Language Models and Interfaces 2015

Lecture 2

Dr Ivan Titov

ILLC, Universiteit van Amsterdam

Recap: Lecture 1

- Language models in the broad sense: models of language

Recap: Lecture 1

- Language models in the broad sense: models of language
- Language models in the narrower sense: probabilistic models

Recap: Lecture 1

- Language models in the broad sense: models of language
- Language models in the narrower sense: probabilistic models
- Language models in the narrowest sense: models that define probability distributions over (parts of) sentences

Recap: Lecture 1

- Claude Shannon, Information Theory (1948)
- Source $\rightarrow$ Encoder $\rightarrow$ Channel $\rightarrow$ Decoder $\rightarrow$ Destination

Recap: Lecture 1

- Claude Shannon, Information Theory (1948)
- Source $\rightarrow$ Encoder $\rightarrow$ Channel $\rightarrow$ Decoder $\rightarrow$ Destination
- Entropy as measure of amount of information:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

Recap: Lecture 1

- Claude Shannon, Information Theory (1948)
- Source → Encoder → Channel → Decoder → Destination
- Entropy as measure of amount of information:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Ngram models as language models in the narrowest sense (Markov-order n-1 = ngram model)

Recap: Lecture 1

- Claude Shannon, Information Theory (1948)
- Source → Encoder → Channel → Decoder → Destination
- Entropy as measure of amount of information:

$$H(X) = -\sum_{x \in X} p(x) \log_2 p(x)$$

- Ngram models as language models in the narrowest sense (Markov-order n-1 = ngram model)
- Noisy channel metaphor: Source Model $\times$ Distortion Model $\approx$ Language Model $\times$ Task model.

Today an example: Spelling Correction

- Task is easily defined, but solving it at human level performance is almost AI complete...
- Illustration of how we design probabilistic models in NLP using noisy channel metaphor

Today an example: Spelling Correction

- Task is easily defined, but solving it at human level performance is almost AI complete...
- Illustration of how we design probabilistic models in NLP using noisy channel metaphor

Two models

(1) Word-level model (per word “Bayesian” model)

- Model without knowledge of the language as a whole

Today an example: Spelling Correction

- Task is easily defined, but solving it at human level performance is almost AI complete...
- Illustration of how we design probabilistic models in NLP using noisy channel metaphor

Two models

(1) Word-level model (per word “Bayesian” model)

- Model without knowledge of the language as a whole

(2) Context-dependent Model

- Model sees sentences as ordered sequences of words
- Syntactic structure of sentences still ignored (Next week: POS tags).

Errors in spelling, OCR and Hand Writing Recognition

Similar models used for other tasks:

Spelling: When typing text (*hi* instead of *he*)

Errors in spelling, OCR and Hand Writing Recognition

Similar models used for other tasks:

Spelling: When typing text (*hi* instead of *he*)

OCR (Optical Character Recognition): in recognition of letters (e.g. O rather than D) in type-written text

- different fonts, old texts
- important for languages with scripts other than the Latin alphabet.
- online recognition of signs / texts (Google Glass)

Errors in spelling, OCR and Hand Writing Recognition

Similar models used for other tasks:

Spelling: When typing text (*hi* instead of *he*)

OCR (Optical Character Recognition): in recognition of letters (e.g. O rather than D) in type-written text

- different fonts, old texts
- important for languages with scripts other than the Latin alphabet.
- online recognition of signs / texts (Google Glass)

Hand-Writing: during recognition of hand-written text

Errors in spelling, OCR and Hand Writing Recognition

Similar models used for other tasks:

Spelling: When typing text (*hi* instead of *he*)

OCR (Optical Character Recognition): in recognition of letters (e.g. O rather than D) in type-written text

- different fonts, old texts
- important for languages with scripts other than the Latin alphabet.
- online recognition of signs / texts (Google Glass)

Hand-Writing: during recognition of hand-written text

How can we correct errors?

We concentrate on spelling.

Sometimes it is hard for humans

Posted by Victor Mair on Language Log

An easier example

The Comunists disdain to conceal their views and aims. They openly declare that their ends can be attaimed only by the forcible overthrow of all existing social conditionns. Let the ruling classes trebmle at a communist revolution. The proletarians have nothing to lose but their chains. They have a world to win.

An easier example

The Comunists disdain to conceal their views and aims. They openly declare that their ends can be attaimed only by the forcible overthrow of all existing social conditionns. Let the ruling classes trebmle at a communist revolution. The proletarians have nothing to lose but their chains. They have a world to win.

The Comunists disdain to conceal their views and aims. They openly declare that their ends can be attaimed only by the forcible overthrow of all existing social conditionns. Let the ruling classes trebmle at a communist revolution. The proletarians have nothing to lose but their chains. They have a world to win.

An easier example

The Comunists disdain to conceal their views and aims. They openly declare that their ends can be attaimed only by the forcible overthrow of all existing social conditionns. Let the ruling classes trebmle at a communist revolution. The proletarians have nothing to lose but their chains. They have a world to win.

The Comunists disdain to conceal their views and aims. They openly declare that their ends can be attaimed only by the forcible overthrow of all existing social conditionns. Let the ruling classes trebmle at a communist revolution. The proletarians have nothing to lose but their chains. They have a world to win.

- How can we classify these errors?

