applied sciences

MDPI

Article

A Sequential and Intensive Weighted Language Modeling Scheme for Multi-Task Learning-Based Natural Language Understanding

Suhyune Son [1,†], Seonjeong Hwang [1,†], Sohyeun Bae [1,†], Soo Jun Park [2] and Jang-Hwan Choi [3,*]

[1] Computer Science and Engineering, College of Engineering, Ewha Womans University, Seoul 03760, Korea; ssh5131@ewhain.net (S.S.); tjswjd0228@ewhain.net (S.H.); webby10@ewhain.net (S.B.)
[2] Welfare & Medical ICT Research Department, Electronics and Telecommunications Research Institute, Daejeon 34129, Korea; psj@etri.re.kr
[3] Division of Mechanical and Biomedical Engineering, Graduate Program in System Health Science and Engineering, College of Engineering, Ewha Womans University, Seoul 03760, Korea
* Correspondence: choij@ewha.ac.kr; Tel.: +82-2-3277-6945
† These authors contributed equally to this work.

Abstract: Multi-task learning (MTL) approaches are actively used for various natural language processing (NLP) tasks. The Multi-Task Deep Neural Network (MT-DNN) has contributed significantly to improving the performance of natural language understanding (NLU) tasks. However, one drawback is that confusion about the language representation of various tasks arises during the training of the MT-DNN model. Inspired by the internal-transfer weighting of MTL in medical imaging, we introduce a Sequential and Intensive Weighted Language Modeling (SIWLM) scheme. The SIWLM consists of two stages: (1) Sequential weighted learning (SWL), which trains a model to learn entire tasks sequentially and concentrically, and (2) Intensive weighted learning (IWL), which enables the model to focus on the central task. We apply this scheme to the MT-DNN model and call this model the MTDNN-SIWLM. Our model achieves higher performance than the existing reference algorithms on six out of the eight GLUE benchmark tasks. Moreover, our model outperforms MT-DNN by 0.77 on average on the overall task. Finally, we conducted a thorough empirical investigation to determine the optimal weight for each GLUE task.

Keywords: language modeling; natural language understanding; neural networks; multi-task learning; supervised learning

Citation: Son, S.; Hwang, S.; Bae, S.; Park, S. J.; Choi, J.-H. A Sequential and Intensive Weighted Language Modeling Scheme for Multi-Task Learning-Based Natural Language Understanding. *Appl. Sci.* **2021**, *11*, 3095. https://doi.org/10.3390/app11073095

Academic Editor: Maxim Mozgovoy

Received: 26 February 2021
Accepted: 29 March 2021
Published: 31 March 2021

1. Introduction

The importance of natural language understanding (NLU) technology based on deep learning is being emphasized. Rule-based NLU is the previous approach of NLU that relied on human-generated features. Therefore, it could not know about the unknown words and requires significant time to extract the features. Conversely, deep-learning-based NLU automatically extracts features and learns language representations [1]. Many previous studies have demonstrated improved performance in various NLU tasks using deep learning. However, as various deep-learning-based NLU models have been proposed, datasets and indicators that can objectively evaluate them are needed. Accordingly, the General Language Understanding Evaluation (GLUE) benchmark [2], which offers diverse genres of NLU datasets, was introduced. Based on experiments that applied several deep learning architectures (e.g., BiLSTM and ELMo [3]), the GLUE dataset became the most popular measure of multi-task language models.

In recent years, multi-task learning (MTL) models—which, having been developed from single-task learning, learn a general representation from various tasks' datasets and prevent overfitting of a specific task—have received considerable attention. Representative MTL models include the multi-task deep neural network (MT-DNN) [4] and bidirectional

associative memory (BAM) [5]. In MT-DNN, various tasks are simultaneously learned by passing through shared layers and each task-specific layer. In contrast, BAM applies a knowledge-distillation method that uses single-task models to teach a multi-task model. However, MTL models have some drawbacks in performing a central task. The model may become confused because all tasks are simultaneously trained with the same weights. Moreover, the model cannot learn the central task thoroughly during MTL.

We compensate for these limitations by proposing a Sequential and Intensive Weighted Language Modeling (SIWLM) scheme, inspired by Yang et al. [6]. SIWLM consists of Sequential Weighted Learning (SWL) and Intensive Weighted Learning (IWL). In SIWLM, central task and auxiliary tasks have an initial task weight and all tasks have individually adjusted weights during training. These adjusted weights are multiplied by the loss functions. In SWL, the focused task is changed sequentially every 20 iterations. Furthermore, the weights of all tasks except for the focused task are multiplied by 0.1. Thus, the model can intensively learn the focused task over a given 20 iterations. In IWL, during all iterations, the central task becomes the focused task. That is, the weights of auxiliary tasks are multiplied by 0.1. Consequently, the model can sufficiently learn the central task.

We propose the optimal weight for each task belonging to the GLUE datasets. By applying that weight, the MTDNN-SIWLM achieves equal or higher performance on all GLUE datasets except CoLA and RTE than the baseline model, MT-DNN, and SMART [7]. Furthermore, we found that the optimal weight is affected more by the data size than the type of the task. Therefore, we propose a generalized range of weights based on the data size.

