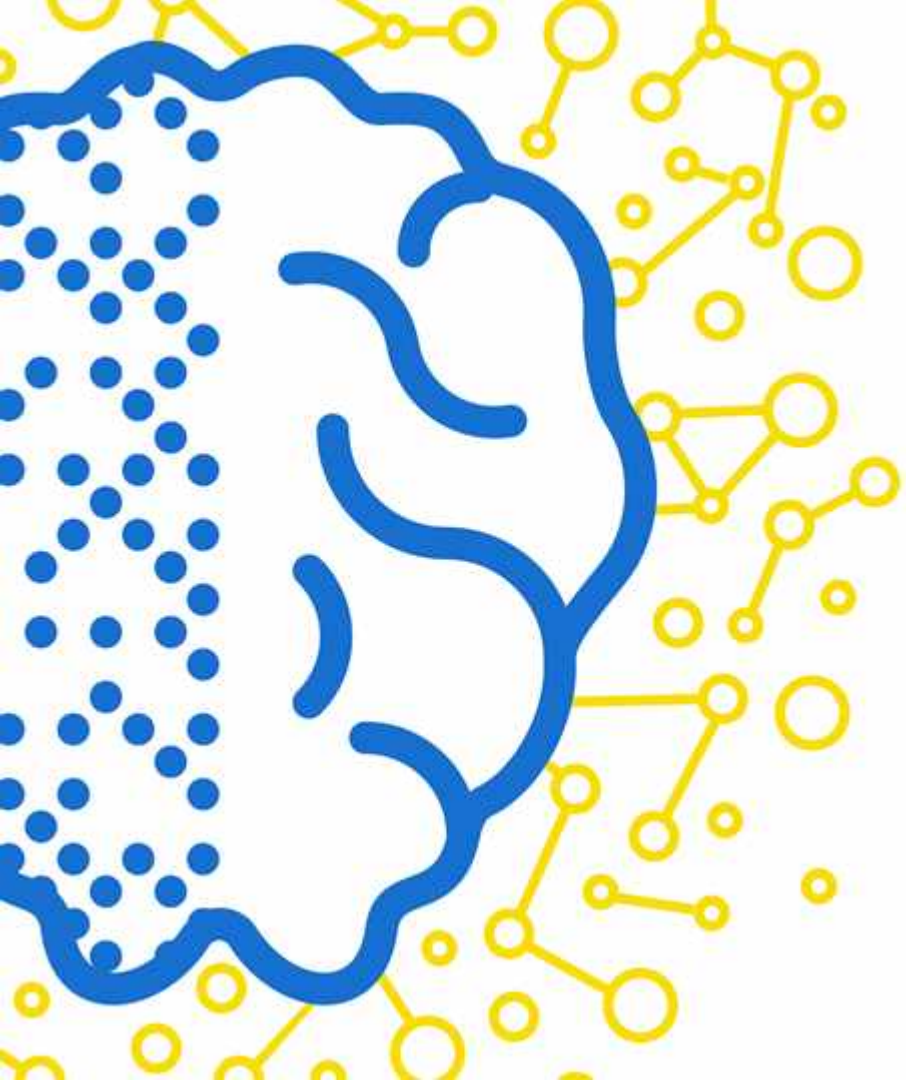

Data pseudo-labeling while adapting BERT for multi-task setting

Dmitry Karpov
Mikhail Burtsev
MIPT

Why do we need to pseudo-label data?

BERT models are widely used for text classification

But: we need to solve several classification tasks

Use a separate BERT for every task? Computationally expensive

Train one BERT for all tasks? But… what about labels?

Labels for task A are labeled only for classes from task A, labels for task B - only for classes from task B, etc.

These labels may or may not overlap.

What was done?

- Different pseudo-labeling methods to train original BERT-Base (without changing architecture) on the following GLUE tasks:
  MNLI, QQP, RTE, SST
- For every method: train on 3 epochs, learning rates 2e-5, 3e-5, 4e-5, 5e-5
- Explored 7 different settings, except for the default one

What settings were explored?