Example from Philip Koehn's slides on Spelling correction using EM

Types of errors

4 types of errors in this text:

Types of errors

4 types of errors in this text:

Deletion : dropping a letter
Communist → Comunist

Types of errors

4 types of errors in this text:

Deletion : dropping a letter
Communist → Comunist

Insertion : adding a letter
conditions → conditiomns

Types of errors

4 types of errors in this text:

Deletion : dropping a letter
Communist → Comunist

Insertion : adding a letter
conditions → conditiomns

Substitution : replacing a letter
attained → attaimed

Types of errors

4 types of errors in this text:

Deletion : dropping a letter
Communist → Comunist

Insertion : adding a letter
conditions → conditiomns

Substitution : replacing a letter
attained → attaimed

Transposition : swapping two letters
tremble → trebmle

Example from Philip Koehn's slides on Spelling correction using EM

“Single-error correction”: Possible Fixes

It appears that 80% of the wrong words can be seen as being operative by one of the above four transformations.

“Single-error correction”: Possible Fixes

It appears that 80% of the wrong words can be seen as being operative by one of the above four transformations.

- Most systems are based on *one* error per word.
- Even with this assumption, there are a large number of corrections that must be considered per word.

Large number of possible corrections

There are many possible corrections, even if we assume only one error per word.

Example: acress

Any idea what it can be?

Large number of possible corrections

There are many possible corrections, even if we assume only one error per word.

Example: acress

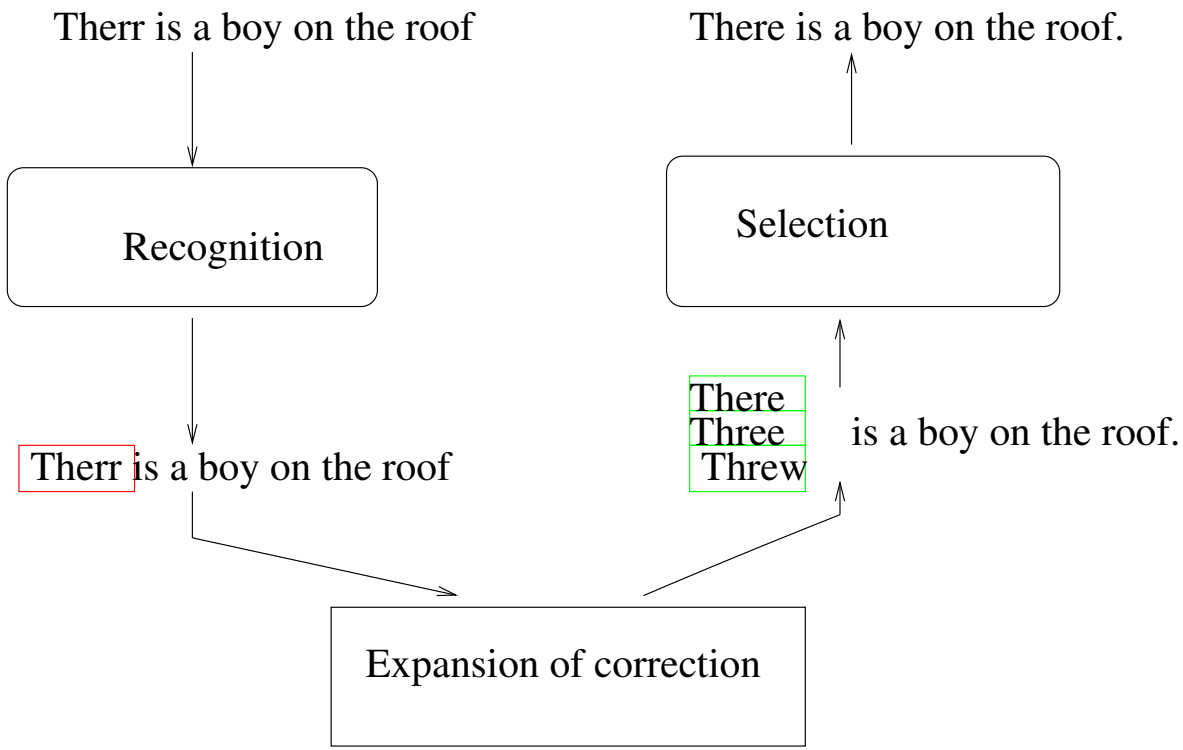
Any idea what it can be?

Error	Correction	Transformation	Type
acress	actress	t→	deletion
acress	cress	→a	insertion
acress	caress	ca→ac	transposition
acress	across	o→e	substitution
acress	acres	→s	insertion

Problem: Disambiguation

Example from Philip Koehn's slides on Spelling correction using EM

General Architecture



Non-word detection (recognition)

How?

Non-word detection (recognition)

How?

Using a dictionary for non-word detection

A `deliverable` `may be composed of smaller deliverables`
`John is famous for` `sloppiness` `in his writing`

Dictionary: too weak even for recognition.

Non-word detection (recognition)

How?

Using a dictionary for non-word detection

A `deliverable` `may be composed of smaller deliverables`
`John is famous for` `sloppiness` `in his writing`

Dictionary: too weak even for recognition.

How to fix this?

Non-word detection (recognition)

How?

Using a dictionary for non-word detection

A `deliverable` may be composed of smaller deliverables
John is famous for `sloppiness` in his writing

Dictionary: too weak even for recognition.

How to fix this?

