Data pseudo-labeling while adapting BERT for multitask approaches

Karpov Dmitry
Moscow Institute of Physics
and Technology
Dolgoprudny, Russia
dimakarp19962008@yandex.ru

Burtsev Mikhail
Moscow Institute of Physics
and Technology
Dolgoprudny, Russia
burtcev.ms@mipt.ru

Abstract

Nowadays, BERT models have found wide use in the NLP field. However, standard BERT architecture training can be stifled by the lack of labels for different tasks while treating multitask settings as a one-task multilabel setting. For every example, we have labels from this example's source task but not from other tasks. This article addressed this issue, exploring eight different data pseudo-labeling approaches in the GLUE 4-task setting. These approaches do not require changes in samples or model architecture. One of the presented techniques excels results on RTE from the original article, by 6.2 %, and falls behind the original article on QQP, MNLI, and SST only by 0.5-1.2 %. This way also excels other pseudo-labeling approaches explored in the article by 0.5-2% on average if we consider similar tasks. However, for tasks that are dissimilar to each other, different proposed approach yields the best results.

Keywords: BERT, multitask, data augmentation, pseudo-labeling
DOI: 10.28995/2075-7182-2021-20-358-366

Псевдоразметка данных при адаптации архитектуры BERT для многозадачных подходов

Карпов Дмитрий
dimakarp19962008@yandex.ru

Бурцев Михаил
burtcev.ms@mipt.ru

Московский физико-технический институт
Долгопрудный, Россия

Аннотация

В настоящее время в обработке естественного языка широко используются модели типа BERT. Однако обучение стандартной архитектуры BERT при многозадачном подходе бывает затруднено недостатком меток для разных задач. В статье исследуются восемь различных способов псевдоразметки данных при обучении на нескольких задачах типа GLUE, не требующих изменений ни в наборе примеров, ни в архитектуре. В частности, представлен такой способ псевдоразметки данных для обучения оригинальной модели для решения четырех задач типа GLUE, который превосходит результаты из оригинальной статьи на датасете RTE на 6.2 % и отстает от нее на QQP, MNLI и SST только на 0.5-1.2 %. Способ превосходит другие методы псевдоразметки, рассмотренные в статье, в среднем на 0.5-2% на похожих задачах, но на разнородных задачах лучше работает другой из предложенных способов.

Ключевые слова: BERT, многозадачная модель, дополнение данных, псевдоразметка

1 Introduction

Transformer-based models, such as BERT, have found their wide use in the task of text classification. Conditions of learning such models are described in the original article [3]. These conditions suppose fitting every model for its task. In such a way, if we need to solve several classification tasks in parallel, we should keep several models for prediction, which increases the demand for computation power. This problem leads us to the idea of training one model that can yield the result for several tasks simultaneously. We explore the ways of training such a model without architecture changes.

2 Literature review

Article [8] shows the approach to auto-selecting tasks while training linear models. However, the authors evaluated the way proposed in the article only for the custom binary classification dataset. This way also cannot be directly compared to the more novel results due to the evaluated model's simplicity. Articles [9], [6] and [10] show the ways of fitting BERT on several tasks at once. However, the ways described there still require utilizing more complex architectures compared to single-task learning. The most basic way of fitting a single-task BERT, described in this article, is the following. We tune the multitask BERT without changing the base model architecture and data the model used for fine-tuning. We only change the available labels and possibly freeze some weights. In [1] while processing images, authors used predictions of models fit on 1 percent of training data for pseudo-labeling (assigning labels for unlabeled samples). In [2] authors used such a pseudo-labeling method (while having models fit for different languages) as data translation from one language to another and backward, and in [7] authors explored the impact of pseudo-labeling approaches for the computer vision tasks as well as for the machine translation. However, we did not find research comparing different pseudo-labeling ways for fitting BERT on the GLUE benchmark[5], so this article fills in this gap. The simplest way of fitting multitask BERT, described in this article, is fitting the multitask model without changing its architecture and data given as an input, but only with editing an array of labels for the multiclass task.

