Language Modeling and Natural Language Generation

CSCI 1460: Computational Linguistics

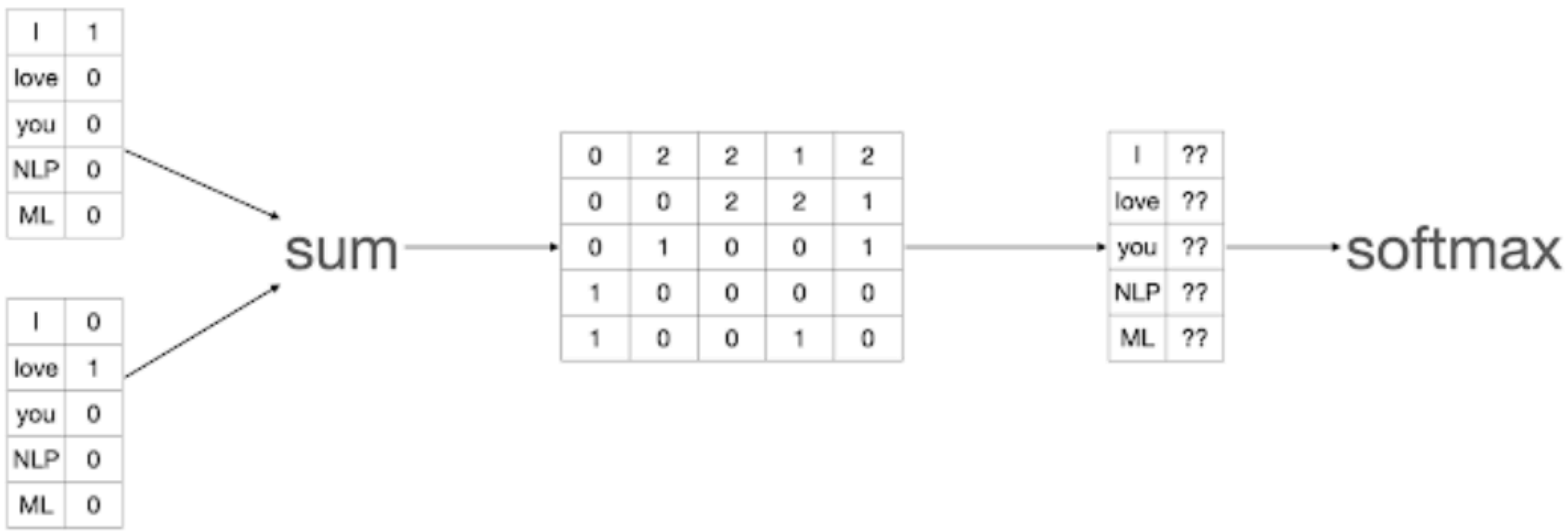
Lecture 9

Ellie Pavlick

Fall 2023

Consider the below network. It is a single-layer perceptron using one-hot encodings. I am using a "bag of vectors" approach, i.e., I summed up the word representations element-wise to obtain a representation for a phrase. What will be the predicted next word? * 1 point

Vocab: {I, love, you, NLP, ML}
Input: I love



Consider the ... one-hot * 1 point

encodings. I a... d up the

word represe... r a phrase.

What will be t...

```python
import numpy as np

v = np.array([1, 1, 0, 0, 0])

w = np.array([[0, 2, 2, 1, 2],
              [0, 0, 2, 2, 1],
              [0, 1, 0, 0, 1],
              [1, 0, 0, 0, 0],
              [1, 0, 0, 1, 0]])


print(v.dot(w))
print(v.T.dot(w))
print(np.matmul(v,w))
print(np.matmul(v.T,w))
```

```
[0 2 4 3 3]
[0 2 4 3 3]
[0 2 4 3 3]
[0 2 4 3 3]
```

I	1
love	0
you	0
NLP	0
ML	0

I	0
love	1
you	0
NLP	0
ML	0

softmax

I am building an MLP classifier to predict whether a review is positive or negative (the same as in question 1). I train my word embedding layer in the process of training the network, and after I've finished training, I cluster words using the trained embeddings. Which of the following should I expect to see? * 1 point

