A Survey on Transfer Learning in NLP

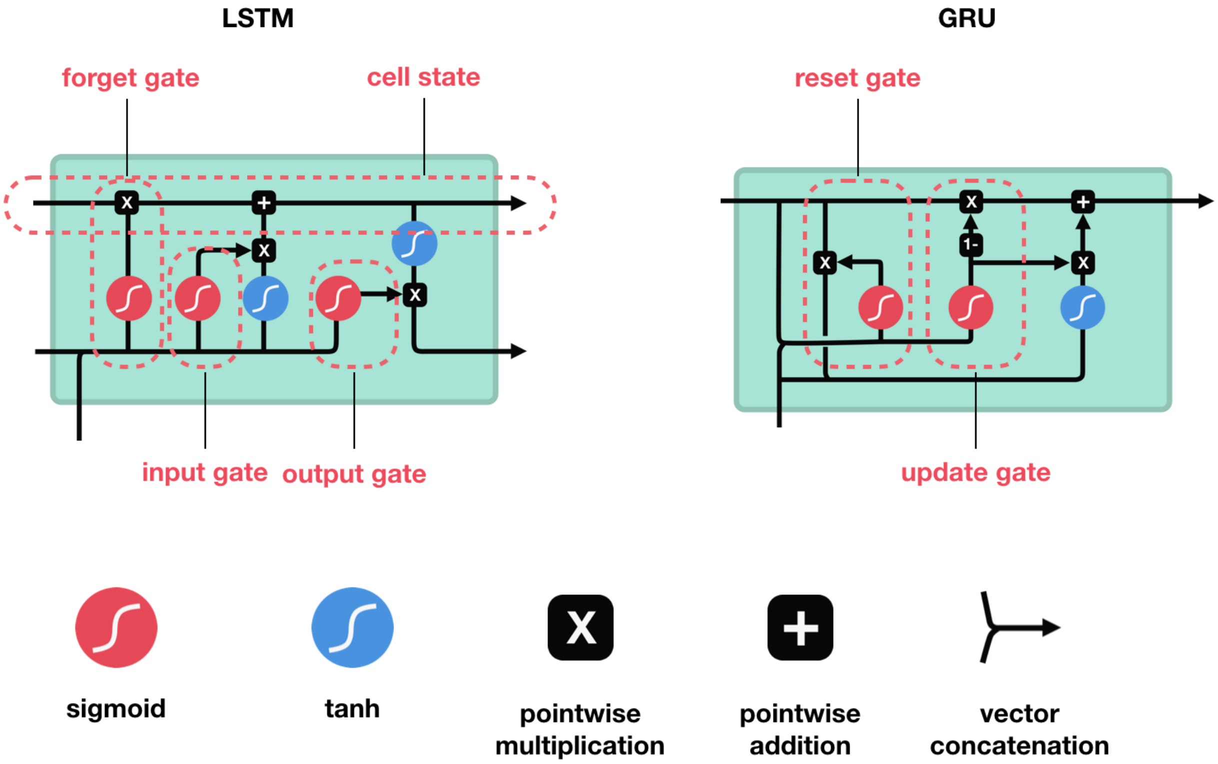
Introduction

Natural language processing had been an ongoing field of research because of its applications in many different fields and domains. In the past decade language models have evolved and broken through with better performance on standard natural language processing tasks such as question-and-answer, named entity recognition, part of speech tagging, among others. Many large language models have been trained and open sourced for public use on each individual's specific task, and transfer learning grew in popularity with the increasing democratization of high-performing pre-trained language models. This article aims to provide an overview of techniques surveyed in *A Survey on Transfer Learning in Natural Language Processing*.

