Page No.:
Date:

Stanford CS224N - II

* Byte-Pair Encoding

Instead of tokenizing text into words, BPE allows the model to break down text into subwords or even characters which helps in handling out-of-vocabulary words are reduces the size of vocab req for training.

EX. 'Banana'
Here 'an' is most freq pair
Replace 'an' with 'X'
→ 'bXaXa'
Repeat, 'Xa' is most freq pair
Replace 'Xa' with 'Y'
→ 'bYY'

This way, the word 'banana' is encoded to 'bYY'

* Word structure, Subword

Common words end up in the vocab, rarer words are split into components.

In worst case, words are split into many subwords as they have characters.

* Pretraining & Fine-tuning

using / with language modelling, the model computes probability distribution over words given their contexts. We train the model to perform language modelling over large amount of text data and save the network parameters.

This model now can be fine tuned ~~over~~ for your req task. It is much more efficient and scalable over training from scratch.

* Pretraining Encoders

We cant do language modelling on encoders bcoz they get bidirectional context. Instead, we mask out certain words from the text and ask the encoder to predict, the prob distribution of masked text over the unmasked one.

* **Bidirectional Encoder Representations for Transformer**

Masked LM for BERT :

i. Predict random 15% of sub(word) tokens.
ii. Replace input word with mask 80% of the time
iii. Replace input with random token 10% of the time.
iv. Leave input unchanged & predict 10% of the time.

* **PEFT - Parameter Efficient FT**

Prefix tuning adds a prefix of parameters, and freezes all pretrained parameters. The prefix parameters are learnable to the model.

* Low Rank Adaptation

Using LoRA FT, most of the models weights are frozen and only necessary parameters are tuned according to the task.

* Pretraining Encoder Decoders

A method called span corruption is used. We mask out diff length spans from the input to encoder and task the decoder to predict the masks.

* Pretraing decoders

Decoders are naturally pretrained as language models like discussed before and then fine tuning them for generative tasks.