Morphology: Using the structure of words

- Example: Hard-ness, shallow-ness, sloppi(y)-ness, the-ness
- Morphology is not as simple

Recognition of errors: Dictionary + Morphology

Non-word detection (recognition)

Is dictionary + morphology sufficient?

Non-word detection (recognition)

Is dictionary + morphology sufficient?

Using a dictionary + morphology for non-word detection

There is a boy on the roof

Therr is a boy on the roof

Three is a boy on the roof

Threw is a boy on the roof

Non-word detection (recognition)

Is dictionary + morphology sufficient?

Using a dictionary + morphology for non-word detection

```
There is a boy on the roof
Therr is a boy on the roof
Three is a boy on the roof
Threw is a boy on the roof
```

Problem: correct words in non-appropriate context.

- Dictionary and morphology do not address this problem
- We should use the context.

Spelling and spelling correction: Subtasks

Three levels of models (Kukich,1992)

- **Recognition:** Non-word error detection(*graff* instead *giraffe*)

Spelling and spelling correction: Subtasks

Three levels of models (Kukich,1992)

- **Recognition:** Non-word error detection(*graff* instead *giraffe*)
- **Correction:** Isolated word-error correction: corrects the word in isolation without context.

Spelling and spelling correction: Subtasks

Three levels of models (Kukich,1992)

- **Recognition:** Non-word error detection(*graff* instead *giraffe*)
- **Correction:** Isolated word-error correction: corrects the word in isolation without context.
- **Recognition and Corrections:** Context-dependent error detection and correction: use the context that the word occurs in (parts of the sentence) to determine correction (*there* instead of *three*).

Models for selection / disambiguation

We need a model that *selects* the best possible correction:

Models for selection / disambiguation

We need a model that *selects* the best possible correction:

Ranking requires a function

$$P : \mathit{errors} \times \mathit{corrections} \rightarrow [0, 1]$$

Models for selection / disambiguation

We need a model that *selects* the best possible correction:

Ranking requires a function

$$P : \mathit{errors} \times \mathit{corrections} \rightarrow [0, 1]$$

Two types of models:

Models for selection / disambiguation

We need a model that *selects* the best possible correction:

Ranking requires a function

$$P : errors \times corrections \rightarrow [0, 1]$$

Two types of models:

Single word: looks just the wrong word : Word-level models: Single Word “Bayesian model” / Noisy Channel model

Models for selection / disambiguation

We need a model that *selects* the best possible correction:

Ranking requires a function

$$P : errors \times corrections \rightarrow [0, 1]$$

Two types of models:

Single word: looks just the wrong word : Word-level models: Single Word "Bayesian model" / Noisy Channel model

Context-dependent: look at context, e.g. entire sentence "Next-word" / Language models.

Noisy Channel Model/Bayesian Model

Metaphor: Metaphor of Communication: Communication and Information Theory (Shannon, 1948)

Communication: Message goes through a “noisy channel”: the observation contains errors and should be corrected.

Task: to restore the original message given the observation

What is the noisy channel in this case?

Probabilistic notion of selection

O : Observation
C : Correction

What we want: Correct message (i.e. word) given the observation
i.e. $P(C \mid O)$

Probabilistic notion of selection

O : Observation
C : Correction

What we want: Correct message (i.e. word) given the observation
i.e. $P(C \mid O)$

Given a set of corrections Γ

$$C^* = argmax_{C \in \Gamma} P(C \mid O)$$

Probabilistic notion of selection

O : Observation
C : Correction

What we want: Correct message (i.e. word) given the observation
i.e. $P(C \mid O)$

Given a set of corrections Γ

$$C^* = argmax_{C \in \Gamma} P(C \mid O)$$

What is $P(C \mid O)$?
How do we estimate it from data?

Word-level model

A collection of Γ corrections for an observation 0

`Example:` $O = \textit{Therr}$ `and` $\Gamma = \{\textit{There}, \textit{Three}, \textit{Threw}\}$

Word-level model

A collection of Γ corrections for an observation 0

Example: $O = \textit{Therr}$ and $\Gamma = \{\textit{There}, \textit{Three}, \textit{Threw}\}$

$C^* = \textit{argmax}_{C \in \Gamma} P(C \mid O)$

$\textit{argmax}_c \{P(C = \textit{Three} \mid \textit{Therr}), P(C = \textit{Threw} \mid \textit{Therr}), P(C = \textit{There} \mid \textit{Therr})\}$

Problems?

Word-level model

A collection of Γ corrections for an observation 0

`Example:` $O = \textit{Therr}$ `and` $\Gamma = \{\textit{There}, \textit{Three}, \textit{Threw}\}$

$C^* = \textit{argmax}_{C \in \Gamma} P(C \mid O)$

$\textit{argmax}_c \{P(C = \textit{Three} \mid \textit{Therr}), P(C = \textit{Threw} \mid \textit{Therr}), P(C = \textit{There} \mid \textit{Therr})\}$

Problems?

- We will need to count over errors

$$P(\textit{Three} \mid \textit{Therr}) = \frac{\textit{count}(\textit{Three}, \textit{therr})}{\textit{count}(\textit{Therr})}$$

Word-level model

A collection of Γ corrections for an observation 0

`Example:` $O = \textit{Therr}$ `and` $\Gamma = \{\textit{There}, \textit{Three}, \textit{Threw}\}$

$C^* = \textit{argmax}_{C \in \Gamma} P(C \mid O)$

$\textit{argmax}_c \{P(C = \textit{Three} \mid \textit{Therr}), P(C = \textit{Threw} \mid \textit{Therr}), P(C = \textit{There} \mid \textit{Therr})\}$

Problems?

