CSE 4392 Special Topics
Natural Language Processing

Language Models

2024 Spring

An Example

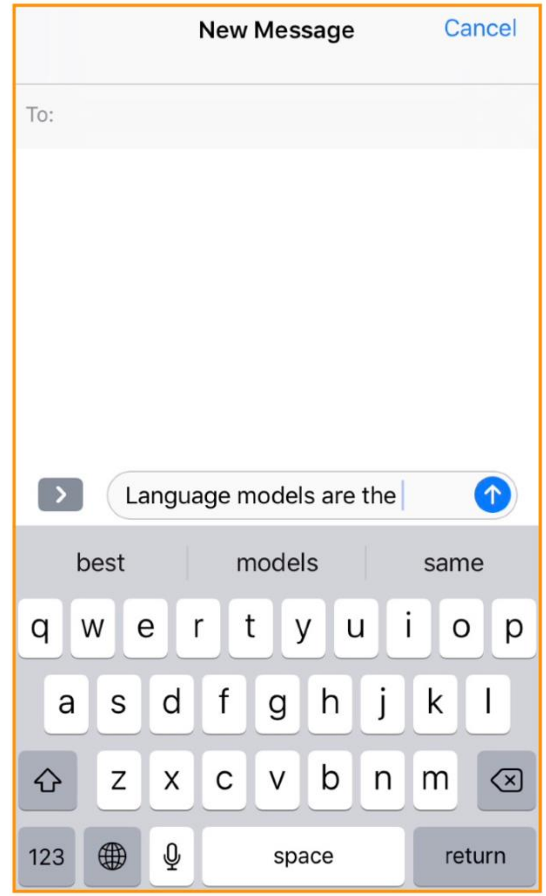
Today in Arlington, TX, it's 45F and <u>sunny</u>.

vs.

Today in Arlington, TX, it's 45F and <u>blue</u>.

- Both are grammatical
- But which is more likely?

Language Models are Everywhere



And Many Applications

- Predicting words is important in many situations
  - Machine translation

    P(a **smooth** finish) > *P*(a **flat** finish)
  - Speech recognition/Spell checking

    P(high school **principal**) > *P*(high school **principle**)
  - Information extraction, question answering

Impact on Downstream Applications

Language Resources	Adaptation	Word		PP
		Cor.	Acc.	
1. Doc-A		54.5%	45.1%	49972
2. Trans-C(L)		63.3%	50.6%	1856.5
3. Trans-B(L)		70.2%	60.3%	318.4
4. Trans-A(S)		70.4%	59.3%	442.3
5. Trans-B(L)+Trans-A(S)	CM	72.6%	63.9%	225.1
6. Trans-B(L)+Doc-A	KW	72.1%	64.2%	247.5
7. Trans-B(L)+Doc-A	KP	73.1%	65.6%	259.7
8. Trans-A(L)		75.2%	67.3%	148.6

(Miki et al. 2006)

New Approach to Language Modeling Reduces Speech Recognition Errors by Up to 15%

Ankur Gandhe

Principal, Applied Scientist

Alexa Speech group, Amazon

WHAT IS A LANGUAGE MODEL?

- Probabilistic model of a sequence of words.
  - How likely is a given phrase/sentence/paragraph/document?
- Joint probability distribution:

$$P(w_1, w_2, \ldots, w_n)$$

Chain Rule

$$\begin{aligned} P(X_1, X_2, \ldots X_n) &= P(X_1)P(X_2|X_1)P(X_3|X_1, X_2)\ldots \\ &= \prod_{i=1}^{n} P(X_i|X_1, \ldots, X_{i-1}) \end{aligned}$$

- Sentence: “the sun rises and shines”

P(the sun rises and shines) = P(the) * P(sun | the) * P(rises | the sun) * P(and | the sun rises) * P(shines | the sun rises and)

Estimating The Probabilities

$$P(rises \mid the\ sun) = \frac{count(the\ sun\ rises)}{count(the\ sun)}$$

$$P(and \mid the\ sun\ rises) = \frac{count(the\ sun\ rises\ and)}{count(the\ sun\ rises)}$$

$$\vdots$$

Maximum Likelihood Estimate (MLE)

- With a vocabulary of size V,
  - number of sequences of length $n = V^n$
- Typical vocab size of 40k words (English):
  - even just considering sentences of <=11 words results in $4*10^{50}$ different sentences (number of atoms on earth only ~10^{50})
- Use a corpus to count these word sequences

Markov Assumption

- Use only recent past in the sequence to predict next word

- Reduce the number of estimated parameters in exchange for model capacity (can model longer sentences now!)

