Distributed Representation

Language Modeling

Motivation: suppose we want to build a speech recognition system.

We'd like to be able to infer a likely sentence $\mathbf{s}$ given the observed speech signal $\mathbf{a}$. The generative approach is to build two components:

- An observation model, represented as $p(\mathbf{a} \mid \mathbf{s})$, which tells us how likely the sentence $\mathbf{s}$ is to lead to the acoustic signal $\mathbf{a}$.
- A prior, represented as $p(\mathbf{s})$, which tells us how likely a given sentence $\mathbf{s}$ is. E.g., it should know that "recognize speech" is more likely that "wreck a nice beach."

Language Modeling

Motivation: suppose we want to build a speech recognition system.

We'd like to be able to infer a likely sentence $\mathbf{s}$ given the observed speech signal $\mathbf{a}$. The generative approach is to build two components:

- An observation model, represented as $p(\mathbf{a} \mid \mathbf{s})$, which tells us how likely the sentence $\mathbf{s}$ is to lead to the acoustic signal $\mathbf{a}$.
- A prior, represented as $p(\mathbf{s})$, which tells us how likely a given sentence $\mathbf{s}$ is. E.g., it should know that "recognize speech" is more likely that "wreck a nice beach."

Given these components, we can use Bayes' Rule to infer a posterior distribution over sentences given the speech signal:

$$p(\mathbf{s} \mid \mathbf{a}) = \frac{p(\mathbf{s})p(\mathbf{a} \mid \mathbf{s})}{\sum_{\mathbf{s}'} p(\mathbf{s}')p(\mathbf{a} \mid \mathbf{s}')}.$$

Language Modeling

In this lecture, we focus on learning a good distribution $p(\mathbf{s})$ of sentences. This problem is known as language modeling.

Assume we have a corpus of sentences $\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(N)}$. The maximum likelihood criterion says we want our model to maximize the probability our model assigns to the observed sentences. We assume the sentences are independent, so that their probabilities multiply.

$$\max \prod_{i=1}^{N} p(\mathbf{s}^{(i)}).$$

Language Modeling

In maximum likelihood training, we want to maximize $\prod_{i=1}^{N} p(\mathbf{s}^{(i)})$.

The probability of generating the whole training corpus is vanishingly small — like monkeys typing all of Shakespeare.

- The log probability is something we can work with more easily. It also conveniently decomposes as a sum:

$$\log \prod_{i=1}^{N} p(\mathbf{s}^{(i)}) = \sum_{i=1}^{N} \log p(\mathbf{s}^{(i)}).$$

- Let's use *negative* log probabilities, so that we're working with positive numbers.

Language Modeling

In maximum likelihood training, we want to maximize $\prod_{i=1}^{N} p(\mathbf{s}^{(i)})$.

The probability of generating the whole training corpus is vanishingly small — like monkeys typing all of Shakespeare.

- The log probability is something we can work with more easily. It also conveniently decomposes as a sum:

$$\log \prod_{i=1}^{N} p(\mathbf{s}^{(i)}) = \sum_{i=1}^{N} \log p(\mathbf{s}^{(i)}).$$

- Let's use *negative* log probabilities, so that we're working with positive numbers.

Language Modeling

- Probability of a sentence? What does that even mean?
  - A sentence is a sequence of words $w_1, w_2, \ldots, w_T$. Using the chain rule of conditional probability, we can decompose the probability as

$$p(\mathbf{s}) = p(w_1, \ldots, w_T) = p(w_1)p(w_2 \mid w_1) \cdots p(w_T \mid w_1, \ldots, w_{T-1}).$$

  - Therefore, the language modeling problem is equivalent to being able to predict the next word!
- We typically make a Markov assumption, i.e. that the distribution over the next word only depends on the preceding few words. I.e., if we use a context of length 3,

$$p(w_t \mid w_1, \ldots, w_{t-1}) = p(w_t \mid w_{t-3}, w_{t-2}, w_{t-1}).$$

  - Such a model is called memoryless.
  - Now it's basically a supervised prediction problem. We need to predict the conditional distribution of each word given the previous K.
  - When we decompose it into separate prediction problems this way, it's called an autoregressive model.