- (selected) words that occur in positive reviews cluster together, and words that occur in negative reviews cluster together (e.g., good/great/awesome vs. bad/awful/terrible)
- nouns cluster together, and verbs cluster together (e.g., food/service/ambiance vs. eat/drink/meet)
- content words cluster together, and stop words cluster together (e.g., food/eat/favorite vs. is/of/and)

Topics

- What is language modeling? When do we use it?
- Ngram language models
- Smoothing
- Perplexity

Topics

- **What is language modeling? When do we use it?**
- Ngram language models
- Smoothing
- Perplexity

Language Modeling

Definition

- Assign a probability to a sequence of words

OR

- Given a sequence of words, predict the most likely next word

OR

- Generate likely sequences of words

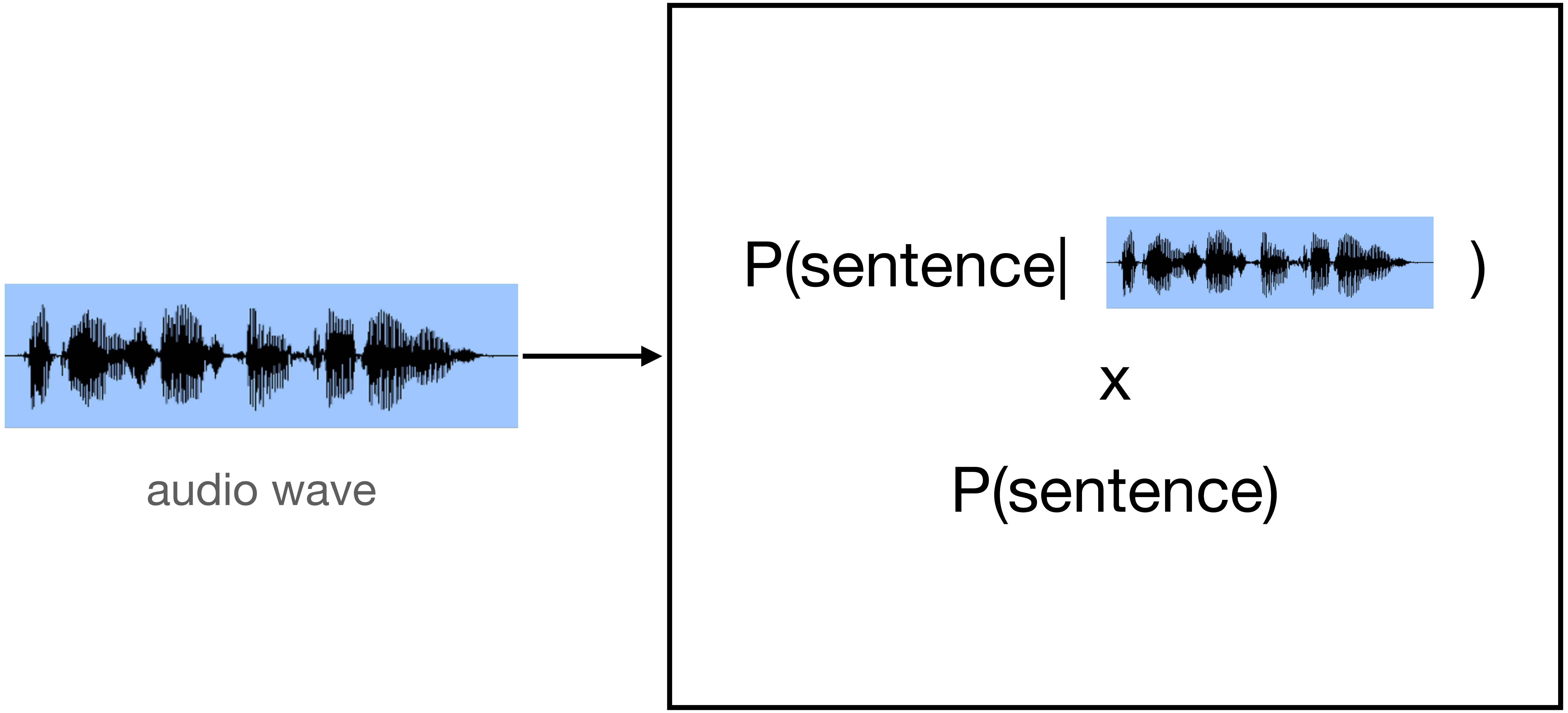
Language Modeling

Applications

- Unconstrained text generation (fun, but not super practical)
- Conditional text generation, e.g.,:
  - Machine translation: e.g., find most likely English sentence given Mandarin sentence
  - Speech recognition: e.g., find most likely English sentence given acoustic input
  - Summarization: e.g., find most likely 50 word English document given a 1000 word English document
  - …
- Representation learning

