Lecture 3:
Language models

Julia Hockenmaier
juliahmr@illinois.edu
3324 Siebel Center

Finite-state transducers

- FSTs define a relation between two regular languages.
- Each state transition maps (transduces) a character from the input language to a character (or a sequence of characters) in the output language

\[ x : y \]

- By using the empty character (\( \varepsilon \)), characters can be deleted (\( x : \varepsilon \)) or inserted (\( \varepsilon : y \))

- FSTs can be composed (cascaded), allowing us to define intermediate representations.

Today’s lecture

How can we distinguish word salad, spelling errors and grammatical sentences?

Language models define probability distributions over the strings in a language. N-gram models are the simplest and most common kind of language model.

We’ll look at how they’re defined, how to estimate (learn) them, and what their shortcomings are.

We’ll also review some very basic probability theory.
Why do we need language models?

Many NLP tasks return output in natural language:
- Machine translation
- Speech recognition
- Natural language generation
- Spell-checking

Language models define probability distributions over (natural language) strings or sentences.

We can use them to score/rank possible sentences:
If $P_{LM}(A) > P_{LM}(B)$, choose sentence A over B

---

**Sampling with replacement**

Pick a random shape, then put it back in the bag.

<table>
<thead>
<tr>
<th>Shape</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>5/15</td>
</tr>
<tr>
<td>Blue</td>
<td>5/15</td>
</tr>
<tr>
<td>□</td>
<td>2/15</td>
</tr>
</tbody>
</table>

$P(\text{Red}) = 5/15$

$P(\text{Blue}) = 5/15$

$P(\text{□}) = 2/15$

---

**Sampling with replacement**

Pick a random shape, then put it back in the bag.

What sequence of shapes will you draw?

$P(\text{Red} \land △) = 3/5 \times 2/15 = 6/50$  
$P(\text{Red} \lor △) = 5/15 \times 3/5 = 15/50$  
$P(\text{Red} \lor △) = 15/50 + 6/50 = 21/50$

---

**Basic Probability Theory**
Sampling with replacement

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'

\[
P(\text{of}) = \frac{3}{66} \quad P(\text{her}) = \frac{2}{66} \\
P(\text{Alice}) = \frac{2}{66} \quad P(\text{sister}) = \frac{2}{66} \\
P(\text{was}) = \frac{2}{66} \quad P(.) = \frac{4}{66} \\
P(\text{to}) = \frac{2}{66} \quad P(') = \frac{4}{66}
\]

In this model, \( P(\text{English sentence}) = P(\text{word salad}) \)

Probability theory: terminology

**Trial:**
Picking a shape, predicting a word

**Sample space** \( \Omega \):
The set of all possible outcomes (all shapes; all words in Alice in Wonderland)

**Event** \( \omega \subseteq \Omega \):
An actual outcome (a subset of \( \Omega \)) (predicting ‘the’, picking a triangle)

The probability of events

Kolmogorov axioms:
1) Each event has a probability between 0 and 1.
2) The null event has probability 0.
   - The probability that any event happens is 1.
3) The probability of all disjoint events sums to 1.

\[
0 \leq P(\omega \subseteq \Omega) \leq 1 \\
P(\emptyset) = 0 \quad \text{and} \quad P(\Omega) = 1 \\
\sum_{\omega_i \subseteq \Omega} P(\omega_i) = 1 \quad \text{if} \quad \forall j \neq i : \omega_i \cap \omega_j = \emptyset \\
\quad \text{and} \quad \bigcup_i \omega_i = \Omega
\]
Discrete probability distributions: single trials

**Bernoulli distribution** (two possible outcomes)
The probability of success (=head,yes)
  - The probability of head is $p$.
  - The probability of tail is $1-p$.

**Categorical distribution** ($N$ possible outcomes)
The probability of category/outcome $c_i$ is $p_i$
  - $0 \leq p_i \leq 1$ $\sum_i p_i = 1$
  - Also often (incorrectly) called Multinomial distribution

---

**Joint and Conditional Probability**

The conditional probability of $X$ given $Y$, $P(X \mid Y)$, is defined in terms of the probability of $Y$, $P(Y)$, and the joint probability of $X$ and $Y$, $P(X,Y)$:

$$P(X \mid Y) = \frac{P(X,Y)}{P(Y)}$$

$P(\text{blue} \mid \square) = 2/5$

---

**Conditioning on the previous word**

*Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do: once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it, 'and what is the use of a book,' thought Alice 'without pictures or conversation?'}
The chain rule

The joint probability $P(X,Y)$ can also be expressed in terms of the conditional probability $P(X \mid Y)$

$$P(X, Y) = P(X \mid Y)P(Y)$$

This leads to the so-called chain rule:

$$P(X_1, X_2, \ldots, X_n) = P(X_1)P(X_2 \mid X_1)P(X_3 \mid X_2, X_1)\ldots P(X_n \mid X_1, \ldots X_{n-1})$$

$$= P(X_1) \prod_{i=2}^{n} P(X_i \mid X_1 \ldots X_{i-1})$$

Independence

Two random variables $X$ and $Y$ are independent if

$$P(X, Y) = P(X)P(Y)$$

If $X$ and $Y$ are independent, then $P(X \mid Y) = P(X)$:

$$P(X \mid Y) = \frac{P(X, Y)}{P(Y)} = \frac{P(X)P(Y)}{P(Y)} \quad (X, Y \text{ independent})$$

$$= P(X)$$

Probability models

Building a probability model consists of two steps:
1. Defining the model
2. Estimating the model’s parameters

Models (almost) always make independence assumptions.

