Plan for Remaining 5 Weeks

Date	Topic	Hw?
April 8	n-gram models	yes
April 10	Python (functions)	
April 15	n-gram models (cont.)	yes
April 17	Python (tokens)	
April 22	Machine learning	yes
April 24	Python (ngrams)	
April 29	Human-like models	practice /Final project
May 01	Python (frequencies)	
May 06	Summary	Final project
May 08	Python Q&A session	
May 14		Final Project due (at midnight)

Language & Technology

Lecture 7: From Unigrams to n-Grams

Alëna Aksënova & Aniello De Santo

Stony Brook University
alena.aksenova@stonybrook.edu
aniello.desanto@stonybrook.edu

Generalizing Unigrams

- Unigram models perform reasonably well:
  - cultuormics
  - stylistic analysis
  - authorship attribution
  - word semantics
  - ad placement
  - web search
- But they only consider words in isolation.
- An n-gram model looks at **sequences of words**.

Example: Word Prediction

Your phone can make suggestions for the most likely next word(s).

A (Bad) Solution with Unigrams

1. Build corpus.
2. For each word type, calculate number of tokens.
3. Calculate the **frequency** of the word in the sample:

$$\text{freq}(\textit{word}, \textit{sample}) = \frac{\text{number of tokens of } \textit{word}}{\text{word length of whole } \textit{sample}}$$

4. Suggest words with highest frequency.

Example Calculation

Sample: 1000 words long

Words: be, bed, bee, bell

Type	be	bed	bee	bell
Tokens	**13**	**2**	**0**	**3**

$$\text{freq}(\text{be}) = \frac{\mathbf{13}}{1000} = 1.3\% \qquad \text{freq}(\text{bee}) = \frac{\mathbf{0}}{1000} = 0.0\%$$

$$\text{freq}(\text{bed}) = \frac{\mathbf{2}}{1000} = 0.2\% \qquad \text{freq}(\text{bell}) = \frac{\mathbf{3}}{1000} = 0.3\%$$

This is a Workable Solution for Word Completion

word completion completing a partially typed word

A frequency-based unigram model can work reasonably well for word completion.

Example

$$\text{freq}(\text{be}) = \frac{\mathbf{13}}{1000} = 1.3\% \qquad \text{freq}(\text{bee}) = \frac{\mathbf{0}}{1000} = 0.0\%$$

$$\text{freq}(\text{bed}) = \frac{\mathbf{2}}{1000} = 0.2\% \qquad \text{freq}(\text{bell}) = \frac{\mathbf{3}}{1000} = 0.3\%$$

Partial input: be
Ranked completions: be, bell, bed, bee

Why This is a Horrible Solution for Word Prediction

word prediction suggesting next word before it is typed

Unigram models are horrible for word prediction:

- always suggest the same words
- only suggest stop words (because they're most frequent)
- do not take context into account

The n-Gram Hypothesis

One can reliably predict the next word based on the **preceding $n-1$ words**.

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	**Name**	**Example**
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	**Name**	**Example**
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie $\boxed{\text{are not}}$ Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are $\boxed{\text{not Bill}}$ and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John $\boxed{\text{and Marie are}}$ not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie $\boxed{\text{are not Bill}}$ and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are $\boxed{\text{not Bill and}}$ Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	**Name**	**Example**
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not $\boxed{\text{Bill and Sue}}$

Defining n-grams

n-gram a contiguous sequence of n words

n	**Name**	**Example**
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

$\boxed{\text{John and Marie are}}$ not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John $\boxed{\text{and Marie are not}}$ Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie $\boxed{\text{are not Bill and}}$ Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie are not Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

$\boxed{\text{John and Marie are not}}$ Bill and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John $\boxed{\text{and Marie are not Bill}}$ and Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and $\boxed{\text{Marie are not Bill and}}$ Sue

Defining n-grams

n-gram a contiguous sequence of n words

n	Name	Example
1	unigram	John
2	bigram	John to
3	trigram	John to be
4	4-gram	John to be in
5	5-gram	John to be in the car

Example

String

John and Marie $\boxed{\text{are not Bill and Sue}}$

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo buffalo buffalo buffalo buffalo buffalo

- **Bigram token list**

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  $\boxed{\text{when buffalo}}$ buffalo buffalo buffalo buffalo buffalo

- **Bigram token list**

  when buffallo,

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when $\boxed{\text{buffalo buffalo}}$ buffalo buffalo buffalo buffalo