Language Modeling

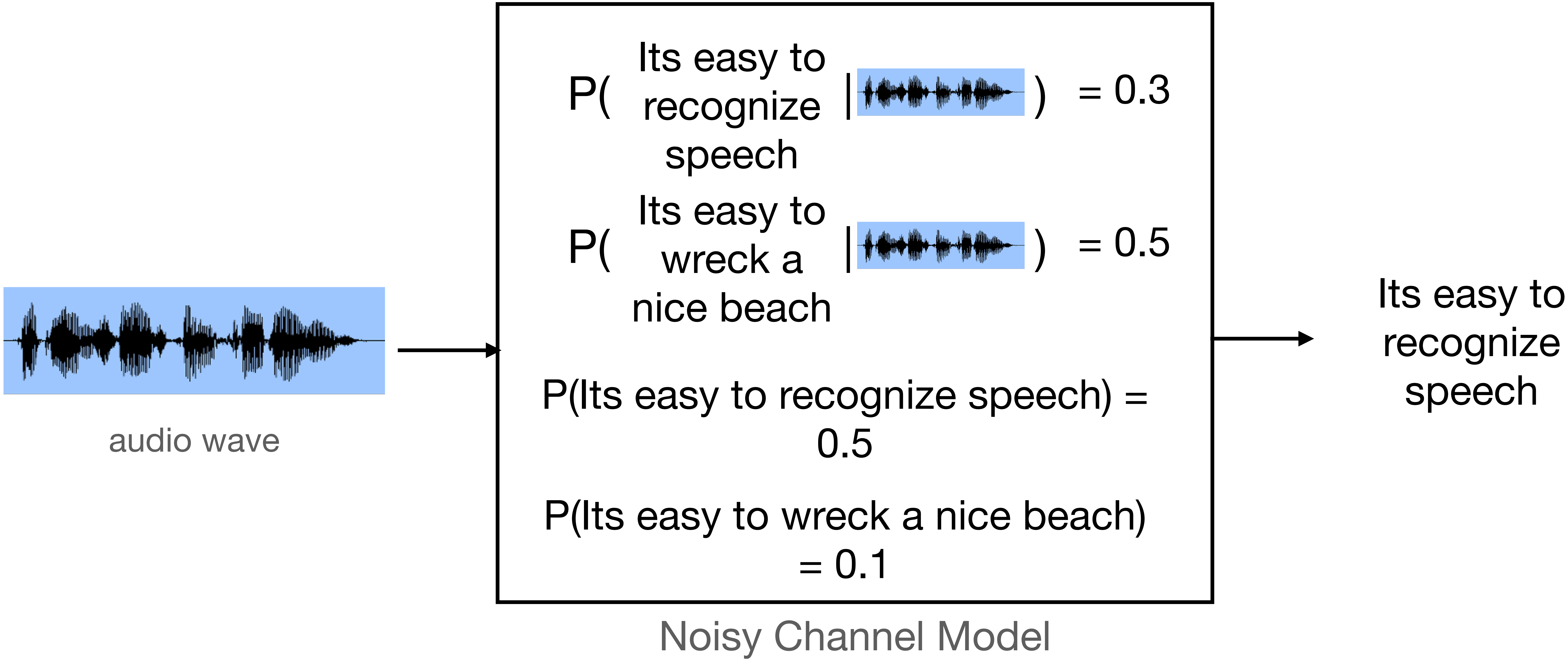
Application: Noisy Channel Speech Recognition Model



