

Probabilistic Language Modeling

Outline

- Language Modeling - What it is
- Why Language Modeling - Use cases
- N-gram Language Models
- Chain Rule
- Challenges
- Language Model Evaluation

How do they do it?

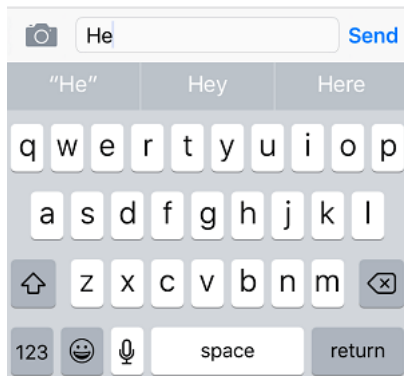


Figure: Word Prediction

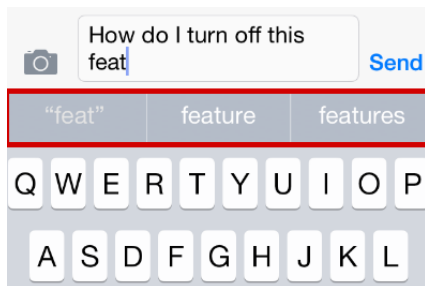


Figure: Word Prediction in the sequence

Downstream Applications - Language Modeling

- Sentence Completion
- Machine Translation
- OCR and Hand-writing recognition
- Speech Recognition

Use Cases - Language Modeling

- Gmail Smart Compose
- Spelling Correction
- Machine Translation

Gmail Smart Compose

- Supports predicting completion of sentence
- Real-time suggestions in Gmail that assists users in writing mails

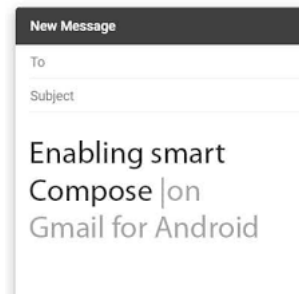


Figure: Gmail Smart Compose¹

¹Source of image- <https://encrypted-tbn0.gstatic.com/>

Sequence Prediction

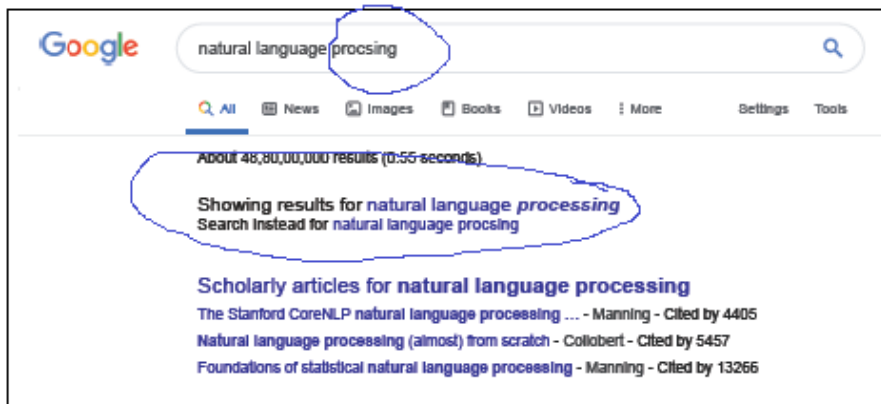


Figure: Google's Sequence Prediction

Spelling Correction

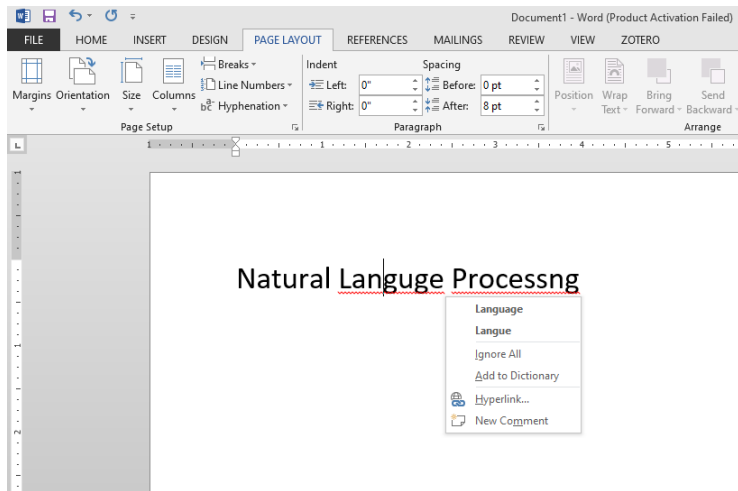


Figure: Microsoft word spelling correction

Context Sensitive Spelling Correction

- Natural Language Proposing
- Natural Language Processing

An Example

- $P(\text{Natural Language Proposing}) < P(\text{Natural Language Processing})$
- $P(\text{Proposing} \mid \text{Natural, Language}) < P(\text{Processing} \mid \text{Natural, Language})$