2. Related Work

Language models based on a pre-training and fine-tuning paradigm achieve high performance on many NLU tasks [8–10], including GLUE tasks. Bidirectional Encoder Representations from Transformers (BERT) [11], a bidirectional model that consists of attention layers, in contrast to the existing recurrent neural network (RNN) based models, has made great strides in Natural Language Processing (NLP). XLNet [12] is another pre-trained language model, combining the advantages of auto-encoding and autoregressive pre-training methods. Along with this paradigm, several other models were introduced that improved the pre-training method of BERT or changed the model structure. StructureBERT [13] extends BERT by including language structure learning to predict the order of tokens and sentences during unsupervised learning. By adding this method, it can capture word and sentence structure bidirectionally. RoBERTa [14] optimizes BERT by altering pre-training strategies and using more data to improve downstream task performance. DeBERTa [15] obtained the second-highest performance for the glue benchmark using a disentangled attention mechanism and an enhanced mask decoder.

In contrast to these approaches, MTL is a learning method that aims to improve the overall performance of several tasks by simultaneously training related tasks on one model [16–18]. MTL has two distinct advantages. First, a task with a relatively large amount of labeled data can assist with the learning of another task that lacks labeled data. Second, it can prevent models from overfitting to a specific task [19]. MT-DNN is an MTL model based on the pre-trained BERT, and it demonstrated impressive performance in general domain NLP tasks. ERNIE 2.0 [20], which uses continual pretraining and MTL methods, obtained state-of-the-art result for the GLUE benchmark. However, MTL can confuse a model and interfere with the learning of the central task [21]. For addressing this problem, HMTL [22] proposed a hierarchical MTL model that trained a low-complexity task first, followed by high-complexity tasks. Wu et al. [23] proposed a method of quantifying similarities between datasets, and they conduct MTL using part of the GLUE dataset.

In the medical computer vision domain, a successful method has been suggested to solve model confusion in MTL. Yang et al. [6], using four auxiliary tasks to improve lung cancer prediction, introduced the Periodic Focusing Learning Policy (PFLP) and Internal-Transfer Weighting (ITW). However, the format of data used in this paper differs from NLP.

In NLP, there are different input data formats and labels for each task. Furthermore, in the medical image MTL, one image has various annotations. Inspired by this approach, we present an SIWLM scheme that can be applied to an MTL model in NLP. With this scheme, our model can concentrate on all tasks sequentially while focusing on the central task. Figure 1 depicts an overview of MTDNN-SIWLM. Each component is covered in more detail in the next section.

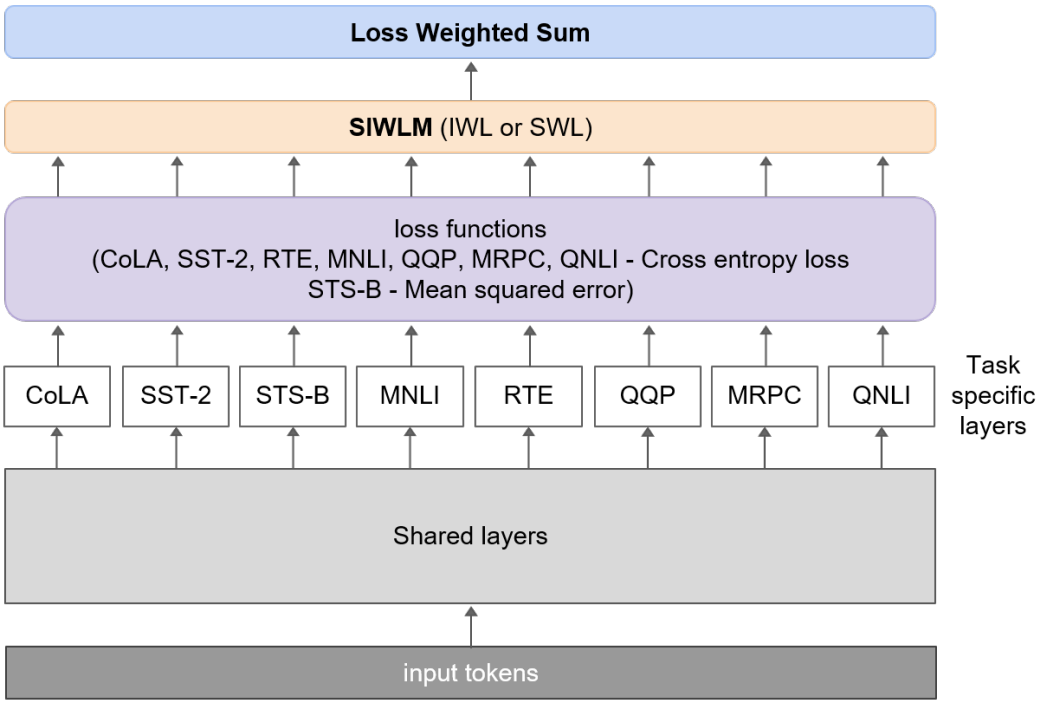

Figure 1. Model architecture used to apply the sequential and intensive weighted language modeling (SIWLM) scheme to MTL.