Noisy Channel Model

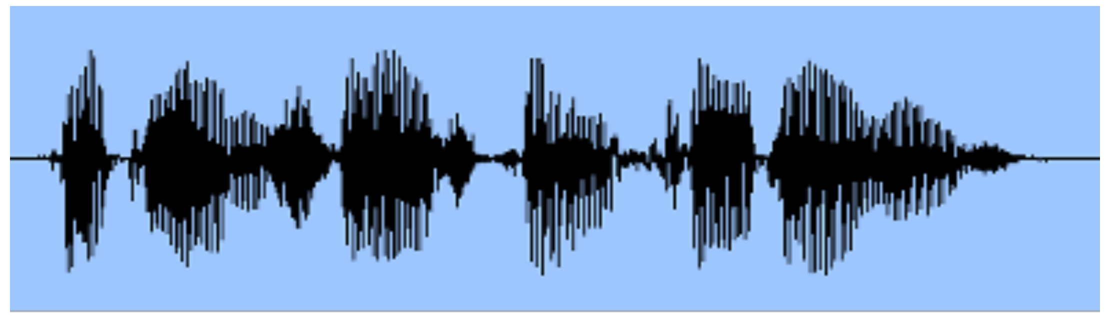
Language Modeling

Application: Noisy Channel Speech Recognition Model

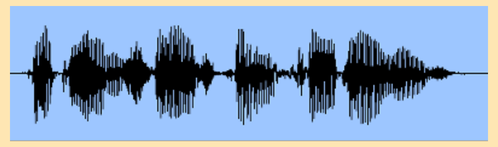


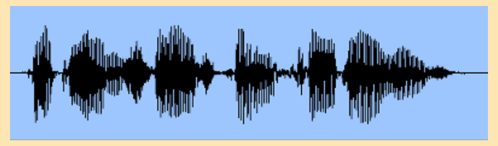
Language Modeling

Application: Noisy Channel Speech Recognition Model



audio wave

P(Its easy to recognize speech | ) = 0.3

P(Its easy to wreck a nice beach | ) = 0.5

Acoustic Model

P(Its easy to recognize speech) = 0.5

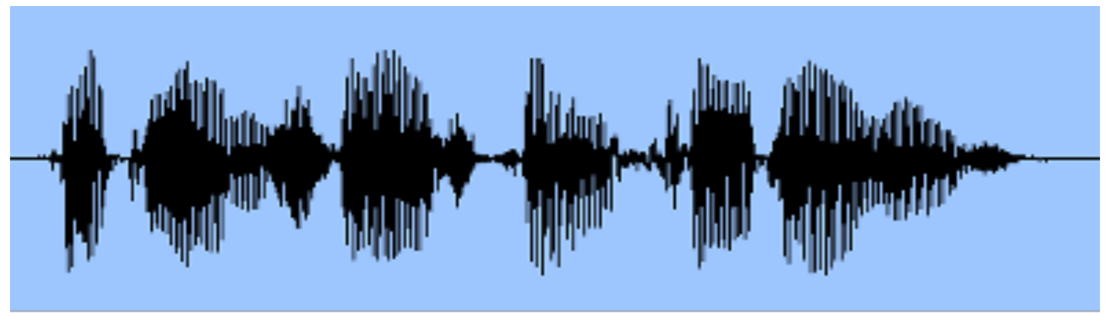
P(Its easy to wreck a nice beach) = 0.1

Noisy Channel Model

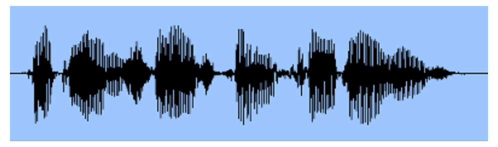
Its easy to recognize speech

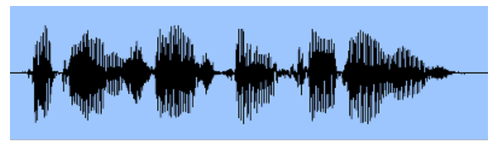
Language Modeling

Application: Noisy Channel Speech Recognition Model



audio wave

P(Its easy to recognize speech | ) = 0.3

P(Its easy to wreck a nice beach | ) = 0.5

P(Its easy to recognize speech) = 0.5

P(Its easy to wreck a nice beach) = 0.1

Noisy Channel Model

Language Model

Its easy to recognize speech

Topics

- What is language modeling? When do we use it?
- **Ngram language models**
- Smoothing
- Perplexity

