Given the pre-trained weights of a Transformer language model, replacing the Transformer encoder-decoder with a Transformer decoder only model for fine-tuning on text summarization is highly sample efficient since there are no more non-pre-trained encoder-decoder attention weights.

**Fine-tuning on Text Summarization**

Encoder-Decoder with pre-trained weights

- source
- transformer encoder
- encoder-decoder attention weights (not pre-trained)
- target-so-far
- transformer decoder
- next word

Transformer Language Model with pre-trained weights

- source
- delim
- target-so-far
- transformer decoder
- next word

Simpler model, fewer parameters and no non-pre-trained attention weights

**Sample Efficiency**

- Encoder-Decoder
- Transformer LM
- Encoder-Decoder + Pretraining
- Transformer LM + Pretraining

Outputs from models fine-tuned on 1% data

**Ground truth:** A man in suburban Boston is selling snow online to customers in warmer states. For $89, he will ship 6 pounds of snow in an insulated Styrofoam box.

**Encoder-Decoder + Pre-training:** NEW: A snowfall of is forecast for New England. NEW: The Massachusetts-based company hopes to sell more than 30,000 bottles of snow. The company says it will use snow from as far as Canada.

**Transformer LM + Pre-training:** Kyle Waring will ship you 6 pounds of Boston-area snow in an insulated Styrofoam box – enough for 10 to 15 snowballs, he says. But not if you live in New England or surrounding states.

The Transformer LM fine-tuned on 1% data successfully extracts facts from the source, while the encoder-decoder hallucinates facts/generates gibberish.

**How much of the sample efficiency is due to copying?**

This Transformer LM is more abstractive (copies less) than the Pointer-Generator model, likely due to lack of the copy mechanism. A drawback of this abstractive is subtle factual inaccuracies!