A Survey of popular Pretrained NLP Models

Poornapragna Vadiraj

For CMPE 297 Sec 49 - Emerging Technologies
Prof. Vijay Eranti

Introduction

Through the advancement of deep learning, numerous neural networks have been commonly used to solve natural language processing (NLP) problems, such as:

1. deep convolutional neural networks (CNNs)
2. recurrent neural networks (RNNs)
3. graph-based neural networks (GNNs)
4. attention mechanisms

- Aids Automated Feature Engineering
- Non-neural NLP methods vs Neural NLP methods
- These representations are studied in the specific tasks of NLP (More later)
- Therefore, neural network-based approaches make it possible for people to build different NLP models.

Paper (Pre-trained Models for Natural Language Processing: A Survey)

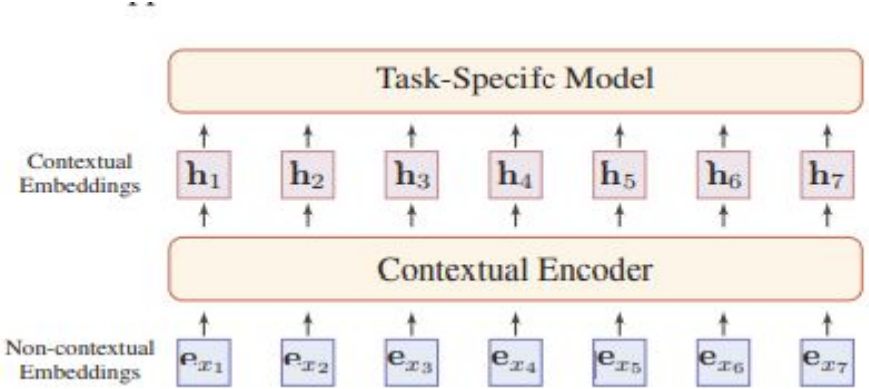
- Pre-trained models can develop universal language representations.
- How?
- With the advancement of computational power
- The advent of deep models (i.e. Transformer), and the ongoing improvement of training skills, the design of PTMs has shifted from shallow to deep.
- The goal of the first generation PTMs is to learn good word embeddings.

Language Representation Learning

- A good representation should articulate general-purpose priors - Bengio, et al.
- Implicit linguistic principles should be caught by a decent representation.
- The central idea of distributed representation is to explain the sense of a piece of text by low-dimensional real-valued vectors.
- There is no equivalent meaning for each part of the vector, while the whole represents a concrete idea.

Generic Neural Architecture for NLP

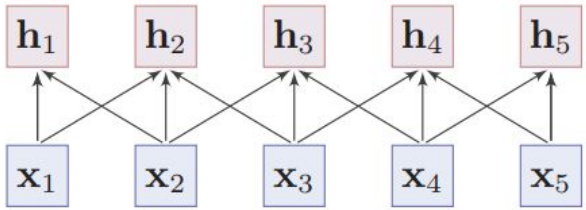
This displays the standardized neural architecture. Two forms of word embedding are available: non-contextual and contextual embedding. The difference between them is whether the embedding for a word dynamically changes according to the context it appears in.



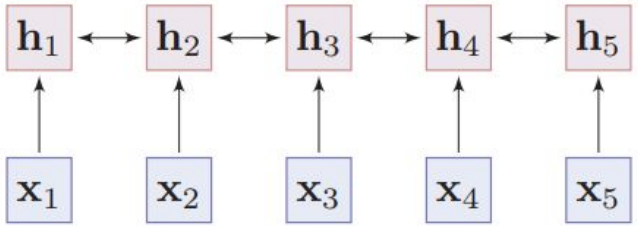
Neural Contextual Encoders

Most of the neural contextual encoders can be classified into two categories:

- sequence models - capture a word's local meaning in sequential order.
  - CNN
  - RNN



(a) Convolutional Model



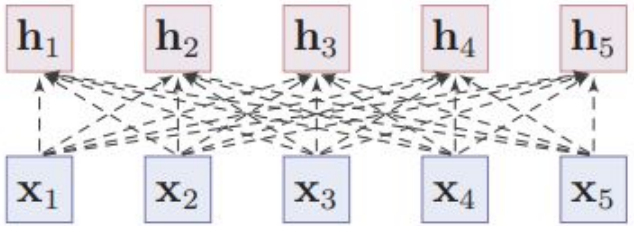
(b) Recurrent Model

Neural Contextual Encoders

Graph-based models:

- Non-sequence models learn the relational representation between words, such as the syntactic structure or semantic relationship, with a pre-defined tree or graph structure.

Fully-Connected Self-Attention Model



(c) Fully-Connected Self-Attention Model

Why Pre-training?

1. Pre-training on the vast corpus of text will learn universal representations of language and assist with the downstream activities.
2. Pre-training produces a smoother initialization of the model, which typically leads to enhanced results for generalization and accelerates convergence on the target task.
3. To prevent overfitting on limited details, pre-training can be seen as a kind of regularization.

PTM History

First-Generation PTMs: Pre-trained Word Embedding:

- There is a long tradition of considering words as dense vectors.
- In the groundbreaking work of the neural network language model (NNLM), the modern concept of embedding is implemented.
- NLP tasks could be dramatically enhanced by pre-trained word embedding on the unlabelled results.