Ngram Language Models

Directly computing corpus stats

- Simple idea: Just compute the probability P(w0, … wn) directly from a corpus!
- I.e.:

$$\frac{\text{\# occurances of w0, … wn}}{\text{\# sequences of length n}}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i’m looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i’m looking for a good place to eat breakfast
when is caffe venezia open during the day

P(tell me about chez panisse) =

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

P(tell me about chez panisse) =

1

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about chez panisse}) = \frac{1}{}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about chez panisse}) = \frac{1}{}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about chez panisse}) = \frac{1}{\qquad\qquad}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about chez panisse}) = \frac{1}{51}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about chez panisse}) = \frac{1}{51}$$

Problems?

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about caffe venezia}) = \frac{0}{51}$$

Ngram Language Models

Directly computing corpus stats

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(\text{tell me about caffe venezia}) = \frac{0}{51}$$

Ngram Language Models

Unigram Language Model

Chain Rule of Probability

$$P(w_0 \ldots w_n) = P(w_0) \times P(w_1 \mid w_0) \times P(w_2 \mid w_0, w_1) \times \ldots \times P(w_n \mid w_0 \ldots w_{n-1})$$

Ngram Language Models

Unigram Language Model

Chain Rule of Probability

$$P(w_0 \ldots w_n) = P(w_0) \times P(w_1 \mid w_0) \times P(w_2 \mid w_0, w_1) \times \ldots \times P(w_n \mid w_0 \ldots w_{n-1})$$

Not helpful (yet). Still requires observing w0...wn

Ngram Language Models

Unigram Language Model

$$P(w_0 \ldots w_n) = P(w_0) \times P(w_1 | w_0) \times P(w_2 | w_0, w_1) \times \ldots \times P(w_n | w_0 \ldots w_{n-1})$$

$$P(w_i | w_0 \ldots w_{i-1}) \approx P(w_i)$$

Independence Assumption
(Just like Naive Bayes)

Ngram Language Models

Unigram Language Model

$$P(w_0 \dots w_n) = P(w_0) \times P(w_1 \mid w_0) \times P(w_2 \mid w_0, w_1) \times \dots \times P(w_n \mid w_0 \dots w_{n-1})$$

$$P(w_i \mid w_0 \dots w_{i-1}) \approx P(w_i)$$

$$P(w_0 \dots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \dots \times P(w_n)$$

Unigram Model

Ngram Language Models

Unigram Language Model

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(w_0 \ldots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \ldots \times P(w_n)$$

P(tell me about caffe venezia) = P(tell) x P(me) x P(about) x P(caffe) x P(venezia)

Ngram Language Models

Unigram Language Model

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(w_0 \ldots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \ldots \times P(w_n)$$

P(tell me about caffe venezia) = 2/56 x 3/56 x 3/56 x 1/56 x 1/56

Ngram Language Models

Unigram Language Model

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(w_0 \ldots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \ldots \times P(w_n)$$

P(tell me about caffe venezia) = 2/56 x 3/56 x 3/56 x 1/56 x 1/56

= 3.26×10^{-8}

Ngram Language Models

Unigram Language Model

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

$$P(w_0 \ldots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \ldots \times P(w_n)$$

P(tell me about caffe venezia)

=log(2/56) + log(3/56)+ log(3/56)+ log(1/56)+ log(1/56)

= -13.9

Ngram Language Models

Unigram Language Model

can you tell me about any good cantonese restaurants close by
mid priced thai food is what i'm looking for
tell me about chez panisse
can you give me a listing of the kinds of food that are available
i'm looking for a good place to eat breakfast
when is caffe venezia open during the day

P(tell me about caffe venezia)

P(caffe about tell venezia me)

Which is more probable?

Ngram Language Models

Bigram Language Model

$$P(w_0 \ldots w_n) = P(w_0) \times P(w_1 | w_0) \times P(w_2 | w_0, w_1) \times \ldots \times P(w_n | w_0 \ldots w_{n-1})$$

$$P(w_i | w_0 \ldots w_{i-1}) \approx P(w_i | w_{i-1})$$

Markov Assumption!