- **Bigram token list**

  when buffallo, buffalo buffalo,

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo $\boxed{\text{buffalo buffalo}}$ buffalo buffalo buffalo

- **Bigram token list**
  when buffallo, buffalo buffalo, buffalo buffalo,

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo buffalo $\boxed{\text{buffalo buffalo}}$ buffalo buffalo

- **Bigram token list**
  when buffallo, buffalo buffalo, buffalo buffalo, buffalo buffalo,

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo buffalo buffalo $\boxed{\text{buffalo buffalo}}$ buffalo

- **Bigram token list**
  when buffallo, buffalo buffalo, buffalo buffalo, buffalo buffalo, buffalo buffalo,

- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo buffalo buffalo buffalo buffalo buffalo

- **Bigram token list**
  when buffallo, buffalo buffalo, buffalo buffalo, buffalo buffalo, buffalo buffalo, buffalo buffalo
- **Bigram counts and frequencies**

Frequencies for n-grams

Frequencies can be computed for n-grams, too.

Example: Calculating Bigram Frequencies

- **String**

  when buffalo buffalo buffalo buffalo buffalo buffalo

- **Bigram token list**
  when buffallo, buffalo buffalo, buffalo buffalo, buffalo buffalo, buffalo buffalo, buffalo buffalo

- **Bigram counts and frequencies**
  1. when buffalo: 1 $\Rightarrow \frac{1}{6} = 16.7\%$
  2. buffalo buffalo: 5 $\Rightarrow \frac{5}{6} = 83.3\%$

How Your Phone Does it

- Frequency database for n-grams ($2 \leq n \leq 5$)
- Look at previous $n - 1$ words.
- Pick **fitting n-gram with highest frequency**.

Example

- **Trigram frequencies**

bus is late	30%	train is late	15%
bus is cheap	25%	train is cheap	8%
bus is early	20%	train is early	2%

- **Input**
  I will text you if the train is
- **Word suggestion**

How Your Phone Does it

- Frequency database for n-grams ($2 \leq n \leq 5$)
- Look at previous $n - 1$ words.
- Pick **fitting n-gram with highest frequency**.

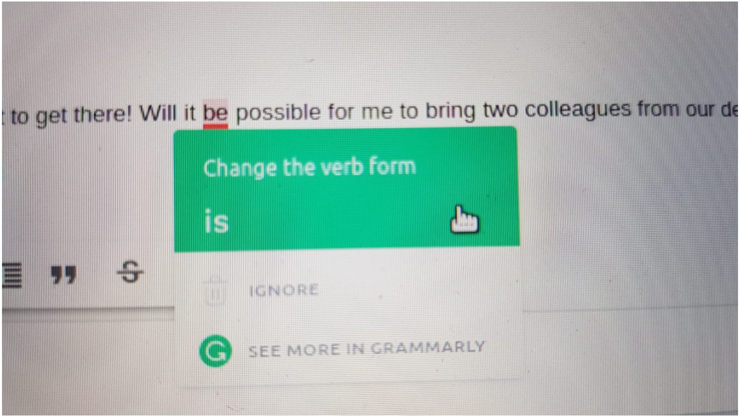
Example

- **Trigram frequencies**

bus is late	30%	train is late	15%
bus is cheap	25%	train is cheap	8%
bus is early	20%	train is early	2%

- **Input**
  I will text you if the train is
- **Word suggestion**
  late

An example from Grammarly ...



Zipf's Law Strikes Again

- As with words, n-grams have a Zipfian distribution.
- This creates a major problem: **sparse data**

The Overwhelming Number of n-Grams

- Suppose English has 5,000 words (it actually has way more)
- Suppose each word has two inflected forms

  see the picture and **see the pictures** are distinct trigrams!
- Then there are $10,000^n = 10^{4n}$ distinct n-grams.

n	number of possible n-grams
2	100 million
3	1 trillion
4	10 quadrillion
5	100 quintillion

Is That a Lot?

- Assuming $10,000$ English word forms, the number of 5-grams rivals the **number of seconds since the Big Bang**!

The Sparse Data Problem

1. We want a large n for better accuracy.
2. But the larger the n, the more data we need.
3. Because of Zipf's law, the majority of the data consists of the same n-grams.
4. Hence most grammatical n-grams have a frequency of 0.
5. This means they will never be suggested, even if there is no grammatical alternative.