Architectures

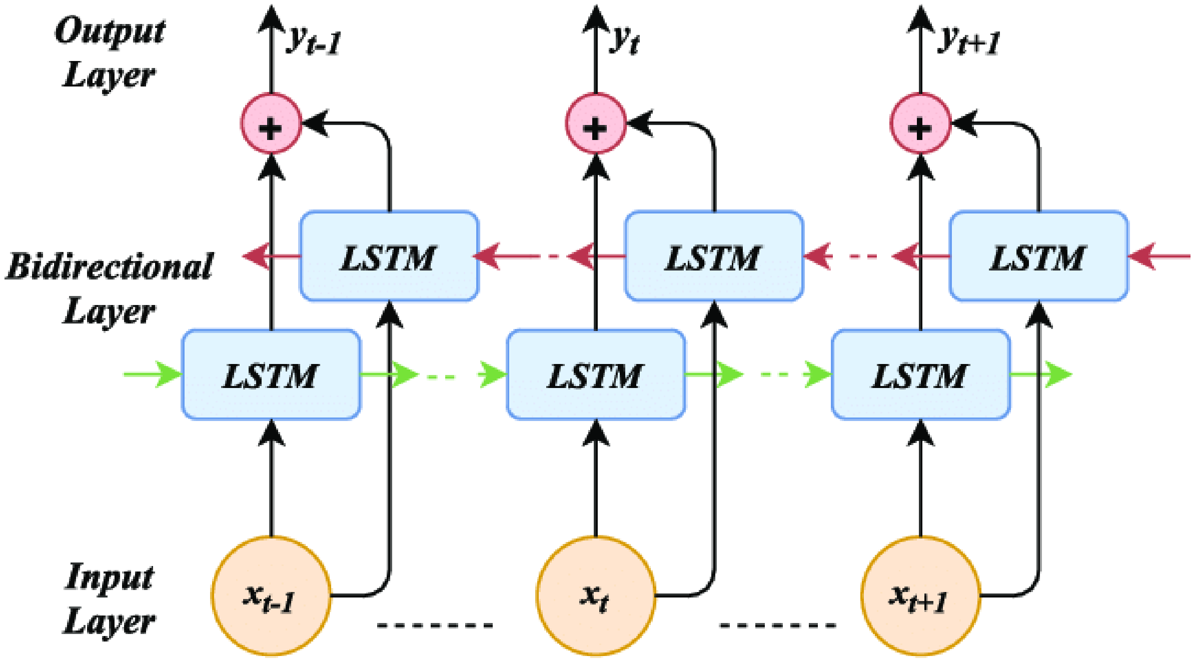
Recurrent Neural Networks

Recurrent neural networks (RNNs) are a neural network architecture which processes sequential data by passing the previous state along with each input. A main problem with RNNs is with their ability to carry information from previous steps in a long sequence. This is called the **vanishing gradient problem**, where backpropagation causes error to decay as the losses are propagated backwards through all the layers causing little to no adjustments in the network weights. Another related issue is the **exploding gradient problem**, where errors accumulate as they are propagated backwards and cause network weights to overflow. Some solutions to this problem were to use Rectified Linear Unit (ReLU) activation layers, and later the bidirectional Long Short Term Memory (LSTM) architecture. The LSTM unit consists of gates and activations which prevent both vanishing and exploding gradients and layers which can store long term information. The gates allow the LSTM unit to keep or forget information.



Credit:
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

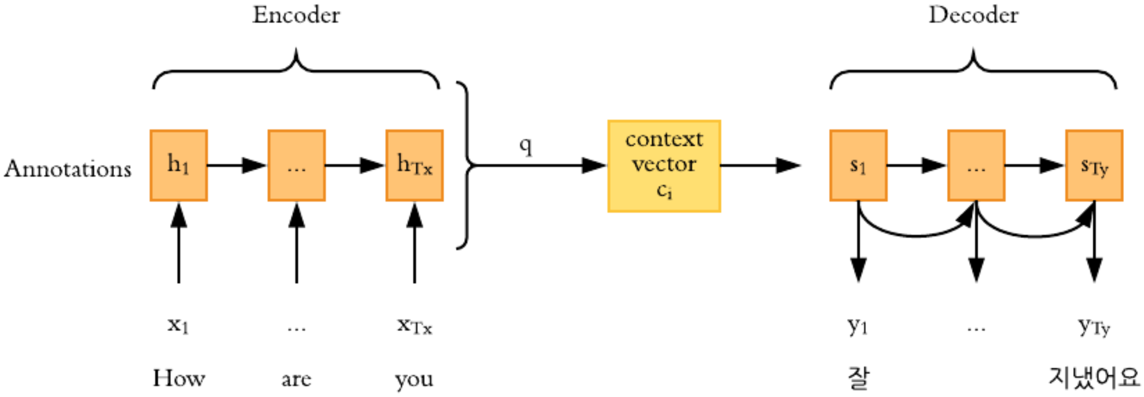
The bidirectional LSTM architecture takes advantage of chaining many LSTM units together in both a left-to-right and right-to-left direction to learn more about the natural language. This provides an advantage in being able to handle long term sequence dependencies and having the context of forward and backward directions, but is quite slow to train. This led to the development of the Gated Recurrent Unit (GRU). GRUs are a faster version of the LSTM