Applications - Probabilistic Language Modeling

Speech Recognition

- $P(\text{Ice cream}) > P(\text{I scream})$

















HOMOPHONES		HOMOPHONES	
Break  I need a break .	Brake  He stopped with a squeal of the brakes .	Right  Keep on the right side of the road.	Write  She had to write a report on the project.
Buy  I want to buy a new coat.	By  The telephone is by the window.	Farther  ...	Father  Andrew was very excited about
HOMOPHONES		HOMOPHONES	
Cite  He was cited for bravery.	Site  A site has been chosen for the new school.	Allowed  Smoking is not allowed here.	Aloud  The pain made him cry aloud .
Cellar  We don't use our coal cellar anymore.	Seller  She is a flower seller .	Altar  The groom left the bride standing	Alter  I can't alter the plans.

Figure: Homophones²

²Source of image- <http://yourdictionary.com/examples-of-homophones.html>

Machine Translation

Machine Translation

- Which phrase is more probable in the target language?
- $P(\text{high winds}) > P(\text{large winds})$

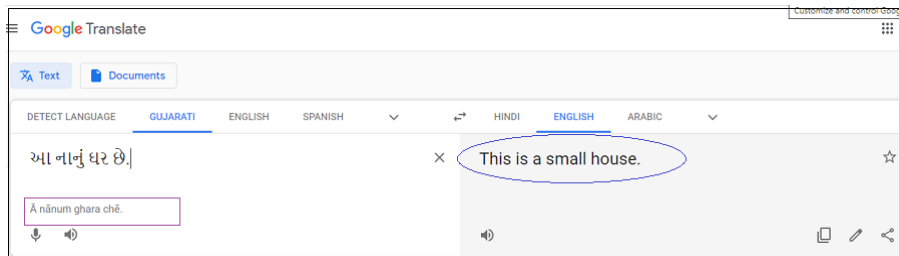


Figure: Google MT - Gujarati to English

Language Modeling

Definition

Language Models are stochastic process for computing probability of a sentence or sequence of words.

$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$

- Probabilistic approach - Finding how probable a sentence may occur given the sequence of words
- In the probabilistic world, the Language Model is used to assign the probability $P(W)$ to every possible word sequence W .

Language Modeling

John Rupert Firth - An English linguist

“You shall know a word by the company it keeps.”

— Firth, J.R.



Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Completion prediction

- Please turn off your cell.... **phone**
- Your program does not... **run**
- I call you... **back**
- Thank you for your kind... **help**

Language Modeling

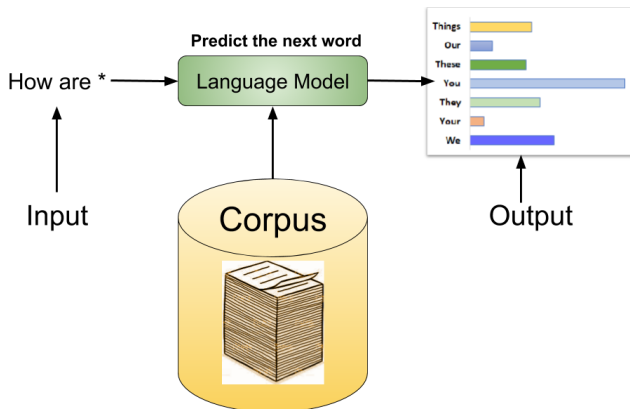


Figure: Language Model

Probabilistic Language Modeling

- **Function 1:** Compute the probability of sequence of words

$$P(W) = P(w_1, w_2, w_3, \dots, w_n)$$

- **Function 2::** Probability of an upcoming word:

$$P(w_4 \mid w_1, w_2, w_3)$$

- A model that computes either of these is called language model.