3. Methodology

3.1. Multi-Task Learning (MTL)

MT-DNN, which we used as the baseline model, is an MTL model that learns representation from several tasks simultaneously. MT-DNN consists of the shared layers and task-specific layers.

The shared layers follow the architecture of the Transformer encoder [24]. Similar to the form of the Transformer input, the input is a sequence of tokens resenting one or two sentences. For the model recognizing the individual sentences, the special token [SEP] is inserted at the end of each sequence. Furthermore, the first token is always [CLS] and its contextual embedding is used in the task-specific layers. We feed the input tokens into the Lexicon encoder and the Transformer encoder successively. In the lexicon encoder, word, segment, and positional embedding vectors are generated from the input tokens and are added as input embedding vectors. Subsequently, we pass the input embedding vectors into the Transformer encoder to extract contextual embedding vectors. MT-DNN learns sufficient language representation by performing MTL on the pre-trained language models, such as BERT and RoBERTa.

The contextual embedding vectors from the shared layers are passed into the task-specific layers using different loss functions based on their task type. Classification tasks, such as CoLA, SST-2, MNLI, MRPC, QQP, RTE, and QNLI, use the cross-entropy loss even though their input formats are different. In contrast, STS-B, which is the regression task, uses the mean squared error.

3.2. Sequential and Intensive Weighted Language Modeling (SIWLM)

We denote the task that we target for improved performance as a central task and the remainder as auxiliary tasks. Furthermore, a task focused on in each iteration is termed a focusing task.

SIWLM is learning scheme that adjusts the weights to be multiplied by the loss functions. In this scheme, all tasks have fixed initial weights, and each task is weight-adjusted from the initial weight during training. SIWLM consists of two stages: Sequential Weighted Learning (SWL) and Intensive Weighted Learning (IWL). In the SWL stage, all tasks used for the MTL have opportunities to be dominantly trained. In the IWL stage, only the central task that aims to obtain higher performance through MTL is intensively learned.

Algorithm 1 describes the process of SIWLM in more detail. Each mini-batch consists of one type of dataset, and the model is aware of the type of each batch during training. Model training has two stages: SWL for odd epochs and IWL for even epochs.

Algorithm 1 Sequential and intensive weighted learning.

Input: Dataset D_n, task type T_n, initial weight iw_n when n is a dataset id ($1 \leq n \leq N$, N is the number of datasets)

Initialize shared layers as BERT pretrained model
for T in T_1, T_2, ..., T_N **do**
    Append a specific layer for T
end for
Merge D_1, D_2, ..., D_N into D
Divide D into mini-batches B=$\{b_1, b_2, ..., b_k\}$ (k is the number of mini-batches and each mini-batch consists of only one dataset)
for *epoch* in 1, 2, ..., N **do**
    if *epoch* is odd **then** ▷ SWL
        $i \leftarrow 0$
        Set the central task to a focusing task
        for b in B **do**
            $n \leftarrow$ dataset id of b
            if i is 20 **then**
                Change the focusing task
                $i \leftarrow 0$
            end if
            $aw_n \leftarrow 1$ (aw_n denotes an adjusted weight)
            if b is not the focusing task **then**
                $aw_n \leftarrow iw_n * 0.1$
            end if
            Compute $loss$
            $loss \leftarrow loss * aw_n$
            Update model
            $i \leftarrow i + 1$
        end for
    else ▷ IWL
        for b in B **do**
            $n \leftarrow$ dataset id of b
            $aw_n \leftarrow 1$
            if b is not the central task **then**
                $aw_n \leftarrow iw_n * 0.1$
            end if
            Compute $loss$
            $loss \leftarrow loss * aw_n$
            Update model
        end for
    end if
end for

In the SWL stage, one task to be trained with high weight is selected as a focusing task every 20 iterations. Because all tasks are sequentially selected as the focusing task, all tasks are assigned equal opportunities to be learned in-depth. We concentrate on the focusing task by computing adjusted weights, multiplying 0.1 to initial task weights except for the focusing task. Moreover, to update the model to reflect the weights of tasks, the computed loss values are multiplied by the adjusted weights. This approach is iterated over all iterations, enabling the model to learn all tasks.

After the SWL stage, we use IWL to concentrate on the central task. In the IWL stage, the weight of the auxiliary tasks is multiplied by 0.1, which is the purpose of performance improvement in the central task. In contrast to in the SWL stage, the adjusted task weights are maintained throughout all iterations without any change. The task weight used in IWL is also multiplied by the loss to adjust the impact of each task on learning representation. In this stage, because the central task has a high weight during all iterations, the central task can be learned intensively.

Figure 2 illustrates an example of applying SIWLM to eight tasks in GLUE. This example demonstrates how SIWLM is applied when RTE is a central task and the initial weights for each task are set to [6:1:1:1:1:1:1:1]. In the SWL stage, all learning tasks used for learning are sequentially selected as focusing tasks every 20 iterations. During the first 20 iterations, RTE is selected as the focusing task, and the weight of each task except RTE is multiplied by 0.1. During the next 20 iterations, SST-2 is selected as the focusing task and the weight of each task except SST-2 is multiplied by 0.1. The training proceeds for 20 iterations with a weight of [0.6:1:0.1:0.1:0.1:0.1:0.1:0.1]. This method is repeated for all iterations in the form of a loop, as depicted in Figure 2. In the IWL stage, learning is focused only on the central task, RTE. Learning is conducted for all iterations by multiplying the remaining task weights, excluding the RTE weight, by 0.1.