Things Get Worse: A More Realistic Estimate

- The Unix dictionary american-english-insane has 650,000 entries.
- This makes the numbers much worse.
  Can you guess how many 5-grams there are then?

Things Get Worse: A More Realistic Estimate

- The Unix dictionary american-english-insane has 650,000 entries.
- This makes the numbers much worse.
  Can you guess how many 5-grams there are then?

$$\text{116 octillion} \approx \mathbf{10^{29}}$$

Things Get Worse: A More Realistic Estimate

- The Unix dictionary american-english-insane has 650,000 entries.
- This makes the numbers much worse.
  Can you guess how many 5-grams there are then?

116 octillion $\approx \mathbf{10^{29}}$

Number	**Real-world counterpart**
10^{14}	distance in millimeters from Earth to Sun
10^{18}	seconds since Big Bang
10^{24}	milliliters of water in the oceans

Things Get Worse: A More Realistic Estimate

- The Unix dictionary american-english-insane has 650,000 entries.
- This makes the numbers much worse.
  Can you guess how many 5-grams there are then?

116 octillion $\approx \mathbf{10^{29}}$

Number	**Real-world counterpart**
10^{14}	distance in millimeters from Earth to Sun
10^{18}	seconds since Big Bang
10^{24}	milliliters of water in the oceans

10^{29} is larger than the number of shotglasses it takes to drain Earth's oceans over 2000 times.

Trick 1: Stemming and Lemmatization

- Removing inflectional markers reduces number of words
- Two solutions:
  - stemming is quick and dirty
  - lemmatization is accurate but complex

stemming cut off word ends that look like inflection

Example

- cats $\Rightarrow$ cat
- tasks $\Rightarrow$ task (noun and verb)
- asking $\Rightarrow$ ask
- meeting $\Rightarrow$ meet (**noun and verb**)

Trick 1: Stemming and Lemmatization [cont.]

lemmatization stemming with context information

Example

- cats $\Rightarrow$ cat
- tasks $\Rightarrow$ task (noun and verb)
- asking $\Rightarrow$ ask
- meeting $\Rightarrow$ meet (**only verb**)

Evaluation

- Stemming/lemmatization reduces the number of words.
- But we still have at least 10,000 words and thus 10^{20} 5-grams.

Trick 2: Statistics

- **Backoff Method**
  If an n-gram has frequency 0, use the frequency of the corresponding $(n-1)$-gram.
- **Good-Turing Smoothing**
  Change frequency from 0 to a very low value while lowering high frequency values.

Evaluation

- These tricks solve the issue of n-grams with 0% frequency.
- But they do not solve the basic problem that n-gram models are incredibly data hungry.

Future of n-Gram Models

- Moving from unigrams to n-gram models increases performance in many applications we discussed.
  - culturomics
  - stylistic analysis
  - web search
  - ad placement
- But we quickly hit diminishing returns.
- Even 5-gram models are **no match for humans**, and it's unlikely we'll be able to move on to 6-grams any time soon.

New Areas of Application

- N-gram models are nearly maxed out in current applications.
- But this still leaves areas where they haven't been used at all.
- Let's briefly look at one example: OCR.

Optical Character Recognition

OCR the process of

1. scanning in images of text and
2. converting it into digital text.

"making computers read"

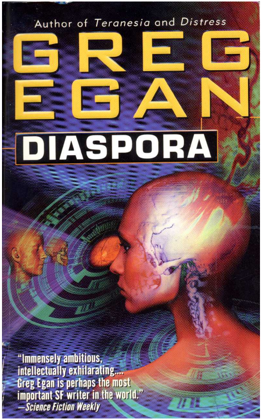
- In a purely digital world, OCR would be superfluous.
- But there's still many analog texts that need to be digitized.
  old books, paper forms, signed contracts, ...
- A special (and much harder) case of OCR is handwriting recognition.

The Quality of Current OCR Software

- OCR sounds trivial; even a 4-year old can recognize letters
- So **why are my ebooks full of mistakes?**

Examples from *Diaspora* (1997)

- That was-n't entirely true;
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- If they can shorten wormholes, the\, might visit us.
- He fell silent, abruptly realizing ”it'll she t, as feeling: electing not to wake up again [. . .]
- Seaweed every twenty -seven -seven I DIASPORA 231 light years.



The Prototype Problem

- Characters have **prototypical shapes**.
- But numerous deviations are possible, with fuzzy borders.