- We will need to count over errors

$$P(\textit{Three} \mid \textit{Therr}) = \frac{\textit{count}(\textit{Three}, \textit{therr})}{\textit{count}(\textit{Therr})}$$

- There can be infinitely many ways of spelling a word incorrectly.
- Direct model: difficult to find statistical data.

Bayesian Inversion

Bayes Rule: Relates conditional probabilities

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

Bayesian Inversion

Bayes Rule: Relates conditional probabilities

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

$$C^* = argmax_{C \in \Gamma} P(C \mid O) \overset{Bayes}{=} argmax_{C \in \Gamma} \frac{P(O \mid C)\, P(C)}{P(O)}$$

$$\overset{max}{=} argmax_{C \in \Gamma} P(O \mid C)\, P(C)$$

Prior: $P(C)$ is the *prior* probability of the correct word.

Likelihood: $P(O \mid C)$ the *misspelling model* (our task model).
Given the correct word C, how likely is it that it is mis-spelt as O.

Estimating the prior and the misspelling model

Prior: $P(C)$ can be estimated from a large text (body) by relative frequency.

$$P(word_1) = \frac{count(word_1)}{N}$$

Estimating the prior and the misspelling model

Prior: $P(C)$ can be estimated from a large text (body) by relative frequency.

$$P(word_1) = \frac{count(word_1)}{N}$$

Likelihood: Given the correct word C, how likely is it that it is mis-spelt as O.

$P(O \mid C)$ is a little harder to estimate, but is estimated using the approximation that errors are *insertion/deletion/substitution/transposition* of "single-letters"

Likelihood of error

- Create a training corpus: find mis-spelt text, correct it and keep track of corrections.
- Estimate $P(O \mid C)$

Likelihood of error

- Create a training corpus: find mis-spelt text, correct it and keep track of corrections.
- Estimate $P(O \mid C)$

Given the correct word w = “problem”, how likely is it that it will be spelt as t=“oroblem”?

Likelihood of error

- Create a training corpus: find mis-spelt text, correct it and keep track of corrections.
- Estimate $P(O \mid C)$

Given the correct word w = “problem”, how likely is it that it will be spelt as t=“oroblem”?

“p” occurs 14568 times. It is mis-spelt as “o” 17 times.

Likelihood of error

- Create a training corpus: find mis-spelt text, correct it and keep track of corrections.
- Estimate $P(O \mid C)$

Given the correct word w = "problem", how likely is it that it will be spelt as t="oroblem"?

"p" occurs 14568 times. It is mis-spelt as "o" 17 times.

$$\frac{sub(t_n, c_n)}{count(c_n)} = \frac{sub(o, p)}{count(p)} = \frac{17}{14568}$$

Likelihood of error

- Create a training corpus: find mis-spelt text, correct it and keep track of corrections.
- Estimate $P(O \mid C)$

Given the correct word w = "problem", how likely is it that it will be spelt as t="oroblem"?

"p" occurs 14568 times. It is mis-spelt as "o" 17 times.

$$\frac{sub(t_n, c_n)}{count(c_n)} = \frac{sub(o, p)}{count(p)} = \frac{17}{14568}$$

$$\begin{aligned}
P(t|c) &= \frac{sub(t_p, c_p)}{count(c_p)} && \text{if substitution} \\
&= \frac{del(c_p)}{count(c_p)} && \text{if deletion} \\
&= \frac{ins(c_{p-1}, t_p)}{count(c_{p-1})} && \text{if insertion} \\
&= \frac{trans(c_p, c_{p+1})}{count(c_p, c_{p+1})} && \text{if transposition}
\end{aligned}$$

What if you do not have a lot of data?

- Hand-corrected data is expensive, you do not normally have a lot of it
- Example:“p” occurs 1456 times. It is mis-spelt as “o” 0 times.

What if you do not have a lot of data?

- Hand-corrected data is expensive, you do not normally have a lot of it
- Example:"p" occurs 1456 times. It is mis-spelt as "o" 0 times.
- We can use an even coarser model, for example:

$$\frac{sub(t_n, c_n)}{count(c_n)} \approx \frac{\sum_{c'} \sum_{t':dist(t',c')=dist(t_n,c_n)} sub(t', c')}{N}$$

- *dist* is, e.g., a 'distance' on the keyboard

Word-level models: Issues?

Do you expect this model to work?

Word-level models: Issues?

Do you expect this model to work? No, the model fails because it is context-free!!

Word-level models: Issues?

Do you expect this model to work? No, the model fails because it is context-free!!

Knowledge about the language and the world is crucial!

Word-level models: Issues?

Do you expect this model to work? No, the model fails because it is context-free!!

Knowledge about the language and the world is crucial!

Therr is a boy on the roof

Which sentence is the most plausible correction?

```
There is a boy on the roof
Three is a boy on the roof
Threw is a boy on the roof
```

Context-Dependent models (1)

- Knowledge about world and semantic knowledge is extremely difficult to model.
  `Three is a boy on the roof`

Context-Dependent models (1)

- Knowledge about world and semantic knowledge is extremely difficult to model.
  `Three is a boy on the roof`
- To make do with context of words : *a word is known by the company it keeps*
  Consider whole sentences/ parts of sentences.