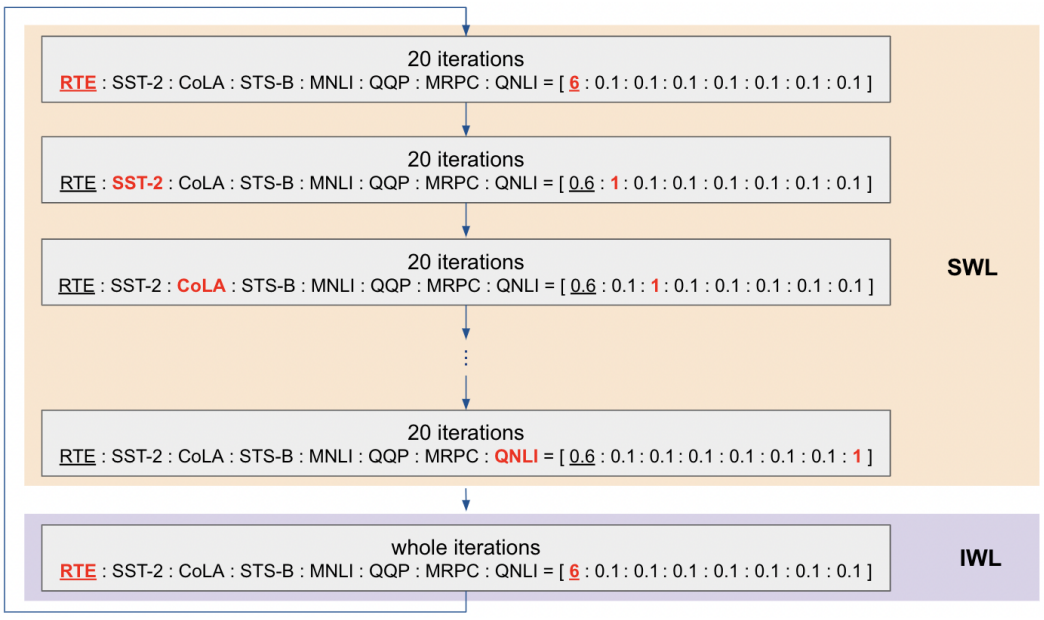

Figure 2. Example of weight-adjustment process applied to loss weighted sum. In this example, RTE is the central task and is underlined in the figure. The focusing tasks in the SWL stage are indicated by the bold red font. For each epoch, the SWL and IWL stages are applied alternately. In the SWL stage, the focusing task is changed every 20 iterations and the initial weight of each task except the focusing task is multiplied by 0.1. In contrast, in the IWL stage, the central task is fixed as a focusing task and the initial weights of all auxiliary tasks are multiplied by 0.1.

3.3. Fine-Tuning

We fine-tune the trained model for each GLUE dataset after training it using MTL with the SIWLM scheme. We use the SMoothness-inducing Adversarial Regularization and BRegman pRoximal poinT opTimization (SMART) learning method proposed by Jiang et al. [7] for fine-tuning. SMART improves generalization with smoothness-inducing regularization, which does not change the model's output if it injects a small perturbation. The researchers also propose a Bregman proximal point optimization that updates only in little neighborhoods of the previous iteration. Accordingly, SMART prevents aggressive training during the fine-tuning process. Finally, these ingredients prevent the model from overfitting even after fine-tuning. We evaluated our SIWLM performance, as depicted in Section 5, with a SMART model that applies the SMART fine-tuning method to the MT-DNN model.

4. Experiments

4.1. Datasets and Metrics

The primary purpose of our experiment is to determine whether the SIWLM scheme can improve the MTL model, such as MT-DNN and SMART. Accordingly, we used the GLUE benchmark commonly used by MTL models as a dataset for the experiment. The number of training examples used in the experiment are listed in Table 1.

Table 1. Number of training examples.

Dataset	The Number of Training Examples
MNLI	393 k
QQP	364 k
QNLI	108 k
SST-2	67 k
CoLA	8.5 k
STS-B	7 k
MRPC	3.7 k
RTE	2.5 k

4.1.1. Single-Sentence Classification

CoLA The Corpus of Linguistic Acceptability [25] is proposed to determine the grammatical acceptability of English sentences from published linguistics literature. Each sentence is annotated with 0 or 1, respectively, representing acceptable or unacceptable.

SST-2 The Stanford Sentiment Treebank [26] is a corpus for a sentiment classification of movie reviews. Each review sentence has a human annotation: positive is 1, and negative is 0.

4.1.2. Pairwise Text Similarity

STS-B The Semantic Textual Similarity Benchmark [27] is a corpus of sentence pairs driven from news headlines, image captions, and user forums. The task of STS-B is to predict the semantic similarity score between two sentences. Each pair is annotated with a similarity score from 1 to 5.

