# Transfer Learning in NLP: A Survey

Nupur Yadav

San Jose State University

Emerging Technologies in ML

# Introduction

- The limitations of deep learning models, such as requiring a large amount of data to train models and demand for huge computing resources, forces research for the knowledge transfer possibilities.
- Many large DL models are emerging that demand the need for transfer learning.

# Models used for NLP

1. Recurrent-Based Models
2. Attention-Based Models
3. CNN-Based Models

# Recurrent-Based Models

- In RNNs we pass the previous model state along with each input to make them learn the sequential context. Works great for many tasks such as speech recognition, translation, text generation, time-series classification, and biological modeling.
- Suffers from the problem of vanishing gradient because of using backpropagation and its sequential nature.
- To overcome this issue many ideas came such as using Rectified Linear Unit (ReLU) as the activation function, then Long Short-Term Memory (LSTM) architecture, bidirectional LSTMs, Gated Recurrent Networks (GRUs).
- GRUs are the fastest versions of LSTMs and can beat LSTMs in some tasks such as automatic capturing the grammatical properties of the input sentences.

# Attention-Based Models

- RNNs aggregates the sequence activations in one vector which causes the learning process to forget about words that were fed in the past.
- Attention-based models on the other hand attend each word differently to inputs based on the similarity score.
- Attention can be applied between different sequences or in the same sequence which is called self-attention.

# CNN-Based Models

- It uses convolutional and max-pooling layers for sub-sampling. Convolutional layers extract features and pooling layers reduce the spatial size of the extracted features.
- In NLP, CNNs have been successfully used for sentence classification tasks such as movie reviews, question classification, etc.
- Also used in language modeling where gated convolutional layers were used to preserve larger contexts and can be parallelized compared to traditional recurrent neural networks.

# Language Models

1. **Unidirectional LM:** Consider tokens either that are to the left of the current context or to the right.
2. **Bidirectional LM**: Each token can attend to any other token in the current context.
3. **Masked LM:** Used in bidirectional LM where we randomly mask some tokens in the current context and then predict these masked tokens.
4. **Sequence-to-sequence LM**: involves splitting the input into two separate parts. In the first part, every token can see the context of any other token in that part but in the second part, every token can only attend to tokens to the left.
5. **Permutation LM:** This language model combines the benefits of both auto-regressive and auto-encoding.
6. **Encoder-Decoder based LM:** In comparison to other approaches that use a single stack of encoder/decoder blocks, this approach uses both the blocks.

# Transfer Learning

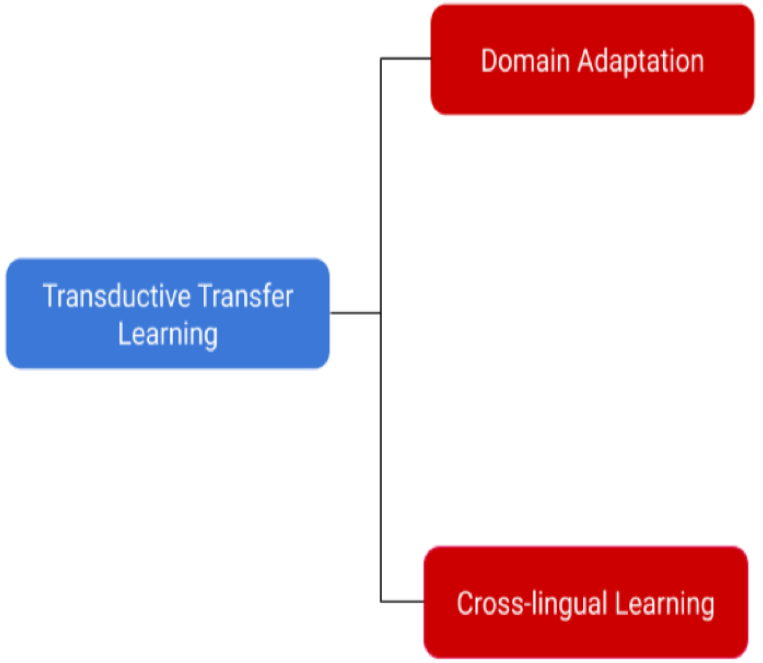
- Given a source domain-task tuple **(Ds, Ts)** and a different target domain-task tuple **(Dt , Tt )**, transfer learning can be defined as the process of using the source domain and task in the learning process of the target domain task.
- Mathematically, the objective of transfer learning is to learn the target conditional probability distribution **P(Yt|Xt)** in ***Dt*** with the information gained from ***Ds*** and where **Ds ≠Dt** or **Ts ≠ Tt**.

