Pre-trained Models for Natural Language Processing

Alex Wang

2020.08.11

- Pre-training
- Pre-trained Models
- Transfer Learning
- Applications
- Future Direction
- Taxonomy of PTMs

Pre-training

- Why Pretrain?
- Pre-training Tasks

Why Pre-training

- Provides better model initialization
- Speeds up convergence on the target task
- Can be regarded as a regularization method and avoid overfitting

Examples

- Improvement in the training and generalization of LSTMs in many text classification tasks
- Initialize the weights of both encoder and decoder with PTMs can improved Seq2Seq models
- ……

Language Modeling (LM)

- Most common unsupervised task
- Can efficiently solve a wide range of down stream NLP problems
- Given a text sequence $\mathbf{x}_{1:T} = [x_1, x_2, \cdots, x_T]$, Predict the next word x_{T+1}

Masked Language Modeling (MLM)

- Masks out some tokens from the input sentences by [MASK]
- Trains the model to predict the masked tokens by the rest of the tokens

Permuted Language Modeling (PLM)

- Solving the problem that [MASK] symbol in MLM won't appear in downstream tasks
- Using a randomly sampled permutation from all possible permutations, predict the last few tokens
- Eg.: Using $[x_1, x_2, \cdots, x_k]$ to predict x_{k+1}, using $[x_1, x_2, \cdots, x_{k+1}]$ to predict x_{k+2}

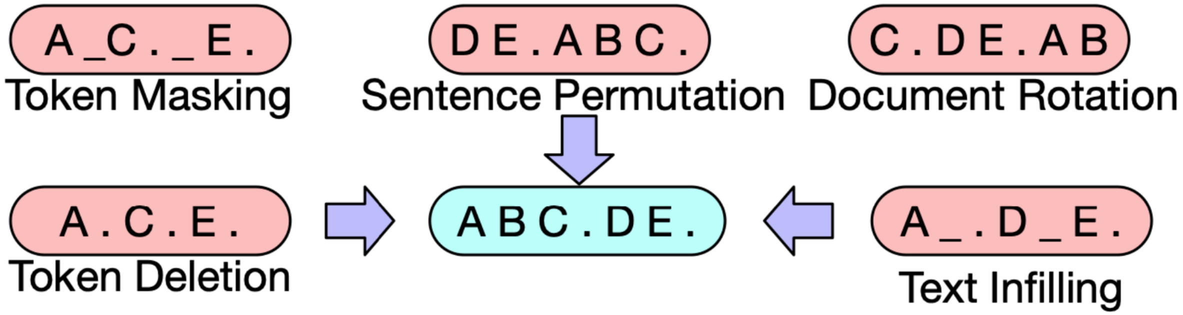
Denoising AutoEncoder (DAE)

- Input a partially corrupted sentence
- Train the model to recover the sentence
- Methods
  - **Token Masking:** Randomly replace some tokens with [MASK]
  - **Text Infilling:** Randomly replace some text span with [MASK]
  - **Token Deletion:** Randomly delete some tokens

Words Level

  - **Sentence Permutation:** Shuffle all sentences in random order
  - **Document Rotation:** Choose a random token as the beginning word

Sentences Level



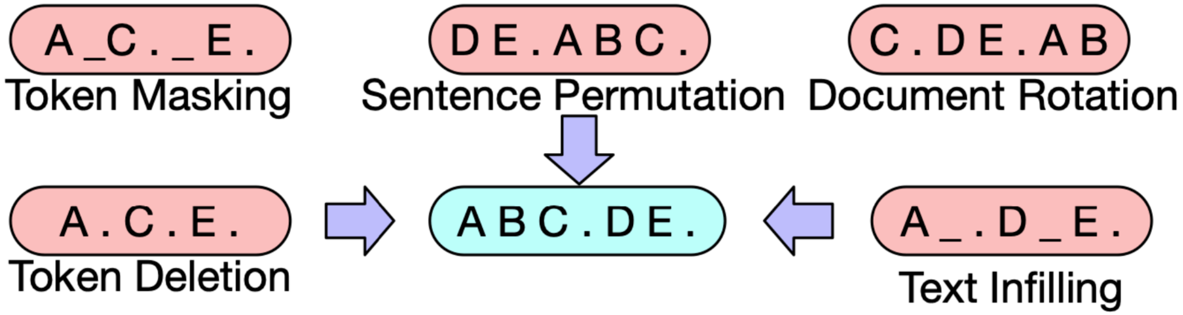
Picture Source: BART: Lewis et al., Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Denoising AutoEncoder (DAE)