4.1.3. Pairwise Text Classification

MNLI The Multi-Genre Natural Language Inference Corpus [28] is a corpus of sentence pairs with textual entailment annotations. A pair of sentences is comprised of a premise sentence and a hypothesis sentence. The task is to determine whether the premise entails the hypothesis, contradicts the hypothesis, or neither. Accordingly, it has three labels: entailment, contradiction, and neutral.

RTE The Recognizing Textual Entailment is a textual entailment dataset, which is a combination of RTE1 [29], RTE2 [30], RTE3 [31], and RTE5 [32]. GLUE's authors modified some datasets annotated with three kinds of labels to two: entailment and not_entailment.

QQP The Quora Question Pairs [2] is a dataset of question pairs drawn from social question-answering domain. This task's goal is to predict whether question pairs are semantically equivalent or not. This dataset consists of two labels: 0 is negative, implying that the questions in a pair are not semantically equivalent, and 1 is positive

MRPC The Microsoft Research Paraphrase Corpus [33] is a corpus of sentence pairs comprised of sentences automatically extended from online news. As in QQP, the task is to determine whether the sentences in the pair are semantically equivalent or not. Each sentence is annotated 0 or 1, where 0 is negative, implying that the sentences in a pair are not semantically equivalent, and 1 is positive.

4.1.4. Pairwise Ranking

QNLI The Question-answering NLI [34] is a corpus of question–context sentence pairs that modify the Stanford QA dataset which consists of question–paragraph pairs. The task with QNLI is to predict whether a context contains the answer to a given question. Each pair is annotated as entailment or not_entailment.

4.2. Experiment Setting

4.2.1. Competing Models

We reproduce the performance of three MTL models and BERT for comparison with our model. The competing models are as follows:

- BERT: A model with an additional task -specific layer added to the uncased BERT-base model. We fine-tune the pretrained model with each GLUE dataset.
- BAM: A model that performs MTL using knowledge distillation and exhibits high performance in the GLUE benchmark.
- MT-DNN: An MTL model that uses a pretrained BERT as an encoder and learns each task in parallel.
- SMART: A model that outperforms MT-DNN in many NLP tasks by applying a new fine-tuning technique to MT-DNN.
- MTDNN-SIWLM: An MT-DNN model that uses our SIWLM scheme.

4.2.2. Implementation Details

We used three V100 GPUs, while SMART used eight V100 GPUs. We compared our model with SMART in the same environment by reproducing performance using SMART with BERT-base. We used the same hyperparameter reported in Liu et al. [4] for MTL: a learning rate of 5×10^{-5}, a batch size of 32, and the Adamax optimizer. The number of epochs is set to 5. Furthermore, for the fine-tuning stage, we followed the hyperparameter range suggested by Jiang et al. [7]: a learning rate $\in \{1\times 10^{-5}\ 2 \times 10^{-5}, 3 \times 10^{-5}, 5 \times 10^{-5}\}$, a batch size $\in \{16, 32, 64\}$ and the Adam optimizer. The maximum number of epochs is set to 5. Consequently, when comparing the SMART model with our model, the possibility that factors other than the SIWLM scheme affect the performance is minimized. The source code is available at https://github.com/sonsuhyune/SIWLM (accessed on 30 March 2021).

We set the initial task weights as {**1**:1:1:1:1:1:1:1:1, **3**:1:1:1:1:1:1:1:1, **6**:1:1:1:1:1:1:1:1, **9**:1:1:1:1:1: 1:1:1, **12**:1:1: 1:1:1:1:1:1, **15**:1:1:1:1:1:1:1:1}. The first values in the task weights are for a central task. The first value of the weight is set to a maximum of 15 because all tasks except RTE exhibited the highest performance before having a weight of 15. The results are reported in Table 2.

Table 2. Validation performance of MTDNN-SIWLM are reported; highest results are in bold font.

Task Weight	MNLI-m/mm Acc	QQP Acc/F1	QNLI Acc	SST-2 Acc	CoLA Acc/Mcc	STS-B P/S Corr	MRPC Acc/F1	RTE Acc
1	**84.65**/84.30	90.61/87.36	91.01	92.54	80.92/52.65	88.01/87.97	87.00/90.20	76.17
3	84.52/84.35	90.76/87.56	91.23	92.54	81.20/53.39	**88.05**/88.16	86.76/90.14	75.81
6	84.34/**84.60**	**90.80**/87.61	91.14	**92.88**	81.11/53.18	87.83/87.90	86.76/90.21	77.25
9	84.51/84.55	90.73/87.49	**91.36**	92.66	80.72/52.06	88.02/**88.20**	87.99/90.90	77.61
12	84.37/84.46	90.79/87.61	91.06	92.77	**81.30/53.64**	87.91/87.99	**87.99/91.04**	77.25
15	84.52/84.54	90.59/**87.70**	91.15	92.20	80.92/52.69	87.83/87.98	87.01/90.31	**77.98**

5. Results and Analysis

We conduct experiments on all datasets with weights of the central task ∈ {1, 3, 6, 9, 12, 15}, which are specified in the experiment setting. The validation performance of MTDNN-SIWLM is depicted in Table 2. Each row of the table represents the weight of the central task, and the weights of the auxiliary tasks are all set to 1. In the case of MNLI, the validation and test datasets had matched or mismatched versions, and the weight settings for the highest performance in each version differed, at 1 and 6. We use accuracy/F1 and Pearson/Spearman for the evaluation metrics of QQP and STS-B, respectively. STS-B performed best in different weights for each metric.