N-Gram Language Models

- One sort of Markov model we can learn uses a conditional probability table, i.e.

	cat	and	city	$\cdots$
the fat	0.21	0.003	0.01	
four score	0.0001	0.55	0.0001	$\cdots$
New York	0.002	0.0001	0.48	
$\vdots$		$\vdots$		

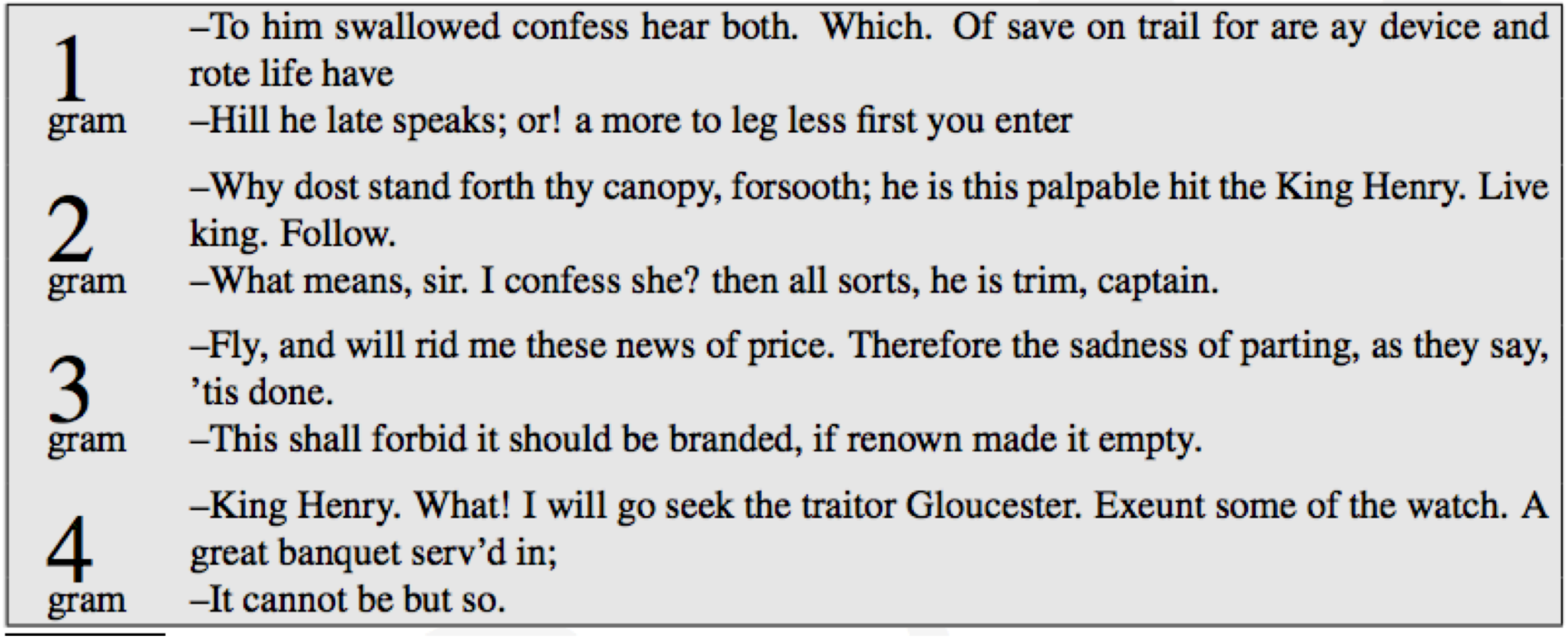
- Maybe the simplest way to estimate the probabilities is from the empirical distribution:

$$p(w_3 = \mathrm{cat} \mid w_1 = \mathrm{the}, w_2 = \mathrm{fat}) = \frac{\mathrm{count}(\mathrm{the\ fat\ cat})}{\mathrm{count}(\mathrm{the\ fat})}$$

- This is the maximum likelihood solution; we'll see why later in the course.
- The phrases we're counting are called n-grams (where n is the length), so this is an n-gram language model.
  - Note: the above example is considered a 3-gram model, not a 2-gram

N-Gram Language Models

Shakespeare:

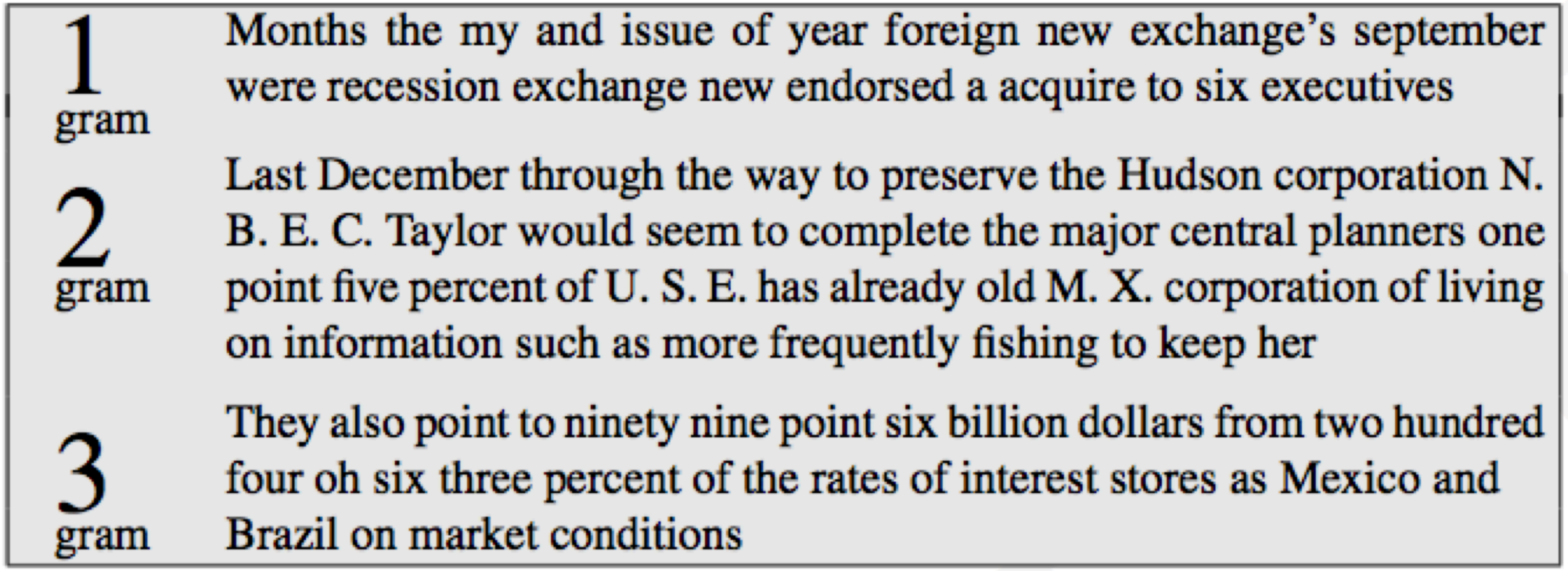


1 gram	–To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have –Hill he late speaks; or! a more to leg less first you enter
2 gram	–Why dost stand forth thy canopy, forsooth; he is this palpable hit the King Henry. Live king. Follow. –What means, sir. I confess she? then all sorts, he is trim, captain.
3 gram	–Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done. –This shall forbid it should be branded, if renown made it empty.
4 gram	–King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; –It cannot be but so.

Jurafsky and Martin, *Speech and Language Processing*

N-Gram Language Models

Wall Street Journal:



1 gram	Months the my and issue of year foreign new exchange's september were recession exchange new endorsed a acquire to six executives
2 gram	Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor would seem to complete the major central planners one point five percent of U. S. E. has already old M. X. corporation of living on information such as more frequently fishing to keep her
3 gram	They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and Brazil on market conditions

Jurafsky and Martin, *Speech and Language Processing*

N-Gram Language Models

- Problems with n-gram language models

N-Gram Language Models

- Problems with n-gram language models
  - The number of entries in the conditional probability table is exponential in the context length.
  - Data sparsity: most n-grams never appear in the corpus, even if they are possible.

N-Gram Language Models

- Problems with n-gram language models
  - The number of entries in the conditional probability table is exponential in the context length.
  - Data sparsity: most n-grams never appear in the corpus, even if they are possible.
- Ways to deal with data sparsity

N-Gram Language Models

- Problems with n-gram language models
  - The number of entries in the conditional probability table is exponential in the context length.
  - Data sparsity: most n-grams never appear in the corpus, even if they are possible.
- Ways to deal with data sparsity
  - Use a short context (but this means the model is less powerful)
  - Smooth the probabilities, e.g. by adding imaginary counts
  - Make predictions using an ensemble of n-gram models with different n

Distributed Representations

- Conditional probability tables are a kind of localist representation: all the information about a particular word is stored in one place, i.e. a column of the table.
- But different words are related, so we ought to be able to share information between them. For instance, consider this matrix of word attributes:

	academic	**politics**	**plural**	**person**	**building**
students	1	0	1	1	0
colleges	1	0	1	0	1
legislators	0	1	1	1	0
schoolhouse	1	0	0	0	1

- And this matrix of how each attribute influences the next word:

	bill	**is**	**are**	**papers**	**built**	**standing**
academic	−			+		
politics	+			−		
plural		−	+			
person						+
building					+	+