3 Experiments setting

In this article, we have researched different methods of training the BERT model for solving various classification tasks simultaneously. The unique feature of every considered approach is that, unlike multitask learning methods such as [10] and [9], we do not change model architecture. But the only thing we change is the array of labels sent as input. In every approach, we trained the model that solves the multilabel classification problem. Specifically, this model predicts probability from 0 to 1 for every class. In this work, we research the quality of pseudo-labeling for such tasks. We have evaluated the model on the following classification tasks: MNLI, Quora Question Pairs (further - QQP), SST-2, and RTE for the GLUE set of tasks[4]. We have chosen MNLI, QQP, and SST-2 as their datasets were large enough (>= 50000 samples for every task). We also have chosen RTE as we need to get entailment in the same way as in MNLI. We have reproduced original article results for each of these tasks. Note that the BERT model in the original article was not multilabel. The examples from all tasks were shuffled and sampled randomly. We should note that we used the BERT-Base model as a benchmark due to the computational restrictions.

4 Notations

We use the following labels in the formulas described in the article:

- $+, -$: labels *positive* and *negative* for SST dataset;
- $d, !d$: labels *duplicate* and *not duplicate* from Quora question pairs dataset;
- e, c, n: labels *entailment*, *contradiction* and *neutral* from MNLI dataset;
- $\varepsilon, !\varepsilon$: labels *entailment* and *not entailment* from RTE dataset;
- $MNLIpred, RTEpred, QQPpred, SSTpred$ - predictions of the model trained on the following task (MNLI, RTE, QQP, SST) for the label from the lower formula index;
- I denotes rounding of probability vector predicted by the original model: we round the largest element of the probability vector to 1 and all other elements to 0;
- $MNLIpred^{!e}$ is the prediction of the plain MNLI model with entailment set to zero. So, we set the predicted probability of $entailment$ to zero and then treat the 3-class classification as 2-class while predicting the plain single-label MNLI model (for classes entailment, contradiction, and neutral);
- $prob_{task}^{label}$ is the vector with probabilities from 0 to 1 that we need to assign to the example from $task$, which was labeled as $label$;
- P_{label} is the probability P of label $label$, where P is from 0 to 1.

It means that, for example:

- $MNLIpred_e$ - is the probability of entailment label, predicted by the model trained on MNLI;
- $I(MNLIpred)_e$ means that it is 1 if entailment is most likely predicted class in the MNLI task, and 0 otherwise.

$$MNLIpred_n^{!e} = MNLIpred^n/(MNLIpred^n + MNLIpred^c) \tag{1}$$

$$MNLIpred_c^{!e} = MNLIpred^c/(MNLIpred^n + MNLIpred^c) \tag{2}$$

We denote the components of probability vectors using brackets.

5 Multitask approaches explored

We considered different approaches for fitting multitask models. We present these methods below.

5.1 Independent labels

In this approach, we fit the model on the united array of RTE, MNLI, QQP, and SST-2. For every example, we consider label arrays for each of the tasks to be independent. In other words, we set for every sample the probability of absolutely all classes, except for already known, as 0. We also set the likelihood of an already known class as 1 (or 100 percent) for every sample. There are nine classes: 3 classes for the MNLI task and two classes for each other tasks. It means that default probability vector for this setting is:

$$prob_{default} = [0_\varepsilon, 0_{!\varepsilon}, 0_e, 0_c, 0_n, 0_d, 0_{!d}, 0_+, 0_-] \tag{3}$$

So for every label, we change in this default vector only the probability of the "correct" label to 100 percent, for example:

$$prob_{RTE}^{\varepsilon} = [1_\varepsilon, 0_{!\varepsilon}, 0_e, 0_c, 0_n, 0_d, 0_{!d}, 0_+, 0_-] \tag{4}$$

Other equations can be written in an analogous way.

5.2 Soft independent labels

This approach is analogous to the **Independent labels**. However, it has the following difference: we do not take down to zero probabilities of absolutely all classes we do not know for every sample. Instead, we take down to zero only the probability of all classes except for the known label for the "own" task. The probabilities of all other classes are labeled to be the same. To label them, we used the following rule: the sum of probabilities of all other classes must be equal to 1, and the probabilities of all other classes (for each task) must be equal to each other. We can quickly obtain probability coefficients for every class from the "other" task if we know these conditions. It means that default probability vector for this setting is:

$$prob_{default} = [1/2_\varepsilon, 1/2_{!\varepsilon}, 1/3_e, 1/3_c, 1/3_n, 1/2_d, 1/2_{!d}, 1/2_+, 1/2_-] \tag{5}$$

So for every label, we change in this default vector only the probability of the "correct" label to 100 percent and the probability of the "incorrect" labels from this task to 0 percent, for example:

$$prob_{MNLI}^{e} = [1/2_\varepsilon, 1/2_{!\varepsilon}, 1_e, 0_c, 0_n, 1/2_d, 1/2_{!d}, 1/2_+, 1/2_-] \tag{6}$$

We can write other equations analogously.