MNLI, QQP, QNLI, and SST-2 with large data sizes generally exhibit higher performance when the weight of the central task is less than 6. For CoLA, STS-B, MRPC, and RTE, which have a small data size, it was found that when a weight of 9 or more was applied, the performance was enhanced more in most cases. Consequently, applying equal weights when performing MTL on a small dataset may increase performance improvement.

Figure 3a illustrates the accuracy variance of each dataset according to the task weight change. The x-axis indicates the weight of the central task, as used in Table 2. The y-axis indicates the difference between the minimum accuracy and the accuracy for each weight setting in one dataset. We use the same evaluation index for all datasets by expressing accuracy in this graph. Moreover, STS-B, which does not use the accuracy metric, is excluded from the graph.

Furthermore, in Figure 3b, only pairwise text classification tasks (MNLI-m, QQP, MRPC and RTE) are represented to exclude performance differences caused by the differences in the task type. The value indicated by each axis is the same as in Figure 3a. As depicted in Figure 3b, for RTE and MRPC with small data sizes, the accuracy variance according to the task weight is relatively large. In contrast, MNLI-m and QQP with large data sizes have low accuracy variance. The smaller the data size, the more performance change is affected by the task weight.

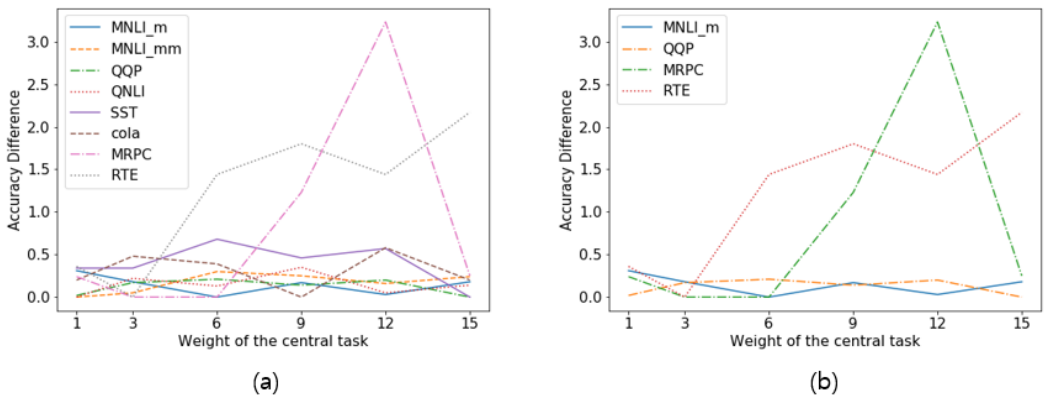

Figure 3. Accuracy variance according to the task weight change. (**a**) amount of change in accuracy for each dataset except STS-B; (**b**) amount of change in accuracy for four datasets belonging to the Pairwise Text Classification. Accuracy difference = (accuracy at each weight − minimum accuracy).

We fine-tune the model with the highest performance in Table 2. We achieve the highest performing model by averaging the performance of MNLI-m and MNLI-mm. For two metrics among QQP, STS-B, and MRPC, we attain the highest performance model by averaging performance. We demonstrate that our model is superior to other MTL-based models by comparing the validation performance the validation performance to three representative MTL models and the BERT model. As shown in Table 3, models using MTL outperform BERT. BAM exhibits the highest performance for MNLI and RTE. MTDNN presents the highest performance for STS-B, and SMART for MNLI and CoLA. Our model—using the SIWLM scheme—exhibits the highest performance for QQP, QNLI, SST-2, and MRPC.

Table 3. Validation performance of reproduced models and MTDNN-SIWLM. The best result is in bold font and the second-best result is underlined. For STS-B, we attain the performance by averaging the Pearson and Spearman correlations.

	MNLI-m/mm Acc	QQP Acc	QNLI Acc	SST-2 Acc	CoLA Mcc	STS-B Avg	MRPC Acc	RTE Acc
BERT [11]	84.1	90.8	91.2	<u>93.0</u>	52.8	83.7	86.5	67.1
BAM [5]	**84.9**	91.0	91.4	92.8	56.3	<u>86.9</u>	88.5	**82.7**
MT-DNN [4]	84.0	89.5	91.4	92.1	53.1	**89.6**	<u>89.7</u>	78.7
SMART [7]	**84.9**	<u>91.4</u>	<u>91.8</u>	91.6	**58.5**	85.7	87.2	<u>79.7</u>
MTDNN-SIWLM	<u>84.8</u>	**91.5**	**92.0**	**93.2**	<u>57.0</u>	86.7	**90.4**	79.0

We compare the performance of our model to MT-DNN and SMART, our baseline models. As depicted in Table 4, the best result for each task is in bold font, and the second-best result is underlined. MTDNN-SIWLM outperforms MT-DNN and SMART in six tasks (MNLI, QQP, QNLI, SST-2, MRPC, and RTE) than MT-DNN and SMART. Moreover, it has the second-best performance in CoLA and STS-B. MTDNN-SIWLM achieves accuracies of 84.8% and 84.0% in MNLI-m and MNLI-mm, outperforming SMART by 0.1% and 0.8%. QQP and SST-2, respectively, present improvements of 0.1% for F1 score and 0.7% for accuracy compared with SMART. STS-B exhibits improved performance for both evaluation indexes, with increases of 1.25% and 1.3% for Pearson and Spearman, respectively. MRPC and RTE, with small data sizes, demonstrate accuracy improvements of 4.1% and 5.6%.