Types of Language Models

- Unigram Language Model [No dependency of the word]
- Bigram Language Model [The word is dependent on the previous word]
- Trigram Language Model [The word is dependent on the previous two words]
- N-gram Language Model [The word is dependent on the previous N-1 words]

N-gram Language model

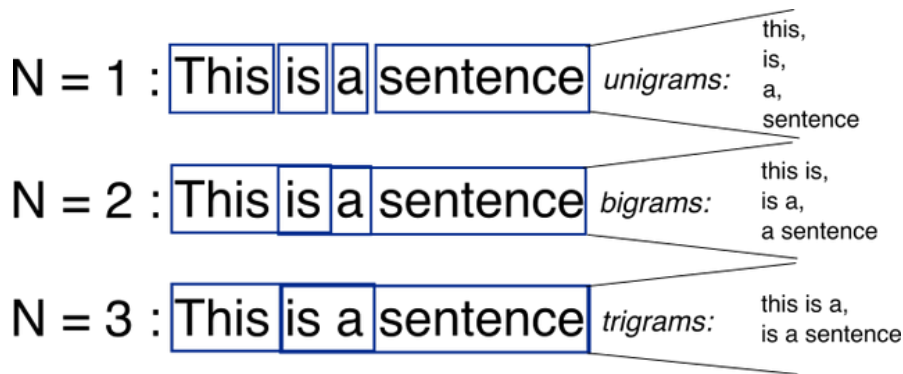


Figure: N-gram Language Modeling ³

Word Count in N-gram Language Models

There are N number of tokens in the collection.

Unigram Language Model

$$P(W_i) = \frac{\text{Count}(W_i)}{N}$$

Bigram Language Model

$$P(W_i) = \frac{\text{Count}(W_{i-1}W_i)}{\text{Count}(W_i)}$$

Trigram Language Model

$$P(W_i) = \frac{\text{Count}(W_{i-2}W_{i-1}W_i)}{\text{Count}(W_{i-2}W_{i-1})}$$

N-gram Language Model

$$P(W_i) = \frac{\text{Count}(W_{i-(N-1)}W_{i-(N-2)}\dots W_{i-1}W_i)}{\text{Count}(W_{i-(N-1)}W_{i-(N-2)}\dots W_i)}$$

Computing $P(W)$

- How to compute the joint probability
- (natural, language, processing, is,)
- It depends on the Chain Rule of probability.

Chain Rule

Conditional Probability

The Probability of occurring B given that A has already occurred.

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

Multiple Variables

The probability of seeing word B given that A, B and C have already been seen in some sentence

$$P(A, B, C, D) = P(A)P(B \mid A)P(C \mid A, B)P(D \mid A, B, C)$$

Generalized form of Chain Rule

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

Chain Rule

Conditional Probability

The Probability of occurring B given that A has already occurred.

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

Multiple Variables

The probability of seeing word B given that A, B and C have already been seen in some sentence

$$P(A, B, C, D) = P(A)P(B \mid A)P(C \mid A, B)P(D \mid A, B, C)$$

Generalized form of Chain Rule

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

Chain Rule

Conditional Probability

The Probability of occurring B given that A has already occurred.

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

Multiple Variables

The probability of seeing word B given that A, B and C have already been seen in some sentence

$$P(A, B, C, D) = P(A)P(B \mid A)P(C \mid A, B)P(D \mid A, B, C)$$

Generalized form of Chain Rule

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

Language Modeling

An Example

P (Natural Language Processing is important)

Computing Probability Values

Product of probability values using chain rule

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i \mid w_1 w_2 \dots w_{i-1})$$

$P(\text{Natural Language Processing is important})$

$$P(\text{Natural}) \times P(\text{Language} \mid \text{Natural}) \times P(\text{Processing} \mid \text{Natural Language}) \times P(\text{is} \mid \text{Natural Language Processing}) \times P(\text{important} \mid \text{Natural Language Processing is})$$

Computing Probability Values

Product of probability values using chain rule

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i \mid w_1 w_2 \dots w_{i-1})$$

$P(\text{Natural Language Processing is important})$

$$P(\text{Natural}) \times P(\text{Language} \mid \text{Natural}) \times P(\text{Processing} \mid \text{Natural Language}) \times P(\text{is} \mid \text{Natural Language Processing}) \times P(\text{important} \mid \text{Natural Language Processing is})$$