Independent labels - all labels are treated as independent, so probabilities of all classes except for the target one are set to zero e.g

$$prob^{\varepsilon}_{RTE} = [1_{\varepsilon}, 0_{!\varepsilon}, 0_e, 0_c, 0_n, 0_d, 0_{!d}, 0_+, 0_-]$$

Soft independent labels - just like **Independent labels**, but probabilities of all classes from the tasks where we don't know labels are set as equal e.g

$$prob^{e}_{MNLI} = [1/2_{\varepsilon}, 1/2_{!\varepsilon}, 1_e, 0_c, 0_n, 1/2_d, 1/2_{!d}, 1/2_+, 1/2_-]$$

Augmented independent labels - just like **Soft independent labels**, but probabilities of all classes from the tasks where we don't know labels are drawn from pretrained model, e.g

$$prob^{d}_{QQP} = [RTEpred_{\varepsilon}, RTEpred_{!\varepsilon}, MNLIpred_e, MNLIpred_c, MNLIpred_n, 1_d, 0_{!d}, SSTpred_+, SSTpred_-]$$

Independent labels, frozen head - the same as **Independent labels**, but with frozen weights of last linear classification layer

Soft independent labels, frozen head - the same as **Soft independent labels**, but with frozen weights of the last linear classification layer

What settings were explored?

On the following 3 settings, we use only 5 classes - positive,negative,entailment,contradiction,neutral.

Soft probability assumption - unite entailment and duplicate labels, not entailment and not duplicate, default probs are as in Soft independent labels e.g

$$prob_{+}^{SST} = [1/3_e, 1/3_c, 1/3_n, 1_+, 0_-]$$

$$prob_{MNLI}^{c} = [0_e, 1_c, 0_n, 1/2_+, 1/2_-]$$

$$prob_{RTE}^{\varepsilon} = prob_{QQP}^{d} = prob_{MNLI}^{e} = [1_e, 0_c, 0_n, 1/2_+, 1/2_-]$$

$$prob_{RTE}^{!\varepsilon} = prob_{QQP}^{!d} = [0_e, 1/2_c, 1/2_n, 1/2_+, 1/2_-]$$

Soft predicted labels - same as Soft probability assumption, but default probs are drawn from the MNLI and SST model prediction e.g

$$prob_{RTE}^{\varepsilon} = prob_{MNLI}^{e} = prob_{QQP}^{d} = [1_e, 0_c, 0_n, SSTpred_+, SSTpred_-]$$

$$prob_{RTE}^{!\varepsilon} = prob_{QQP}^{!d} = [0_e, MNLIpred_c^{!e}, MNLIpred_n^{!e}, SSTpred_+, SSTpred_-]$$

$$prob_{MNLI}^{n} = [0_e, 0_c, 1_n, SSTpred_+, SSTpred_-]$$

Hard predicted labels - same as **Soft predicted labels**, but from labels received from the original model prediction the maximal probability for each task is rounded to 1, all other probabilities are rounded to 0.

Test results

Setting name	Average by 4 tasks	RTE	QQP	MNLI(m/mm)	SST
Plain(from original article)	78.8	66.4	71.2	84.6/83.4	93.5
Plain(reproduced)	77.6	62.7	71.0	83.1/ 82.7	93.5
Independent labels	79.0	71.5	70.9	82.7/81.7	91.3
Soft independent labels	78.9	69.3	71.3	82.8/ 82.1	92.6
Augmented independent labels	77.6	64.2	**71.8**	81.2/ 80.7	**93.2**
Soft probability assumption	**79.7**	**72.7**	70.7	**83.4/82.3**	92.5
Soft predicted labels	78.8	70.3	70.7	81.7/ 81.7	92.5
Hard predicted labels	79.1	71.3	71.1	81.7/ 81.4	92.6
Independent labels frozen head	78.2	66.9	**71.8**	82.6/81.8	91.9
Soft independent labels frozen head	79.1	70.0	71.5	83.0/ **82.3**	92.4

Results

- **Soft probability assumption** method is the best for the most similar tasks: RTE and MNLI, so while we solve similar tasks we can unite labels.

- But this method constrained by the similarity of tasks. Also, the absence of the accuracy growth on QQP while uniting labels can tell that unification in this task was too rough. It shows the constraints for the proposed method with label unification.

- On different tasks, such as SST and QQP,**Augmented independent labels** method yields the best result, which is explained by the effect of knowledge transfer while solving different tasks.From the considered methods, it is the best choice, as it can be expanded to a great variety of tasks.

Thanks for your attention