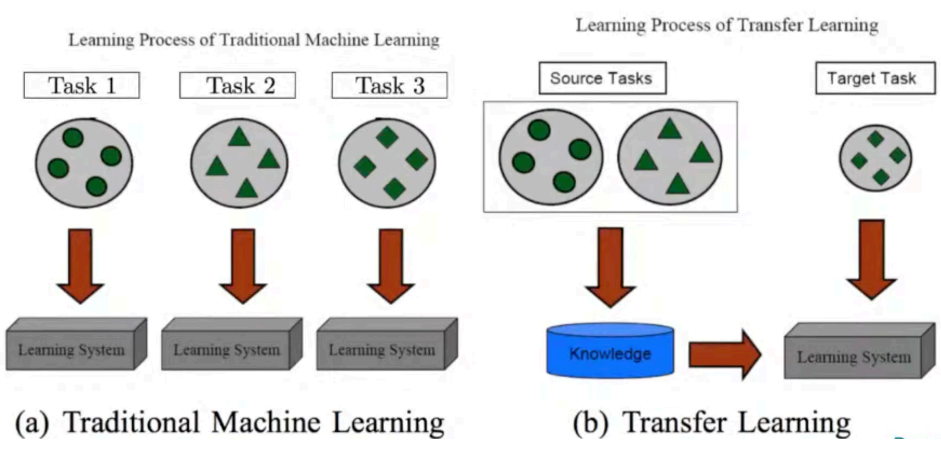
# Transfer Learning in NLP

1/13/21 4:27 PM

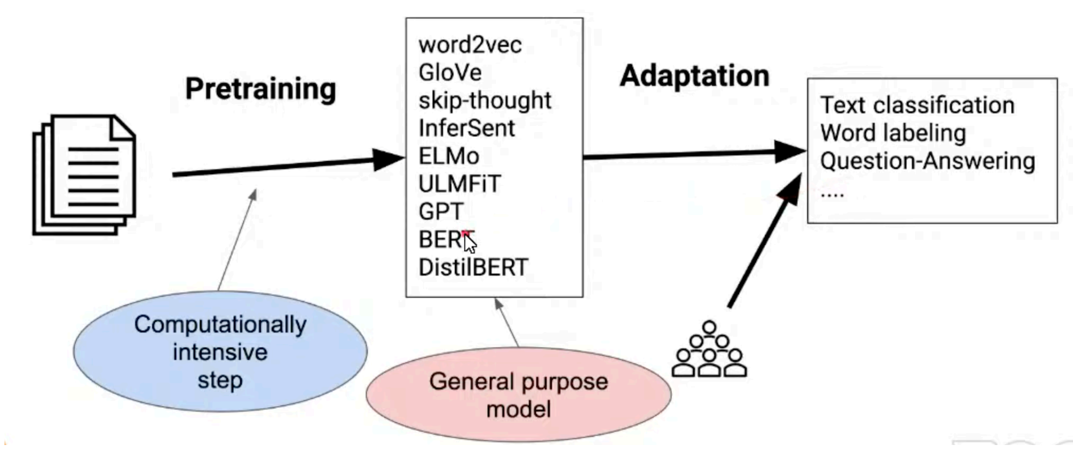
*Speaker: Thomas Wolf (HuggingFace)*
*Date: 11/24/2020*