Computing Probability Values

Product of probability values using chain rule

$$P(w_1 w_2 \dots w_n) = \prod_i P(w_i \mid w_1 w_2 \dots w_{i-1})$$

$P(\text{Natural Language Processing is important})$

$$P(\text{Natural}) \times P(\text{Language} \mid \text{Natural}) \times P(\text{Processing} \mid \text{Natural Language}) \times P(\text{is} \mid \text{Natural Language Processing}) \times P(\text{important} \mid \text{Natural Language Processing is})$$

Language Modeling

Example

Language Modeling also computes the probability of next word in the sequence

$$P(w_4 \mid w_1, w_2, w_3)$$

Google Ngram Example

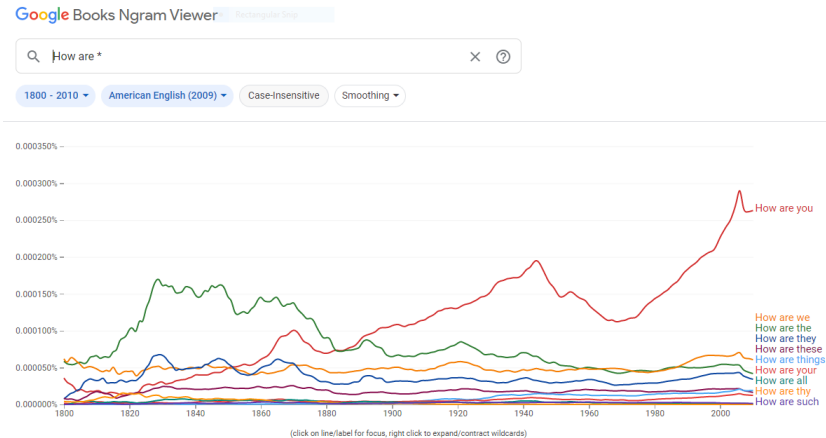


Figure: Google Ngram - How are *

Google Ngram Example

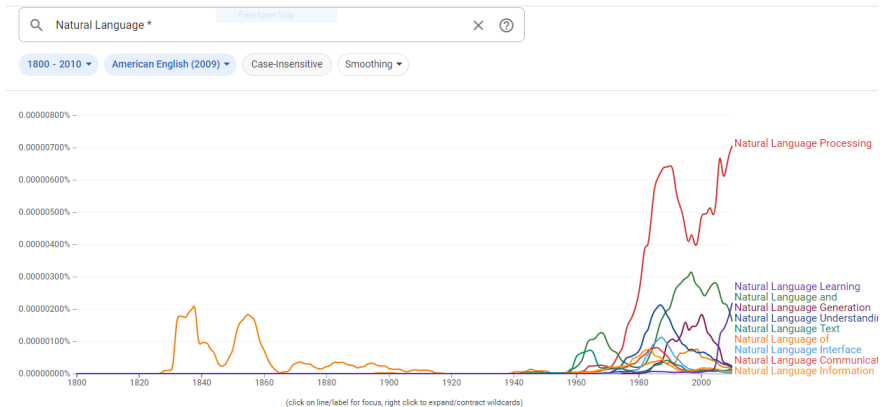


Figure: Google Ngram - Natural Language *

Google Ngram Example

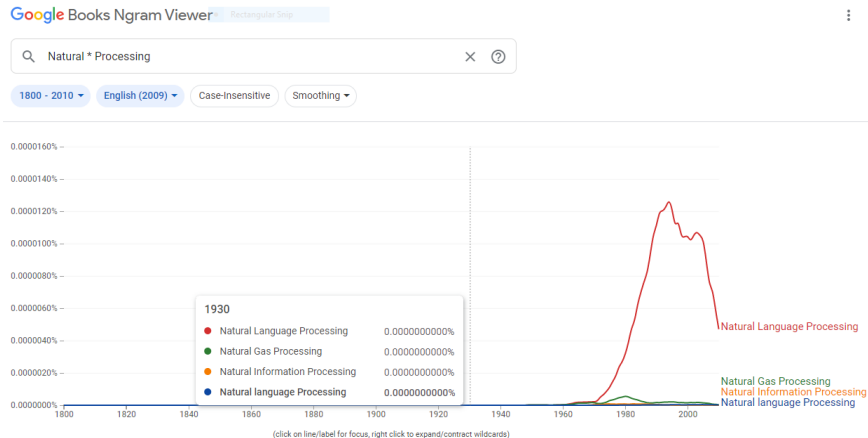


Figure: Google Ngram - Natural * Processing

Maximum Likelihood Estimation (MLE)