Ngram Language Models

Bigram Language Model

$$P(w_0 \ldots w_n) = P(w_0) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_0, w_1) \times \ldots \times P(w_n \,|\, w_0 \ldots w_{n-1})$$

$$P(w_i \,|\, w_0 \ldots w_{i-1}) \approx P(w_i \,|\, w_{i-1})$$

$$P(w_0 \ldots w_n) \approx P(w_0 \,|\, \text{<s>}) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_1) \times \ldots \times P(w_n \,|\, w_{n-1})$$

Bigram Language Model

Ngram Language Models

Unigram Language Model

<s> can you tell me about any good cantonese restaurants close by </s>
<s> mid priced thai food is what i'm looking for </s>
<s> tell me about chez panisse </s>
<s> can you give me a listing of the kinds of food that are available </s>
<s> i'm looking for a good place to eat breakfast </s>
<s> when is caffe venezia open during the day </s>

$$P(w_0 \dots w_n) \approx P(w_0 \mid \text{<s>}) \times P(w_1 \mid w_0) \times P(w_2 \mid w_1) \times \dots \times P(w_n \mid w_{n-1})$$

P(tell me about caffe venezia) = P(tell|<s>) x P(me|tell) x P(about|me) x P(caffe|about) x P(venezia|caffe) x P(</s>|venezia)

Ngram Language Models

Ngram Language Models

Unigram Language Model

$$P(w_0 \ldots w_n) \approx P(w_0) \times P(w_1) \times P(w_2) \times \ldots \times P(w_n)$$

Bigram Language Model

$$P(w_0 \ldots w_n) \approx P(w_0 \,|\, \text{<s>}) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_1) \times \ldots \times P(w_n \,|\, w_{n-1})$$

n-gram Language Model

$$P(w_0 \ldots w_n) \approx \prod_{i=0}^{n} P(w_i \,|\, w_{i-(n-1)} \ldots w_{i-1})$$

Ngram Language Models

Ngram Language Models

To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have

Unigram

Ngram Language Models

Ngram Language Models

To him swallowe
both. Which. Of
are ay device a

Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow

Unigram

Bigram

Ngram Language Models

Ngram Language Models

Trigram

To him swallowe
both. Which. Of
are ay device a

Why dost st
forsooth; h
the King He

Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, 'tis done

Unigram

Bigram

Ngram Language Models

Ngram Language Models

Unigram

To him swallowe
both. Which. Of
are ay device a

Bigram

Why dost st
forsooth; h
the King He

Trigram

Fly, and w
price. Th
parting.

4-gram

King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv'd in;

Topics

- What is language modeling? When do we use it?
- Ngram language models
- **Smoothing**
- Perplexity

Ngram Language Models

<s> can you tell me about any good cantonese restaurants close by </s>
<s> mid priced thai food is what i'm looking for </s>
<s> tell me about chez panisse </s>
<s> can you give me a listing of the kinds of food that are available </s>
<s> i'm looking for a good place to eat breakfast </s>
<s> when is caffe venezia open during the day </s>

$$P(w_0 \ldots w_n) \approx P(w_0 \,|\, \text{<s>}) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_1) \times \ldots \times P(w_n \,|\, w_{n-1})$$

P(tell me about caffe venezia) = P(tell|<s>) x P(me|tell) x P(about|me) x
P(caffe|about) x P(venezia|caffe) x P(</s>|venezia)

Ngram Language Models

<s> can you tell me about any good cantonese restaurants close by </s>
<s> mid priced thai food is what i'm looking for </s>
<s> tell me about chez panisse </s>
<s> can you give me a listing of the kinds of food that are available </s>
<s> i'm looking for a good place to eat breakfast </s>
<s> when is caffe venezia open during the day </s>

$$P(w_0 \ldots w_n) \approx P(w_0 \,|\, \text{<s>}) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_1) \times \ldots \times P(w_n \,|\, w_{n-1})$$

P(tell me about caffe venezia) = P(tell|<s>) x P(me|tell) x P(about|me) x
P(caffe|about) x P(venezia|caffe) x P(</s>|venezia)