Context-Dependent models (1)

- Knowledge about world and semantic knowledge is extremely difficult to model.
  `Three is a boy on the roof`
- To make do with context of words : *a word is known by the company it keeps*
  Consider whole sentences/ parts of sentences.

The Noisy-Channel metaphor at sentence level

Sentence level O and C

$$O = o_1, \ldots, o_n \qquad C = w_1, \ldots, w_n$$

Bayesian model: Sentence level

Observation: $O = o_1, \ldots, o_n = 0_1^n$
Correct sentence: $C = w_1, \ldots, w_n = w_1^n$

We are interested in $P(w_1^n \mid o_1^n)$

Bayesian model: Sentence level

Observation: $O = o_1, \ldots, o_n = o_1^n$
Correct sentence: $C = w_1, \ldots, w_n = w_1^n$

We are interested in $P(w_1^n \mid o_1^n)$

Bayesian model

Given a set of corrections Γ

$$
\begin{aligned}
C^* &= \mathrm{argmax}_{w_1^n \in \Gamma} P(w_1^n \mid o_1^n) \\
&\overset{Bayes}{=} \mathrm{argmax}_{w_1^n \in \Gamma} \frac{P(o_1^n \mid w_1^n) P(w_1^n)}{P(o_1^n)} \\
&\overset{max}{=} \mathrm{argmax}_{w_1^n \in \Gamma} P(o_1^n \mid w_1^n)\ P(w_1^n)
\end{aligned}
$$

Two Models: Task Model and Language Model

$$argmax_{w_1^n \in \Gamma} P(o_1^n \mid w_1^n) \;\; P(w_1^n)$$

Two Models: Task Model and Language Model

$$argmax_{w_1^n \in \Gamma} P(o_1^n \mid w_1^n) \;\; P(w_1^n)$$

$P(o_1^n \mid w_1^n)$ *Task* Model

- How likely is o_1^n as a result of typos in w_1^n?
- *What plays a role:*

Two Models: Task Model and Language Model

$$argmax_{w_1^n \in \Gamma} P(o_1^n \mid w_1^n) \; P(w_1^n)$$

$P(o_1^n \mid w_1^n)$ *Task* Model

- How likely is o_1^n as a result of typos in w_1^n?
- *What plays a role: knowledge of keyboard, knowledge of ins/del/sub/tran!*

$P(w_1^n)$ *Language* Model

- How likely is it that w_1^n is a sentence in the language?
- *Here, knowledge of the language (word order,syntax, semantics can be included..)*

Note how the two models are merged elegantly!

Independence Assumptions (Task model)

Task Model We make the assumption that misspelling in a word is independent of misspellings in other words (a reasonable assumption!)

How do we write it down?

$P(o_1^n \mid w_1^n) \approx$

Independence Assumptions (Task model)

Task Model We make the assumption that misspelling in a word is independent of misspellings in other words (a reasonable assumption!)

How do we write it down?

$P(o_1^n \mid w_1^n) \approx \prod_{i=1}^{n} P(o_i \mid w_i)$

Independence Assumptions (Task model)

Task Model We make the assumption that misspelling in a word is independent of misspellings in other words (a reasonable assumption!)

How do we write it down?

$P(o_1^n \mid w_1^n) \approx \prod_{i=1}^{n} P(o_i \mid w_i)$

This model is consistent with the previous *word-level* model.
Single-point transforms (ins / del / sub / tran).

What are language models

A language model (in the narrowest sense) is a probability distribution over a sequence of words (such as a sentence)!

What are language models

A language model (in the narrowest sense) is a probability distribution over a sequence of words (such as a sentence)!

Fomal Description

Given a finite vocabulary V of words:

Formal Language: a set Ω of sequences of words from V.

For all $x \in \Omega$, x is called a *sentence*

Language Model: A probability distribution over the formal language Ω,

$$P : \Omega \to [0, 1] \qquad \sum_{x \in \Omega} P(x) = 1$$

What do we want from a language model?

Sentence probability

Which sequences of words seem more likely?

I would like to eat.	*I would like eat to.*
I like would eat to.	*I to like would eat.*
I would like to eat.	*I would like to open.*

Sentence probability

Which sequences of words seem more likely?

I would like to eat.	*I would like eat to.*
I like would eat to.	*I to like would eat.*
I would like to eat.	*I would like to open.*

We should expect that “regular” sentences occur more often in text and speech than other weird word sequences.

Sentence probability

Which sequences of words seem more likely?

I would like to eat.	*I would like eat to.*
I like would eat to.	*I to like would eat.*
I would like to eat.	*I would like to open.*

We should expect that “regular” sentences occur more often in text and speech than other weird word sequences.

Can sentence probability capture sentence “regularity” (or “grammaticality”)?
(Recall examples from the previous lecture).

Word prediction by Language Models

Can word prediction be useful?

Word prediction by Language Models

Can word prediction be useful?

Example of spelling and other errors in text:

> 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.
> I need to notified the bank of this problem.
> He is trying to fine out what happened.

Relation between sentence probability and prediction

Suppose we have

- a vocabulary V (a finite set of words)
- a probability model over word sequences $P : V^+ \rightarrow [0, 1]$

How do we predict the next word w_n, given the preceding $w_1, \ldots, w_{n-1}$ words?