MLE

Values are converted to probability... that computes the "most probable" from observed data

$$P(W_i) = \frac{\text{Count}(W_{i-1}W_i)}{\text{Count}(W_i)}$$

$$P(W_i) = \frac{C(W_{i-1}W_i)}{C(W_i)}$$

An example

$$P(W_i) = \frac{C(W_{i-1}W_i)}{C(W_i)}$$

- $\langle s \rangle$ I am here $\langle /s \rangle$
- $\langle s \rangle$ who am I $\langle /s \rangle$
- $\langle s \rangle$ I would like to know $\langle /s \rangle$

Estimating bigrams

- $P(I \mid \langle s \rangle) = 2/3$
- $P(\langle /s \rangle \mid \text{here}) = 1$
- $P(\text{would} \mid I) = 1/3$
- $P(\text{here} \mid \text{am}) = 1/2$
- $P(\text{know} \mid \text{like}) = 0$

An example

$$P(W_i) = \frac{C(W_{i-1}W_i)}{C(W_i)}$$

- $\langle s \rangle$ I am here $\langle /s \rangle$
- $\langle s \rangle$ who am I $\langle /s \rangle$
- $\langle s \rangle$ I would like to know $\langle /s \rangle$

Estimating bigrams

- $P(I \mid \langle s \rangle) = 2/3$
- $P(\langle /s \rangle \mid \text{here}) = 1$
- $P(\text{would} \mid I) = 1/3$
- $P(\text{here} \mid \text{am}) = 1/2$
- $P(\text{know} \mid \text{like}) = 0$

Bigram counts from 9222 Restaurant Sentences

	i	want	to	eat	chinese	food	lunch	spend
i	5	827	0	9	0	0	0	2
want	2	0	608	1	6	6	5	1
to	2	0	4	686	2	0	6	211
eat	0	0	2	0	16	2	42	0
chinese	1	0	0	0	0	82	1	0
food	15	0	15	0	1	4	0	0
lunch	2	0	0	0	1	0	0	0
spend	1	0	1	0	0	0	0	0

Table: Bigrams Counts for Restaurant Corpus

Normalized by Unigram - Counts to Probabilities

Unigram Counts

i	want	to	eat	chinese	food	lunch	spend
2533	927	2417	746	158	1093	341	278

Bigram Probabilities

	i	want	to	eat	chinese	food	lunch	spend
i	0.002	0.33	0	0.0036	0	0	0	0.00079
want	0.0022	0	0.66	0.0011	0.0065	0.0065	0.0054	0.0011
to	0.00083	0	0.0017	0.28	0.00083	0	0.0025	0.087
eat	0	0	0.0027	0	0.021	0.0027	0.056	0
chinese	0.0063	0	0	0	0	0.52	0.0063	0
food	0.014	0	0.014	0	0.00092	0.0037	0	0
lunch	0.0059	0	0	0	0	0.0029	0	0
spend	0.0036	0	0.0036	0	0	0	0	0

Computing Sentence Probabilities

$$P(<s>I \text{ want english food } </s>)$$

$$= P(I \mid <s>) \times P(\text{want} \mid I) \times P(\text{english} \mid \text{want}) \times P(\text{food} \mid \text{english}) \\ \times P(</s> \mid \text{food})$$

$$= 0.000031$$

Computing Sentence Probabilities

$$P(<s>I \text{ want english food } </s>)$$

$$= P(I \mid <s>) \times P(\text{want} \mid I) \times P(\text{english} \mid \text{want}) \times P(\text{food} \mid \text{english}) \\ \times P(</s> \mid \text{food})$$

$$= 0.000031$$

What knowledge does n-gram represent?

- $P(\text{english} \mid \text{want}) = .0011$
- $P(\text{chinese} \mid \text{want}) = .0065$
- $P(\text{to} \mid \text{want}) = .66$
- $P(\text{eat} \mid \text{to}) = .28$
- $P(\text{food} \mid \text{to}) = 0$
- $P(\text{want} \mid \text{spend}) = 0$
- $P(i \mid \langle s \rangle) = .25$

Challenges

- N-gram model might be missing some words - no matter how training corpus is large
- Probability computation might results in zero for some sequence of words
- The trained model considers the sequence impossible if it is not in the training set

An Example

Test Data: I want to spend money.

