their/there ...

- Example:
  "On Tuesday, the whether …''
  "On Tuesday, the weather …"

Other Applications

- Machine translation
- Text summarization
- Optical character recognition

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
-

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
-       W

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- Wh

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- Wha

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What d

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
- What do

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
- What do you think the next letter is?

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
-     What do you think the next letter is?
- Guess the next word:
-

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do you

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do you think

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do you think the

Letter-based Language Models

- Shannon’s Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do you think the next

Letter-based Language Models

- Shannon's Game
- Guess the next letter:
- What do you think the next letter is?
- Guess the next word:
- What do you think the next word is?

Approximating Natural Language Words

- zero-order approximation: letter sequences are independent of each other and all equally probable:
    - xfoml rxkhrjffjuj zlpwcwkcy ffjeyvkcqsghyd

Approximating Natural Language Words

- first-order approximation: letters are independent, but occur with the frequencies of English text:

  - ocro hli rgwr nmielwis eu ll nbnesebya th eei alhenhtppa oobttva nah

Approximating Natural Language Words

- second-order approximation: the probability that a letter appears depends on the previous letter
  - on ie antsoutinys are t inctore st bes deamy achin d ilonasive tucoowe at teasonare fuzo tizin andy tobe seace ctisbe

Approximating Natural Language Words

- third-order approximation: the probability that a certain letter appears depends on the two previous letters
  - in no ist lat whey cratict froure birs grocid pondenome of demonstures of the reptagin is regoactiona of cre

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 (V)**: number of distinct words that might appear in a corpus (vocabulary size)
- **Tokens (N_T)**: total number of words in a corpus
- **Types seen so far (T)**: number of distinct words seen so far in corpus (smaller than V and N_T)

Word-based Language Models

- A model that enables one to compute the probability, or likelihood, of a sentence S, P(S).
- Simple: Every word follows every other word w/ equal probability (0-gram)
  - Assume |V| is the size of the vocabulary V
  - Likelihood of sentence S of length n is = $1/|V| \times 1/|V| \ldots \times 1/|V|$
  - If English has 100,000 words, probability of each next word is 1/100000 = .00001

Word Prediction: Simple vs. Smart

- Smarter: probability of each next word is related to word frequency (unigram)
  - Likelihood of sentence S = $P(w_1) \times P(w_2) \times \ldots \times P(w_n)$
  - Assumes probability of each word is independent of probabilities of other words.

- Even smarter: Look at probability *given* previous words (N-gram)
  - Likelihood of sentence S = $P(w_1) \times P(w_2|w_1) \times \ldots \times P(w_n|w_{n-1})$
  - Assumes probability of each word is dependent on probabilities of other words.

Chain Rule

- Conditional Probability
  - $P(w_1,w_2) = P(w_1) \cdot P(w_2|w_1)$
- The Chain Rule generalizes to multiple events
  - $P(w_1, \ldots,w_n) = P(w_1)\ P(w_2|w_1)\ P(w_3|w_1,w_2)\ldots P(w_n|w_1\ldots w_{n-1})$
- Examples:
  - P(the dog) = P(the) P(dog | the)
  - P(the dog barks) = P(the) P(dog | the) P(barks| the dog)

Relative Frequencies and Conditional Probabilities

- Relative word frequencies are better than equal probabilities for all words
  - In a corpus with 10K word types, each word would have P(w) = 1/10K
  - Does not match our intuitions that different words are more likely to occur (e.g. the)
- Conditional probability more useful than individual relative word frequencies
  - dog may be relatively rare in a corpus
  - But if we see barking, P(dog|barking) may be very large

For a Word String

- In general, the probability of a complete string of words $w_1^n = w_1 \ldots w_n$ is
- $P(w_1^n)$
- $= P(w_1)P(w_2|w_1)P(w_3|w_1..w_2)\ldots P(w_n|w_1\ldots w_{n-1})$
-
- $= \prod_{k=1}^{n} P(w_k \mid w_1^{k-1})$

- But this approach to determining the probability of a word sequence is not very helpful in general – gets to be computationally very expensive

Markov Assumption

- How do we compute $P(w_n|w_1^{n-1})$?
  Trick: Instead of P(rabbit|I saw a), we use P(rabbit|a).
  - This lets us collect statistics in practice
  - A bigram model: P(the barking dog) = P(the|<start>)P(barking|the)P(dog|barking)
- 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
  - Specifically, for N=2 (bigram):
  - $P(w_1^n) \approx \prod_{k=1}^{n} P(w_k|w_{k-1})$; $w_0 = \text{<start>}$
- Order of a Markov model: length of prior context
  - bigram is first order, trigram is second order, …

Counting Words in Corpora

- What is a word?
  - e.g., are cat and cats the same word?
  - September and Sept?
  - zero and oh?
  - Is seventy-two one word or two? AT&T?
  - Punctuation?
- How many words are there in English?
- Where do we find the things to count?