5.3 Augmented independent labels

This approach is similar to the **Independent labels** and **Soft independent labels**. However, it has the following difference. For every sample, we do not consider the probability of every class from a "different" task to be the same, but instead, we define it by the prediction of the base model. The base model was trained preliminarily on this "different" task to reproduce the original article results.

It means that default probability vector for this setting is:

$$prob_{default} = [RTEpred_{\varepsilon}, RTEpred_{!\varepsilon}, MNLIpred_{e}, MNLIpred_{c}, MNLIpred_{n}, QQPpred_{d}, QQPpred_{!d}, SSTpred_{+}, SSTpred_{-}] \quad (7)$$

So for every label, we change in this default vector only the probability of the "correct" label to 100 percent and the probability of the "incorrect" labels from this task to 0 percent, for example:

$$prob^{d}_{QQP} = [RTEpred_{\varepsilon}, RTEpred_{!\varepsilon}, MNLIpred_{e}, MNLIpred_{c}, MNLIpred_{n}, 1_{d}, 0_{!d}, SSTpred_{+}, SSTpred_{-}] \quad (8)$$

We can write other equations analogously.

5.4 Soft probability assumption

In this setting, as well in the settings **Independent labels**, **Soft independent labels** and **Augmented independent labels**, we trained the model on the united array of RTE, MNLI, QQP, and SST-2. However, we considered labels for these tasks to be dependent. Specifically, we downsized the number of classes for the model to 5: *positive*, *negative*, *entailment*, *contradiction*, *neutral*. We have transferred classes of that datasets to the probabilities of these five classes by the following rules. We consider labels from RTE, MNLI, and QQP to be 50 percent positive and 50 percent negative, and in that time:

- We use MNLI labels "as it is" for classes *entailment*, *contradiction* and *neutral*: one of the classes *entailment/neutral/contradiction* has probability 100 percent and other two have probability 0 percent.
- We consider QQP label *duplicate* to be *entailment* with probability 100 percent and *neutral/contradiction* with 0 percent probability. The label *not duplicate* is considered to be *neutral/contradiction* with probabilities 0.5 and 0.5 (as they need to be equal to each other and their sum must be equal to 1), and its probability to be *entailment* is set to 0.
- We consider RTE label *entailment* to be entailment with probability 100 percent and neutral/contradiction with zero probability. The label *not entailment* is considered to be *neutral/contradiction* with probabilities 0.5 and 0.5(as they need to be equal to each other and their sum must be equal to 1), and its probability to be *entailment* is set to 0.

We consider all labels on SST-2 as belonging to classes *entailment/neutral/contradiction* with the same probability equal to 1/3. At the same time, we assign labels positive/negative according to the initial SST-2 dataset.

It means that default probability vector for this setting is:

$$prob_{default} = [1/3_{e}, 1/3_{c}, 1/3_{n}, 1/2_{+}, 1/2_{-}] \quad (9)$$

And for examples from SST task and MNLI task, we just set in the default vector the probabilities of "correct" labels to 1 and of "incorrect" labels to 0, respectively, for example:

$$prob^{SST}_{+} = [1/3_{e}, 1/3_{c}, 1/3_{n}, 1_{+}, 0_{-}] \quad (10)$$

$$prob^{c}_{MNLI} = [0_{e}, 1_{c}, 0_{n}, 1/2_{+}, 1/2_{-}] \quad (11)$$

For RTE task and QQP task, we handle entailment in an analogous way to the:

$$prob^{\varepsilon}_{RTE} = prob^{d}_{QQP} = prob^{e}_{MNLI} = [1_{e}, 0_{c}, 0_{n}, 1/2_{+}, 1/2_{-}] \quad (12)$$