$P(< s> \text{I want to spend money } < /s>)$

$$= P(\text{I} \mid < s>) \times P(\text{want} \mid \text{I}) \times P(\text{to} \mid \text{want}) \times P(\text{spend} \mid \text{to}) \times \\ P(\text{money} \mid \text{to}) \times P(< /s> \mid \text{money})$$

Probability of the sentence

$$= 0.25 \times 0.33 \times 0.66 \times \underline{0} \times \underline{0} = 0$$

An Example

Test Data: I want to spend money.

$P(< s> \text{I want to spend money } < /s>)$

$$= P(\text{I} \mid < s>) \times P(\text{want} \mid \text{I}) \times P(\text{to} \mid \text{want}) \times P(\text{spend} \mid \text{to}) \times \\ P(\text{money} \mid \text{to}) \times P(< /s> \mid \text{money})$$

Probability of the sentence

$$= 0.25 \times 0.33 \times 0.66 \times \underline{0} \times \underline{0} = 0$$

Laplace Smoothing (Additive smoothing)

- Laplace Smoothing is introduced to solve the problem of zero probability.
- Pretend as if we saw each word (N-gram) one more time that we actually did
- Just add one to all the counts!

Laplace smoothed bigrams counts

	i	want	to	eat	chinese	food	lunch	spend
i	6	828	1	10	1	1	1	3
want	3	1	609	2	7	7	6	2
to	3	1	5	687	3	1	7	212
eat	1	1	3	1	17	3	43	1
chinese	2	1	1	1	1	83	2	1
food	16	1	16	1	2	5	1	1
lunch	3	1	1	1	2	1	1	1
spend	2	1	2	1	1	1	1	1

Table: Laplace smoothed bigram counts

Smoothing techniques

- Laplacian smoothing (Add-1 smoothing)
- Add-K smoothing
- Backoff and Interpolation
- Kneser-Ney Smoothing
 - ① Absolute discounting
 - ② Kneser-Ney discounting

Choosing Vocabulary

- Open Vs. Closed Vocabulary
- Criteria for building closed vocabulary
 - 1 Minimum word frequency (e.g 2)
 - 2 Upper bound for number of words in vocabulary (e.g $|V| = 10000$)

Out Of Vocabulary (OOV) words

- Words may appear that does not appear while building the model
- These words are ‘Out Of Vocabulary (OOV)’ words
- It causes the overall sentence or phrase probabilities to zero
- Replace the word which is not in the vocabulary by a special token
- $\langle \text{UNK} \rangle$
- Count the probability with $\langle \text{UNK} \rangle$ as any other words in the corpus

Language Model Evaluation

Does it predict good symbol/word/sequence?

Assigns higher probability to frequently observed word (sentence) over rarely seen word (sentence)

Training and Testing Set

- Model is trained on a training set, parameters are learned
- Model is tested on held-out (Test data) to check the prediction accuracy of the model

Extrinsic Evaluation

- Two models, A and B, are built on some dataset
- Use each Model for some NLP applications, such as Statistical Machine Translation or spelling correction
- Measure the accuracy value of A and B
- Compare the accuracy of A and B

Extrinsic Evaluation

- Two models, A and B, are built on some dataset
- Use each Model for some NLP applications, such as Statistical Machine Translation or spelling correction
- Measure the accuracy value of A and B
- Compare the accuracy of A and B

Extrinsic Evaluation

- Two models, A and B, are built on some dataset
- Use each Model for some NLP applications, such as Statistical Machine Translation or spelling correction
- Measure the accuracy value of A and B
- Compare the accuracy of A and B

Extrinsic Evaluation

- Two models, A and B, are built on some dataset
- Use each Model for some NLP applications, such as Statistical Machine Translation or spelling correction
- Measure the accuracy value of A and B
- Compare the accuracy of A and B

Intrinsic Evaluation - Perplexity

- Perplexity is a measurement of how well a probability model predicts a word or a sequence of words.
- Intuitively, perplexity can be understood as a measure of uncertainty.

$$\text{For Bigram Model, } PP(W_N) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(W_i) \mid W_{i-1}}} \quad (1)$$

$$\text{For Trigram Model, } PP(W_N) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(W_i) \mid W_{i-1} W_{i-2}}} \quad (2)$$

A minimum perplexity is the indication of a better language model.

Summary

- Word or sequence prediction models based on the probabilities in the training data
- Spelling Correction, Speech recognition, Smart email compose, Machine Translation
- Neural Network based language model performs better

Thank You All.....