A Related Compression Methods

A.1 BERT Compression Methods

Many techniques have been proposed to compress BERT (Devlin et al., 2018). Ganesh et al. (2020) provide a survey on various compression methods for BERT. Most existing methods focus on alternative architectures in transformer layers or learning strategies.

In our work, we use DistilBERT and ALBERT-base as light pretrained language model encoders for semantic parsing. DistilBERT (Sanh et al., 2019) uses distillation to pretrain a model that is 40% smaller and 60% faster than BERT-base, while retaining 97% of its downstream performances. ALBERT (Lan et al., 2019) factorizes the embedding and shares parameters among the transformer layers in BERT and results in better scalability than BERT. ALBERT-xxlarge outperforms BERT-large on GLUE (Wang et al., 2018), RACE (Lai et al., 2017), and SQUAD (Rajpurkar et al., 2016) while using less parameters.

We use compositional code learning (Shu and Nakayama, 2017) to compress the model embeddings, which contain a substantial amount of model parameters. Previously ALBERT uses factorization to compress the embeddings. We find more compression possible with code embeddings.

A.2 Embedding Compression Methods

Varied techniques have been proposed to learn compressed versions of non-contextualized word embeddings, such as, Word2Vec (Mikolov et al., 2013) and GLoVE (Pennington et al., 2014). Subramanian et al. (2018) use denoising k-sparse autoencoders to achieve binary sparse intrepretable word embeddings. Chen et al. (2016) achieve sparsity by representing the embeddings of uncommon words using sparse linear common combination of common words. Lam (2018) achieve compression by quantization of the word embeddings by using 1-2 bits per parameter. Faruqui et al. (2015) use sparse coding in a dictionary learning setting to obtain sparse, non-negative word embeddings. Raunak (2017) achieve dense compression of word embeddings using PCA combined with a post-processing algorithm. Shu and Nakayama (2017) propose to represent word embeddings using compositional codes learnt directly in end-to-end fashion using neural networks. Essentially few common basis vectors are learnt and embeddings are reconstructed using their composition via a discrete code vector specific to each token embedding. This results in 98% compression rate in sentiment analysis task and 94% - 99% in machine translation tasks without performance loss while applied to LSTM based models. All the above techniques are applied to embeddings such as WordVec and Glove, or LSTM models.

We aim at learning space-efficient embeddings for transformer-based models. We focus on compositional code embeddings (Shu and Nakayama, 2017) since they maintain the vector dimensions, do not require special kernels for calculating in a sparse or quantized space, can be finetuned with transformer-based models end-to-end, and achieve extremely high compression rate. Chen et al. (2018) explores similar idea as Shu and Nakayama (2017) and experiment with more comples composition functions and guidances for training the discrete codes. Chen and Sun (2019) further show that end-to-end training from scratch of models with code embeddings is possible. Given various pretrained language models, we find that the method proposed by Shu and Nakayama (2017) is straightforward and perform well in our semantic parsing experiments.

References

Ting Chen, Martin Renqiang Min, and Yizhou Sun. 2018. Learning k-way d-dimensional discrete codes for compact embedding representations. *arXiv preprint arXiv:1806.09464*.

Ting Chen and Yizhou Sun. 2019. Differentiable product quantization for end-to-end embedding compression. *arXiv preprint arXiv:1908.09756*.

Yunchuan Chen, Lili Mou, Yan Xu, Ge Li, and Zhi Jin. 2016. Compressing neural language models by sparse word representations. *arXiv preprint arXiv:1610.03950*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Manaal Faruqui, Yulia Tsvetkov, Dani Yogatama, Chris Dyer, and Noah Smith. 2015. Sparse overcomplete word vector representations. *arXiv preprint arXiv:1506.02004*.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Deming Chen, Marianne Winslett, Hassan Sajjad, and Preslav Nakov. 2020. Compressing large-scale transformer-based models: A case study on bert. *arXiv preprint arXiv:2002.11985*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. *arXiv preprint arXiv:1704.04683*.

Maximilian Lam. 2018. Word2bits-quantized word vectors. *arXiv preprint arXiv:1803.05651*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.

Vikas Raunak. 2017. Simple and effective dimensionality reduction for word embeddings. *arXiv preprint arXiv:1708.03629*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Raphael Shu and Hideki Nakayama. 2017. Compressing word embeddings via deep compositional code learning. *arXiv preprint arXiv:1711.01068*.

Anant Subramanian, Danish Pruthi, Harsh Jhamtani, Taylor Berg-Kirkpatrick, and Eduard Hovy. 2018. Spine: Sparse interpretable neural embeddings. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. *CoRR*, abs/1804.07461.