Language Modeling

Prof. Sameer Singh

CS 295: STATISTICAL NLP
WINTER 2017

January 24, 2017
Outline

- Wrapup Word Embeddings
- Introduction to Language Models
- N-Gram Based Language Models
- Smoothing Language Models
Predict surrounding words

$$P(w_{t+j} | w_t) \neq j \in \{-m, \ldots, m \} \cup \{0\}$$

$$P(o|c) = \frac{e^{u_0 \cdot v_c}}{\sum_{v} e^{u_0 \cdot v_c}}$$
Negative Sampling

\[ \hat{p}(0|c) = \frac{e^{u_0 v_c}}{1 + e^{u_0 v_c}} \]

\[ \hat{p}(w|c) \geq 1 \]

\[ \arg\max_{w} \sum_{j} \log \hat{p}(w_{t+j}|w_t) + \sum_{k} \frac{1}{k} \log (1 - \hat{p}(w_{k+1}|w_t)) \]

Negative Sampling

weeks \rightarrow minutes
Neural View of Embeddings
Word embeddings

Variations

• Skip-gram: predict context from word
• CBOW: predict word from context bag of words
• Dependencies: a better description of context

Uses

• Similarity:
• Grammar:
• Analogies
  • Gender:
  • Facts:

\[ w_i, w_j \sim \cos(V_i, V_j) \]

walking \rightarrow \text{swam}

King \rightarrow \text{male} + \text{female} \rightarrow \text{queen}

Doctor \rightarrow m + f \rightarrow \text{nurse}

Capital \rightarrow Country + France \rightarrow \text{Paris}
Outline

- Wrapup Word Embeddings
- Introduction to Language Models
- N-Gram Based Language Models
- Smoothing Language Models
Language Models

Probability of a Sentence

- Is a given sentence something you would expect to see?
- Syntactically (grammar) and Semantically (meaning)

\[ P(W) = P(w_1, w_2, \ldots, w_n) \]

Probability of the Next Word

- Predict what comes next for a given sequence of words.
- Think of it as V-way classification

\[ P(w_i | w_1, w_2, \ldots, w_{i-1}) \in \{1, \ldots, V\} \]
Task: Speech Recognition

“eyes awe of an”
OR
“I saw a van”

\[ \Pr(\mathbf{w}_2) > \Pr(\mathbf{w}_1) \]

<table>
<thead>
<tr>
<th>word sequence</th>
<th>( \log p(\text{acoustics} \mid \text{word sequence}) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>the station signs are in deep in english</td>
<td>-14732</td>
</tr>
<tr>
<td>the stations signs are in deep in english</td>
<td>-14735</td>
</tr>
<tr>
<td>the station signs are in deep into english</td>
<td>-14739</td>
</tr>
<tr>
<td>the station ’s signs are in deep in english</td>
<td>-14740</td>
</tr>
<tr>
<td>the station signs are in deep in the english</td>
<td>-14741</td>
</tr>
<tr>
<td>the station signs are indeed in english</td>
<td>-14757</td>
</tr>
<tr>
<td>the station ’s signs are indeed in english</td>
<td>-14760</td>
</tr>
<tr>
<td>the station signs are indians in english</td>
<td>-14790</td>
</tr>
<tr>
<td>the station signs are indian in english</td>
<td>-14799</td>
</tr>
<tr>
<td>the stations signs are indians in english</td>
<td>-14807</td>
</tr>
<tr>
<td>the stations signs are indians and english</td>
<td>-14815</td>
</tr>
</tbody>
</table>
Task: Machine Translation

Quiero ir a la playa más bonita.

I try | to leave | per | the most lovely | open space.  

I want | to go | to | the prettiest | beach.  