However, we handle "not entailment" from RTE and "not duplicate" from QQP differently:

$$prob^{!\varepsilon}_{RTE} = prob^{!d}_{QQP} = [0_e, 1/2_c, 1/2_n, 1/2_+, 1/2_-] \tag{13}$$

5.5 Soft predicted labels

This approach is analogous to the **Soft probability assumption** with the following difference. We obtain the missing labels (*contradiction/neutral* and positive/negative on tasks RTE and QQP, positive/negative on the MNLI task, *entailment/contradiction/neutral* on the SST-2 task) by the additional labeling made by the model for each task, specifically:

- If an example is not from the SST-2, we get positive/negative labels from the SST-2 model. Otherwise, we get them from the original dataset;
- If an example is from MNLI, we get labels *entailment*, *contradiction*, or *neutral* from the original dataset;
- If an example is from RTE with the label *entailment* or from QQP with the label *duplicate*, we assign the label *entailment* with probability 1 in an analogous way to the previous point;
- If an example is from RTE with the label *not entailment* or from QQP with the label *not duplicate*, we assign the probability of label *entailment* as 0. In that way, we also take probabilities of label *contradiction* or *neutral* from predictions of the model trained on MNLI, and we normalize that probabilities for the sum of probability of *contradiction* and the probability of *neutral*.

It means that default probability vector for this setting is:

$$prob_{default} = [MNLIpred_e, MNLIpred_c, MNLIpred_n, SSTpred_+, SSTpred_-] \tag{14}$$

For the SST task and MNLI task, we just set in this vector the probabilities of "correct" labels to 1 and of "incorrect" labels to 0, respectively, in the same way as in previous approach, for example:

$$prob^n_{MNLI} = [0_e, 0_c, 1_n, SSTpred_+, SSTpred_-] \tag{15}$$

And the formulas for probabilities of RTE and QQP tasks look as the following:

$$prob^{\varepsilon}_{RTE} = prob^{e}_{MNLI} = prob^{d}_{QQP} = [1_e, 0_c, 0_n, SSTpred_+, SSTpred_-] \tag{16}$$

$$prob^{!\varepsilon}_{RTE} = prob^{!d}_{QQP} = [0_e, MNLIpred^{!e}_c, MNLIpred^{!e}_n, SSTpred_+, SSTpred_-] \tag{17}$$

5.6 Hard predicted labels

This approach is analogous to the **Soft predicted labels** approach with the following changes. From labels received from the original model prediction ($MNLIpred$, $SSTpred$, $QQPpred$, $RTEpred$), the maximal probability for each task is rounded to 1, all other probabilities are rounded to 0.

It means that default probability vector for this setting is:

$$prob_{default} = [I(MNLIpred_e), IMNLIpred_c), I(MNLIpred_n), I(SSTpred_+), I(SSTpred_-)] \tag{18}$$

And for the SST task and MNLI task, we make in this vector the same changes as in previous approaches. Changes for QQP task and RTE task look in the following way:

$$prob^{\varepsilon}_{RTE} = prob^{d}_{QQP} = [1_e, 0_c, 0_n, I(SSTpred_+), I(SSTpred_-)] \tag{19}$$

$$\begin{aligned} prob^{!\varepsilon}_{RTE} = prob^{!d}_{QQP} = [0_e, MNLIpred^{!e}_c, MNLIpred^{!e}_n, \\ I(SSTpred_+), I(SSTpred_-)] \end{aligned} \tag{20}$$

Setting name	Average by 4 tasks	RTE	QQP	MNLI	SST
Plain(reproduced)	81.3	64.6	90.8	77.3	92.7
Independent labels	82.8	**78.3**	90.6	75.8	92.0
Soft independent labels	82.2	69.7	89.5	75.9	**92.6**
Augmented independent labels	81.4	68.1	90.5	75.6	92.4
Soft probability assumption	**84.2**	78.2	**90.7**	76.2	91.9
Soft predicted labels	83.2	76.3	90.5	76.0	92.2
Hard predicted labels	82.9	77.4	90.6	75.3	90.7
Independent labels frozen head	82.5	76.1	90.5	75.7	91.4
Soft independent labels frozen head	82.6	74.4	90.4	**76.7**	91.2

Table 1: Best accuracy on validation data (best learning rate for every setting, average by 3 runs)

5.7 Independent labels frozen head

This approach is the same as **Independent labels**, with the following exception: the head of the model (linear layer for classification) does not learn; only the body does. Formulas for this approach are the same as for **Independent labels**.

