Statistical NLP 2019 - Assignment 1

Consider n-gram language modeling:

$$p(w_i \mid w_1 \ldots w_{i-1}) \approx p(w_i \mid w_{i-n+1} \ldots w_{i-1}) = p(w_i \mid w_{i-n+1}^{i-1})$$

let V denote the Vocabulary size (typically around a million), n the n-gram order (typically around 5), and C the number of tokens in the corpus used to compute the relevant counts (typically around 1 billion).

1. Characterize the memory complexity of Kneser-Ney language models in big-O notation. You should decide what variables are important to model. (Hint: think about how to store the relevant counts efficiently)

Solution: The key is to find a way to store the count of all (w', w) pairs for all k-grams for $k = 1, 2, \ldots, n$, where w' is the history and w is the next word.

Consider storing the relevant counts in a matrix for a specific k-gram, the number of different histories (number of rows) is bounded by: $\min(V^k, O(C))$, because each token can take on V possible values, and the corpus length is C. Since V is around a million, which is 1×10^6, C is around a billion, which is 1×10^9, we have

$$\min(V^k, d(C)) = \begin{cases} V^k, & \text{if } k=1, \\ d(C), & \text{otherwise} \end{cases}$$

The number of different words (number of columns) is the vocabulary size, which is V.

Using a dense matrix representation, the space complexity of the matrix for a specific k-gram is $O(VC)$ for $k=2,\cdots,n$, and $O(V^2)$ for $k=1$.

Thus the total space complexity is:

$$(n-1)\,O(VC) + O(V^2) = O(nVC)$$

However, $O(VC)$ is around 1×10^{15}, so using the dense matrix representation may not work.

Instead, consider the sparse matrix representation, the space complexity of the matrix for a specific k-gram is the number of non-empty entries in the corresponding dense matrix, which is $O(C)$ for $k=1,2,\cdots,n$.

Thus the total space complexity is:

$$n\,O(C) = O(nC),$$

which is around $n\times10^9$ for n around 5.

The sparse matrix representation might be feasible.

2. The term inference time refers to one call to the language model, i.e., computing $P(w_i \mid w_{i-n+1}^{i-1})$ under the language model for one choice of w_{i-n+1}^{i-1}, w_i. Characterize the inference time complexity of Kneser-Ney language models in big-O notation. (Hint: Think about which quantities can be cached efficiently to speed up inference time).

Solution: The recursive equation for computing P_{KN} is:

$$P_{KN}(w_i \mid w_{i-n+1}^{i-1}) = \frac{\max(C_{KN}(w_{i-n+1}^{i}) - d, 0)}{\sum_v C_{KN}(w_{i-n+1}^{i-1} v)} + \lambda(w_{i-n+1}^{i-1}) P_{KN}(w_i \mid w_{i-n+2}^{i-1})$$

where $C_{KN}(w') = \begin{cases} \text{Count}(w'), & \text{for } w' = w_{i-n+1}^{i}, \\ \text{Continuation count}(w'), & \text{otherwise} \end{cases}$

Continuation count (w') = the number of unique single word contexts for w'.

$$\lambda(w_{i-n+1}^{i-1}) = \frac{d}{\sum_v C(w_{i-n+1}^{i-1} v)} \left|\{w : C(w_{i-n+1}^{i-1} w) > 0\}\right|$$

We can cache the following quantities in advance:

① For each (n, i) pair, store $\sum_v C_{KN}(w_{i-n+1}^{i-1} v)$. This requires $O(nC)$ space.

② $\left|\{w : C(w_{i-n+1}^{i-1} w) > 0\}\right|$ for each (n, i) pair. This requires $O(nC)$ space.

We cannot store each (W_{i-n+1}^{i-1}, w^i) pair since this requires $O(VC)$ space. Instead, to compute $C_{KN}(W_{i-n+1}^{i}) = C_{KN}(W_{i-n+1}^{i-1} w^i)$, we search in the row indexed by W_{i-n+1}^{i-1} for the column w^i in the corresponding n-gram count matrix. Indexing into an element in the sparse matrix takes time proportional to the logarithm of the length of its columns, which is $O(\log V)$, usually around $\log_2(1\times 1$ ≈ 20. This overhead is acceptable.

To sum up, if we cache $\sum_v C_{KN}(W_{i-n+1}^{i-1} v)$ and $|\{w : C(W_{i-n+1}^{i-1} w) > 0\}|$ for each (n, i) pair in advance, requiring $O(nc)$ space overhead, we can compute $\sum_v C_{KN}(W_{i-n+1}^{i-1} v)$ and $\lambda(W_{i-n+1}^{i-1})$ in $O(1)$ time. We can also compute $\max(C_{KN}(W_{i-n+1}^{i}) - d, 0)$ in $O(\log v)$ time.

Thus we have the recurrence:

$$T(n) = O(\log v) + T(n-1)$$

which gives $T(n) = O(n \log v)$

So the time complexity for inference time is $O(n \log v)$

3. Recall absolute discounting for $n=2$:

$$P_{ad}(w_i \mid w_{i-1}) = \frac{\max(C(w_{i-1}, w_i) - d, 0)}{\sum_w C(w_{i-1}, w)} + \alpha(w_{i-1})\,\hat{p}(w_i)$$

where $\hat{p}(w_i) = \frac{C(w_i)}{\sum_w C(w)}$ is the empirical unigram dist'n,

d is the constant and α is the left-over weight assigned to the lower order distribution:

$$\alpha(w_{i-1}) = \frac{d \times |\{w_i : C(w_{i-1}, w_i) > 0\}|}{\sum_w C(w_{i-1}, w)}$$

For $n=2$ (a bigram model), give an example of a small vocabulary V and corpus C where absolute discounting does not preserve the marginal constraint, i.e.,

$$\hat{p}(w_i) \neq \sum_{w_{i-1}} P_{ad}(w_i \mid w_{i-1}) \cdot \hat{p}(w_{i-1}) \qquad (1)$$

Solution:

Corpus: <s> I am A I am B </s>

Vocabulary: {<s>, I, am, A, B, </s>}

$$\hat{p}(I) = \frac{C(I)}{\sum_w C(w)} = \frac{2}{8} = \frac{1}{4}$$

let $\tilde{p}(I) = \sum_{w_{i-1}} P_{ad}(I \mid w_{i-1}) \cdot \hat{p}(w_{i-1})$

$$= P_{ad}(I \mid \langle s\rangle) \cdot \hat{p}(\langle s\rangle) + P_{ad}(I \mid A) \cdot \hat{p}(A)$$

$$= \left(\frac{1-d}{1} + \frac{d \times 1}{1} \times \frac{1}{4}\right) \times \frac{1}{8} \times 2$$

$$= \frac{1}{4} - \frac{9}{16} d$$

We have $\frac{1}{4} = \hat{p}(I) \neq \tilde{p}(I) = \frac{1}{4} - \frac{9}{16} d$ for $0 < d < 1$