## What is Transfer Learning?




### *Sequential Transfer Learning*

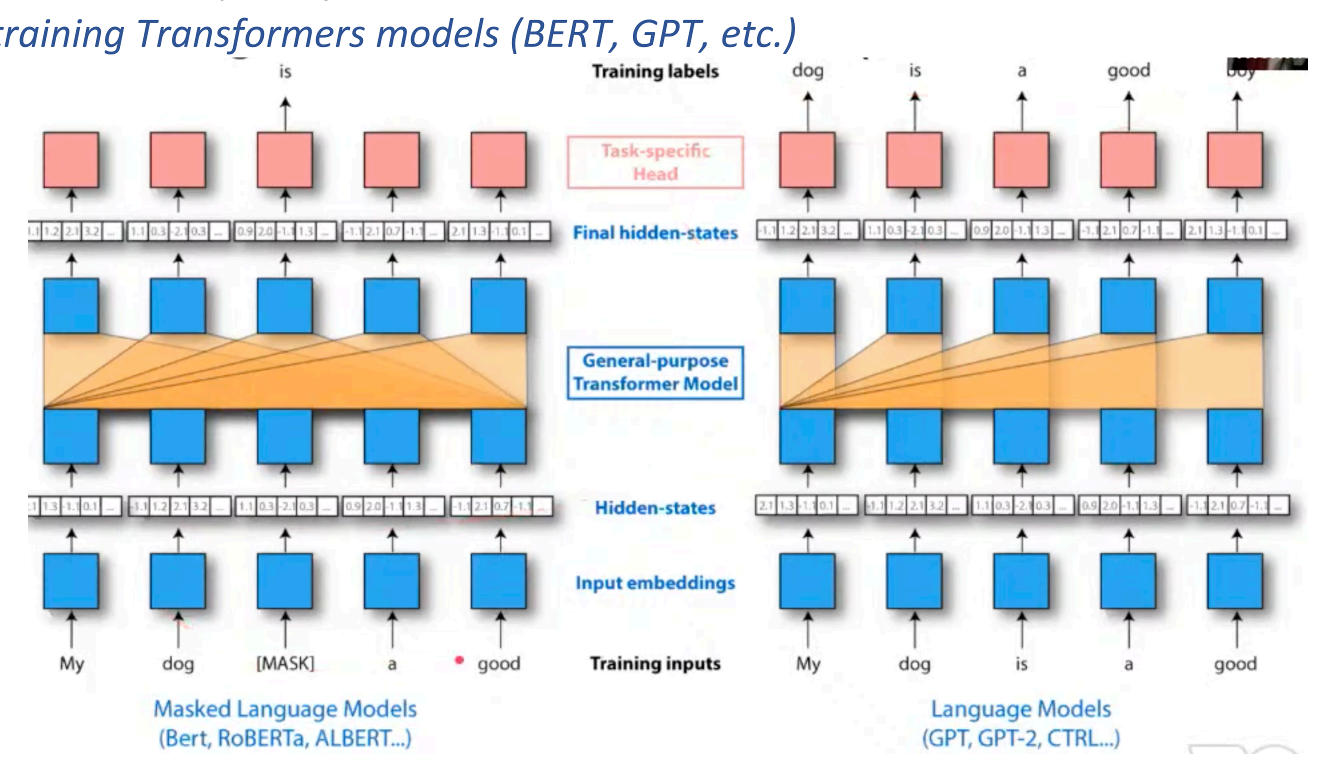
- Learn on one task/dataset, transfer to another task/dataset




## Training: the rise of language modeling pretrain

- Many currently successful pretraining approaches are **based on language modeling (LM)**
- Advantages:
    - Doesn't require human annotation (self-supervised)
    - Many languages have enough text to learn high capacity model
    - Versatile - can be used to learn both sentence and word representations with a variety of objective functions

### *Pretraining Transformers models (BERT, GPT, etc.)*




Left:

- One word is masked, the real word appears as label only
- With attention mask from both directions, predict mask token

Right: Auto-regressive way

- Try to predict the next word
- Need to change attention, can only use the left context
- Less powerful, because it can't use right context
- But advantage is we have more labels (every word), so trains much faster

### *Model: Adapt for target task*

1. Remove pretraining task head (if not used for target task)
2. Add target task-specific elements on top/bottom
    Simple: linear layer(s)
    Complex: full LSTM on top

Sometimes very complex: adapting to a structurally different task

### *Example 1 - Transfer Learning for text classification*

### *Procedure*

Input sentence -> tokenization -> Convert to vocabulary indices -> Pretrained Model -> Classifier model -> Prediction