- 1st order:
  $$P(shines|the\ sun\ rises\ and) \cong P(shines|and)$$
- 2nd order:
  $$P(shines|the\ sun\ rises\ and) \cong P(shines|rises\ and)$$

K-th Order Markov Chain

- Consider only the last k words from the context:

$$P(w_i \mid w_1 w_2 \ldots w_{i-1}) \approx P(w_i \mid w_{i-k} \ldots w_{i-1})$$

which implies the probability of a sequence is:

$$P(w_1 w_2 \ldots w_n) \approx \prod_i P(w_i \mid w_{i-k} \ldots w_{i-1})$$

k+1 gram

N-gram Language Models

- Unigram $$P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i)$$
- Bigram $$P(w_1, w_2, ...w_n) = \prod_{i=1}^{n} P(w_i | w_{i-1})$$
- And trigram, 4-gram, etc.
- Larger the *n*, more accurate and better the language model (but at a higher cost)
- Remember the data is *infinite*!

Text Generations using N-grams

Unigram *release millions See ABC accurate President of Joe Will cheat them a CNN megynkelly experience @ these word out- the*

Bigram *Thank you believe that @ABC news, New Hampshire tonight and the false editorial I think the great people Nikki Haley . ''*

Trigram *We are going to MAKE AMERICA GREAT AGAIN! #MakeAmericaGreatAgain https: //t.co/DjkdAzT3WV*

$$\arg \max_{(w_1, w_2, \ldots, w_n)} \Pi_{i=1}^{n} P(w_i | w_{<i})$$

Text Generations using N-grams

Unigram	*release millions See ABC accurate President of Joe Will cheat them a CNN megynkelly experience @ these word out- the*
Bigram	*Thank you believe that @ABC news, New Hampshire tonight and the false editorial I think the great people Nikki Haley . ''*
Trigram	*We are going to MAKE AMERICA GREAT AGAIN! #MakeAmericaGreatAgain https: //t.co/DjkdAzT3WV*

Typical LMs are not sufficient to handle long-range dependencies:

"Alice/Bob could not go to work that day because she/he had a doctor's appointment"

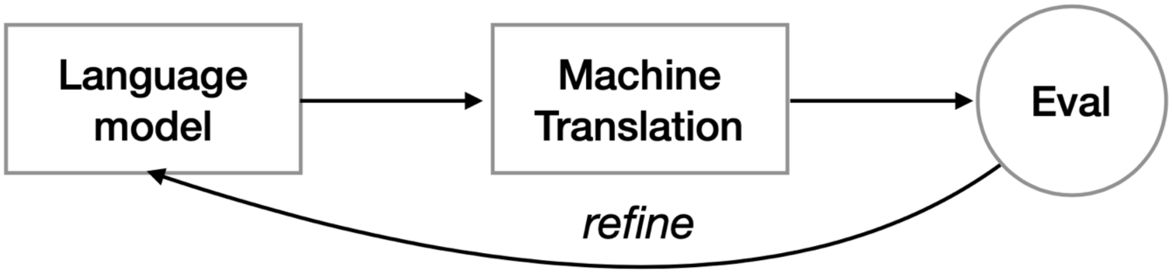
Evaluating Language Models

- A good language model should assign higher probability to typical, grammatically correct sentences

- Research process:
  - Train parameters on a suitable training corpus
    - Assumption: observed sentences ~ good sentences
  - Test on *different, unseen* corpus
    - Training on **any part** of test set not acceptable!
  - Evaluation metric

Extrinsic Evaluation

- Train LM → Apply to task → Observe accuracy



- Directly optimized for downstream tasks
  - Higher accuracy → better model

- Expensive, time consuming
- Hard to optimize downstream objective (indirect feedback)

Perplexity (per word)

- Measures how well a probability distribution (or a model) predicts a sample
- For a corpus S with sentences S_1, S_2, … S_n.

$$\text{ppl}(S) = 2^x \text{ where } x = -\frac{1}{W}\sum_{i=1}^{n}\log_2 P(S_i)$$

A form of cross entropy

where W is the total number of words in test corpus

- Unigram model: $$x = -\frac{1}{W}\sum_{i=1}^{n}\sum_{j=1}^{m} log_2 P(w_j^i)$$

j^{th} word in i^{th} sentence

- Minimizing perplexity ~ maximizing probability

Intuition of Perplexity

- If our n-gram model (with vocabulary V) has the following probability:

$$P(w_i|w_{i-n}, ...w_{i-1}) = \frac{1}{|V|} \quad \forall w_i$$

what is the perplexity on the test corpus?

$$\text{ppl} = 2^{-\frac{1}{W} W * log(1/|V|)} = |V|$$

- The model is "fine" with observing any word at every step!

Pros and Cons of Perlexity

Pros	**Cons**
Fast to compute, eliminate ”bad” models that can’t perform well in expensive real-world testing	Not good for final evaluation: measures model’s confidence, not accuracy
Model’s uncertainty/information density is useful information	Not fair comparison across models trained on different datasets
Statistically robust (not easily influenced by a single outlier sentence in the dataset)	Can reward models trained on toxic or outdated dataset

Quiz: PPL of Bigrams

- Given the following training corpus:

  S1: you have five apples

  S2: you have no oranges

  S3: no apples have you

- What is the ppl of the bigram language model on this test sentence:

  S4: you have no apples

$$\boxed{\mathrm{ppl(S)} = 2^x \text{ where } x = -\frac{1}{W}\sum_{i=1}^{n}\log_2 P(S_i)}$$