Advanced Text-As-Data - Winter School - Iesp UERJ

Day 3: Pre-Trained Models

Sebastián Vallejo Vera [1]

July 6, 2025

[1] Dept. of Political Science – Western University

Table of Contents

Transformers in Encoders

Pre-Trained models i

- The Transformers architecture *embed* tokens with meaning, creating accurate representations (i.e., embeddings).
- As with embeddings, the representations are relational, they provide meanings to tokens in relation to other tokens.
- This means that, the more tokens we can run through the architecture, the more accurate these representations are likely to be.

Pre-Trained models ii

- Pre-trained models like BERT, leverage the Transformers architecture, but they also need (1) information and a way to (2) learn from that information.
- They learn, as we previously learned, through *masking*.
- Getting information is relatively easy (and unethical) and computational expensive:
    - BERT uses 11,038 books from the Toronto BookCorpus and all of English Wikipedia–a total of 16GB worth of text (Devlin et al., 2018).
    - RoBERTa uses the same data as BERT and added more data from Common Crawl (CC-News/Stories), and Open WebText for a total of 160GB of text (Liu et al., 2019).
    - Cross-lingual RoBERTa, XLM-R, was trained using Wikipedia for all languages and data from Common Crawl (Conneau et al., 2019).

Bidirectional Encoder Representations from Transformers

- Google AI's BERT, Facebook AI's RoBERTa and XLM-RoBERTa, and Microsoft's DeBERTa are the encoders of a Transformers model.
- BERT stands for 'Bidirectional Encoder Representations from Transformers'. RoBERTa stands for 'Robustly optimized BERT approach' and XLM-RoBERTa stands for "Cross-lingual RoBERTa."
- As a note, *bidirectional* which points to its ability to read text forwards *and* backwards, relating each word to all words in a sentence.
- And because of the attention mechanism, the representations change according to the context.

Remember the Encoder Mechanism

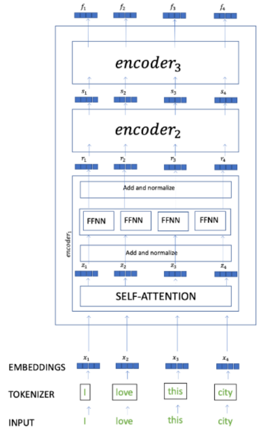

Figure 1: Diagram of a three-stack encoder of a Transformers model. Input text is tokenized and given an initial embedding (vectorized representation) simplified in our figure as x_1 through x_4. The initial embeddings are transformed as they enter the first $encoder_1$. In it, the self-attention mechanism updates the embeddings (z_1 through z_4), which are then passed through a feed-forward neural network. They exit the encoder as a more accurate set of embeddings (r_1 through r_4). The process is repeated for all encoders in the neural network. For example, pre-trained BERT-base models use 12 encoder layers.

How Are Pre-Trained Models Trained

- BERT-base consists of 12 encoder layers and 12 attention heads, and 24 encoder layers and 16 attention heads in its BERT-large configuration (Ravichandiran, 2021).
    - The attention heads repeat their computations multiple times in parallel. The attention head splits its Query, Key, and Value parameters N-ways and passes each split independently through a separate Head. All of these similar Attention calculations are then combined together to produce a final Attention score.
- BERT-base produce outputs word vectors of length 768, while the BERT-large model outputs word vectors of length 1,024 (the same applies to the base and large versions of RoBERTa and XLM-R).

Fine-Tuning and Further Pre-Training

Fine-Tuning a Pre-Trained Model

- Once a model has been pre-trained (e.g., BERT), we can modify the last layer to perform classifications tasks. This is called fine-tuning.
- To this is end, we need a training set and your target data (i.e., data to predict).
- Most considerations when training supervised-models apply, even though Transformers have shown a number of benefits over previous approaches:
  - Better at understanding out-of-context vocabulary.
  - Higher out-of-domain robustness (Hendrycks et al., 2020).
  - Less observations per category required for improved performance.

To the code!

Fine-Tuning: Other Considerations

- Different layers mean different things (Jawahar, Sagot and Seddah, 2019; Zhang et al., 2025). It is unclear what each layer means, or where is the most accurate representation of a particular token, but this is worth considering. However, overall, the consensus is that the more layers the better.
- When predicting labels, we will be using a softmax logistic regression, but we can use an ordinal regression if this is the case.

Further Pre-Training

What if you have a corpus with specialized language that is unlikely to appear in the same context as the data used to pre-train BERT and co.?

Further Pre-Training

What if you have a corpus with specialized language that is unlikely to appear in the same context as the data used to pre-train BERT and co.? What if I have tokens that are highly informative but that are not in the the data used to pre-train BERT and co.?

Further Pre-Training

What if you have a corpus with specialized language that is unlikely to appear in the same context as the data used to pre-train BERT and co.? What if I have tokens that are highly informative but that are not in the the data used to pre-train BERT and co.? One neat thing about pre-trained models is that we can further pre-train them and increase performance Timoneda and Vallejo Vera (2025) .

There are four steps to further train a pre-trained model:

1. We add new tokens to the original model (Optional).
2. (If 1, then) we assign the mean representation of similar words to the newly added tokens.
3. We feed a new large unstructured text corpus (if 1 and 2, containing the new tokens) and train it again to improve the representations for those tokens.
4. We save the new model and apply it to our classification task through fine-tuning in the same way we would apply the original.

To the code!

Further Pre-Training: Other Considerations

- We can further pre-train a model using different masking strategies (see Timoneda and Vera, 2025).
- You need A LOT of text to see improvements in performance.
- You need A LOT of computational resources to further pre-train a model.

Questions? Comments? Break?

References

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer and Veselin Stoyanov. 2019. "Unsupervised cross-lingual representation learning at scale." *arXiv preprint* .

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. 2018. "Bert: Pre-training of bidirectional transformers for language understanding." *arXiv preprint* .

Hendrycks, Dan, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan and Dawn Song. 2020. "Pretrained transformers improve out-of-distribution robustness." *arXiv preprint arXiv:2004.06100* .

Jawahar, Ganesh, Benoît Sagot and Djamé Seddah. 2019. What Does BERT Learn about the Structure of Language? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, ed. Anna Korhonen, David Traum and Lluís Màrquez. Florence, Italy: Association for Computational Linguistics pp. 3651–3657.
URL: *https://aclanthology.org/P19-1356/*

Liu, Yinhan, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer and Veselin Stoyanov. 2019. "Roberta: A robustly optimized bert pretraining approach." *arXiv preprint arXiv:1907.11692* .

Ravichandiran, Sudharsan. 2021. *Getting Started with Google BERT*. Packt Publishing.

Timoneda, Joan C and Sebastián Vallejo Vera. 2025. "Behind the mask: Random and selective masking in transformer models applied to specialized social science texts." *PloS one* 20(2):e0318421.

Timoneda, Joan C. and Sebastián Vallejo Vera. 2025. "BERT, RoBERTa or DeBERTa? Comparing Performance Across Transformer Models in Political Science Text." *The Journal of Politics* 00(00):00.

Zhang, Cheng, Jinxin Lv, Jingxu Cao, Jiachuan Sheng, Dawei Song and Tiancheng Zhang. 2025. "Unravelling the semantic mysteries of transformers layer by layer." *The Computer Journal* p. bxaf034.