$P(\tilde{m}_2) > P(\tilde{m}_1)$
Task: Handwriting Recognition

http://www.cedar.buffalo.edu/handwriting/HROverview.html
Task: Image Captioning

A person skiing down a snow covered slope.
Task: Spelling Correction

The office is about fifteen minuets from my house

\[ P(\text{about fifteen minutes from}) \gg P(\text{about fifteen minuets from}) \]
Other Applications

- Summarization
- Question Answering
- Dialog Systems
Evaluating Language Models

Best choice: Extrinsic

2nd choice: Intrinsic

\[ P_A(\bar{w}) \quad P_B(\bar{w}) \]

Application: MT

\[ \text{acc}(P_A) \quad \text{acc}(P_B) \]

Train

Dev

Test

\[ p(\bar{w}) \]

learn P

tune
Perplexity $PP$

$$P(W) = \prod_i P(\bar{w}_i)$$

$$\frac{1}{n} \log_2 P(W) = \frac{1}{n} \sum_i \log_2 P(\bar{w}_i)$$

$$PP(W) = 2^{-\frac{1}{n} \sum_i \log P(\bar{w}_i)}$$

$$= n \sqrt[n]{\prod_i P(\bar{w}_i)}$$

Random: $P(\bar{w}_i) = \frac{1}{V}$, $PP(W) = V$

Perfectly: $PP(W) = 1$
Generating Text from an LM

\[ S = [] \quad \# \text{prefix} \]

\[
\text{do} \quad w \leftarrow P(w|s) \\
S \leftarrow w \\
\text{while } w \neq "Eos" \text{ or } \text{maxLength}
\]
Outline

- Wrapup Word Embeddings
- Introduction to Language Models
- N-Gram Based Language Models
- Smoothing Language Models
Direct Language Modeling

\[ P(\text{"I do not like green eggs and ham"}) = \frac{\#(\text{"I do not like"} \ldots \text{""})}{N \text{ number of sentences}} \]

\[ P(w \mid \text{"I do not like green eggs and"}) = \frac{\#(\text{"I do not"} \ldots \text{""} + w)}{\#(\text{"I do not"} \ldots \text{"" and"})} \]

\[ \text{ham} \]
\[ w \text{ 0} \]
Applying the Chain Rule

\[ p(w_1, w_2, \ldots, w_n) = p(w_1) p(w_2 | w_1) p(w_3 | w_1, w_2) \ldots p(w_n | w_1, w_2, \ldots, w_{n-1}) \]

\[ p(\text{"I do not like eggs"}) = p(\text{"I"} | <s>) \]
\[ \quad \times p(\text{"do"} | \text{"I"}) \]
\[ \quad \times \ldots \times p(\text{"eggs"} | \text{"I do not like"}) \]
Markov Assumption

\[ P(w_i | w_{i-k}, \ldots, w_{i-1}) = P(w_i | w_{i-k}, \ldots, w_{i-1}) \]

1st Order Markov
Unigram Language Model

\[ p(w_i | w_1, \ldots, w_{i-1}) = p(w_i) = \frac{\# w_i}{N \rightarrow \text{number of words}} \]

\[ p(\text{"the a an is the"}) > p(\text{"I love food"}) \]
Bigram Language Model

\[
P(w_i | w_1, w_2, \ldots, w_{i-1}) = P(w_i | w_{i-1})
\]

\[
= \frac{\#"w_{i-1} w"}{\# "w_{i-1}"
}\]

Corpus: 800k
Vocab: 30k

30k x 30k
300k bigrams obs.
Berkeley Restaurant Project

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>5</td>
<td>827</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>want</td>
<td>2</td>
<td>0</td>
<td>608</td>
<td>1</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>to</td>
<td>2</td>
<td>0</td>
<td>4</td>
<td>686</td>
<td>2</td>
<td>0</td>
<td>6</td>
<td>211</td>
</tr>
<tr>
<td>eat</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>16</td>
<td>2</td>
<td>42</td>
<td>0</td>
</tr>
<tr>
<td>chinese</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>82</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>food</td>
<td>15</td>
<td>0</td>
<td>15</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>lunch</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>spend</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
## Berkeley Restaurant Project