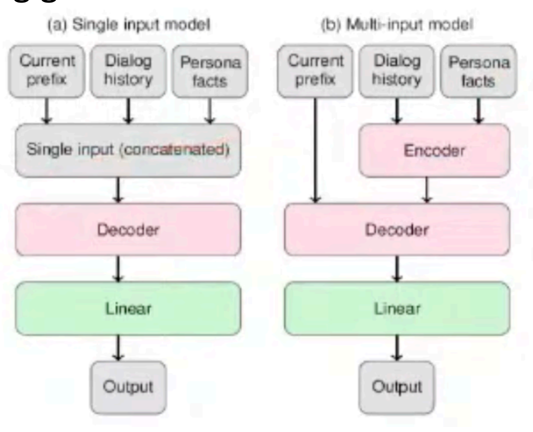
*Pretrained Model (e.g. BERT) convert vocabulary indices (integer) to vector of high dimensions (e.g. BERT 768)

### *Remarks*

- The error rate goes down quickly. After one epoch we already have >90% accuracy
    - So Fine-tuning is highly data efficient in Transfer Learning
- We took our pre-training & fine-tuning hyper-parameters straight from the literature on related models
    - Fine-tuning is often robust to the exact choice of hyper-parameters

### *Example 2 - Transfer Learning for Language Generation*

A dialog generation task: chatbox




## Trends and limits of Transfer Learning in NLP

### *Recent Trends*

Going big on model sizes - over 1 billion parameters as become the norm for SOTA

### *Model Size Problems:*

- Narrowing the research competition filed
- Environmental costs
- Is bigger-is-better a scientific research program?

Three main techniques currently investigated:

- Distillation
    - Use a big model to teach a smaller model how to generalize
    - DistilBert: 95% of Bert performances in a model 40% smaller and 60% faster
- Pruning
    - Take large pre-train model, remove some weights
    - Keep a few weights remaining, but able to keep performance
- Quantization
    - From FP32 to INT8 (float to integer)

### *Generalization Problem:*

- Models are **brittle**: fail when text is modified, even with meaning preserved
- Models are **spurious**: memorize artifacts and biases instead of truly learning

### *Shortcoming of language modeling in general*

Need for grounded representations

- Limits of distributional hypothesis - difficult to learn certain types of information from raw text
    - Example: "while sheep" rarely appear in raw text, because sheep are usually white so they don't need to be indicated specifically. So if you ask a GPT what's the color of sheep, based on the training context, they would answer black.
    - Human reporting bias: not stating the obvious
    - Common sense isn't written down
    - Facts about named entities
    - No relation with other modalities
- Possible solutions:
    - Incorporate structured knowledge (e.g. ERNIE)
    - Multimodal learning (e.g. visual representations - VideoBERT)
    - Interactive/human-in-the-loop approaches (e.g. dialog)

### *Current transfer learning performs adaptation once*

- Example: model just trained before COVID, but now impossible to add COVID into it
- Ultimately, we'd like to have models that continue to retain and accumulate knowledge across many tasks
- No distinction between pretraining and adaptation, just one stream of tasks
- Main challenge: catastrophic forgetting
    - When we try to add more piece of knowledge, the model forgets lots of exisitng information

## HuggingFace Libraries

### *Transformers library*

SOTA general-purpose tools for NLU and Generation

Features:

- Super easy to use, fast to on-board
- For everyone
- SOTA performances
- Deep interoperability between TensorFlow 2.0 and PyTorch

### *Tokenizers library*

Now that NN have fast implementations, a bottleneck in DL based NLP pipelines is often tokenization

Converting strings -> model inputs

Features:

- Encode 1GB in 20sec
- BPE/byte-level-BPE/WordPiece/...
- Bindings in python/js/rust...

### *Datasets library*

Datasets library is a lightweight and extensible library to easily access and process datasets and evaluation metrics for NLP

Features:

- One-line access to 150+ datasets and metrics - Open/collaborative hub
- Built-in interoperability with Numpy, Pandas, PyTorch and Tensorflow 2
- Strive on large datasets: Wikipedia (18GB) only take 9 MB of RAM
- Smart catching: never wait for your data to process several times