- Input a partially corrupted sentence
- Train the model to recover the sentence
- Methods
  - **Token Masking:** Randomly replace some tokens with [MASK]
  - **Text Infilling:** Randomly replace some text span with [MASK]
  - **Token Deletion:** Randomly delete some tokens
  - **Sentence Permutation:** Shuffle all sentences in random order
  - **Document Rotation:** Choose a random token as the beginning word

Words Level

Sentences Level



Picture Source: BART: Lewis et al., Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Contrastive Learning (CTL)

- Distinguish observed pairs of text and randomly sample text pair
- Methods
  - Deep InfoMax (DIM): Maximizing the mutual information between global and local representations
  - Replaced Token Detection (RTD): Whether a token is replaced given its surrounding context
  - Next Sentence Prediction (NSP): Distinguish whether two sentences are continuous segments
  - Sentence Order Prediction (SOP):
    - Positive examples: Two consecutive segments from the same document
    - Negative examples: The same two consecutive segments but with their order swapped

❖ NSP conflates topic prediction and coherence prediction in a single task
❖ SOP only focus on topic prediction

Pre-trained Models

- Non-contextual Embeddings
- Contextual Embeddings
- Extensions of PTMs

Distributed Representation

- Describe the meaning of a piece of text by low-dimensional real-valued vectors.
- Models: Word2Vec (Skip-Gram), Sentence2Vec, Context2Vec

Distributional Representation

- Describe the meaning of a piece of text in metric space
- Model: GloVe

Pros

- Can capture both syntactic and semantic word relationships

$$\begin{aligned} &\text{vec}(\text{“China”}) - \text{vec}(\text{“Beijing”}) \\ \approx\ &\text{vec}(\text{“Japan”}) - \text{vec}(\text{“Tokyo”}) \end{aligned}$$

- No need for deep neural networks to build good word embeddings
- Word2vec embeddings implicitly encode referential attributes of entities

Cons

- Static embedding, not based on context
- Fail to capture polysemous disambiguation, syntactic structures, semantic roles, anaphora.

Pros

- Can capture both syntactic and semantic word relationships
- No need for deep neural networks to build good word embeddings
- Word2vec embeddings implicitly encode referential attributes of entities

GloVe can't do this!

Referential Attributes:
The population, GDP of Italy are referential attributes for "Italy"

Cons

- Static embedding, not based on context
- Fail to capture polysemous disambiguation, syntactic structures, semantic roles, anaphora.

Sequence Models

- Convolutional Models: Aggregating the local information from its neighbors by convolution operations
- Recurrent Models: LSTMs and GRUs

Pros & Cons

- Pros: Easy to train and get good results for various NLP tasks
- Cons: Based on neighbor words or suffered from long-term dependency

Non-Sequence Models

Syntactic Structure or Semantic Relation

- Learning contextual representation based on pre-defined tree or graph structure
- Models: Transformer, Recursive NN, TreeLSTM, and GCN

Pros & Cons

- Pros: Suitable to model long range dependency of language
- Cons: Good graph structure is hard to find, Easy to overfit

BERT

- Bidirectional Encoder Representation from Transformers
- Pre-train tasks: Masked Language Modeling & Document-Level Next Sentence Prediction

Pros

- Good performance on syntactic tasks
- Ability to learn subject-verb agreement and semantic roles
- Ability to encode syntax structure

Cons

- Not good enough at semantic and fine-grained syntactic tasks
- Hard to generate language