Relation between sentence probability and prediction

Suppose we have

- a vocabulary V (a finite set of words)
- a probability model over word sequences $P : V^+ \rightarrow [0, 1]$

How do we predict the next word w_n, given the preceding $w_1, \ldots, w_{n-1}$ words?

This is the *conditional probability* $P(w_n|w_1, \ldots, w_{n-1})$

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

In $P(w_n|w_1^{n-1})$, we call w_1^{n-1} the history or conditioning context.

Sentence probability models allow word prediction.

The Construction of Language Models

Probability of a sentence $w_1 ... w_n$ (Joint probability)

$$
\begin{aligned}
P(w_1, \ldots, w_n) &= P(w_n | w_1 \ldots w_{n-1})\ P(w_1 \ldots w_{n-1}) \\
&= P(w_n | w_1 \ldots w_{n-1})\ P(w_{n-1} | w_1 \ldots w_{n-2})\ P(w_1 \ldots w_{n-2}) \\
&= P(w_1)\ \prod_{i=2}^{n} P(w_i | w_1, \ldots, w_{i-1}) \text{ — from Chain Rule}
\end{aligned}
$$

The Construction of Language Models

Probability of a sentence $w_1 ... w_n$ (Joint probability)

$$
\begin{aligned}
P(w_1, \ldots, w_n) &= P(w_n | w_1 \ldots w_{n-1})\ P(w_1 \ldots w_{n-1}) \\
&= P(w_n | w_1 \ldots w_{n-1})\ P(w_{n-1} | w_1 \ldots w_{n-2})\ P(w_1 \ldots w_{n-2}) \\
&= P(w_1)\ \prod_{i=2}^{n} P(w_i | w_1, \ldots, w_{i-1}) \text{ — from Chain Rule}
\end{aligned}
$$

Example: *a look back at history*

$P(a\ look\ back\ at\ history) =$
$P(history \mid a\ look\ back\ at)\ P(at \mid a\ look\ back)\ P(back \mid a\ look)\ P(a)$

Language Models: Concepts

Probability Concepts

Joint Probability $P(A, B)$

Language Models: Concepts

Probability Concepts

Joint Probability $P(A, B)$

Conditional Probability $P(B|A)$ $P(A|B) = \frac{P(A\,B)}{P(B)}$

Language Models: Concepts

Probability Concepts

Joint Probability $P(A, B)$

Conditional Probability $P(B|A)$ $P(A|B) = \frac{P(A\,B)}{P(B)}$

Generalization $P(A_1, \ldots, A_n) = P(A_1)P(A_2 \ldots A_n|A_1)$
$= P(A_1)P(A_2|A_1)P(A_3 \ldots A_n|A_1 A_2)$
$= P(A_1)P(A_2|A_1) \ldots P(A_n|A_1, \ldots, A_{n-1})$

Language Models: Concepts

Probability Concepts

Joint Probability $P(A, B)$

Conditional Probability $P(B|A)$ $P(A|B) = \frac{P(A\,B)}{P(B)}$

Generalization $P(A_1, \ldots, A_n) = P(A_1)P(A_2 \ldots A_n|A_1)$
$= P(A_1)P(A_2|A_1)P(A_3 \ldots A_n|A_1 A_2)$
$= P(A_1)P(A_2|A_1) \ldots P(A_n|A_1, \ldots, A_{n-1})$

Language Models: Concepts

Probability Concepts

Joint Probability $P(A, B)$

Conditional Probability $P(B|A)$ $P(A|B) = \frac{P(A\,B)}{P(B)}$

Generalization $P(A_1, \ldots, A_n) = P(A_1)P(A_2 \ldots A_n|A_1)$
$= P(A_1)P(A_2|A_1)P(A_3 \ldots A_n|A_1 A_2)$
$= P(A_1)P(A_2|A_1) \ldots P(A_n|A_1, \ldots, A_{n-1})$

Probability of a sentence $w_1 \ldots w_n$

Joint Probability of the sequence of words (order important).

Expectations: Next-Word Prediction

Language Model A distribution over all word-sequences $w_1, \ldots, w_n$

$$\sum_{\langle w_1, \ldots, w_n \rangle} P(w_1, \ldots, w_n) = 1$$

Expectations: Next-Word Prediction

Language Model A distribution over all word-sequences $w_1, \ldots, w_n$

$$\sum_{\langle w_1, \ldots, w_n \rangle} P(w_1, \ldots, w_n) = 1$$

Derivation $P(w_1, \ldots, w_n) = P(w_1) \prod_{i=2}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$

Expectations: Next-Word Prediction

Language Model A distribution over all word-sequences $w_1, \ldots, w_n$

$$\sum_{\langle w_1, \ldots, w_n \rangle} P(w_1, \ldots, w_n) \;=\; 1$$

Derivation $P(w_1, \ldots, w_n) \;=\; P(w_1) \prod_{i=2}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$

Prediction A language model contains a "next-word prediction" model

$$\forall\, 1 \leq\; i \;\leq\; n : P(w_i \mid w_1, \ldots, w_{i-1})$$

Expectations: Next-Word Prediction

Language Model A distribution over all word-sequences $w_1, \ldots, w_n$

$$\sum_{\langle w_1, \ldots, w_n \rangle} P(w_1, \ldots, w_n) \;=\; 1$$

Derivation $P(w_1, \ldots, w_n) \;=\; P(w_1) \prod_{i=2}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$

