Indian Institute of Technology Kharagpur
Department of Computer Science & Engineering

CS60075
Natural Language Processing
Autumn 2020

Lecture 2A : Language Models

Sep 9 2020

Sudeshna Sarkar

Language Understanding

How likely is a sentence?

- P (the baby is taking classes on the computer)
- P (about fifteen minutes from)
  P (about fifteen minuets from)
- P (I saw a bus) >> P (eyes awe a boss)

Language Model Definition

- How likely is a sentence $(w_1, w_2, \ldots, w_n)$?

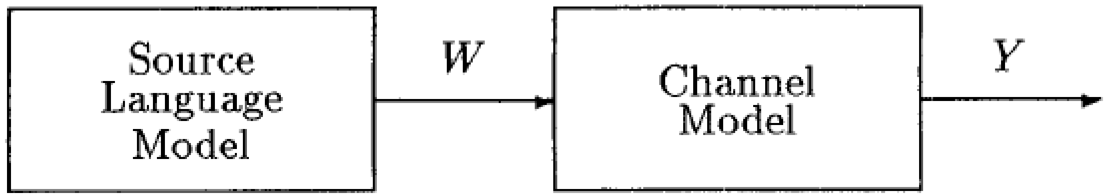
- A statistical language model is a probability distribution over sequences of words.

$$P(w_1, w_2, \ldots, w_n) = P(w_n | w_{n-1}, w_{n-2}, .., w_1)$$

Application

Application	Signal Y
speech recognition	acoustic signal
machine translation	sequence of words in a foreign language
spelling correction	sequence of characters produced by a possibly imperfect typist

source-channel model



$$\Pr(W) \quad \times \quad \Pr(Y \mid W) = \Pr(W, Y)$$

Goal: to determine W from Y

Completion Prediction

- A language model also supports predicting the completion of a sentence.
  - Please turn off your cell _____
  - Your program does not ______
  - *Stocks plunged this ….*
  - Let’s meet in Times ….
- *Predictive text input* systems can guess what you are typing and give choices on how to complete it.

Probabilistic Language Models

The goal: assign a probability to a sentence

Applications:

- Machine Translation:
  - P(**high** winds tonite) > P(**large** winds tonite)
- Spelling Correction
  - The office is about fifteen **minuets** from my house
  - P(about fifteen **minutes** from) > P(about fifteen **minuets** from)
- Speech Recognition
  - P(I saw a van) >> P(eyes awe of an)
- + Summarization, question-answering, …

- Input: a training set of example sentences
- Output: a probability distribution p over L

A naïve method

- Assume we have N training sentences
- Let $w_1, w_2, \ldots, w_n$ be a sentence,
- $c(w_1, w_2, \ldots, w_n)$ be the number of times it appeared in the training data.
- Define a language model:

$$\mathrm{p}(w_1, w_2, \ldots, w_n) = \frac{c(w_1, w_2, \ldots, w_n)}{N}$$

- Given a sequence of n random variables:
- Model the probability of sequences:

$$p(X_1 = w_1, X_2 = w_2, \dots X_n = w_n)$$

- How many sequences possible?

Chain Rule

- Recall the definition of conditional probabilities

p(B|A) = P(A,B)/P(A)   Rewriting: P(A,B) = P(A)P(B|A)

- More variables:

    P(A,B,C,D) = P(A)P(B|A)P(C|A,B)P(D|A,B,C)

- Chain rule in general:

$$p(X_1 = w_1, X_2 = w_2, \ldots X_n = w_n)$$

$$= p(X_1 = w_1).\prod_{i=2}^{n} p(X_i = w_i | X_1 = w_1 \ \ldots X_{i-1} = w_{i-1})$$

First-order Markov process

Andrei Markov

- Chain rule

$$p(X_1 = w_1, X_2 = w_2, \dots X_n = w_n)$$

$$= p(X_1 = w_1).\prod_{i=2}^{n} p(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1})$$

- Markov assumption

$$p(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1}) = p(X_i = w_i | X_{i-1} = w_{i-1})$$

Second-order Markov process

- Chain rule

$$p(X_1 = w_1, X_2 = w_2, \dots X_n = w_n)$$

$$= p(X_1 = w_1). \prod_{i=2}^{n} p(X_i = w_i | X_1 = w_1, \dots, X_{i-1} = w_{i-1})$$

$$p(X_1 = w_1, X_2 = w_2, \dots X_n = w_n) =$$

$$p(X_1 = w_1) \times p(X_2 = w_2 | X_1 = w_1) \times \prod_{i=3}^{n} p(X_i = w_i | X_{i-2} = w_{i-2}, X_{i-1} = w_{i-1})$$

How to handle variable length sentences?

