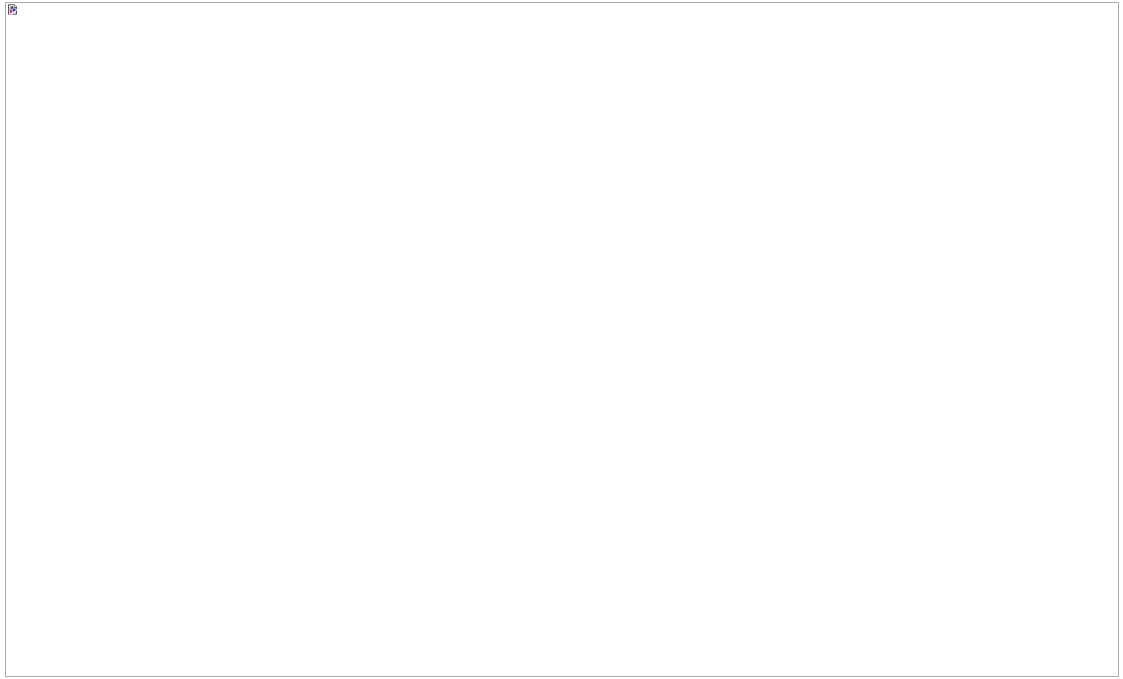
What are the main computational and statistical bottlenecks in NMT

Daniil Anastasyev, Eugene Zakharov,
Ivan Provilkov, Olga Kalinichenko, Ramil Yarullin

Generic Seq2seq

Issues:

1. Computational
    - Slow softmax computation
    - Large embeddings matrices - hard to fit in cheap video card :)
2. Statistical
    - Even very large embeddings cannot cover all words in a language



Approximating the Softmax

We perform Logistic Regression on |V| classes in the output layer: $P(w_k|c) = \frac{\exp(v_{w_k}^\top h)}{\sum_{w_j \in V} \exp(v_{w_j}^\top h)}$

Let's replace it with noise classifier: correct word vs random subset of the rest:

- E.g., Negative Sampling, Noise Contrastive Estimation or Importance Sampling
- Faster training but the same inference speed

Let's replace it with hierarchical prediction, e.g.: $P(w_k|c) = P(V_i|c) \cdot P(w_k|V_i)$

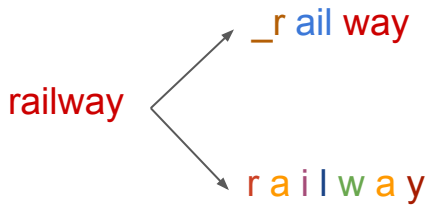
- E.g., Hierarchical Softmax, Adaptive Softmax
- Faster training but approximately same inference speed

Let's replace it with logistic regression on carefully constructed (in a hacky way) subset of V

- E.g. "Vocabulary Manipulation for Neural Machine Translation"
- Faster training and inference speed

$$P(w_k|c) = \frac{\exp(v_{w_k}^\top h)}{\sum_{w_j \in V'} \exp(v_{w_j}^\top h)}$$

Replacing words



We represented words as indices in dictionary space

Let’s replace single index with a sequence of indices

- Each word can be mapped to a sequence of subwords (or even chars)
  - Out-of-vocabulary words have embeddings similar to embeddings of words with similar spelling
- Size of subwords dictionary can be controlled
  - More subwords - shorter sequences but harder to compute softmax
  - Less subwords - harder to compute attention on long sequences
    - Can be solved by applying convolutions and pooling before attention
- Shared dictionary for source and target subwords can be used
  - Shared embeddings for source and target subwords can be used!

Summary

1. Softmax approximation
   + Faster training (and sometime inference) which allows large vocabulary and leads to + 1-2 BLEU
   - Doesn’t solve Out-of-Vocabulary Words problem
   - Model size tends to be very large
2. Words representation with sequence of subwords
   + Embedding of every word can be computed - also + 1-2 BLEU
   + Vocabulary size may be much smaller
   + Segmentation is not required in fully character-level setup
   + Subwords can have shared representations across different languages
   - The computation speed is worse
   - Model tends to be more complicated