Some known problems when training RNNs:
- MLE is typically used for training RNNs, where Maximum Likelihood Estimation (MLE) is used.
- This surrogate function is known to suffer from certain issues:
  - Different from the test loss
  - All-or-Nothing loss
  - Teacher forcing can be exposure bias

Can we build on existing approaches?
- Existing approaches use ideas from Reinforcement Learning to tackle these problems.
- Instead, we use ideas from Structured Prediction to revisit the Learning To Search literature, introducing a new algorithm based on MLE.

Contributions:
- Link between RNNs and Learning To Search
- Introduction of a new algorithm: SEARN [3]
- Experiments on real structured prediction tasks with substantial improvement over MLE

SEARN [3]
- Integrates roll-outs in the decoder to compute the cost of every possible action at every step.
- Leverage these costs to enable better training losses.

Overview:
Learning To Search
- Structured prediction
  - Learn a mapping $f$ between inputs $X$ and structured outputs $Y$ made of interrelated parts often subject to constraints.
- Learning To Search (L2S)
  - Reduces the structured problem down to cost-sensitive classification with theoretical guarantees.

How does it work?
- A unique shared classifier makes predictions one by one, conditioned on the input and the previous tokens. This classifier is trained on an intermediate dataset.

Links between Learning To Search and RNNs
- Decomposition of structured tasks in sequential predictions conditioned on the past
- Unique shared classifier for all decisions using predecessors

SEARN
- For each action $a_t$, compute the following:
  - Compute costs with roll-outs
  - Derive a loss from the costs
  - Use the loss to take a gradient step
  - Rinse and repeat

The devil in the details [7]
- Which cost-sensitive loss? How can we scale?

Why is it better than MLE?
- Makes direct use of the evaluation metric, and leverages structured information by comparing costs, contrary to MLE.
- Global-local losses, with global information at each local cell, vs. MLE (local information) and RL approaches (global loss).

Experiments
- Full algorithm:
  - Dataset: S, M, L, XL, XX
  - Cost: 0/1, MLE
  - MLE vs. learned
  - LL: learned
  - LUCAS: learned

- Scaling approach:
  - Significant improvements over MLE on all 4 tasks.
  - The harder the task, the bigger the improvement.
  - Learned/mixed is the best strategy for roll-in/out.
  - The best performing losses (for now) are those structurally close to MLE.
  - SeaRNN does not require warm start.
  - The proposed sampling strategy works, maintaining improvements at a fraction of the cost.

Machine Translation (in progress): ISWT 14 Ger/Eng
- Depth
  - MLE[4] 17.7
  - MLE[5] 22.5
  - BSO[5] 23.8
  - MLE 22.7
  - SeaRNN (ref/mixed) 28.0

References