Knowledge-Enriched PTMs

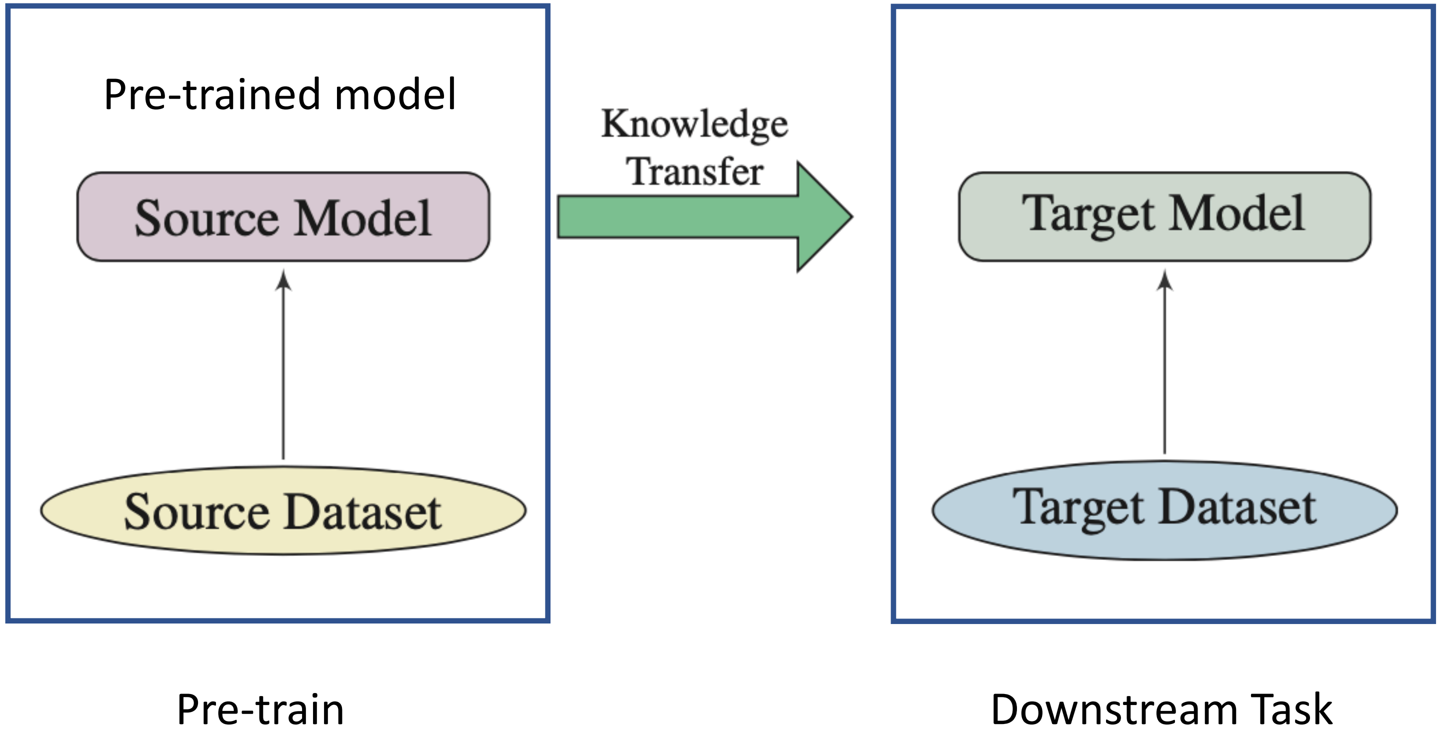
- Inject linguistic, commonsense, domain-specific knowledge into PTMs
- By adding extra Pre-train task
- When injecting multiple kinds of knowledge, PTMs may suffer from catastrophic forgetting

…

Transfer Learning

- How to Transfer?
- Fine-Tuning Strategies

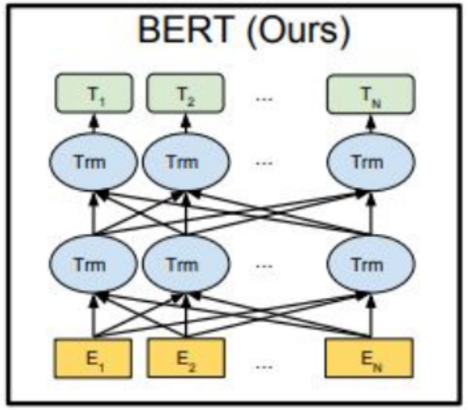
What is Transfer Learning



Picture Source: QIU XP, et al. Pre-trained Models for Natural Language Processing: A Survey

Notice

- Corpus: Should from similar domain
- Pre-train task should be similar to downstream task
  - E.g.: Next Sentence Prediction for Question Answering
- Model structure: BERT is good at language understanding tasks, but hard to generate language
- Layers:
  - **Embedding Only:** Only use the pre-trained static embeddings
  - **Top Layer:** Feed the representation at the top layer
  - **All Layers:** Like ELMo, weighted average of some layers, usually choose the last four layers



Picture Source: BERT: Devlin, et al. Pre-training of Deep Bidirectional Transformers for Language Understanding