(Figure credits: https://www.researchgate.net/figure/Bidirectional-LSTM-model-showing-the-input-and-output-layers-The-red-arrows-represent_fig3_344554659)

Encoder-Decoder

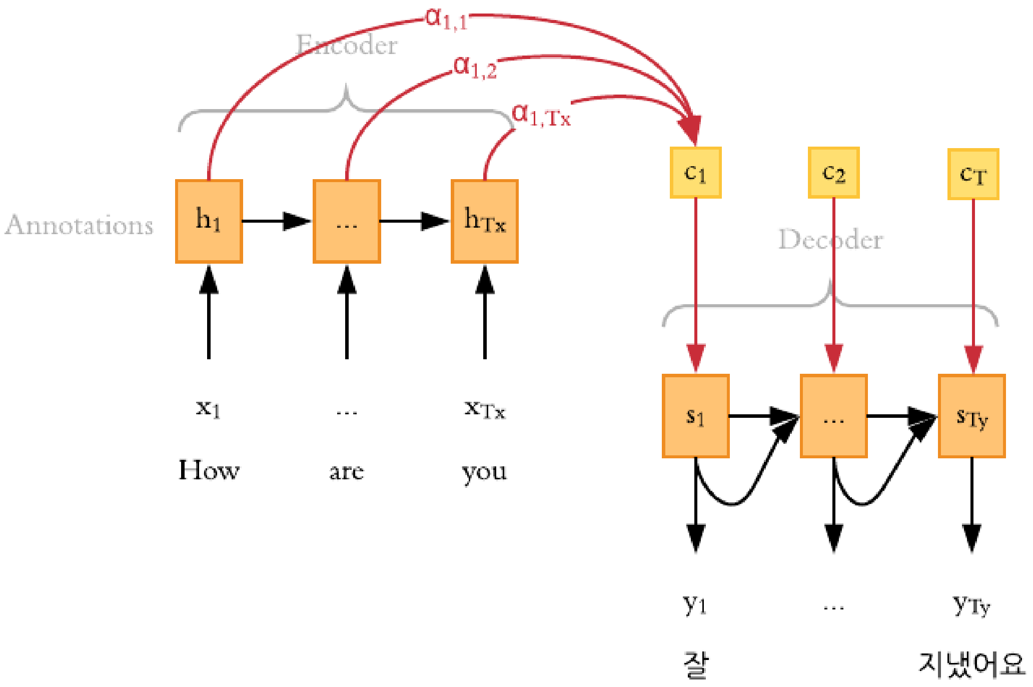
The encoder-decoder architecture is a sequence to sequence model. The encoder component encodes textual input into a context vector. The decoder component decodes the vector back into the output sequence text.



Credit: https://medium.com/@edloginova/attention-in-nlp-734c6fa9d983

Attention

In an attention-based architecture, the encoder remains the same. The decoder on the other hand attends to different parts of the source sentence at each step of the output generation. Its hidden state is computed with the context vector, the previous output and the previous hidden state.



