Université de Montréal
Faculté des arts et des sciences
Département d'**informatique et de recherche opérationnelle**

Pretrained Language Models on Low Ressources

Gao Yinghan, Wei Wei, Frederic Boileau

February 5, 2021

Introduction to attention and its history

It was conjectured by Bahdanau et al[1] that the dominant encoder decoder paradigms to Neural Machine Translation presented one major drawback. All the information necessary to the transduction mechanism had to be encoded into a fixed length vector. The dominant approach was to use a RNN to encode the sequence to transduce (or more precisely the
sentence to translate in the case of Bahdanau et al) into a fixed length vector c which was a function of the encoder's hiddens states, usually simply extracting the last hidden parameter. To remedy the situation the authors tackled the problem with an innovative approach: making the context vector depend on a linear combinations of **all** the hidden states, where the weights assigned respectively were learned through training a feedforward network. This enabled to effectively learn what part of the input sentence the output had to *attend to*. This idea was called attention.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \qquad \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \qquad e_{ij} = a(s_{i-1}, h_j) \tag{1}$$

The Transformer revolution

Later on Vaswani et al [2] introduced a revolutionnary architecture, the transformer. Still based on the idea of an encoder and decoder for transduction, they based their architecture solely on an attention mechanism enables to model the long range dependencies in a sentence without the inherently sequential constraint which RNNs in their various forms imply. The massive parallelization enabled massive improvements in training time to achieve state of the art results in machine translation

Pretrained Language Models

We will focus in our project not precisely on transduction tasks but on large pretrained language models (PLMs) which can then be fined tuned for downstream tasks such as …

BERT is …[3]

Architecture	Number of parameters
BERT	340M
GPT-2	1.5B
MegatronLM	8.3B
T5	11B
T-NLG	17B
GShard	600B

Table 1: PLMs and their sizes

Compression approaches

Model compression consists of ...

Knowledge distillation

Knowledge Distillation (KD)[4]

tinyBERT

TinyBERT[5]

Other model commpression techniques

We have the following other model compression techniques[6]

- Pruning
- Quantization
- Parameter Sharing
- Tensor Decomposition

PLMs can be evaluated on GLUE for general LM performance and more specific downstream tasks for which they are fine tuned such as Squad.

- SquAD[7]
- GLUE[8]

References

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2016.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015.

[5] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling BERT for natural language understanding. *CoRR*, abs/1909.10351, 2019.

[6] Manish Gupta and Puneet Agrawal. Compression of deep learning models for text: A survey, 2020.

[7] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. *CoRR*, abs/1806.03822, 2018.

[8] Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. SWAG: A large-scale adversarial dataset for grounded commonsense inference. *CoRR*, abs/1808.05326, 2018.