Trigram language model

- A vocabulary V
- Non-negative parameters $q(w|u,v)$ for every trigram
- The probability of a sentence

$$p(w_1, w_2, \dots, w_n) = \prod_{i=1}^{n} q(w_i \mid w_{i-2}, w_{i-1})$$

$w_n = \text{STOP}$

$p(\text{the cat bites STOP}) =$

Estimating Probabilities

- N-gram conditional probabilities can be estimated from raw text based on the ***relative frequency*** of word sequences.
- Maximum likelihood for estimating q
- Let $c(w_1, w_2, \ldots, w_n)$ be the number of the n-gram appeared in the corpus.

$$\mathrm{q}(w_i \mid w_{i=2}, w_{i-1}) = \frac{c(w_{i-2}, w_{i-1}, w_i)}{c(w_{i-2}, w_{i-1})}$$

Corpora

- Corpora are online collections of text and speech
  - Brown Corpus
  - Wall Street Journal
  - AP newswire
  - Hansards
  - DARPA/NIST text/speech corpora (Call Home, ATIS, switchboard, Broadcast News, TDT, Communicator)
  - TRAINS, Radio News

Google N-Gram Release, August 2006

AUG
3

All Our N-gram are Belong to You

Posted by Alex Franz and Thorsten Brants, Google Machine Translation Team

Here at Google Research we have been using word n-gram models for a variety of R&D projects,

…

That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.

Google N-Gram Release

- serve as the incoming 92
- serve as the incubator 99
- serve as the independent 794
- serve as the index 223
- serve as the indication 72
- serve as the indicator 120
- serve as the indicators 45
- serve as the indispensable 111
- serve as the indispensible 40
- serve as the individual 234

Google 1-T Corpus

- *1 trillion word tokens*
  - Number of tokens –1,024,908,267,229
  - Number of sentences –95,119,665,584
  - Number of unigrams –13,588,391
  - Number of bigrams –314,843,401
  - Number of trigrams –977,069,902
  - Number of fourgrams– 1,313,818,354
  - Number of fivegrams– 1,176,470,663

Data Sparsity

- Data sparsity:
- # of all possible n-grams: $|V|^n$, where $|V|$ is the size of the vocabulary. Most of them never occur.

Training Set:
... denied the allegations
... denied the reports
... denied the claims
... denied the request

Test Set:
... denied the offer
... denied the loan

$P(\text{offer} \mid \text{denied the}) = 0$

Generative Model & MLE

- An N-gram model can be seen as a probabilistic automata for generating sentences.

- Relative frequency estimates are ***maximum likelihood estimates*** (MLE) since they maximize the probability that the model *M* will generate the training corpus *T*.

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmax}}\, P(T \mid M(\lambda))$$

Evaluation of the Model

- Does our language model prefer good sentences to bad ones?
  - Assign higher probability to "real" or "frequently observed" sentences
    - Than "ungrammatical" or "rarely observed" sentences?
- We train parameters of our model on a **training set**.
- We test the model's performance on data we haven't seen.
  - A **test set** is an unseen dataset that is different from our training set, totally unused.
  - An **evaluation metric** tells us how well our model does on the test set.

Extrinsic evaluation of N-gram models

- Put each model in a task
  - spelling corrector, speech recognizer, MT system
- Run the task, get an accuracy for A and for B
  - How many misspelled words corrected properly
  - How many words translated correctly
- Compare accuracy for A and B
- Extrinsic evaluation

Time-consuming

Intrinsic Evaluation

- Sometimes use **intrinsic** evaluation: **perplexity**
- Intuition: The Shannon Game:
  - How well can we predict the next word?

Perplexity

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

- Gives the highest P(sentence)

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

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

Perplexity is the inverse probability of the test set, normalized by the number of words:

Chain rule: $$\mathrm{PP}(W) = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i | w_1 \ldots w_{i-1})}}$$

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

Minimizing perplexity is the same as maximizing probability

Train and Test Corpora

- A language model is trained on a large corpus of text to estimate good parameter values.
- Model can be evaluated based on its ability to predict a high probability for a disjoint (held-out) test corpus
- May need to ***adapt*** a general model to a small amount of new (***in-domain***) data by adding highly weighted small corpus to original training data.

False independence assumption

- We assume that each word is only conditioned on the previous n-1 words
- “The dogs chasing the cat bark”.
- The tri-gram probability P (bark|the cat) is very low

Human Word Prediction

- The ability to predict future words in an utterance.
- How?
  - Domain knowledge
  - Syntactic knowledge
  - Lexical knowledge