Table 4. Fine-tuned results for eight tasks of GLUE. The best result is in bold font and the second-best result is underlined. For two metrics, we attain the highest performance model by averaging two performances. These GLUE test set results were measured using the GLUE evaluation server on 19 January 2021.

	MNLI-m/mm Acc	QQP Acc/F1	QNLI Acc	SST-2 Acc	CoLA Mcc	STS-B P/S Corr	MRPC Acc/F1	RTE Acc
MT-DNN [4]	84.2/83.0	71.4/89.1	90.9	<u>93.1</u>	50.1	**87.4/86.8**	86.4/89.8	<u>75.5</u>
SMART [7]	<u>84.7/83.2</u>	**89.5/71.6**	**91.7**	92.5	**55.2**	82.6/81.0	<u>86.6/89.8</u>	72.7
MTDNN-SIWLM	**84.8/84.0**	**89.4/71.7**	**91.7**	**93.8**	<u>53.1</u>	<u>83.8/82.3</u>	**90.7/87.6**	**77.1**

6. Conclusions

In this study, we introduce an SIWLM scheme consisting of SWL and IWL to solve a model confusion problem in MTL in NLP. With this scheme, the model can thoroughly learn all tasks while still strengthening its understanding of the central task. In comparing our model with the MT-DNN and SMART models, our model demonstrated the highest performance for the GLUE benchmark tasks excluding CoLA and STS-B. This result confirms that the SIWLM scheme contributes to the performance improvement of the MTL model. Furthermore, based on experiments using various weights, when the data size is large, the performance improves more when the weight of the central task is less than 6. Conversely, when the data size is small, the performance is improved even more when a weight of 9 or more is applied. Moreover, we find that the performance variance is large in the small data size task, whereas the weight application method is more effective under

such a condition. The proposed SIWLM-based learning scheme can be applied to other MTL models to improve learning efficiency and prediction accuracy, especially for tasks with small datasets.

While our approach outperforms the MT-DNN model in most GLUE datasets, there are still several possible directions for improving our model. We searched optimal weights for each task through several experiments. However, our optimization method is expensive, and we must find optimal weights when new tasks are added. Therefore, we seek to investigate a scheme that enables a general weight regardless of tasks. Future studies can apply our SIWLM scheme to MTL architectures in various fields [35–37] to improve their performance. Furthermore, additional training on other NLU datasets such as SuperGLUE [38] may help reinforce our model.

Author Contributions: S.S., S.H., S.B., and J.-H.C. designed the method. S.S., S.H., and S.B. implemented the method and wrote the paper. S.J.P. and J.-H.C. contributed to the supervision of the work, analysis of the method, and paper writing. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Collaborative Genome Program for Fostering New Post-Genome Industry of the National Research Foundation (NRF) funded by the Ministry of Science and ICT (MSIT) (Grant No. NRF-2014M3C9A3064706); the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP: Ministry of Science, ICT, and Future Planning) (Grant Nos. NRF-2020R1A4A1016619, NRF-2020R1F1A1073774); and by the Korea Medical Device Development Fund grant funded by the Korea government (the Ministry of Science and ICT, the Ministry of Trade, Industry and Energy, the Ministry of Health & Welfare, the Ministry of Food and Drug Safety) (Project Number: KMDF_PR_20200901_0016, 9991006689). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Jang, B.; Kim, M.; Harerimana, G.; Kang, S.u.; Kim, J.W. Bi-LSTM model to increase accuracy in text classification: combining Word2vec CNN and attention mechanism. *Appl. Sci.* **2020**, *10*, 5841. [CrossRef]
2. Wang, A.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv* **2018**, arXiv:1804.07461.
3. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep contextualized word representations. *arXiv* **2018**, arXiv:1802.05365.
4. Liu, X.; He, P.; Chen, W.; Gao, J. Multi-task deep neural networks for natural language understanding. *arXiv* **2019**, arXiv:1901.11504.
5. Clark, K.; Luong, M.T.; Khandelwal, U.; Manning, C.D.; Le, Q.V. Bam! born-again multi-task networks for natural language understanding. *arXiv* **2019**, arXiv:1907.04829.
6. Yang, Y.; Gao, R.; Tang, Y.; Antic, S.L.; Deppen, S.; Huo, Y.; Sandler, K.L.; Massion, P.P.; Landman, B.A. Internal-transfer weighting of multi-task learning for lung cancer detection. In *Medical Imaging 2020: Image Processing*; International Society for Optics and Photonics: Bellingham, WA, USA, 2020; Volume 11313, p. 1131323.
7. Jiang, H.; He, P.; Chen, W.; Liu, X.; Gao, J.; Zhao, T. Smart: Robust and efficient fine-tuning for pre-trained natural language models through principled regularized optimization. *arXiv* **2019**, arXiv:1911.03437.
8. Lee, C.; Yang, K.; Whang, T.; Park, C.; Matteson, A.; Lim, H. Exploring the Data Efficiency of Cross-Lingual Post-Training in Pretrained Language Models. *Appl. Sci.* **2021**, *11*, 1974. [CrossRef]
9. Jwa, H.; Oh, D.; Park, K.; Kang, J.M.; Lim, H. exBAKE: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Appl. Sci.* **2019**, *9*, 4062. [CrossRef]
10. Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; Liu, P.J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv* **2019**, arXiv:1910.10683.
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
12. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv* **2019**, arXiv:1906.08237.