<table>
<thead>
<tr>
<th></th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>2533</td>
<td>927</td>
<td>2417</td>
<td>746</td>
<td>158</td>
<td>1093</td>
<td>341</td>
<td>278</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>0.002</td>
<td>0.33</td>
<td>0</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.00079</td>
</tr>
<tr>
<td>want</td>
<td>0.0022</td>
<td>0</td>
<td>0.66</td>
<td>0.0011</td>
<td>0.0065</td>
<td>0</td>
<td>0.0054</td>
<td>0.0011</td>
</tr>
<tr>
<td>to</td>
<td>0.00083</td>
<td>0</td>
<td>0.0017</td>
<td>0.28</td>
<td>0.00083</td>
<td>0</td>
<td>0.0025</td>
<td>0.087</td>
</tr>
<tr>
<td>eat</td>
<td>0</td>
<td>0</td>
<td>0.0027</td>
<td>0</td>
<td>0.021</td>
<td>0.0027</td>
<td>0.52</td>
<td>0</td>
</tr>
<tr>
<td>chinese</td>
<td>0.0063</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0027</td>
<td>0.52</td>
<td>0</td>
</tr>
<tr>
<td>food</td>
<td>0.014</td>
<td>0</td>
<td>0.014</td>
<td>0</td>
<td>0.00092</td>
<td>0.0037</td>
<td>0.0029</td>
<td>0</td>
</tr>
<tr>
<td>lunch</td>
<td>0.0059</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.0037</td>
<td>0.0029</td>
<td>0</td>
</tr>
<tr>
<td>spend</td>
<td>0.0036</td>
<td>0</td>
<td>0.0036</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**N-Gram Language Models**

\[ p(w_i | w_1, \ldots, w_{i-1}) = p(w_i | w_{i-n}, \ldots, w_{i-1}) \]

\[ n = 3 \quad \text{Trigram} \]

\[ = 4 \quad \text{Quadgram} \]

“The computer which I had just put into the dining room on the fifth floor **crashed.**”

“The computer which I had just put into the dining room on the fifth floor **had lunch.**”
Shakespeare

<table>
<thead>
<tr>
<th>Unigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>To him swallowed confess hear both. Which. Of save on trail for are ay device and rote life have Every enter now severally so, let Hill he late speaks; or! a more to leg less first you enter Are where exeunt and sighs have rise excellency took of.. Sleep knave we. near; vile like</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Bigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>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?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sweet prince, Falstaff shall die. Harry of Monmouth’s grave. This shall forbid it should be branded, if renown made it empty. Indeed the duke; and had a very good friend. Fly, and will rid me these news of price. Therefore the sadness of parting, as they say, ’tis done.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Quadrigram</th>
</tr>
</thead>
<tbody>
<tr>
<td>King Henry. What! I will go seek the traitor Gloucester. Exeunt some of the watch. A great banquet serv’d in; Will you not tell me who I am? It cannot be but so. Indeed the short and the long. Marry, ’tis a noble Lepidus.</td>
</tr>
<tr>
<td>Unigram</td>
</tr>
<tr>
<td>---------</td>
</tr>
<tr>
<td>Bigram</td>
</tr>
<tr>
<td>Trigram</td>
</tr>
</tbody>
</table>
Implementation Tips

Use Logs

- Prevent underflow
- Sums, instead of products

\[
\log p(w_i | w_{i-1}, w_{i-2}) \\
\prod_i p(w_i) \Rightarrow \int \log p(w_i)
\]

Filter out n-grams

- Rare n-grams are noisy/have low prob
- Use unigrams to filter bigrams...

\[
\text{count} > Y = 1, 2 \\
\text{egg soup} > Y \\
\text{egg} > Y \\
\text{soup} > Y
\]
Zero Probability Problem

Training set:
  ... denied the allegations
  ... denied the reports
  ... denied the claims
  ... denied the request

P(“offer” | denied the) = 0

Test set
  ... denied the offer
  ... denied the loan

Rare words/combinations
  • Because corpus is finite..

Mispellings
  • “minuets”

New words
  • Truthiness
  • #letalonethehashtag
  • bigly
Laplace Smoothing

\[ p(w_i | w_{i-1}) = \frac{\#(w_{i-1}, w_i) + 1}{\#(w_{i-1}) + \lambda \cdot \#|V|} \]

Add $\lambda$ smoothing

\[ \frac{\#(w_{i-1}, w_i) + \lambda}{\#(w_{i-1}) + \lambda \cdot \#|V|} \]
Intuition Behind Smoothing

When we have sparse statistics:

\[ P(w \mid \text{denied the}) \]
- 3 allegations
- 2 reports
- 1 claims
- 1 request
- 7 total

Steal probability mass to generalize better

\[ P(w \mid \text{denied the}) \]
- 2.5 allegations
- 1.5 reports
- 0.5 claims
- 0.5 request
- 2 other
- 7 total
## Berkeley Restaurant Project

