Deep Learning Within Natural Language Inference

INTRODUCTION

Task: Natural Language Inference

Approaches:

1) BiLSTM Sentence Embeddings with Heuristics, Iterative Refinement Encoders and FNN Classifier
- **#1 non-transformer solution and #5 overall for SciTail**

2) Fine-tuned DeBERTa pre-trained on Replaced Token Detection with Gradient-Disentangled Embeddings Sharing
- **#5 and #7 solution for RTE (Recognizing Textual Entailment)**

METHODOLOGY: APPROACH 1

- Traditional NLI is a three-way classification problem, but we train the model only with 2 (contradiction and entailment).

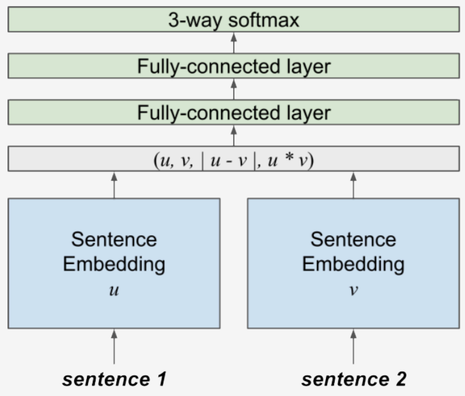

Figure 1: Approach 1 architecture [2]

- ***Heuristics*** for sentence embeddings [1]; for each sentence pair embeddings *u* and *v* we perform:
  - concatenation (*u, v*),
  - absolute element-wise difference |*u*−*v*|,
  - element-wise product *u* ⋆ *v*

- **Iterative Refinement Encoding**: input embeddings are taken (repeatedly) at each layer, as well as output from previous layer

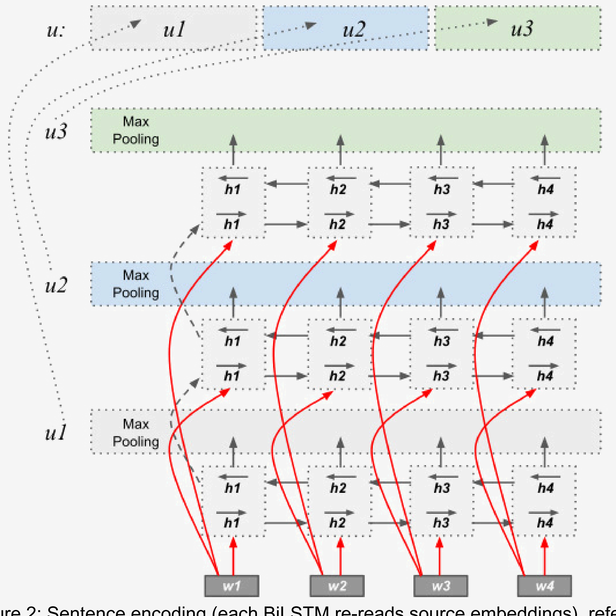

Figure 2: Sentence encoding (each BiLSTM re-reads source embeddings), referred to as *iterative refinement architecture* [2]

METHODOLOGY: APPROACH 2

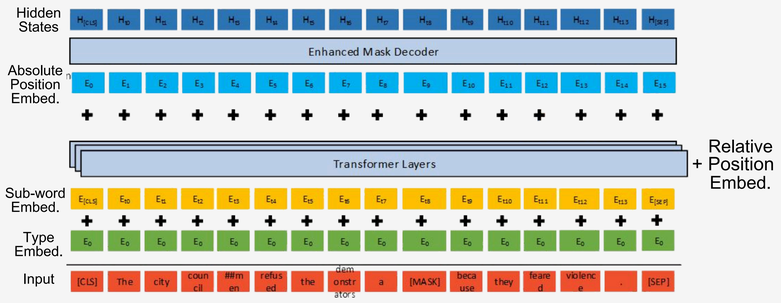

Figure 3: DeBERTa architecture [3]

- Based on **RoBERTa**.
- **Disentangled attention**: improved relative-position encoding mechanism.

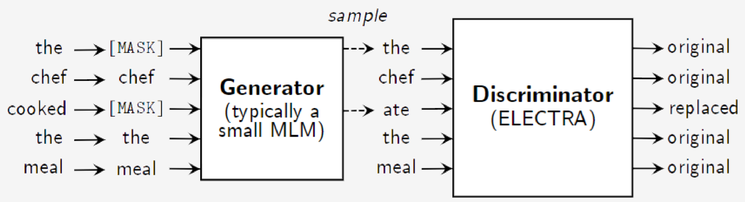

Figure 4: Replaced Token Detection (RTD) task [4]

- **Replaced Token Detection (RTD)**: generate ambiguous corruptions with a ***generator*** and distinguish the ambiguous tokens from the original inputs with a ***discriminator***.
- **Gradient-disentangled Embedding Sharing (GDES)**: the generator shares its embeddings with the discriminator but stops the gradients from the discriminator to the generator embeddings.

