A Supplemental Material

A.1 Implementation Details

We use the same settings as the best performing model for unsupervised unlabeled constituency parsing from the DIORA paper (Drozdov et al., 2019). We vary only in the batch size, using 32 instead of 128. A summary of the settings is below:

- Batch Size: 32
- Cell Size: 400
- Learning Rate: 0.002
- Optimization Algorithm: Adam
- Gradient Clipping: 5 (max-L2-norm)
- Training Data: SNLI (Bowman et al., 2015) + MultiNLI (Williams et al., 2018b)
- Negative Samples: 100 per batch
- Maximum Sentence Length: 20
- Composition Function: 2-layer MLP

All experiments were written using Pytorch (Pytorch Core Team, 2019) and using the publicly available DIORA codebase.[8]

A.2 ELMo/BERT Variants

For ELMo, we use the heuristic from Peters et al. (2018b) for each of the 3 layers individually and all combined, resulting in 4 variants.

With ELMo$_{CI}$, there is no notion layers and each token representation is completely context independent. For this reason, we use the same heuristic as for ELMo, but we also try averaging all tokens in a phrase, using the max, and concatenating the average with max. The latter of the 4 approaches worked best.

With BERT, each token is contextualized but retrieving a phrase representation is less clear cut. In addition, BERT operates over subword-tokens, so in all our methods, we tried aggregating subword-tokens using their average (same as in Hewitt and Manning (2019)) or using them as provided. To extract phrases, we tried 6 different approaches: using the max, using the average, concatenating the max and the average, using the heuristic, concatenating the max with the heuristic, and concatenating the max with the CLS and EOS tokens. We tried every variant with every layer of BERT (we used the base model that has 12 layers), and tried concatenating representations from each consecutive third of layers. This resulted in $2 \times 6 \times 15 = 180$ variants of the BERT baseline.

A.3 Codebook Notes

For the codebook, we use equation:

$$x' = x + C^\top \sigma(CWx)$$

In practice, we found that for σ it is effective to use the identity function. The matrix W is a learned bi-linear mapping, but if it is an identity matrix, than the equation above reduces to a much simpler form:

$$x' = (C^\top C + I)\, x$$

When interpreted this way, there could be some interesting properties to explore or enforce in the codebook equation, although we leave this for future work.

[8] https://github.com/iesl/diora