Knowledge Distillation Techniques for Biomedical Named Entity Recognition

Tahir Mehmood[1,2], Ivan Serina[1], Alberto Lavelli[2], and Alfonso Gerevini[1]

[1] University of Brescia, 25121 Brescia, Italy
{t.mehmood,ivan.serina,alfonso.gerevini}@unibs.it
[2] Fondazione Bruno Kessler, 38123 Povo, Trento, Italy
{t.mehmood,lavelli}@fbk.eu

Abstract. The limited amount of annotated biomedical literature and its peculiar characteristics make biomedical named entity recognition more challenging than standard named entity recognition. The multi-task learning approach overcomes these limitations by training different related tasks simultaneously. It learns common features among different tasks by sharing some layers of the neural network architecture. For this reason, the multi-task model attains more generalization properties than a single task learning. The generalization of the multi-task model can be utilized to enhance other models' results. In particular, knowledge distillation techniques make this possible in which one model supervises, through its learned generalization, another model during the training. This research analyzes the knowledge distillation approach and shows that a simple deep learning model performance can be leveraged through distilling the multi-task model's generalization. Results show that our approach outperformed compared with the multi-task model and single task model. This demonstrates that our model learns more diverse features using the knowledge distillation approach. We also found our approach statistically better than multi-task model and single task model.

Keywords: Biomedical Named Entity Recognition · Multi-task Learning · Knowledge Distillation.

1 Introduction

The biomedical named entity recognition (BioNER) task has gained more attention with the increasing availability of large amounts of unstructured biomedical text data. BioNER is also a preliminary task of many other tasks e.g. the relation extraction task (e.g., chemical induced disease relation, drug-drug interaction, . . .) [20]. However, biomedical texts are more complex than normal texts and carry unusual characteristics, e.g. spelling alternations (e.g., *10-Ethyl-5-methyl-5,10-dideazaaminopterin* vs *10-EMDDA*) [1], long multi-word expressions (*10-ethyl-5-methyl-5,10-dideazaaminopterin*), and ambiguous words (*TNF alpha* can be used for both DNA and Protein) [8]. The above-mentioned characteristics make BioNER even a more difficult task than traditional named entity recognition.

Traditional machine learning approaches that include e.g., Hidden Markov Models (HMMs), Conditional Random Fields (CRFs), and Support Vector Machine (SVM), have been used to overcome the limitations faced by the BioNER task [4]. These machine learning methods have shown some promising results. However, these approaches strongly rely on feature engineering. On the other hand, deep learning models usually do not require hand-crafted feature engineering since this is done implicitly. Simultaneously, the deep learning models' results are very appealing for the BioNER task. However, due to the biomedical literature's peculiar characteristics mentioned at the beginning of the section, these systems' performance is still limited. Another challenge to deep learning models is the limited availability of annotated biomedical text data to train these systems as deep learning models require substantial amounts of training data.

The multi-task and transfer learning approaches have shown results improvement for BioNER task [16][17], but these techniques still have some limitations. The multi-task model (MTM) [18] does not always produce noticeable increase in performance compared to their counterpart single task model (STM) [3][5]. The MTM could also learn the features that are more task-specific and which can lead to biased feature learning [13]. Similarly, transfer learning [7] also faces limitations e.g., catastrophic forgetting or catastrophic interference problem [23]. In catastrophic forgetting, the deep learning model starts forgetting what it has learned from the previous domain. The forgetting of the previously learned source information happens, even if both source and target domains are heterogeneous [10]. It is also an empirical dilemma to choose the number of new layers for the model used on the target datasets along with pretrained layers or weights of the pretrained layers need to be frozen in the pretrained model as it is applied to the target dataset. The transfer learning approach is therefore not always a feasible solution to transfer previous knowledge into the new task.

Furthermore, in general, a common issue with the deep learning models is their complex structure. The deep learning methods have brought much success in numerous fields and have shown results breakthrough. To achieve state-of-the-art results, the complex structure of the deep learning models is often observed in many fields. Sutskever et al. [25] model comprised of 4-layers of long short-term memory (LSTM) and each layer had 1000 hidden units. Similarly, Zhou et al. [33] proposed a model that contains multi-level LSTM and each layer had 512 hidden units. These deep learning models have millions of parameters, and training such models require much more computational power. These complex models also require more storage space and which is also not very suitable to deploy on the systems where available storage capacity is limited e.g., cell phones. In such situations, implementation of these complex models requires compression while, in the meantime, not to compromise their performances and keep the generalization they have learned.