Prediction A language model contains a "next-word prediction" model

$$\forall\, 1 \leq\; i \;\leq\; n : P(w_i \mid w_1, \ldots, w_{i-1})$$

Expectations: Next-Word Prediction

Language Model A distribution over all word-sequences $w_1, \ldots, w_n$

$$\sum_{\langle w_1, \ldots, w_n \rangle} P(w_1, \ldots, w_n) = 1$$

Derivation $P(w_1, \ldots, w_n) = P(w_1) \prod_{i=2}^{n} P(w_i \mid w_1, \ldots, w_{i-1})$

Prediction A language model contains a "next-word prediction" model

$$\forall\, 1 \leq i \leq n : P(w_i \mid w_1, \ldots, w_{i-1})$$

Where do we get $P(w_i \mid w_1, \ldots, w_{i-1})$ from? How?

Estimation

Estimating language models from corpora: Ngrams

Estimation from Corpora

We want a model of sentence probability $P(w_1, \ldots, w_n)$ for all word sequences $w_1, \ldots, w_n$ over the vocabulary V:

$$P(w_1, \ldots, w_n) = P(w_1) \prod_{i=2}^{n} P(w_i | w_1, \ldots, w_{i-1})$$

Tasks to do:

- Estimate $P(w_1)$
- Estimate probabilities $P(w_i | w_1, \ldots, w_{i-1})$ for all $w_1, \ldots, w_i$!

Estimation from Corpora II

Relative Frequency from a corpus

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{P(w_1^i)}{P(w_1^{i-1})} = \frac{Count(w_1, \ldots, w_{i-1}, w_i) \ / \ N}{Count(w_1, \ldots, w_{i-1}) \ / \ N'}$$

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{Count(w_1, \ldots, w_{i-1}, w_i) \ / \ N}{\sum_{w \in V} Count(w_1, \ldots, w_{i-1}, \ w) \ / \ N}$$

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{Count(w_1, \ldots, w_{i-1}, w_i)}{\sum_{w \in V} Count(w_1, \ldots, w_{i-1}, \ w)}$$

where N is number of all sequences of length i in corpus

Can we really do this?

Estimation from Corpora II

Relative Frequency from a corpus

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{P(w_1^i)}{P(w_1^{i-1})} = \frac{Count(w_1, \ldots, w_{i-1}, w_i) \ / \ N}{Count(w_1, \ldots, w_{i-1}) \ / \ N'}$$

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{Count(w_1, \ldots, w_{i-1}, w_i) \ / \ N}{\sum_{w \in V} Count(w_1, \ldots, w_{i-1}, \ w) \ / \ N}$$

$$P(w_i \mid w_1, \ldots, w_{i-1}) = \frac{Count(w_1, \ldots, w_{i-1}, w_i)}{\sum_{w \in V} Count(w_1, \ldots, w_{i-1}, \ w)}$$

where N is number of all sequences of length i in corpus

Can we really do this?

Suppose $|V| = 1000$, sentences are ≈ 10 words long:
1000^{10} possible sequences (probability values to estimate): no corpus is large enough!

What to do in order to estimate these probabilities?

- Bucketing histories: Markov models
- Smoothing techniques against sparse-data.

Markov Assumption and N-grams

The Markov assumptions:

Time invariance: independent and identical trials!

Markov Assumption and N-grams

The Markov assumptions:

Time invariance: independent and identical trials!

Limited history: There is a fixed finite k such that for all w_1^{i+1}:

$$P(w_{i+1}|w_1, \ldots, w_i) \approx P(w_{i+1}|w_{i-k}, \ldots, w_i)$$

Markov Assumption and N-grams

The Markov assumptions:

Time invariance: independent and identical trials!

Limited history: There is a fixed finite k such that for all w_1^{i+1}:

$$P(w_{i+1}|w_1, \ldots, w_i) \approx P(w_{i+1}|w_{i-k}, \ldots, w_i)$$

Markov Assumption and N-grams

The Markov assumptions:

Time invariance: independent and identical trials!

Limited history: There is a fixed finite k such that for all w_1^{i+1}:

$$P(w_{i+1}|w_1,\ldots,w_i) \approx P(w_{i+1}|w_{i-k},\ldots,w_i)$$

For $k \geq 0$

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

Model: A k^{th}-order Markov Model

N-gram: The statistics of a k-order Markov model is $k+1$-gram model!

$$P(w_i \mid w_{i-k},\ldots,w_{i-1}) = \frac{Count(w_{i-k},\ldots,w_{i-1},w_i)}{\sum_{w\in V} Count(w_{i-k},\ldots,w_{i-1},\ w)}$$

Markov model-order and N-grams

The order of a Markov model is defined by the length of its history:

0^{th}-order: $P(w_1, \ldots, w_n) \approx P(w_1) \prod_{i=1}^{n-1} P(w_{i+1})$

$|history| = 0$ $\approx P(w_1)\, P(w_2)\, P(w_3) \ldots P(w_n)$

1^{st}-order: $P(w_1, \ldots, w_n) \approx P(w_1) \prod_{i=1}^{n-1} P(w_{i+1}|w_i)$