<table>
<thead>
<tr>
<th>i</th>
<th>want</th>
<th>to</th>
<th>eat</th>
<th>chinese</th>
<th>food</th>
<th>lunch</th>
<th>spend</th>
</tr>
</thead>
<tbody>
<tr>
<td>i</td>
<td>6</td>
<td>1</td>
<td>10</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
<tr>
<td>want</td>
<td>3</td>
<td>1</td>
<td>609</td>
<td>2</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>to</td>
<td>3</td>
<td>1</td>
<td>5</td>
<td>687</td>
<td>3</td>
<td>1</td>
<td>7</td>
</tr>
<tr>
<td>eat</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>1</td>
<td>17</td>
<td>3</td>
<td>43</td>
</tr>
<tr>
<td>chinese</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>83</td>
<td>2</td>
</tr>
<tr>
<td>food</td>
<td>16</td>
<td>1</td>
<td>16</td>
<td>1</td>
<td>2</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>lunch</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>spend</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>i</td>
<td>want</td>
<td>to</td>
<td>eat</td>
<td>chinese</td>
<td>food</td>
<td>lunch</td>
</tr>
<tr>
<td>-------</td>
<td>--------</td>
<td>-------</td>
<td>--------</td>
<td>--------</td>
<td>---------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>i</td>
<td>0.0015</td>
<td>0.21</td>
<td>0.00025</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0.0025</td>
<td>0.0025</td>
</tr>
<tr>
<td>want</td>
<td>0.0013</td>
<td>0.00042</td>
<td>0.26</td>
<td>0.00084</td>
<td>0.0029</td>
<td>0.0029</td>
<td>0.0025</td>
</tr>
<tr>
<td>to</td>
<td>0.00078</td>
<td>0.00026</td>
<td>0.0013</td>
<td>0.18</td>
<td>0.00078</td>
<td>0.00026</td>
<td>0.0018</td>
</tr>
<tr>
<td>eat</td>
<td>0.00046</td>
<td>0.00046</td>
<td>0.0014</td>
<td>0.00046</td>
<td>0.0078</td>
<td>0.0014</td>
<td>0.02</td>
</tr>
<tr>
<td>chinese</td>
<td>0.0012</td>
<td>0.00062</td>
<td>0.00062</td>
<td>0.00062</td>
<td>0.0062</td>
<td>0.052</td>
<td>0.0012</td>
</tr>
<tr>
<td>food</td>
<td>0.0063</td>
<td>0.00039</td>
<td>0.0063</td>
<td>0.00039</td>
<td>0.0079</td>
<td>0.002</td>
<td>0.00039</td>
</tr>
<tr>
<td>lunch</td>
<td>0.0017</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.00056</td>
<td>0.0011</td>
<td>0.00056</td>
</tr>
<tr>
<td>spend</td>
<td>0.0012</td>
<td>0.00058</td>
<td>0.0012</td>
<td>0.00058</td>
<td>0.00058</td>
<td>0.00058</td>
<td>0.00058</td>
</tr>
</tbody>
</table>
Backoff and Interpolation

**Backoff**

- Use trigram, unless rare
- Then use bigram, unless rare
- Then use unigram..

**Interpolation**

- Combine all three!
- Linear function with parameters
- Learn on held out data

\[
p(w_i|w_{i-2}w_{i-1}) = \begin{cases} 
  p(w_i|w_{i-2}w_{i-1}) & \text{if } "w_{i-2}w_{i-1}w_i" > 0 \\
  p(w_i|w_{i-1}) & \text{if } "w_{i-1}w_i" > 0 \\
  p(w_i) & \text{if } w_i > 0 \\
\end{cases}
\]

\[
p(w_i|w_{i-2}w_{i-1}) = \lambda_1 p(w_i|w_{i-2}w_{i-1}) + \lambda_2 p(w_i|w_{i-1}) + \lambda_3 p(w_i).
\]

\[\sum \lambda = 1, \text{ context}\]
Upcoming...

**Homework**
- Homework 1 is due: **January 26, 2017**
- Write-up, data, and code for Homework 2 is up
- Homework 2 is due: **February 9, 2017**

**Project**
- Proposal is due: **February 7, 2017** (~2 weeks)
- Make things more concrete: approach, metrics, baselines
- Mention progress, and address my concerns, if any
- Only 2 pages