Ngram Language Models

<s> can you tell me about any good cantonese restaurants close by </s>
<s> mid priced thai food is what i'm looking for </s>
<s> tell me about chez panisse </s>
<s> can you give me a listing of the kinds of food that are available </s>
<s> i'm looking for a good place to eat breakfast </s>
<s> when is caffe venezia open during the day </s>

$$P(w_0 \ldots w_n) \approx P(w_0 \,|\, \text{<s>}) \times P(w_1 \,|\, w_0) \times P(w_2 \,|\, w_1) \times \ldots \times P(w_n \,|\, w_{n-1})$$

P(tell me about caffe venezia) = P(tell|<s>) x P(me|tell) x P(about|me) x
P(caffe|about) x P(venezia|caffe) x P(</s>|venezia)

Zero counts!

Generalization in LMs

Smoothing Strategies

- Laplace Smoothing (i.e., “Add-One” smoothing)
- Backoff
- Kneser-Ney Smoothing

Generalization in LMs

Smoothing Strategies

- **Laplace Smoothing (i.e., “Add-One” smoothing)**
- Backoff
- Kneser-Ney Smoothing

Smoothing

Laplace ("Add One")

Bigram Probabilities

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

Smoothing

Laplace ("Add One")

P(want|i)

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

Smoothing

Laplace ("Add One")

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

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

Smoothing

Laplace ("Add One")

Simple Idea: Just add 1 to everything!

	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	1	2	1	1
spend	2	1	2	1	1	1	1	1

Smoothing
Laplace ("Add One")

Need to renormalize to keep it a probability distribution

	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	1	2	1	1
spend	2	1	2	1	1	1	1	1

Smoothing

Laplace ("Add One")

$$P(w_n \mid w_{n-1}) = \frac{\#(w_{n-1}w_n) + 1}{\sum_w (\#(w_{n-1}w) + 1)}$$

	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	1	2	1	1
spend	2	1	2	1	1	1	1	1

Smoothing

Laplace ("Add One")

$$P(w_n \mid w_{n-1}) = \frac{\#(w_{n-1}w_n) + 1}{\#w_{n-1} + V}$$

	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	1	2	1	1
spend	2	1	2	1	1	1	1	1

Smoothing

Laplace ("Add One")

$$P(w_n \mid w_{n-1}) = \frac{\#(w_{n-1}w_n) + 1}{\#w_{n-1} + V}$$

	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	1	2	1	1
spend	2	1	2	1	1	1	1	1

Often interpreted as "discounting".

I.e., we borrow probability mass from high-count words in order to make room for unseen words.

Generalization in LMs

Smoothing Strategies

- Laplace Smoothing (i.e., “Add-One” smoothing)
- **Backoff**
- Kneser-Ney Smoothing

Generalization in LMs

Smoothing Strategies

- Laplace Smoothing (i.e., “Add-One” smoothing)
- **Backoff/Interpolation**
- Kneser-Ney Smoothing

Smoothing

Backoff

- Intuition: We can estimate the probability of a longer sequence from the probabilities of its subsequences
- If an ngram of length n is not observed, use the corresponding length n-1 ngram instead
- P (“tell me about caffe venezia”) ≅ P (“me about caffe venezia”)

≅ P (“about caffe venezia”) ≅ P (“caffe venezia”) ≅ P (“venezia”)

Smoothing

Interpolation

- All counts are estimated using a weighted combination of smaller ngrams
- P (“tell me about caffe venezia”) = λ_1P (“tell me about caffe venezia”) x λ_2P (“me about caffe venezia”) x λ_3P (“about caffe venezia”) x λ_4P (“caffe venezia”) x λ_5P (“venezia”)
- Requires some renormalization (like in Laplace Smoothing)

Generalization in LMs

Smoothing Strategies

- Laplace Smoothing (i.e., “Add-One” smoothing)
- Backoff
- **Kneser-Ney Smoothing**

Smoothing
Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

Smoothing
Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

$$P_{\text{AbsoluteDiscounting}}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}\,v)} + \lambda(w_{i-1})P(w_i)$$

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

$$P_{\text{AbsoluteDiscounting}}(w_i|w_{i-1}) = \boxed{\frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}\,v)}} + \lambda(w_{i-1})P(w_i)$$

Similar to Laplace, except we discount factor doesnt necessarily correspond to adding 1...