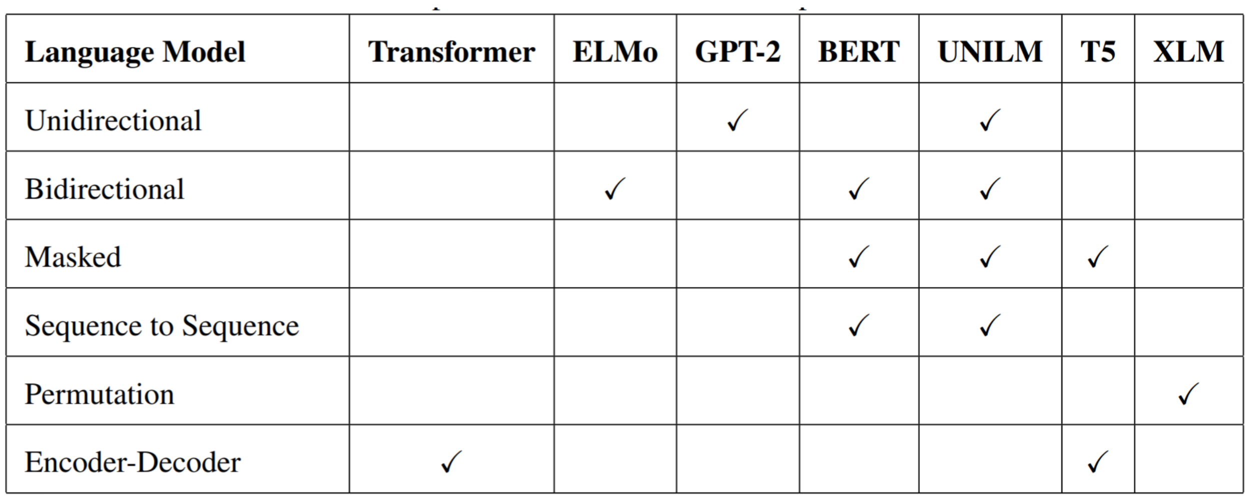
Credit: https://medium.com/@edloginova/attention-in-nlp-734c6fa9d983

Convolutional Neural Networks

Convolutional Neural Networks (CNNs) use convolutional and max-pooling layers to subsample and extract features from image data. It has been used in language modelling for a variety of tasks.

Types of Language Models

- Unidirectional Language Models: every token only uses left or right context to train
- Bidirectional Language Models: every token can use any token in the context
- Masked Language Models: in a bidirectional LSTM mask some tokens and predict them
- Sequence-to-sequence Language Models: converts sequence from one domain to sequence in another domain of different length
- Encoder-decoder Language Models: encode a sequence with an encoder and predict next sequence with decoder



Language Model	Transformer	ELMo	GPT-2	BERT	UNILM	T5	XLM
Unidirectional			✓		✓		
Bidirectional		✓		✓	✓		
Masked				✓	✓	✓	
Sequence to Sequence				✓	✓		
Permutation							✓
Encoder-Decoder	✓					✓	

Credit: https://arxiv.org/pdf/2007.04239.pdf

Transfer Learning

Now that we've discussed various types of natural language models, we can discuss how to apply these techniques to more specific tasks. Transfer learning is a commonly used technique to take preexisting pre-trained model weights which have been trained on a generic task, and transferring those same weights to continue training on a more specific dataset.