Simple N-Grams

- An N-gram model uses the previous N-1 words to predict the next one:
  - $P(w_n \mid w_{n-N+1}\, w_{n-N+2\ldots}\, w_{n-1})$
- unigrams: P(dog)
- bigrams: P(dog | big)
- trigrams: P(dog | the big)
- quadrigrams: P(dog | chasing the big)

Using N-Grams

- Recall that
  - Bigram: $P(w_1^n) \approx \prod_{k=1}^{n} P(w_k \mid w_{k-1})$
- For a bigram grammar
  - P(sentence) can be approximated by multiplying all the bigram probabilities in the sequence
- Example:

P(I want to eat Chinese food) =

P(I | <start>) P(want | I) P(to | want) P(eat | to) P(Chinese | eat) P(food | Chinese)

A 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

Additional Grammar

<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

Computing Sentence Probability

- P(I want to eat British food) = P(I|<start>) P(want|I) P(to|want) P(eat|to) P(British|eat) P(food|British) = .25×.32×.65×.26×.001×.60 = .000080
- vs.
- P(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
- N-gram models can be trained by counting and normalization

Training and Testing

- N-Gram probabilities come from a training corpus
  - 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; development test set
  - cross validation
  - results tested for statistical significance

Example

- Training Set
  - <s> The Arabian Knights
  - <s> These are the fairy tales of the east
  - <s> The stories of the Arabian knights are translated in many languages

P(the|<s>)

- <s> The Arabian Knights
- **<s> These** are the fairy tales of the east
- <s> The stories of the Arabian knights are translated in many languages

- $P(\text{the}|\text{<s>}) = 2/3 = 0.67$

P(these|<s>)

- **The** Arabian Knights
- **These** are the fairy tales of the east
- **The** stories of the Arabian knights are translated in many languages

- P(these|<s>) = 1/3 = 0.34

P(Arabian|the)

- **The Arabian** Knights
- These are **the fairy** tales of **the east**
- **The stories** of **the Arabian** knights are translated in many languages

- $P(\text{Arabian}|\text{the}) = 2/5 = 0.4$

P(knights|Arabian)

- The **Arabian Knights**
- These are the fairy tales of the east
- The stories of the **Arabian knights** are translated in many languages

- P(knights|Arabian) = 2/2 = 1.0

P(are|these)

- The Arabian Knights
- **These are** the fairy tales of the east
- The stories of the Arabian knights are translated in many languages

- $P(are|these) = 1/1 = 1.0$

P(the|are)

- The Arabian Knights
- These **are the** fairy tales of the east
- The stories of the Arabian **knights are** translated in many languages

- $P(the|are) = 1/2 = 0.5$

P(fairy|the)

- **The Arabian** Knights
- These are **the fairy** tales of **the east**
- **The stories** of **the Arabian** knights are translated in many languages

- $P(fairy|the) = 1/5 = 0.2$

P(tales|fairy)

- The Arabian Knights
- These are the **fairy tales** of the east
- The stories of the Arabian knights are translated in many languages

- P(tales|fairy) = 1/1 = 1.0

$P(of|tales)$

- The Arabian Knights
- These are the fairy **tales of** the east
- The stories of the Arabian knights are translated in many languages

- $P(of|tales) = 1.0$

P(the|of)

- The Arabian Knights
- These are the fairy tales **of the** east
- The stories of the Arabian knights are translated in many languages

- $P(the|of) = 1/1 = 1.0$

P(east|the)

- **The Arabian** Knights
- These are **the fairy** tales of **the east**
- **The stories** of **the Arabian** knights are translated in many languages

- P(east|the) = 1/5 = 0.2

P(stories|the)

- **The Arabian** Knights
- These are **the fairy** tales of **the east**
- **The stories** of **the Arabian** knights are translated in many languages

- $P(stories|the) = 1/5 = 0.2$

$P(of|stories)$

- The Arabian Knights
- These are the fairy tales of the east
- The **stories of** the Arabian knights are translated in many languages

- $P(of|stories) = 1/1 = 1.0$

P(are|knights)

- The Arabian Knights
- These are the fairy tales of the east
- The stories of the Arabian **knights are** translated in many languages

- $P(are|knights) = 1/1 = 1.0$

P(translated|are)

- The Arabian Knights
- These **are the** fairy tales of the east
- The stories of the Arabian knights **are translated** in many languages

- P(translated|are) = 1/2 =0.5

$P(in|translated)$

- The Arabian Knights
- These are the fairy tales of the east
- The stories of the Arabian knights are **translated in** many languages

- $P(in|translated) = 1/1 = 1.0$

P(many|in)

- The Arabian Knights
- These are the fairy tales of the east
- The stories of the Arabian knights are translated **in many** languages

- $P(\text{many}|\text{in}) = 1/1 = 1.0$

P(languages|many)

- The Arabian Knights
- These are the fairy tales of the east
- The stories of the Arabian knights are translated in **many languages**

- $P(\text{languages}|\text{many}) = 1.0$

Bi-gram Model

P(the|<s>) = 0.67

P(are|these) = 1.0

P(tales|fairy) = 1.0

P(east|the) = 0.2

P(are|knights) = 1.0

P(many|in) = 1.0

P(fairy|the) = 0.2

P(of|stories) = 1.0

P(languages|many) = 0.4

P(Arabian|the) = 0.5

P(of|tales) = 1.0

P(stories|the) = 0.2

P(translated|are) = 0.5

P(knights|Arabian) = 1.0

P(the|of) = 1.0

P(in|translated) = 1.0

Test Sentence

- The Arabian knights are the fairy tales of the east.
- $P(\text{the}|\text{<s>}) \times P(\text{Arabian}|\text{the}) \times P(\text{knights}|\text{Arabian}) \times P(\text{are}|\text{knights}) \times P(\text{the}|\text{are}) \times P(\text{fairy}|\text{the}) \times P(\text{tales}|\text{fairy}) \times P(\text{of}|\text{tales}) \times P(\text{the}|\text{of}) \times P(\text{east}|\text{the})$
- $0.67 \times 0.5 \times 1.0 \times 1.0 \times 0.5 \times 0.2 \times 1.0 \times 1.0 \times 1.0 \times 0.2$
- 0.0067

N-gram Training Sensitivity

- If we repeat the experiment but trained our n-grams on a Wall Street Journal corpus (1 Billion words), what would we get?
- This has major implications for corpus selection or design

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 zeros, But others are simply low frequency events you haven't seen yet.