$|history| = 1$

2^{nd}-order: $P(w_1, \ldots, w_n) \approx P(w_1)\, P(w_2|w_1) \prod_{i=2}^{n-1} P(w_{i+1}|w_i, w_{i-1})$

$|history| = 2$

$\vdots \quad \vdots \quad \vdots \quad \vdots$

k^{th}-order: $P(w_1, \ldots, w_n) \approx P(w_1) \prod_{i=1}^{n-1} P(w_{i+1}|w_{i-k}^{i})$

$|history| = k$

Estimation: Unigrams, Bigrams and Trigrams

$$P(w_{i+1}|w_{i-k}^{i}) = \frac{Count(w_{i-k}, \ldots, w_i,\ w_{i+1})}{\sum_{w \in V} Count(w_{i-k}, \ldots, w_i, w)}$$

A sequence of words of length $n > 0$ is called an n-gram: $w_{i-n+1}, \ldots, w_i$

An $(n-1)^{th}$-order Markov model demands *n-gram* statistics!

Estimation: Unigrams, Bigrams and Trigrams

$$P(w_{i+1}|w_{i-k}^{i}) = \frac{Count(w_{i-k}, \ldots, w_i,\ w_{i+1})}{\sum_{w \in V} Count(w_{i-k}, \ldots, w_i, w)}$$

A sequence of words of length $n > 0$ is called an n-gram: $w_{i-n+1}, \ldots, w_i$

An $(n-1)^{th}$-order Markov model demands *n-gram* statistics!

Example: $< s >$ The boy went home to eat . $< /s >$

0^{th}-order	$P(< s >)\ P(The)\ P(boy)\ P(went)\ P(home)$
(Unigram)	$P(to)\ P(eat)\ P(.)\ P(< /s >)$
1^{th}-order	$P(< s >)\ P(The \mid < s >)\ P(boy \mid The)\ P(went \mid boy)$
(Bi-gram)	$P(home \mid went)\ P(to \mid home)\ P(eat \mid to)\ P(. \mid eat)\ P(< /s > \mid .)$

Estimation (from a corpus)

The statistics of an k-order Markov model is $(k+1)$-grams (Relative Frequency Estimate)

$$P(w_i \mid w_{i-k}, \ldots, w_{i-1}) = \frac{Count(w_{i-k}, \ldots, w_{i-1}, w_i)}{\sum_{w \in V} Count(w_{i-k}, \ldots, w_{i-1}, \ w)}$$

Addition of START and STOP

$$P(w_1, \ldots, w_n) = \prod_{i=1}^{i=n+1} P(w_i | w_{i-n+1}, \ldots, w_{i-1})$$

where $w_j =< s >$ (START) for $j \leq 0$ and $w_{n+1} =< /s >$ (STOP)

$$\sum_{w \in V \cup \{STOP\}} Count(w_{i-k}, \ldots, w_{i-1}, \ w) = Count(w_{i-k}, \ldots, w_{i-1})$$

Where and how to get word statistics?

- Corpora (plural of corpus): large bodies of texts or speech utterances.
  Example: Brown corpus 1 million word collection of different texts
- Number of *tokens* vs. number of *types* in a corpus!
  Brown corpus: 1 million wordform tokens and 61805 wordform types

What are words?

Where and how to get word statistics?

- Corpora (plural of corpus): large bodies of texts or speech utterances.
  Example: Brown corpus 1 million word collection of different texts
- Number of *tokens* vs. number of *types* in a corpus!
  Brown corpus: 1 million wordform tokens and 61805 wordform types

What are words?

- Are commas and other non alfanumeric characters words?
- Are the voices “uh” and “um” words?
- Should we consider the lemmas instead of wordforms: “work” instead of “working”?

Words are defined by the task at hand!

A 2^{nd}-order Markov model (trigrams) of $P(w_1, \ldots w_m)$ can be modelled using two tables:

Trigram	**count**
$< s >$ The boy	Count($< s >$ The boy)
The boy went	Count(The boy went)
boy went home	Count(boy went home)
$\vdots$	$\vdots$

Bigram	**counts**
$< s >$ The	Count($< s >$ The)
The boy	Count(The boy)
boy went	Count(boy went)
$\vdots$	$\vdots$

The probability $P(w_i \mid w_{i-2}, w_{i-1})$ is estimated from counts in two tables:

table	Contains entries for
Table for 3-grams	$Count(w_{i-2}, w_{i-1}, w_i)$
Table for 2-grams	$Count(w_{i-2}, w_{i-1})$

In fact, 3-gram table is enough on its own (why?).

$$P(w_i \mid w_{i-2}, w_{i-1}) = \frac{Count(w_{i-2}, w_{i-1}, w_i)}{Count(w_{i-2}, w_{i-1})} \tag{1}$$

N-gram models of language: Pros and Cons

Pros

- Simple, easy and cheap
- useful for many applications
- availablity of statistics over the internet
- well understood math.

N-gram models of language: Pros and Cons

Pros

- Simple, easy and cheap
- useful for many applications
- availablity of statistics over the internet
- well understood math.

Cons

Language: they do not capture non-local dependencies
The boy (who was here yesterday) went home

Sparsity: not enough data to estimate large $n > 3$ values

Further: Markov assumption might be too strong

Next Lecture

- The construction of Language models (over words and over Part-of-speech tags)
- Any problems with current assignments?
- Deadlines are at 23:59 on deadline days