That is, even though $X$ and $Y$ are not actually independent, our model may treat them as independent.

This reduces the number of model parameters that we need to estimate (e.g. from $n^2$ to $2n$)

Language modeling with n-grams
Language modeling with N-grams

A language model over a vocabulary $V$ assigns probabilities to strings drawn from $V^*$.

Recall the chain rule:

$$P(w_1...w_i) = P(w_1)P(w_2|w_1)P(w_3|w_1w_2)...P(w_i|w_1...w_{i-1})$$

An n-gram language model assumes each word depends only on the last $n-1$ words:

$$P_{ngram}(w_1...w_i) := P(w_1)P(w_2|w_1)...P\left( \underbrace{w_i}_{\text{nth word}} \mid \underbrace{w_{i-n+1}...w_{i-1}}_{\text{prev. } n-1 \text{ words}} \right)$$

### Estimating N-gram models

1. Bracket each sentence by special start and end symbols:
   ```
   <s> Alice was beginning to get very tired... </s>
   (We only assign probabilities to strings <s>... </s>)
   ```

2. Count the frequency of each n-gram:
   
   $$C(<s> \text{ Alice}) = 1, C(\text{Alice was}) = 1,....$$  

3. .... and normalize to get the probability:

   $$P(w_i|w_{i-1}...w_{i-n+1}) = \frac{C(w_i|w_{i-1}...w_{i-n+1})}{\sum_{j=1}^{n} C(w_j|w_{j-1}...w_{j-n+1})}$$

   This is called a relative frequency estimate of $P(w_i | w_{i-1})$

### N-gram models

- **Unigram model**
  $$P(w_1)P(w_2)...P(w_i)$$

- **Bigram model**
  $$P(w_1)P(w_2|w_1)...P(w_i|w_{i-1})$$

- **Trigram model**
  $$P(w_1)P(w_2|w_1)...P(w_i|w_{i-2}w_{i-1})$$

- **N-gram model**
  $$P(w_1)P(w_2|w_1)...P(w_i|w_{i-n+1}...w_{i-1})$$

N-gram models assume each word (event) depends only on the previous $n-1$ words (events). Such independence assumptions are called Markov assumptions (of order $n-1$).

$$P(w_i|w_{i-1}...w_{i-n+1}) \approx P(w_i|w_{i-n}...w_{i-1})$$

### Using n-gram models to generate language
Generating from a distribution

How do you generate text from an \( n \)-gram model?

That is, how do you sample from a distribution \( P(X \mid Y=y) \)?
- Assume \( X \) has \( N \) possible outcomes (values): \( \{x_1, \ldots, x_N\} \)
- \( P(X=x_i \mid Y=y) = p_i \)
- Divide the interval \([0,1]\) into \( N \) smaller intervals according to the probabilities of the outcomes
- Generate a random number \( r \) between 0 and 1.
- Return the \( x_i \) whose interval the number is in.

Generating Shakespeare

- To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have
- Every enter now severely so, lot
- 'Hill he late speak; oh! a more to leg less first you enter
- Are where excent and sichts have rise excellency took of. Sleep knowe we. near; vile like
- What means, sir. I confess she? then all sorts, he is trim, captain.
- Why dost stand forth thy canopy, forsooth, he is this palpable hit the King Henry. Live king. Follow.
- What we, hath got so she that I rest and sent to scold and nature bankrupt, nor the first gentleman?
- Thee comes, if so many good direction found st thou art a strong upon command of fear not a liberal largess given away. Falstaff! Exeunt

Shakespeare as corpus

The Shakespeare corpus consists of \( N=884,647 \) word tokens and a vocabulary of \( V=29,066 \) word types

Shakespeare produced 300,000 bigram types out of \( V^2=844 \) million possible bigram types. 99.96% of the possible bigrams were never seen

4-grams look like Shakespeare because they are Shakespeare
Unseen events matter

We estimated a model on 440K word tokens, but:

Only 30,000 word types occurred. Any word that does not occur in the training data has zero probability!

Only 0.04% of all possible bigrams occurred. Any bigram that does not occur in the training data has zero probability!

Dealing with unseen events

Relative frequency estimation assigns all probability mass to events in the training corpus

But we need to reserve some probability mass to events that don’t occur in the training data

Unseen events = new words, new bigrams

Important questions:

What possible events are there?

How much probability mass should they get?

Zipf’s law: the long tail

How many words occur once, twice, 100 times, 1000 times?

In natural language:

- A small number of events (e.g. words) occur with high frequency
- A large number of events occur with very low frequency

To recap….
Today’s key concepts

N-gram language models
Independence assumptions
Relative frequency estimation
Unseen events
Zipf’s law

Today’s reading:
Jurafsky and Martin, Chapter 4, sections 1-4