13. Wang, W.; Bi, B.; Yan, M.; Wu, C.; Bao, Z.; Xia, J.; Peng, L.; Si, L. Structbert: Incorporating language structures into pre-training for deep language understanding. *arXiv* **2019**, arXiv:1908.04577.
14. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
15. He, P.; Liu, X.; Gao, J.; Chen, W. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv* **2020**, arXiv:2006.03654.
16. Zhang, Y.; Yang, Q. A survey on multi-task learning. *arXiv* **2017**, arXiv:1707.08114.
17. Guo, H.; Pasunuru, R.; Bansal, M. Soft layer-specific multi-task summarization with entailment and question generation. *arXiv* **2018**, arXiv:1805.11004.
18. Ruder, S.; Bingel, J.; Augenstein, I.; Søgaard, A. Latent multi-task architecture learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 4822–4829.
19. Ruder, S. An overview of multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1706.05098.
20. Sun, Y.; Wang, S.; Li, Y.K.; Feng, S.; Tian, H.; Wu, H.; Wang, H. ERNIE 2.0: A Continual Pre-Training Framework for Language Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 8968–8975.
21. Bingel, J.; Søgaard, A. Identifying beneficial task relations for multi-task learning in deep neural networks. *arXiv* **2017**, arXiv:1702.08303.
22. Sanh, V.; Wolf, T.; Ruder, S. A hierarchical multi-task approach for learning embeddings from semantic tasks. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 6949–6956.
23. Wu, S.; Zhang, H.R.; Ré, C. Understanding and Improving Information Transfer in Multi-Task Learning. *arXiv* **2020**, arXiv:2005.00944.
24. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *arXiv* **2017**, arXiv:1706.03762.
25. Warstadt, A.; Singh, A.; Bowman, S.R. Neural network acceptability judgments. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 625–641. [CrossRef]
26. Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C.D.; Ng, A.Y.; Potts, C. Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; pp. 1631–1642.
27. Cer, D.; Diab, M.; Agirre, E.; Lopez-Gazpio, I.; Specia, L. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv* **2017**, arXiv:1708.00055.
28. Williams, A.; Nangia, N.; Bowman, S.R. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv* **2017**, arXiv:1704.05426.
29. Dagan, I.; Glickman, O.; Magnini, B. The PASCAL recognising textual entailment challenge. In *Machine Learning Challenges Workshop*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 177–190.
30. Haim, R.B.; Dagan, I.; Dolan, B.; Ferro, L.; Giampiccolo, D.; Magnini, B.; Szpektor, I. The second pascal recognising textual entailment challenge. Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment, Venezia, Italy, 10 April 2006.
31. Giampiccolo, D.; Magnini, B.; Dagan, I.; Dolan, B. The third pascal recognizing textual entailment challenge. In Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing, Prague, Czech Republic, 28–29 June 2007; pp. 1–9.
32. Bentivogli, L.; Clark, P.; Dagan, I.; Giampiccolo, D. The Fifth PASCAL Recognizing Textual Entailment Challenge. In Proceedings of the Text Analysis Conference, Gaithersburg, MD, USA, 16–17 November 2009.
33. Dolan, W.B.; Brockett, C. Automatically constructing a corpus of sentential paraphrases. In Proceedings of the Third International Workshop on Paraphrasing (IWP2005), Jeju Island, Korea, 14 October 2005.
34. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv* **2016**, arXiv:1606.05250.
35. Kochkina, E.; Liakata, M.; Zubiaga, A. All-in-one: Multi-task learning for rumour verification. *arXiv* **2018**, arXiv:1806.03713
36. Majumder, N.; Poria, S.; Peng, H.; Chhaya, N.; Cambria, E.; Gelbukh, A. Sentiment and sarcasm classification with multitask learning. *IEEE Intell. Syst.* **2019**, *34*, 38–43. [CrossRef]
37. Crichton, G.; Pyysalo, S.; Chiu, B.; Korhonen, A. A neural network multi-task learning approach to biomedical named entity recognition. *BMC Bioinform.* **2017**, *18*, 368. [CrossRef] [PubMed]
38. Wang, A.; Pruksachatkun, Y.; Nangia, N.; Singh, A.; Michael, J.; Hill, F.; Levy, O.; Bowman, S.R. Superglue: A stickier benchmark for general-purpose language understanding systems. *arXiv* **2019**, arXiv:1905.00537.