5.8 Soft independent labels frozen head

This approach is the same as **Soft independent labels**, with the following exception: the head of the model (linear layer for classification) does not learn; only the body does.

Formulas for this approach are the same as for **Soft Independent labels**.

6 Results

We have made four reproduction attempts for every approach described above, including reproducing the original article results. In a similar way to the original article, these attempts had learning rates 2e-5, 3e-5, 4e-5, and 5e-5 accordingly. As the final learning rate, we chose the learning rate for which we had the maximal accuracy on the validation set. We have defined accuracy on the validation set as the average accuracy for all four tasks. We restricted the learning to 3 epochs for all tasks. Complete obtained data or the validation set are attached below in the Appendix A. Results on the validation set and the test set are described below in Table 1 and Table 2. We should note that we achieved all these results with 10-13 % fewer parameters and without any changes to basic architecture despite the yielding lower results than [10].

7 Discussion

As we can see, considered methods yield results similar to the original BERT model results, or even better if we descry the RTE task. The reason for this exceeding for the RTE task is its similarity with other tasks from the GLUE benchmark, for which we have much more data than for RTE.

From all considered methods, **Soft probability assumption** method yields the best results on the most similar tasks: RTE and MNLI. This result shows that uniting labels while solving similar tasks is justified.

However, on different tasks, such as SST and QQP, **Augmented independent labels** method yield the best result, which is explained by the effect of knowledge transfer while solving different tasks. Nonetheless, this effect was weakly expressed or even absent while we united labels, and its reason remains unclear. 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.

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

Table 2: Best accuracy on test data (best F1 on the QQP task)

Notably, the similarity of tasks poses some constraints on applying the **Soft probability assumption**. If they were entirely dissimilar, we could not unite the labels in this task. Therefore, in that case, **Augmented independent labels** would have been the best choice, as it can be expanded to a great variety of tasks. Exploring the broader range of architectures for which this conclusion remains valid will be the subject of future research. We also leave unexplored the impact of different sampling ways on the process of learning the model. Looking at how the result varies when we try the same sampling ways as in [4] is also a subject of future research.

8 Conclusion

After considering eight different data pseudo-labeling approaches in the GLUE 4-task setting, we can single out the method **Soft probability assumption** as the best for similar tasks such as RTE and MNLI. This method excels results on RTE from the original article by 6.2 % and falls behind the original article on QQP, MNLI, and SST only by 0.5-1.2 %. However, method **Augmented independent labels** works as the best method for solving different tasks such as SST and QQP.

Acknowledgements

This work was supported by National Technology Initiative and PAO Sberbank project ID 0000000007417F63000.

References

[1] Arachie Chidubem, Huang Bert. Constrained Labeling for Weakly Supervised Learning // arXiv preprint arXiv:2009.07360. — 2021.

[2] Aroyehun Segun Taofeek, Gelbukh Alexander. Aggression Detection in Social Media: Using Deep Neural Networks,Data Augmentation, and Pseudo Labeling.Proceedings of the First Workshop on Trolling // Proceedings of the First Workshop on Trolling, Aggression and Cyberbullying. — 2018. — P. 90:97. — Access mode: https://www.aclweb.org/anthology/W18-4411.pdf.

[3] BERT: Pre-training of deep bidirectional transformers for language understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — 2019. — P. 4171:4186. — Access mode: https://arxiv.org/abs/1810.04805.

[4] Dynamic Sampling Strategies for Multi-Task Reading Comprehension / Ananth Gottumukkala, Dheeru Dua, Sameer Singh, Matt Gardner // Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. — 2020. — P. 920:924. — Access mode: https://www.aclweb.org/anthology/2020.acl-main.86/.

[5] GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding / Alex Wang, Amanpreet Singh, Julian Michael et al. // Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP. — 2018. — P. 353:355. — Access mode: https://arxiv.org/abs/1804.07461.