Fine-Tuning Strategies

- **Two-Stage Fine-Tuning:** Transferred to intermediate task or corpus before Fine-tuning on target tasks
- **Fine-Tuning with Extra Adaptation Modules**
  - Every downstream task has its own parameters - Parameter inefficiency
  - Projected Attention Layers: Equip model with a task-specific adaptation module, only train PALs
- **Self-Ensemble and Self-Distillation**
- **Gradual Unfreezing**
- **Sequential Unfreezing**

Applications

- General Evaluation Benchmark
- Named Entity Recognition
- Others

General Language Understanding Evaluation (GLUE)

- Single-sentence classification tasks: CoLA and SST-2
- Pairwise text classification tasks: MNLI, RTE, WNLI, QQP, and MRPC
- Text similarity task: STS- B
- Relevant ranking task: QNLI

SuperGLUE

- More challenging tasks and more diverse task formats
- E.g.: Coreference Resolution and Question Answering

Named Entity Recognition (NER)

- Most of NER methods are in the sequence-labeling framework.
- The entity information in a sentence will be transformed into the sequence of labels, and one label corresponds to one word.
- The model is used to predict the label of each word.
- Generator (HMM), Classifier (CFR)

Examples

- Akbik et al. produced word-level embedding for NER.
- TagLM and ELMo Use a PTM’s last layer output and weighted-sum of each layer output as a part of word embedding.
- Pires et al. realized zero-shot NER through multilingual BERT.

Others

- Question Answering
- Sentiment Analysis
- Machine Translation
- Context Summarization

Future Direction

- Upper Bound of PTMs
- Architecture of PTMs
- Knowledge Transfer Beyond Fine-tuning
- Interpretability and Reliability of PTMs

Upper Bound of PTMs

- Larger corpora
- Challenging pre-training tasks, Self-supervised pre-training tasks
- More training steps
- Increasing the depth of models
- More efficient model architecture
- Optimizers, training skills

Architecture of PTMs

- Improve the architecture of the Transformer, such as Transformer-XL
- Automatic design of deep architecture, such as Neural Architecture Search

Knowledge Transfer Beyond Fine-tuning

- Fine-tunable adaption modules
- Feature extraction
- Knowledge distillation
- Data augmentation

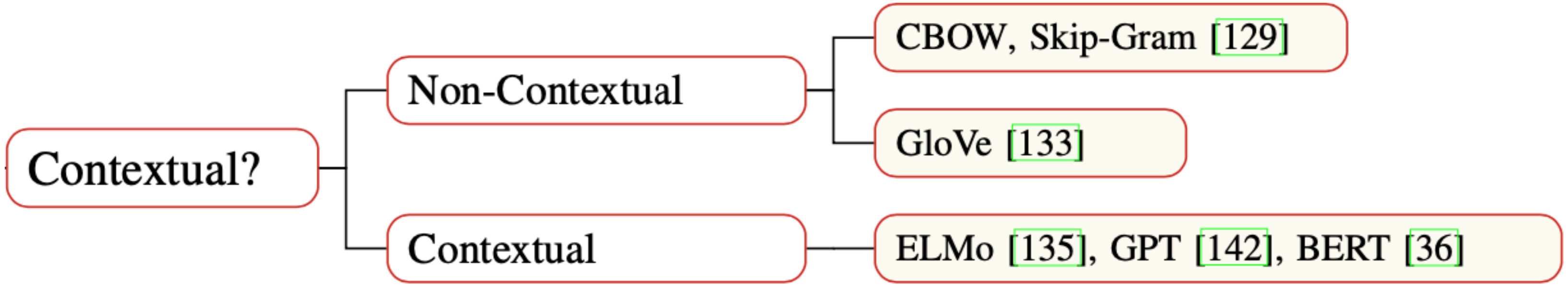
Interpretability and Reliability of PTMs