# Types of Transfer Learning

**Transductive Transfer Learning:** Transductive Transfer Learning is when for the same task, the target domain or task doesn't have labeled data or has very few labeled samples.

It can further be divided into the following sub-categories:

- ***Domain Adaptation***
- ***Cross-lingual transfer learning***




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

# Domain Adaptation

- Process of adapting to a new domain. It usually happens when we want to learn a different data distribution in the target domain.
- Useful if the new task to train-on has a different distribution or the amount of labeled data is scarce.
- One of the recent work involves using adversarial domain adaptation for the detection of duplicate questions. This approach had three main components: an encoder, a similarity function, and a domain adaptation module.
- The encoder encoded the question and was optimized to fool the domain classifier that the question was from the target domain.
- The similarity function calculated the probability for a pair of questions to find they were similar or duplicate.
- And the domain adaptation component was used to decrease the difference between the target and source domain distributions.
- This approach proved better and achieved an average improvement of around 5.6% over the best benchmark for different pairs of domains.

# Cross-lingual transfer learning

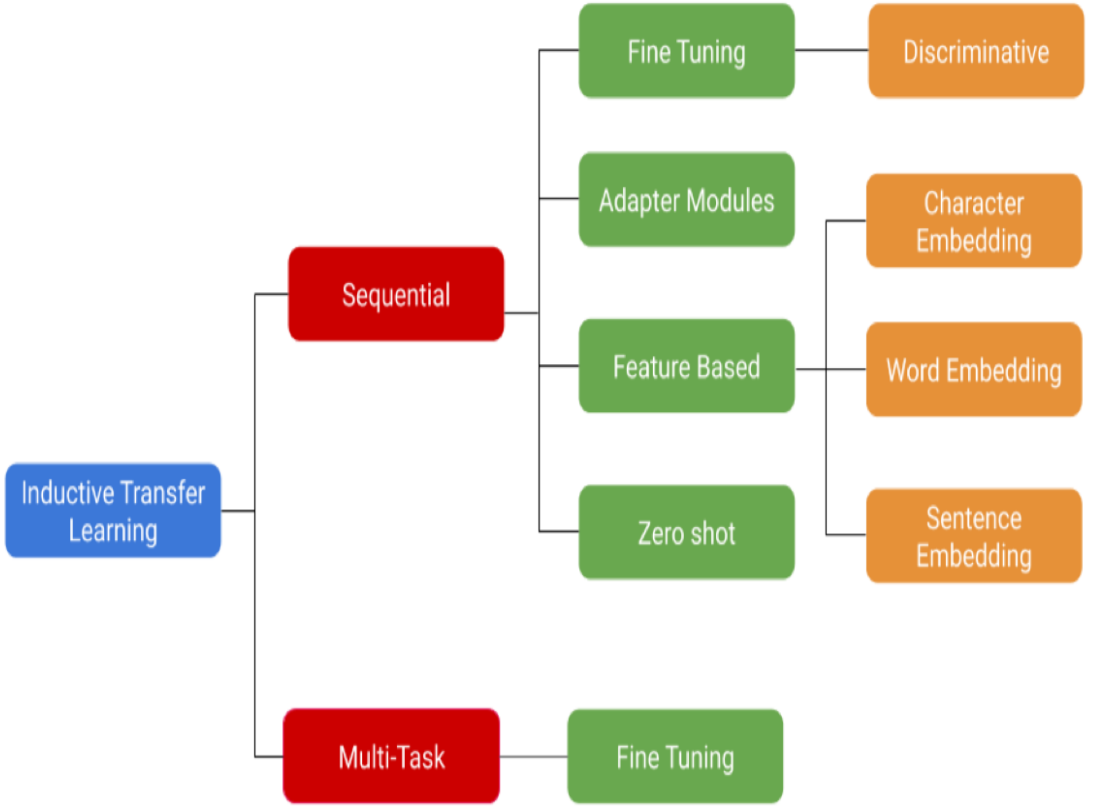
- This involves adapting to a different language in the target domain.
- Useful when we want to use a high-resource language to learn corresponding tasks in a low-resource language.
- One of the recent work involves using a new dataset to evaluate three different cross-lingual transfer methods on the task of user intent classification and time slot detection.
- The dataset contained 57k annotated utterances in English, Thai, and Spanish and was categorized into three domains which were reminder, weather, and alarm.
- The three cross-lingual transfer methods used were translating the training data, using cross-lingual pre-trained embeddings, and novel methods of using multilingual machine translation encoders as contextual word representations.
- The latter two methods outperformed the translation method on the target language that had only several hundred training examples, i.e., a low resource target language.

