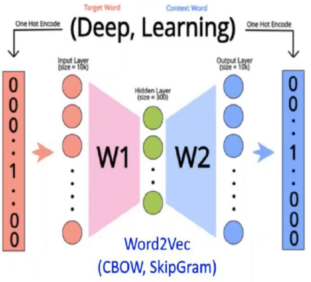
Plan of the Tutorial

Dramatic Entry of Deep Learning!



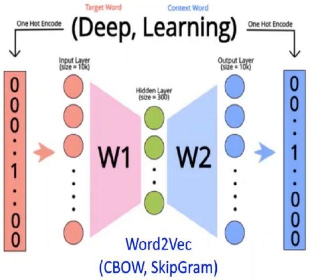
Advantages

- Unsupervised
- Can work with less data

Problems

- No shared representations at sub-word levels
- Scaling to new languages requires new embedding matrices
- Positions of tokens are overlooked

Dramatic Entry of Deep Learning!



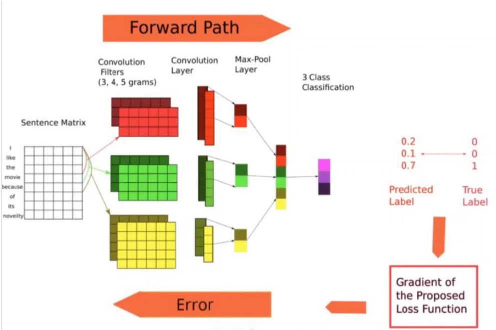
Advantages

- Unsupervised
- Can work with less data

Problems

- No shared representations at sub-word levels
- Scaling to new languages requires new embedding matrices
- Positions of tokens are overlooked

Inspiration from Computer Vision



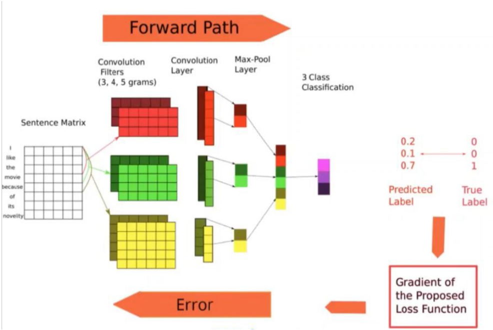
Advantages

- Captures local structures
- Performs well on the classification task
- Very fast (In GPUs)

Problems

- Can't capture long-range dependencies (words often don't need to be adjacent to be related) in POS tagging, entity extraction, etc.
- Can't capture sequential/temporal information.

Inspiration from Computer Vision



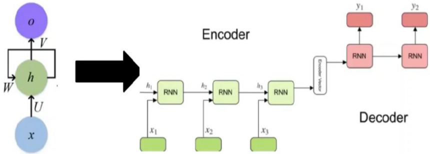
Advantages

- Captures local structures
- Performs well on the classification task
- Very fast (In GPUs)

Problems

- Can't capture long-range dependencies (words often don't need to be adjacent to be related) in POS tagging, entity extraction, etc.
- Can't capture sequential/temporal information.

Seq2Seq Model



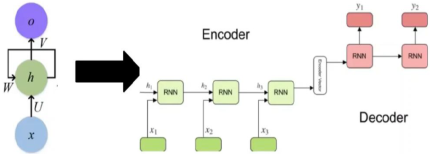
Advantages

- Captures sequential structures
- Process inputs of any length
- Even if the input sizes increases, the model size remains same
- Weights can be shared across the time steps

Problems

- Dealing with long-range dependencies
- The sequential nature of the model prevents parallelization (slow to train)

Seq2Seq Model



Advantages

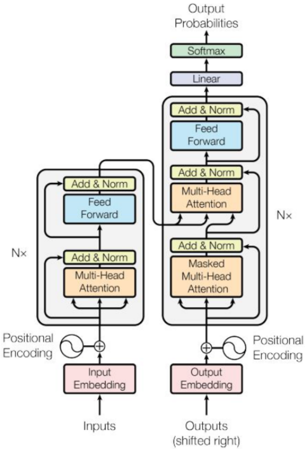
- Captures sequential structures
- Process inputs of any length
- Even if the input sizes increases, the model size remains same
- Weights can be shared across the time steps

Problems

- Dealing with long-range dependencies
- The sequential nature of the model prevents parallelization (slow to train)

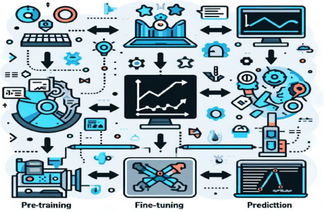
Transformer

- No Convolutions or recurrence
- Easy to Parallelize than recurrent network.
- Captures more long-range Dependencies.



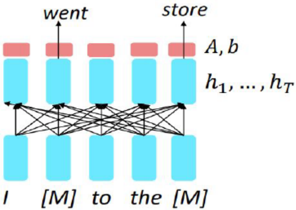
The Uprising of Language Models

- After the invention of the Transformer in the year 2017. The language model comes in the post-2017 era.
- Early 2018 era most of the language models follow the paradigm of **Pretrain** $\rightarrow$ **Finetune** $\rightarrow$ **Predict**.
- Encoder Only
- Encoder- Decoder
- Decoder Only



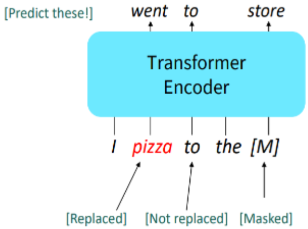
Pretraining Encoder

- Idea: replace some fraction of words in the input with a special [MASK] token; predict these words.
- Only add loss terms from words that are "masked out."



BERT: Bidirectional Encoder Representations from Transformers

- Devlin et al., 2018 proposed the "Masked LM" objective and released the weights of a pretrained Transformer, a model they labeled BERT.
- Predict a random 15% of word tokens.
    - Replace input word with [MASK] 80% of the time.
    - Replace input word with a random token 10% of the time
    - Leave input word unchanged 10% of the time
- Trained on Bookcorpus (800M words) and English Wikipedia (2500M words)

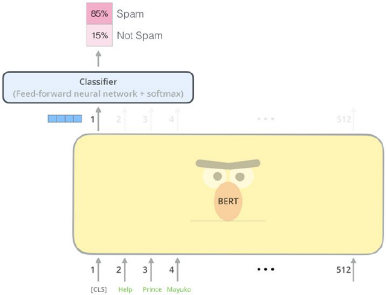


Finetuning BERT

Input

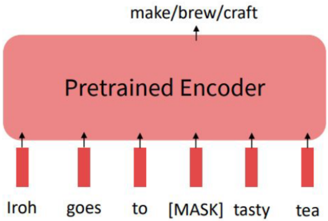
Features

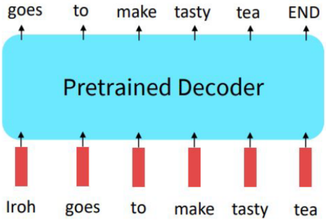
Help Prince Mayuko Transfer Huge Inheritance



Limitations of pretrained encoders

- If our tasks involves generating sequences, consider using a pretrained decoder, BERT, and other pretrained encoders don't naturally lead to nice autoregressive generation methods.





Pretraining encoder-decoders

- For encoder-decoders, we could do something like language modeling, but where a prefix of every input is provided to the encoder and is not predicted.
- This is implemented in text preprocessing: It's still an objective that looks like language modeling on the decoder side.