- Much work is on the attention mechanism, which is still controversial

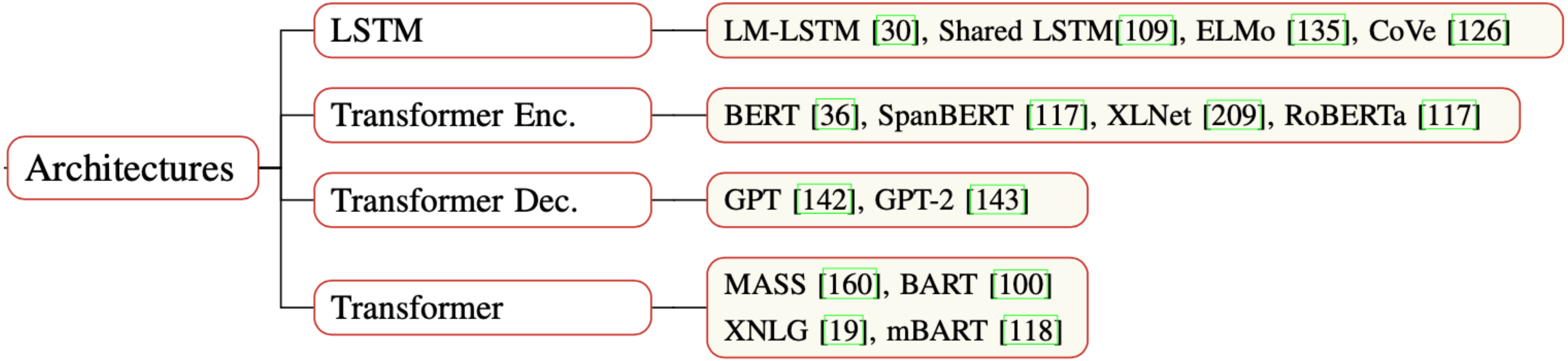
Taxonomy of PTMs

- Contextual & Non-Contextual
- Architectures
- Task Types
- Extensions

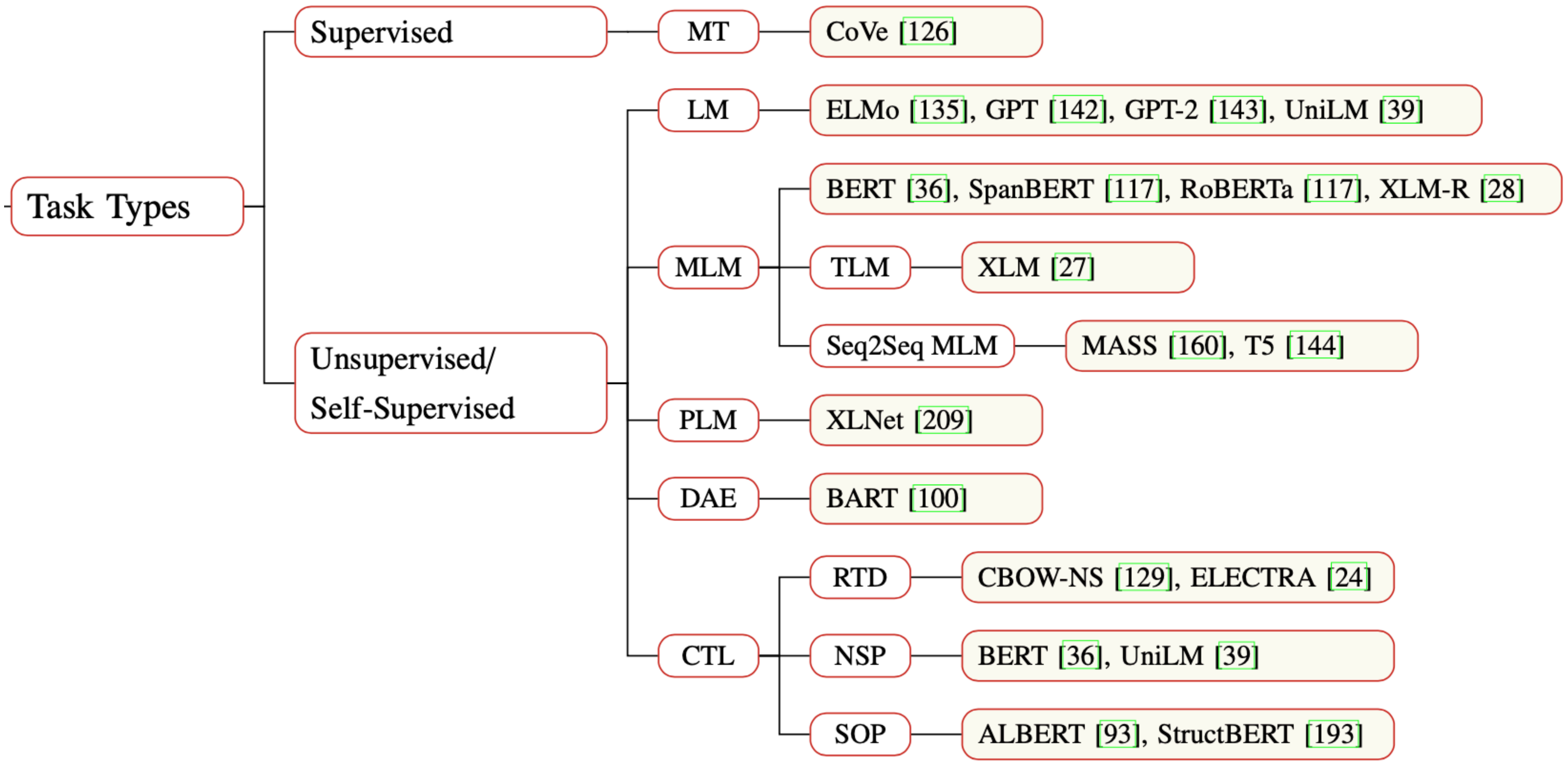
Contextual & Non-Contextual



Architectures

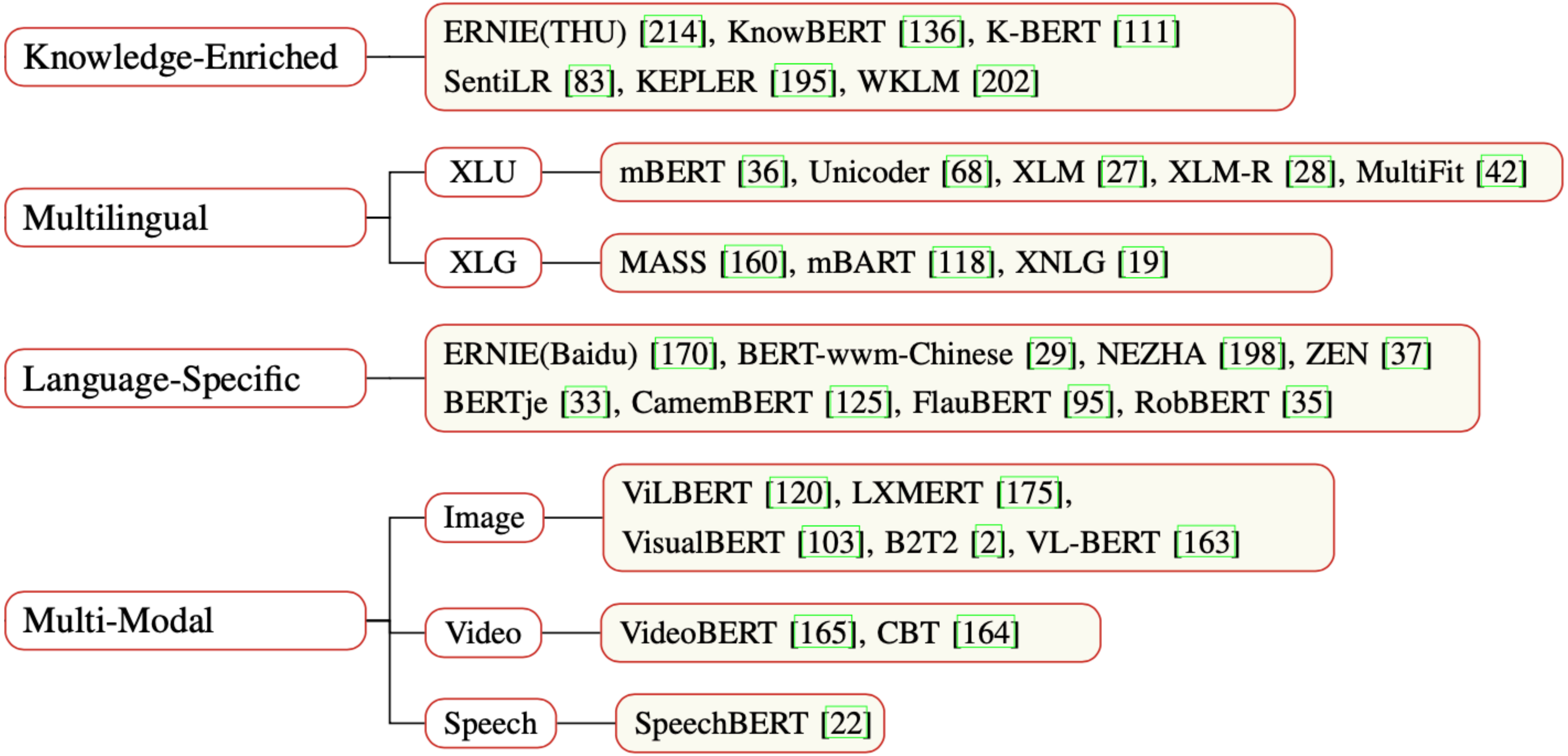


Task Types