Cons:

- While pre-trained word embedding has been shown to be successful in NLP tasks, they are context-independent and shallow models are mainly trained.
- The rest of the entire model also has to be learned from scratch as used on a downstream task.

PTM History

Second-Generation PTMs: Pre-trained Contextual Encoders

Neural encoders' output vectors are often called contextual word embeddings since, based on their meaning, they represent the word semantics.

The first active instance of PTM for NLP was suggested by Dai and Le. They initialized LSTMs with a language model (LM) or sequence autoencoder and found that in many text classification tasks, pre-training would boost the training and generalization of LSTMs.

In the multi-task learning (MTL) system, Liu et al. pre-trained a shared LSTM encoder with LM and fine-tuned it. For many text classification activities, they found that pre-training and fine-tuning could further boost the efficiency of MTL. The Seq2Seq models can be greatly enhanced by unsupervised pre-training, as observed by Ramachandran et al.

PTM History

Now (Third) Generation of PTMs

More recently, very deep PTMs have shown their strong abilities to learn universal representations of language: e.g. OpenAI GPT (Generative Pre-training) and BERT (Transformer Bidirectional Encoder Representation).

In addition to LM, a growing number of self-supervised tasks are proposed to allow large-scale text data of PTMs that capture more information.

Loss Functions of PTMs

Task	Loss Function	Description
LM	$\mathcal{L}_{\text{LM}} = -\sum_{t=1}^{T} \log p(x_t \mid \mathbf{x}_{<t})$	$\mathbf{x}_{<t} = x_1, x_2, \cdots, x_{t-1}$.
MLM	$\mathcal{L}_{\text{MLM}} = -\sum_{\hat{x} \in m(\mathbf{x})} \log p\left(\hat{x} \mid \mathbf{x}_{\backslash m(\mathbf{x})}\right)$	$m(\mathbf{x})$ and $\mathbf{x}_{\backslash m(\mathbf{x})}$ denote the masked words from $\mathbf{x}$ and the rest words respectively.
Seq2Seq MLM	$\mathcal{L}_{\text{S2SMLM}} = -\sum_{t=i}^{j} \log p\left(x_t \mid \mathbf{x}_{\backslash \mathbf{x}_{i:j}}, \mathbf{x}_{i:t-1}\right)$	$\mathbf{x}_{i:j}$ denotes an masked n-gram span from i to j in $\mathbf{x}$.
PLM	$\mathcal{L}_{\text{PLM}} = -\sum_{t=1}^{T} \log p(z_t \mid \mathbf{z}_{<t})$	$\mathbf{z} = perm(\mathbf{x})$ is a permutation of $\mathbf{x}$ with random order.
DAE	$\mathcal{L}_{\text{DAE}} = -\sum_{t=1}^{T} \log p(x_t \mid \hat{\mathbf{x}}, \mathbf{x}_{<t})$	$\hat{\mathbf{x}}$ is randomly perturbed text from $\mathbf{x}$.
DIM	$\mathcal{L}_{\text{DIM}} = s(\hat{\mathbf{x}}_{i:j}, \mathbf{x}_{i:j}) - \log \sum_{\tilde{\mathbf{x}}_{i:j} \in \mathcal{N}} s(\hat{\mathbf{x}}_{i:j}, \tilde{\mathbf{x}}_{i:j})$	$\mathbf{x}_{i:j}$ denotes an n-gram span from i to j in $\mathbf{x}$, $\hat{\mathbf{x}}_{i:j}$ denotes a sentence masked at position i to j, and $\tilde{\mathbf{x}}_{i:j}$ denotes a randomly-sampled negative n-gram from corpus.
NSP/SOP	$\mathcal{L}_{\text{NSP/SOP}} = -\log p(t \mid \mathbf{x}, \mathbf{y})$	$t = 1$ if $\mathbf{x}$ and $\mathbf{y}$ are continuous segments from corpus.
RTD	$\mathcal{L}_{\text{RTD}} = -\sum_{t=1}^{T} \log p(y_t \mid \hat{\mathbf{x}})$	$y_t = \mathbf{1}(\hat{x}_t = x_t)$, $\hat{\mathbf{x}}$ is corrupted from $\mathbf{x}$.

[1] $\mathbf{x} = [x_1, x_2, \cdots, x_T]$ denotes a sequence.

Contextual Embeddings

A large number of studies have probed and classified different types of knowledge in contextual embeddings. In general, there are two types of knowledge:

Eg: **Linguistic** knowledge - An bird

World knowledge - Dante was born in [MASK]

Although PTMs capture the general language knowledge from a large corpus, how effectively adapting their knowledge to the downstream task is still a key problem.

Transfer Learning

How to Transfer?

1. Choosing appropriate pre-training task, model architecture, and the corpus -
2. Choosing appropriate layers
3. Ask “To tune or not to tune?”
    a. Feature Extraction
    b. Fine-tuning
        i. Two-stage fine-tuning
        ii. Multi-task fine-tuning

Applications & Demo

- Question Answering
- Sentiment Analysis
- Named Entity Recognition
- Machine Translation
- Summarization
- Adversarial Attacks and Defenses

https://demo.allennlp.org/named-entity-recognition

Material submitted on:

- Medium
- Github
- Canvas

Thank you