Natural Language Processing

Lecture 5: Language Models and Smoothing

Language Modeling

- Is this sentences good?
  - This is a pen
  - Pen this is a
- Help choose between options, help score options
  - 他向记者介绍了发言的主要内容
  - He briefed to reporters on the chief contents of the statement
  - He briefed reporters on the chief contents of the statement
  - He briefed to reporters on the main contents of the statement
  - He briefed reporters on the main contents of the statement

One-Slide Review of Probability Terminology

- Random variables take different values, depending on chance.
- Notation:

$p(X = x)$ is the probability that r.v. X takes value x
$p(x)$ is shorthand for the same
$p(X)$ is the distribution over values X can take (a function)

- Joint probability: $p(X = x, Y = y)$
  - Independence
  - Chain rule
- Conditional probability: $p(X = x \mid Y = y)$

Unigram Model

- Every word in Σ is assigned some probability.
- Random variables W_1, W_2, … (one per word).

$$
\begin{aligned}
p(\boldsymbol{W} = \boldsymbol{w}) &= p(W_1 = w_1, W_2 = w_2, \ldots, W_{L+1} = stop) \\
&= \left(\prod_{\ell=1}^{L} p(W_\ell = w_\ell) \right) p(W_{L+1} = stop) \\
&= \left(\prod_{\ell=1}^{L} p(w_\ell) \right) p(stop)
\end{aligned}
$$

Part of A Unigram Distribution

[rank 1]
p(the) = 0.038
p(of) = 0.023
p(and) = 0.021
p(to) = 0.017
p(is) = 0.013
p(a) = 0.012
p(in) = 0.012
p(for) = 0.009
…

…
[rank 1001]
p(joint) = 0.00014
p(relatively) = 0.00014
p(plot) = 0.00014
p(DEL1SUBSEQ) = 0.00014
p(rule) = 0.00014
p(62.0) = 0.00014
p(9.1) = 0.00014
p(evaluated) = 0.00014
…

Unigram Model as a Generator

first, from less the This different 2004), out which goal 19.2 Model their It ~(i?1), given 0.62 these (x0; match 1 schedule. x 60 1998. under by Notice we of stated CFG 120 be 100 a location accuracy If models note 21.8 each 0 WP that the that Nov?ak. to function; to [0, to different values, model 65 cases. said - 24.94 sentences not that 2 In to clustering each K&M 100 Boldface X))] applied; In 104 S. grammar was (Section contrastive thesis, the machines table -5.66 trials: An the textual (family applications. We have for models 40.1 no 156 expected are neighborhood

Full History Model

- Every word in Σ is assigned some probability, *conditioned on every history*.

$$
\begin{aligned}
p(\boldsymbol{W} = \boldsymbol{w}) &= p(W_1 = w_1, W_2 = w_2, \ldots, W_{L+1} = stop) \\
&= \left(\prod_{\ell=1}^{L} p(W_\ell = w_\ell \mid \boldsymbol{W}_{1:\ell-1} = \boldsymbol{w}_{1:\ell-1}) \right) p(W_{L+1} = stop \mid \boldsymbol{W}_{1:L} = \boldsymbol{w}_{1:L}) \\
&= \left(\prod_{\ell=1}^{L} p(w_\ell \mid history_\ell) \right) p(stop \mid history_L)
\end{aligned}
$$

Bill Clinton's unusually direct comment Wednesday on the possible role of race in the election was in keeping with the Clintons' bid to portray Obama, who is aiming to become the first black U.S. president, as the clear favorite, thereby lessening the potential fallout if Hillary Clinton does not win in South Carolina.