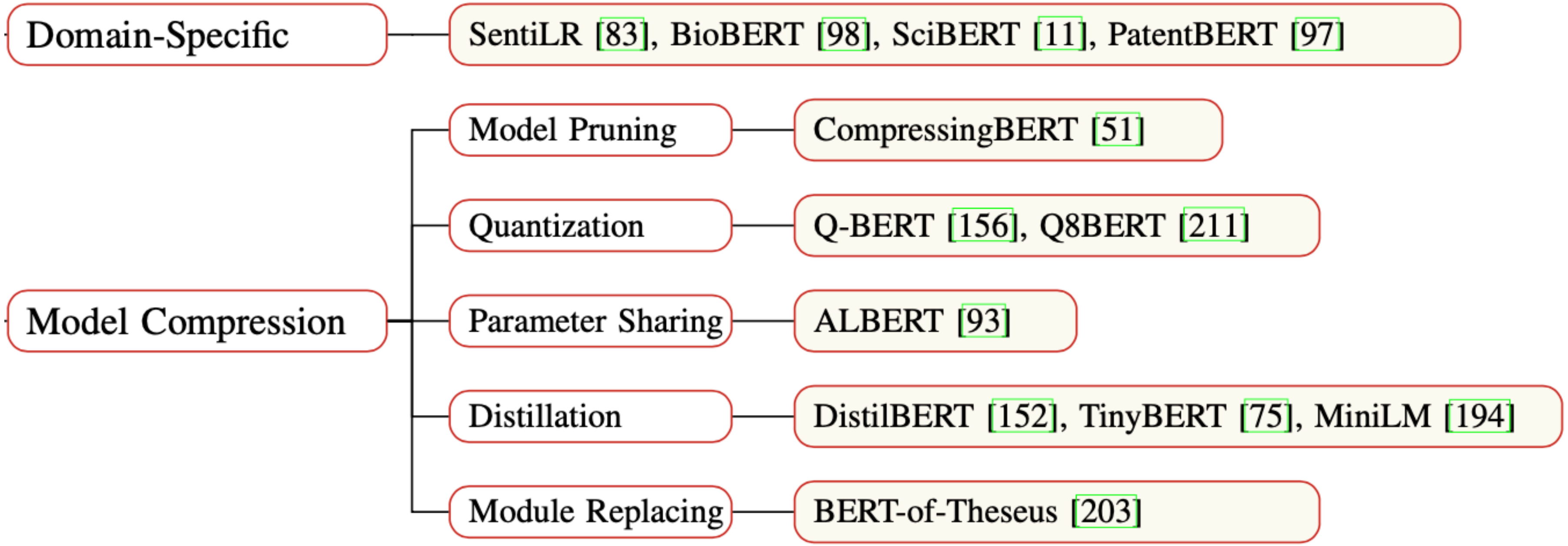

Task Types
Supervised
MT
CoVe [126]
Unsupervised/
Self-Supervised
LM
ELMo [135], GPT [142], GPT-2 [143], UniLM [39]
MLM
BERT [36], SpanBERT [117], RoBERTa [117], XLM-R [28]
TLM
XLM [27]
Seq2Seq MLM
MASS [160], T5 [144]
PLM
XLNet [209]
DAE
BART [100]
CTL
RTD
CBOW-NS [129], ELECTRA [24]
NSP
BERT [36], UniLM [39]
SOP
ALBERT [93], StructBERT [193]

Taxonomy of PTMs

Extensions



Extensions


Domain-Specific
SentiLR [83], BioBERT [98], SciBERT [11], PatentBERT [97]
Model Compression
Model Pruning
CompressingBERT [51]
Quantization
Q-BERT [156], Q8BERT [211]
Parameter Sharing
ALBERT [93]
Distillation
DistilBERT [152], TinyBERT [75], MiniLM [194]
Module Replacing
BERT-of-Theseus [203]

Thanks