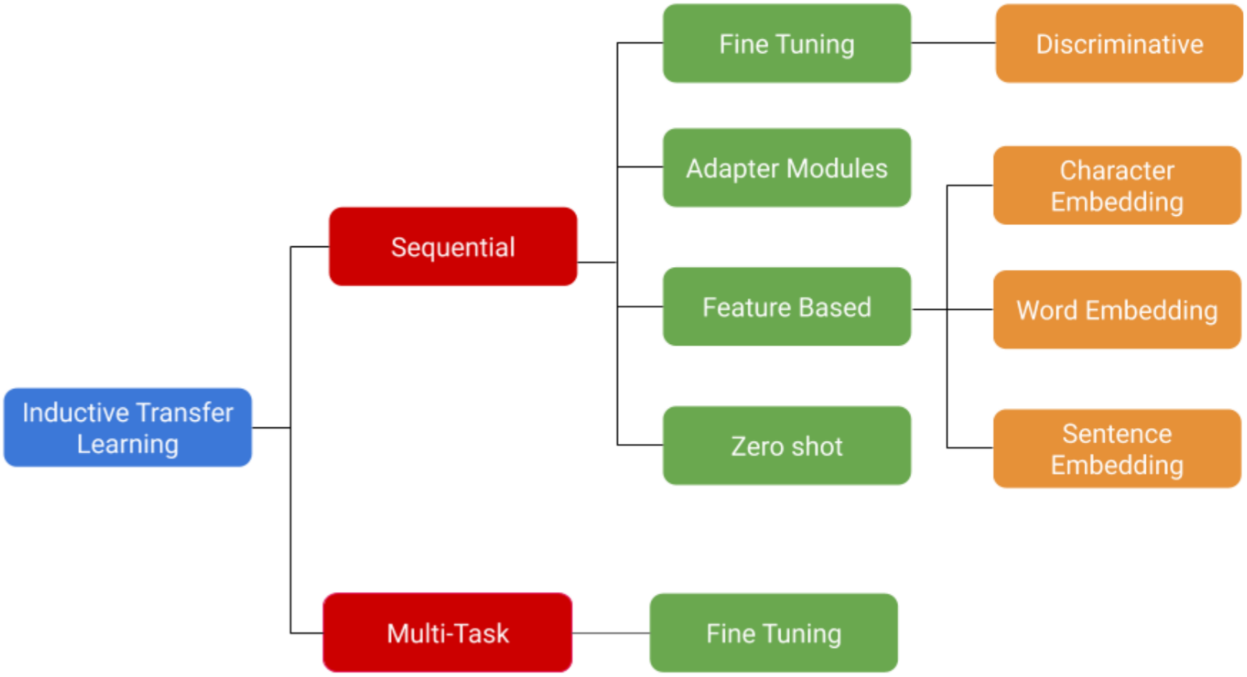
There are two types of transfer learning:

1. Transductive Transfer Learning: same task to learn, but the target domain is different from the domain trained on
2. Inductive Transfer Learning: different task to learn, but we have labelled data in the target domain

Transductive Transfer Learning

We often have the same objective that a previously trained model had, but we want to apply the model onto a different dataset. An example of this would be a sentiment analysis model that was trained on movie reviews, but we want to apply it onto restaurant reviews.

Inductive Transfer Learning



Credit: https://arxiv.org/pdf/2007.04239.pdf

Fine-tuning

A pre-trained model's weights will be reused to learn a new task. The original model's parameters at each layer may change

Feature Embeddings

We can learn vector representations for words or sentences, and use these as input to a model to learn additional weights for a downstream task.

Zero-Shot Learning

We can also apply a pre-trained model to a brand new task without any further fine-tuning or training and evaluate its performance as-is on the new task. This means we can apply a model to a problem it was not explicitly trained to do.

Language Models

Various popular language models have been learned to allow for further transfer learning and fine-tuning on natural language tasks.

ULMFiT

ULMFiT trained initially using a AWD-LSTM model on a large dataset and added a few classification layers to train for text classification tasks. It is a model meant for universal text classification fine-tuning.

BERT

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based architecture which learns bidirectional representations of words. It is trained on a masked language task and a next sentence prediction task.

ELMo

ELMo is a bidirectional LSTM model which learns deep representations of natural language.

RoBERTa

This model optimizes on top of BERT by training on longer sequences with more data after finding that BERT was undertrained. They removed the next sentence prediction task and altered the masked language task

GPT-2

The GPT model is based on the transformer architecture and can perform zero-shot learning on a variety of downstream tasks. It is typically recommended for generative tasks.

Conclusion

Recent advances in natural language processing have continuously improved the state of the art results in common natural language tasks such as question-and-answer, text summarization, translation, and text classification. From recurrent neural networks, to LSTMs, to the newer transformer architectures, improvements are being made on models to allow for improved model fine-tuning on downstream tasks.