Beam Search

Shahrzad Kiani and Zihao Chen

CSC2547 Presentation
Beam Search

Greedy Search: Always go to **top 1** scored sequence (seq2seq)

Beam Search: Maintain the **top K** scored sequences (this paper)
Seq2Seq Train and Test Issues

gold sequence $y_{1:t} = [y_1, ..., y_t]$
predicted sequence $\hat{y}_{1:t} = [\hat{y}_1, ..., \hat{y}_t]$

Word level
- $p_{train}(\hat{y}_t | y_{1:t-1}) = \text{Softmax}(\text{decoder}(y_{1:t-1}))$
- $p_{test}(\hat{y}_t | \hat{y}_{1:t-1}) = \text{Softmax}(\text{decoder}(\hat{y}_{1:t-1}))$

Sentence level
- $p_{train}(\hat{y}_{1:t} = y_{1:t}) = \prod_{t=1}^{T} p(\hat{y}_t = y_t | y_{1:t-1})$
Seq2Seq Train and Test Issues (continued)

Training Loss

- Maximize $p_{train}(\hat{y}_{1:t} = y_{1:t}) = \prod_{t=1}^{T} p(\hat{y}_t = y_t | y_{1:t-1})$
- Minimize Negative Log Likelihood (NLL)

$$NLL = - ln \prod_{t=1}^{T} p(\hat{y}_t = y_t | y_{1:t-1}) = - \sum_t ln(p(\hat{y}_t = y_t | y_{1:t-1}))$$

Testing Evaluation

- **Sequence level** metrics like BLEU
Seq2Seq Train and Test Issues (continued)

Training Loss

- Maximize $p_{\text{train}}(\hat{y}_{1:t} = y_{1:t}) = \prod_{t=1}^{T} p(\hat{y}_t = y_t | y_{1:t-1})$
- Minimize Negative Log Likelihood (NLL)

$$\text{NLL} = -\ln \prod_{t=1}^{T} p(\hat{y}_t = y_t | y_{1:t-1}) = -\sum_{t} \ln(p(\hat{y}_t = y_t | y_{1:t-1}))$$

Testing Evaluation

- **Sequence level** metrics like BLEU
- **word level** loss

2. Loss-Evaluation Mismatch
Optimization Approach

1. **Exposure Bias**: model is not exposed at its errors at training
   - Train with beam search

2. **Loss-Evaluation Mismatch**: loss on word level, evaluation on sequence
   - Define score for sequence
   - Define search-based sequence loss
Sequence Score

• $\text{score}(\hat{y}_{1:T}) = \text{decoder}(t)$

• Hard constraint $\text{score}(\hat{y}_{1:t}) = -\infty$

Constrained Beam Search Optimization (ConBSO)

• Sequence with K-th ranked score $\hat{y}_{1:t}^{(K)}$
Search-Based Sequence Loss

\[ \mathcal{L}(\theta) = \sum_{t} \Delta (\hat{y}^{(K)}_{1:t}) \left[ 1 + \text{score}(\hat{y}^{(K)}_{1:t}) - \text{score}(y_t) \right] \]

When \( 1 + \text{score}(\hat{y}^{(K)}_{1:t}) - \text{score}(y_t) > 0 \):

• The gold sequence \( y_{1:t} \) doesn’t have a K highest score

• Margin Violation
Search-Based Sequence Loss (continued)

\[
\mathcal{L}(\theta) = \sum_{t} \Delta \left( \hat{y}_{1:t}^{(K)} \right) \left[ 1 + \text{score}(\hat{y}_{1:t}^{(K)}) - \text{score}(y_t) \right]
\]

\[
\Delta \left( \hat{y}_{1:t}^{(K)} \right)
\]

- scaling factor of penalizing the prediction
- \( = 1 \) when margin violation; \( = 0 \) when no margin violation

Goals:

- When \( t < T \), avoid margin violation, force the gold sequence to be \textbf{top} \( K \)
- When \( t = T \), force the gold sequence to be \textbf{top} 1, so set \( K = 1 \)
Backpropagation Through Time (BPTT)

• Recall loss function:

\[ \mathcal{L}(\theta) = \sum_t \Delta \left( \hat{y}_{1:t}^{(K)} \right) \left[ 1 + \text{score}(\hat{y}_{1:t}^{(K)}) - \text{score}(y_t) \right] \]

• When margin violation, backpropagate for \( \text{score}(\hat{y}_{1:t}^{(K)}) \) and \( \text{score}(y_t) \): \( O(T) \)

• A margin violation at each time step: worst case \( O(T^2) \)
Learning as Search Optimization (LaSO)

- Normal case: update beam with $\hat{y}_{1:t}^{(K)}$
- Margin violation case: update beam with $y_{1:t}$ instead

Each incorrect sequence is an extension of the partial gold sequence

Only maintain two sequences, $O(2T) = O(T)$
# Experiment on Word Ordering

fish cat eat -> cat eat fish

## Features

- Non-exhaustive search
- Hard constraint

## Settings

- Dataset: PTB dataset
- Metrics: BLEU
- $\Delta(\hat{y}_{1:t}^K)$ scaler: 0/1

<table>
<thead>
<tr>
<th></th>
<th>Word Ordering (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$K_{te} = 1$</td>
</tr>
<tr>
<td>seq2seq</td>
<td>25.2</td>
</tr>
<tr>
<td>BSO</td>
<td>28.0</td>
</tr>
<tr>
<td>ConBSO</td>
<td><strong>28.6</strong></td>
</tr>
<tr>
<td>LSTM-LM</td>
<td>15.4</td>
</tr>
</tbody>
</table>

Table 1: Word ordering. BLEU Scores of seq2seq, BSO, constrained BSO, and a vanilla LSTM language model (from Schmaltz et al, 2016). All experiments above have $K_{tr} = 6$.  

[Image credit: Sequence-to-Sequence Learning as Beam Search Optimization, Wiseman et al., EMNLP’16]
Conclusion

Alleviate the issues of seq2seq

- Exposure Bias: Beam Search
- Loss-Evaluation Mismatch: sequence level cost function with $O(T)$ BPTT with hard constraint

A variant of seq2seq with beam search training scheme
Related Works and References

- https://guillaumegenthial.github.io/sequence-to-sequence.html
- https://medium.com/@sharaf/a-paper-a-day-2-sequence-to-sequence-learning-as-beam-search-optimization-92424b490350
- https://vimeo.com/239248437
  - Propose Sequence-to-Sequence learning with deep neural networks
  - Propose a framework for learning as search optimization, and two parameter updates with convergence theorems and bounds
  - Propose the Gumbel-Greedy Decoding, which trains a generative network to predict translation under a trained model