N-Gram Model

- Every word in Σ is assigned some probability, conditioned on a *fixed-length* history ($n - 1$).

$$
\begin{aligned}
p(\boldsymbol{W} = \boldsymbol{w}) &= p(W_1 = w_1, W_2 = w_2, \ldots, W_{L+1} = stop) \\
&= \left(\prod_{\ell=1}^{L} p(W_\ell = w_\ell \mid \boldsymbol{W}_{\ell-n+1:\ell-1} = \boldsymbol{w}_{\ell-n+1:\ell-1}) \right) \\
&\quad \times p(W_{L+1} = stop \mid \boldsymbol{W}_{L-n+1:L} = \boldsymbol{w}_{L-n+1:L}) \\
&= \left(\prod_{\ell=1}^{L} p(w_\ell \mid history_\ell) \right) p(stop \mid history_{L+1})
\end{aligned}
$$

Bigram Model as a Generator

e. (A.33) (A.34) A.5 ModelS are also been completely surpassed in performance on drafts of online algorithms can achieve far more so while substantially improved using CE. 4.4.1 MLEasaCaseofCE 71 26.34 23.1 57.8 K&M 42.4 62.7 40.9 44 43 90.7 100.0 100.0 100.0 15.1 30.9 18.0 21.2 60.1 undirected evaluations directed DEL1 TRANS1 neighborhood. This continues, with supervised init., semisupervised MLE with the METU- SabanciTreebank 195 ADJA ADJD ADV APPR APPRART APPO APZR ART CARD FM ITJ KOUI KOUS KON KOKOM NN NN NN IN JJ NNTheir problem is y x. The evaluation offers the hypothesized link grammar with a Gaussian

Trigram Model as a Generator

top(xI ,right,B). (A.39) vine0(X, I) rconstit0(I 1, I). (A.40) vine(n). (A.41) These equations were presented in both cases; these scores u<AC>into a probability distribution is even smaller(r =0.05). This is exactly fEM. During DA, is gradually relaxed. This approach could be efficiently used in previous chapters) before training (test) K&MZeroLocalrandom models Figure4.12: Directed accuracy on all six languages. Importantly, these papers achieved state- of-the-art results on their tasks and unlabeled data and the verbs are allowed (for instance) to select the cardinality of discrete structures, like matchings on weighted graphs (McDonald et al., 1993) (35 tag types, 3.39 bits). The Bulgarian,

What’s in a word

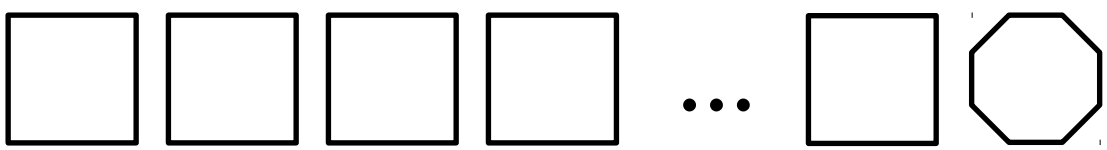
- Is punctuation a word?
  - Does knowing the last “word” is a “,” help?
- In speech
  - I do uh main- mainly business processing
  - Is “uh” a word?

For Thought

- Do N-Gram models “know” English?
- Unknown words
- N-gram models and finite-state automata

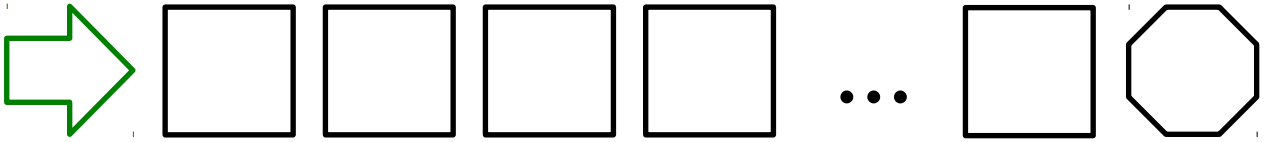
Starting and Stopping

Unigram model:



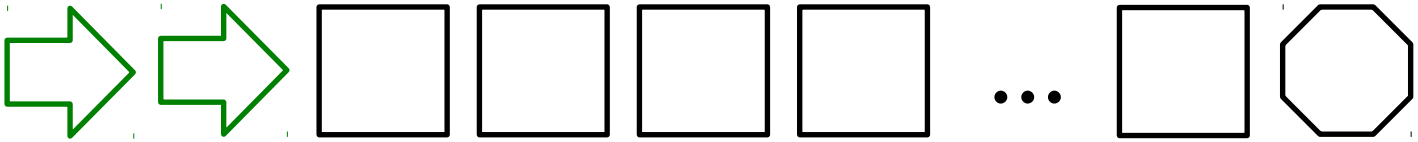
$$\prod_{i=1}^{L+1} p(w_i)$$

Bigram model:



$$\prod_{i=1}^{L+1} p(w_i \mid w_{i-1})$$

Trigram model:



$$\prod_{i=1}^{L+1} p(w_i \mid w_{i-2}, w_{i-1})$$

Evaluation

Which model is better?

- Can I get a number about how good my model is for a test set?
- What is the P(test_set | Model)
- We measure this by Perplexity
- Perplexity is the probability of test set normalized by the number of words

Perplexity

$$
\begin{aligned}
\text{perplexity}(p(\cdot); \boldsymbol{w}) &= 2^{\left(-\frac{\log_2 p(\boldsymbol{w})}{|\boldsymbol{w}|}\right)} \\
&= p(\boldsymbol{w})^{-\frac{1}{|\boldsymbol{w}|}} \\
&= \sqrt[|\boldsymbol{w}|]{\frac{1}{p(\boldsymbol{w})}} \\
&= \sqrt[|\boldsymbol{w}|]{\frac{1}{\prod_{i=1}^{|\boldsymbol{w}|} p(w_i \mid w_{i-N+1}, \ldots, w_{i-1})}}
\end{aligned}
$$

Perplexity of different models

- Better models have lower perplexity
  - WSJ: Unigram 962; Bigram 170; Trigram 109
- Different tasks have different perplexity
  - WSJ (109) vs Bus Information Queries (~25)
- Higher the conditional probability,
  lower the perplexity
- Perplexity is the average branching rate

What about open class

- What is the probability of unseen words?
  - (Naïve answer is 0.0)
- But that's not what you want
  - Test set will usually include words not in training
- What is the probability of
  - P(Nebuchadnezzur | son of)

LM smoothing

- Laplace or add-one smoothing
  - Add one to all counts
  - Or add “epsilon” to all counts
  - You still need to know all your vocabulary
- Have an OOV word in your vocabulary
  - The probability of seeing an unseen word

Good-Turing Smoothing

- Good (1953) From Turing.
  - Using the count of things you've seen once to estimate count of things you've never seen.
- Calculate the frequency of frequencies of Ngrams
  - Count of Ngrams that appear 1 times
  - Count of Ngrams that appear 2 times
  - Count of Ngrams that appear 3 times
  - …
  - Estimate new c = (c+1) (N_c + 1)/N_c)
- Change the counts a little so we get a better estimate for count 0

Good-Turing's Discounted Counts

	AP Newswire Bigrams		Berkeley Restaurants Bigrams		Smith Thesis Bigrams	
c	N_c	c^*	N_c	c^*	N_c	c^*
0	74,671,100,000	0.0000270	2,081,496	0.002553	*x*	38,048 / *x*
1	2,018,046	0.446	5,315	0.533960	38,048	0.21147
2	449,721	1.26	1,419	1.357294	4,032	1.05071
3	188,933	2.24	642	2.373832	1,409	2.12633
4	105,668	3.24	381	4.081365	749	2.63685
5	68,379	4.22	311	3.781350	395	3.91899
6	48,190	5.19	196	4.500000	258	4.42248

$$c^* = \frac{(c+1) \times N_{c+1}}{N_c}$$

Backoff

- If no trigram, use bigram
- If no bigram, use unigram
- If no unigram … smooth the unigrams

Estimating $p(w \mid \text{history})$

- Relative frequencies (count & normalize)
- Transform the counts:
  - Laplace/“add one”/“add λ”
  - Good-Turing discounting
- Interpolate or “backoff”:
  - With Good-Turing discounting: Katz backoff
  - “Stupid” backoff
  - Absolute discounting: Kneser-Ney