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm t
  1. Absolute discounting (estimated fi
  2. Replace ngram probabilities with c

Bigram count in training set	Bigram count in heldout set
0	0.0000270
1	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21
9	8.26

$$P_{\text{AbsoluteDiscounting}}(w_i|w_{i-1}) = \boxed{\frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}\,v)}} + \lambda(w_{i-1})P(w_i)$$

Instead, we estimate it from data!

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm t
  1. Absolute discounting (estimated fi
  2. Replace ngram probabilities with c

Bigram count in training set	Bigram count in heldout set
0	0.0000270
1	0.448
2	1.25
3	2.24
4	3.23
5	4.21
6	5.23
7	6.21
8	7.21
9	8.26

$$P_{\text{AbsoluteDiscounting}}(w_i|w_{i-1}) = \boxed{\frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}\,v)}} + \lambda(w_{i-1})P(w_i)$$

Can be fixed (e.g., 0.75) or a function of n

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

$$P_{\text{AbsoluteDiscounting}}(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i) - d}{\sum_v C(w_{i-1}\,v)} + \lambda(w_{i-1})P(w_i)$$

Interpolated with observed unigram probability

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

Consider: I can't see without my reading ____ .

In corpus:

- P(reading glasses) = P(reading Kong) = 0
- P(Kong) > P(glasses)!
- What to do?

Smoothing
Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

unique contexts for “glasses” > # unique contexts for “Kong”

so we assume:

P(glasses|as-yet-unseen-ctx) > P(Kong|as-yet-unseen-ctx)

Smoothing

Kneser-Ney

- State-of-the-art smoothing algorithm that combines several ideas:
  1. Absolute discounting (estimated from data)
  2. Replace ngram probabilities with *continuation* probabilities

$$P_{\mathrm{KN}}(w_i|w_{i-1}) = \frac{\max(C(w_{i-1}w_i) - d, 0)}{C(w_{i-1})} + \lambda(w_{i-1}) P_{\mathrm{CONTINUATION}}(w_i)$$

Topics

- What is language modeling? When do we use it?
- Ngram language models
- Smoothing
- **Perplexity**

Perplexity

- How do we decide if a language model is “good”?
- A good language model should assign high probability to sentences that actually appear
- Instead of using probability directly, we use a metric called “perplexity”
    - Inverse probability of test set, normalized by # of words

Perplexity

$$ppl(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_1 \ldots w_n)}}$$

Perplexity

$$ppl(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_1 \ldots w_n)}} = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i \mid w_{i-1})}}$$

(for bigram model)

Perplexity

Intuition

$$ppl(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_1 \ldots w_n)}} = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i \mid w_{i-1})}}$$

language with digits 0-9, all equally probable

W = 1 3 5 6 7 2 3 5 4

Perplexity

Intuition

$$ppl(W) = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_1 \ldots w_n)}} = \sqrt[n]{\prod_{i=1}^{n} \frac{1}{P(w_i \mid w_{i-1})}}$$

language with
digits 0-9, all
equally probable

W = 1 3 5 6 7 2 3 5 4

$$ppl(W) = (\frac{1}{10}^{n})^{-\frac{1}{n}} = \frac{1}{10}^{-1} = 10$$

Perplexity

Intuition

- “Weighted average branching factor”, i.e., how many next words can follow any given word?
- In PPL, **lower is better!** A model with lower PPL is less “surprised” by new data
- I.e., a model with lower PPL has more certainty about true sequences. It considers branching factors to be lower, because it has a good sense of what should come next

Perplexity
Intuition

- In natural language, distributions are highly non-uniform, so branching factors are (relatively) low
- PPL will never be zero! Natural language has inherent uncertainty
- PPL is <u>*not*</u> comparable across different datasets! Some datasets/languages/corpora are “easier”/lower uncertainty than others

Perplexity

Intuition

- Higher-order n-grams lead to lower ppl in general, but:
  - More likely to overfit to training data
  - Require more memory
  - Result in many more zero-counts