DATA AUGMENTATION

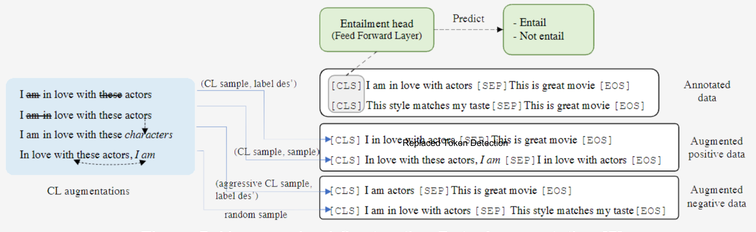

Figure 5: Unsupervised Contrastive Data Augmentation [5]

Augmentation techniques, given S1 [SEP] S2 sample:

- Generate ***positive samples*** S1 [SEP] S1' and S2 [SEP] S2', by creating S1' and S2' from S1 and S2 by applying four augmentations.
  - *word deletion*: randomly remove 10% percent of words
  - *span deletion*: randomly a two-worded span
  - *reordering*: switch positions of pair two-worded spans
  - *substitution*: sample 10% words and replace them with synonyms
- Generate ***negative samples*** P1 [SEP] P2 and H1 [SEP] H2, by creating randomly sampling from premises P and hypotheses H.

Generate 136 augmented samples from 8 original training samples and test DeBERTa model with full training set, 10% of training set, and augmented dataset. **Technique used by #1 solution for SNLI.**

EVALUATION

Hyperparameter tuning by grid search, repeating three times for both.

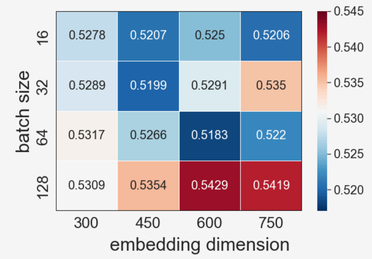

Figure 6: BiLSTM sentence embedding grid search using validation loss (averaged)

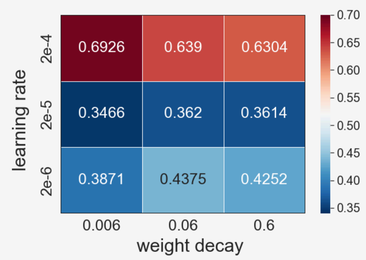

Figure 7: DeBERTa grid search using validation loss (averaged)

BiLSTM Embeddings Approach

Gridspace defined by:
- batch size: {16, 32, 64, 128}
- embed. dim: {300, 450, 600, 750}
- epochs: 1-8

Best setup:
- **batch size: 128**
- **embed. dim: 600**
- **epochs: 2**

DeBERTa Approach

Gridspace defined by:
- learning rate: {2e-4, 2e-5, 2e-6}
- weight decay: {0.6, 0.06, 0.006}
- epochs: 1-6

Best setup:
- **learning rate: 2e-5**
- **weight decay: 0.006**
- **epochs: 2**

Both models evaluated for 20 epochs; DeBERTa evaluated with three different datasets.

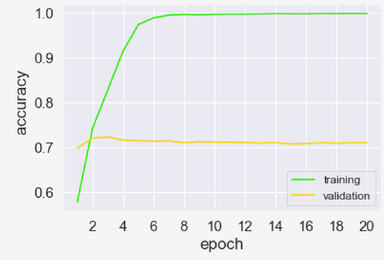

Figure 8: BiLSTM embeddings, FNN classifier training and validation accuracy over 20 epochs

Figure 9: DeBERTa validation accuracy over 20 epoch for three training datasets

	BiLSTM Embeddings w/ FNN	Fine-tuned DeBERTaV3
Validation Acc	72.09 (± 0.1)	90.78 (± 0.08)

Figure 10: Validation accuracy for both models with optimal hyperparameter setups

CONCLUSION

Deep learning continues to be SOTA for NLI, and while it has been shown bigger models have the capacity for better performance, it will be more intriguing to see how models can learn more efficiently, rather than become bigger.

REFERENCES

[1] Mou, L., et. al, (2016). Natural Language Inference by Tree-Based Convolution and Heuristic Matching. In The 54th Annual Meeting of the Association for Computational Linguistics, pp.130-137.

[2] Talman, A., et. al, (2019). Sentence embeddings in NLI with iterative refinement encoders. Natural Language Engineering, 25, pp.467-482.

[3] He, P., et. al, (2021). Microsoft DeBERTa surpasses human performance on the SuperGLUE benchmark. Microsoft Research Blog.

[4] Clark, K., et. al, (2019). ELECTRA: Pre-training Text Encoders as Discriminators Rather Than Generators. In: International Conference on Learning Representations, pp.90-108.

[5] Wang, S., et. al, (2021). Entailment as Few-Shot Learner. arXiv:2104.14690. https://arxiv.org/pdf/2104.14690.pdf