# Types of Transfer Learning

**Inductive Transfer Learning:** Inductive Transfer Learning is when for different tasks in the source and target domain we have labeled data in the target domain only.

It can further be divided into the following sub-categories:

- ***Sequential Transfer Learning***
- ***Multi-task Transfer Learning***




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

# Sequential Transfer Learning

It involves learning multiple tasks in a sequential fashion. It is further divided into five sub-categories:

**1. Sequential Fine Tuning:**

- Fine-tuning involves the training of the pre-trained model on the target task.
- One of the recent works involves the model for a unified pre-trained language model i.e., UNILM.
- It combines three different training objectives to pre-train a model in a unified way which includes Unidirectional, Bidirectional, and Sequence-to-Sequence.
- The UNILM model achieved state-of-the-art results on different tasks including generative question answering, abstractive summarization, and document-grounded dialog response generation.

# Sequential Transfer Learning

**2. Adapter Modules:**

- They are a compact and extensible transfer learning method for NLP .
- Provides parameter efficiency by only adding a *few* trainable parameters per task, and as new tasks are added previous ones don't require revisiting.
- In the latest work, adapter modules were used to share the parameters between different tasks by fine-tuning the BERT model.
- The model was evaluated against GLUE tasks and obtained state-of-the-art results on text entailment while achieving parameter efficiency.

# Sequential Transfer Learning

**3. Feature-Based:**

- The representations of a pre-trained model are fed to another model. It provides the benefit of using the task-specific model again for similar data.
- Also, extracting feature once saves a lot of computing resources if the same data is used repeatedly.
- In one of the recent work, researchers used a semi-supervised approach for the task of sequence labeling.
- A pre-trained neural language model was used that was trained in an unsupervised approach. It was a bidirectional language model where both forward and backward hidden states are concatenated together.
- The output was then augmented to token representations and fed to a supervised sequence tagging model (TagLM) which was then trained in a supervised way to output the tag of each sequence. The datasets used were CoNLL 2003 NER and CoNLL 200 chunking.
- The model achieved state-of-the-art results on both tasks compared to other forms of transfer learning.

# Sequential Transfer Learning

**3. Zero-shot:**

- Simplest approach where for a given pre-trained model, we don't apply any training procedure to optimize/learn new parameters.
- In a recent study, researchers used the zero-shot transfer on text classification.
- Each classification task was modeled as a text entailment problem where the positive class meant an entailment and the negative class meant there was non.
- Then a pre-trained Bert model on text classification in a zero-shot scenario was used to classify texts in different tasks like emotion detection, topic categorization, and situation frame detection.
- This approach achieved better accuracy in two out of the three tasks in comparison to unsupervised tasks like Word2Vec.

# Multi-task Transfer Learning

It involves learning multiple tasks at the same time. For instance, if we are given a pre-trained model and want to transfer the learning to multiple tasks then all tasks are learned in a parallel fashion.

**Multi-task Fine Tuning:**

- In recent work, the researchers used this approach to explore the effect of using a unified text-to-text transfer transformer (T5).
- The architecture used was similar to the Transformers model with an encoder-decoder network. But it used fully-visible masking instead of casual masking especially for inputs that require predictions based on a prefix like translation.
- The dataset used for training the models was created from the common crawl dataset which was around 750GB. The model required around 11 billion parameters to be trained on such a large dataset.
- Multi-task pre-trained models were used to perform well on different tasks where the models were trained on different tasks by using prefixes like: “Translate English to German”. By fine-tuning the model achieved state-of-the-art results on different tasks like text classification, summarization, and question answering.

# Conclusion

- We see that attention-based models are more popular compared to RNN-based and CNN-based language models.
- BERT appears to be the default architecture for language modeling due to its bidirectional architectures which makes it successful in many downstream tasks.
- In transfer learning techniques used for NLP, sequential fine-tuning seems to be the most popular approach.
- Multi-task fine tuning seems to gain popularity in recent years as many studies found that training on multiple tasks at the same time yields better results.
- Also, text classification datasets seems to be widely used compared to other tasks in NLP because fine-tuning models in such tasks are easier.

# Future Work

- For future work, it is recommended to use bidirectional models like BERT for specific tasks like abstractive question answering, sentiment classification, and parts-of-speech tagging and models like GPT-2, T5 for generative tasks like generative question answering, summarization, and text generation.
- Adapter modules can replace sequential fine-tuning as they show comparable results to traditional fine tuning and are faster and compact due to parameter sharing.
- Also, extensive research should be done to reduce the size of these bigger language models so that they can be easily deployed on embedded devices and on the web.