Conventional Transformer Architecture

“Attention is all you need”, Vaswani et al. (2017)

• full self-attention: all words attend to each other
• produces great results
• but doesn’t scale to large documents!
• costs are quadratic in document length
• recurrence is cheaper, but less flexible

• central idea:
  • maybe attending only to the most important words so far is enough?
  • only keep the m most important words in memory

Memory Transformer Architecture

• reads document in chunks
• builds up memory first
• solves tasks next using the memory
  e.g. go back and label the words in all chunks
• effectively enables global attention
• only linear computational cost!
• also saves memory during inference
• outperforms baseline without memory
• outperforms baseline with simpler memory

side note:
• this approach can be taken further!
• we can apply this “learned importance-score-based sparsity” at every self-attention head in the original transformer
• giving rise to Sparse Transformer Networks, with only linearly scaling costs!

How is such a mechanism implemented and trained?

• embeddings are linearly transformed into importance scores
• we will keep only the m words with the highest importance scores
• this step is not differentiable! How can we train it via backprop?
• simple trick:
  • add the importance scores to the attention scores in transformer layers, before applying the softmax to get the attention weights
  • thus, all words are forced to attend to words in memory that were assigned a high importance
  • this way, the model learns to reduce the importance for less important words from the gradients in the attention layer
  • works well in practice!

Task: Large-scale version of SQuAD task, obtained by concatenating coming from the same Wikipedia articles
Objective: Classify words in contexts as being part of the answer or not. Max log-lik weighted by inverse class frequency
y-axis: “Pseudo F1 score”, which corresponds to F1 score with classes weighted by inverse frequency
pink: memory transformer
green: baseline without memory
orange: baseline with naive memory