Non-Mandatory Properties of the Letter A

- two angular strokes, meeting at top
- cross bar
- no horizontal top stroke
- no horizontal bottom stroke
- no curves or arcs
- no disconnected parts

Recognizing Characters is Not Enough

- Mapping pixels in an image to characters is a probabilistic process that is affected by many parameters
  - font
  - low-quality printing process
  - stains on page

  ⋮
- Some misidentifications are unavoidable.
- Why don't humans run into the same problems?
  Because **humans do not read character by character**.

Basic Properties of Human Reading

- **Saccades**
  reading proceeds not character by character,
  eyes move in **saccades**:
  1. focus on several words at once and identify words,
  2. once done, move eyes to next cluster of words to the right,
  3. focus, absorb, then move again, and so on.
- **Word Identification**
  pattern-match whole words rather than character sequences
  ⇒ order of character usually of little relevance

  I cnduo't bvleiee taht I culod aulaclty uesdtannrd
  waht I was rdnaieg.
- **Predictive**
  speakers use information about sentence to predict next word
  ⇒ unexpected words read more slowly

Taking a Hint: Adding Unigrams to OCR

Proposal: OCR-ed sequence of characters must be a word in our dictionary

Examples from *Diaspora* (1997)

- That was-n't entirely true;
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- If they can shorten wormholes, the\, might visit us.
- He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [...]
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Unigrams to OCR

Proposal: OCR-ed sequence of characters must be a word in our dictionary

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- If they can shorten wormholes, the\, might visit us.
- He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Unigrams to OCR

Proposal: OCR-ed sequence of characters must be a word in our dictionary

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- ~~If they can shorten wormholes, the\, might visit us.~~
- He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Unigrams to OCR

Proposal: OCR-ed sequence of characters must be a word in our dictionary

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- ~~If they can shorten wormholes, the\, might visit us.~~
- ~~He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]~~
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Bigrams to OCR

Proposal: only pick words that yield licit bigram

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- 1 want sharp borders, right now.
- The carpets seem to he vulnerable.
- ~~If they can shorten wormholes, the\, might visit us.~~
- ~~He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]~~
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Bigrams to OCR

Proposal: only pick words that yield licit bigram

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- ~~1 want sharp borders, right now.~~
- The carpets seem to he vulnerable.
- ~~If they can shorten wormholes, the\, might visit us.~~
- ~~He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]~~
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Bigrams to OCR

Proposal: only pick words that yield licit bigram

Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- ~~1 want sharp borders, right now.~~
- ~~The carpets seem to he vulnerable.~~
- ~~If they can shorten wormholes, the\, might visit us.~~
- ~~He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]~~
- Seaweed every twenty seven seven I DIASPORA 231 light years.

Taking a Hint: Adding Bigrams to OCR

Proposal: only pick words that yield licit bigram

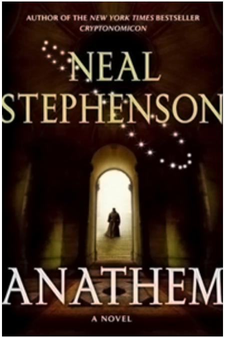
Examples from *Diaspora* (1997)

- ~~That was-n't entirely true;~~
- ~~1 want sharp borders, right now.~~
- ~~The carpets seem to he vulnerable.~~
- ~~If they can shorten wormholes, the\, might visit us.~~
- ~~He fell silent, abruptly realizing "it'll she t, as feeling: electing not to wake up again [. . .]~~
- ~~Seaweed every twenty seven seven I DIASPORA 231 light years.~~

Problems of the Approach

Why don't OCR models use n-grams?
They **create new problems**.

- Lexical creativity
  Neal Stephensons's *Anathem*: speely captor, jeejah, fraa, suur, cartabla, orth, saunt, suvin
- Grammatical creativity
  Diaspora: gender-neutral pronoun ve/vis/ver
- (Deliberately) Archaic language
  Tolkien's *LotR*: anon, askance, ere, furlong, lissom, recreant, thraldom
- Multiple languages
  German and French in *The Magic Mountain*
- Typos



Summary: OCR Needs Fixing

- Current OCR models operate purely character-by-character.
- They do not produce stellar results.
  even 99% accuracy means at least one mistake every other page
- Humans are much more competent and use linguistic insights.
- Adding n-grams is a first step in the same direction.