[6] Multi-Task Deep Neural Networks for Natural Language Understanding / Xiaodong Liu, Pengcheng He, Weizhu Chen, Jianfeng Gao // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — 2019. — P. 4487:4496. — Access mode: https://www.aclweb.org/anthology/P19-1441/.

[7] Müller Rafael, Kornblith Simon, Hinton Geoffrey. When Does Label Smoothing Help? // Proceedings of the 33th NeurIPS. — 2020. — Access mode: https://proceedings.neurips.cc/paper/2019/file/f1748d6b0fd9d439f71450117eba2725-Paper.pdf.

[8] Pentina Anastasia, Lampert Christoph H. Multi-Task Learning with Labeled and Unlabeled Tasks // Proceedings of the 34th International Conference on Machine Learning. — Vol. 70. — 2017. — P. 2807:2816. — Access mode: http://proceedings.mlr.press/v70/pentina17a.html.

[9] Pilault Jonathan, Elhattami Amine, Pal Christopher. Conditionally Adaptive Multi-Task Learning: Improving Transfer Learning in NLP Using Fewer Parameters Less Data. — 2020. — Access mode: 2009.09139.

[10] Stickland Asa Cooper, Murray Iain. BERT and PALs: Projected Attention Layers for Efficient Adaptation in Multi-Task Learning // Proceedings of the 36th International Conference on Machine Learning. — Vol. 97. — 2019. — P. 5986:5995. — Access mode: https://arxiv.org/abs/1902.02671.

9 Appendix A. Validation set accuracies for different attempts

RTE valid accuracies

Setting name	Attempt 1	Attempt 2	Attempt 3	Attempt 4
Plain(reproduced)	64.6	57.8	63.2	62.7
Independent labels	66.8	68.2	66.4	74.0
Soft independent labels	71.1	69.3	65.0	66.8
Augmented independent labels	67.5	67.1	64.6	65.3
Soft probability assumption	76.9	76.9	71.8	71.1
Soft predicted labels	72.6	74.4	70.8	73.3
Hard predicted labels	73.3	71.8	72.9	72.5
Independent labels frozen head	70.0	72.9	70.8	69.3
Soft independent labels frozen head	71.8	66.4	65.0	67.9

SST valid accuracies

Setting name	Attempt 1	Attempt 2	Attempt 3	Attempt 4
Plain(reproduced)	92.7	92.1	91.7	89.3
Independent labels	91.6	91.1	90.7	90.4
Soft independent labels	90.8	91.4	91.5	89.5
Augmented independent labels	92.3	91.5	91.5	91.7
Soft probability assumption	92.3	91.6	90.5	90.4
Soft predicted labels	92.1	91.9	91.2	90.5
Hard predicted labels	92.4	91.4	90.5	90.8
Independent labels frozen head	89.9	90.6	90.8	91.9
Soft independent labels frozen head	91.6	90.9	89.1	90.5

QQP valid accuracies

Setting name	Attempt 1	Attempt 2	Attempt 3	Attempt 4
Plain(reproduced)	90.8	90.5	87.1	89.8
Independent labels	87.9	89.1	87.5	90.4
Soft independent labels	90.4	89.8	86.0	89.1
Augmented independent labels	90.3	89.6	90.4	90.3
Soft probability assumption	90.5	90.3	90.0	89.6
Soft predicted labels	90.0	90.4	90.1	90.0
Hard predicted labels	90.1	89.9	89.3	89.9
Independent labels frozen head	90.1	90.5	90.5	89.6
Soft independent labels frozen head	90.2	90.5	89.9	89.1

MNLI valid accuracies

Setting name	Attempt 1	Attempt 2	Attempt 3	Attempt 4
Plain(reproduced)	76.7	77.3	76.4	72.7
Independent labels	73.8	75.0	72.6	76.3
Soft independent labels	76.5	75.8	72.62	74.5
Augmented independent labels	75.5	75.1	75.3	75.0
Soft probability assumption	77.0	76.4	75.4	75.3
Soft predicted labels	75.9	76.0	76.0	75.5
Hard predicted labels	75.9	75.4	75.0	74.5
Independent labels frozen head	76.1	76.0	76.3	73.6
Soft independent labels frozen head	76.8	76.2	75.0	73.5