In this regard, the knowledge distillation approach is utilized where the cumbersome model is compressed into the simple model, which is more feasible to set up in the end devices [11]. In the knowledge distillation technique [9], one model teaches another model through its learned knowledge. This supervision is done through prediction, where the learning model mimics the prediction of the teacher model. The learning model, therefore, uses two gradients, i.e., the gradient of itself and gradient of the teacher model, and for this reason, it can produce better results. Romero et al. [22] showed that the intermediate layer of the teacher model gives useful information to the student model during the training. Liu et al. [14] improved the performance of the single model using knowledge distillation from an ensemble of different deep neural networks. Tang et al. [26] showed performance gain by distilling knowledge from a single machine translation model to train the multilingual translation model. Zhang et al. [32] demonstrated an increase in performance when different student models were trained mutually and teach each other through knowledge distillation. Sun et al. [24] showed performance gain using knowledge distillation approach in which the intermediate layers of the teacher model were used to train the task specific student model.

This research also proposes the distillation knowledge approach to enhance the performance of the deep learning models for BioNER task. Therefore, the purpose of this research is to increase the performance of the model instead of compression. The multi-task model is used to perform knowledge distillation for the single task model using its logits. In other words, single task model matches the true labels as well as the logits of the multi-task model during its training. Logits are the inputs to the softmax output layer [9] which carries more information and its value ranges from $[-\infty, +\infty]$. This helps the single task model to not only learn from the true labels but also optimize logits for multi-task model.

The rest of the paper is organized as follows. Section 2 gives an introduction of the knowledge distillation approach which is followed by our proposed methodology in Section 3. The experimental setup is described in Section 4 whereas results are discussed in Section 5. Finally, the research is concluded in Section 6.

2 Knowledge Distillation

In transfer learning, the learned representation from source domain is utilized in another related domain. In contrast, the objective of knowledge distillation is to train a model with the knowledge learned by another model. The idea of the knowledge distillation is to train a simple (student) model on the knowledge learned by the complex (teacher) model. More specifically, the knowledge distillation approach addresses how to transfer the generalization of one model, usually a complex model (teacher), to another model, usually a simple model (student).

The complex models or ensemble approaches usually produce better results than the simple single-task model, but it is computationally expensive to train them. The knowledge distillation approach helps the simple model (student) to produce better results than the stand alone single model and the ensemble models. This way student model can be trained on fewer training examples since it will also consume the knowledge learned by the teacher model during training. The idea is that the complex model has already been generalized on the data during its training. This helps the student model to achieve or nearly achieve the generalization of the teacher model. The student model not only learns through the gradient of itself but also though the gradient of another knowledge.

Transferring knowledge from a teacher model is usually done in the shape of the probabilities predicted by the teacher model. The objective of any learning model is to predict the correct class for the input example and assign a high probability to that class whereas allocating small probability values to the rest of the classes. Associating the probabilities to the rest of the incorrect classes is not performed randomly. These side probabilities also carry information which depicts how a specific model has generalized the classes presented in the dataset. For instance, there is very little chance of miss-classifying a motorbike image into a car image but the probability would still be higher for miss-classifying it into the truck image. The softmax activation function outputs the probability distribution of the possible classes for the specific instance. The sum of these softmax probability distributions sums to 1.

These softmax probabilities give more information compared to the one-hot "hard labels". For instance, the softmax probabilities, [0.7, 0.2, 0.1], show ranking of the classes. Such information cannot be examined in the hard labels e.g, [1, 0, 0] where we cannot extract any such information. The posterior probabilities can pass an extra useful signal to the student model during its training. However, training the student model to match these probabilities could not be so much useful as the student model can only pay more attention to the highest probability value. To overcome this barrier, it is better to soften these final softmax output probabilities through normalizing them [9]. The normalized probabilities represents soft labels which provides some knowledge distillation to the student model [29]. The student model then pay attention to other values as well along with the highest probable class. Hinton et al. proposed a term temperature, T, to soften the posterior probabilities [9]. Keeping $T = 1$ makes it standard softmax function as represented in equation 1. The large value of T more softens the softmax output and enhances the non-target class output probability [19]. On the downside, it also reduces the probability value of the target class. Therefore, it is vital to choose the right value for the temperature parameter.

$$\text{Softmax}(z_i) = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)} \tag{1}$$

3 Our Approach

Figure 1 introduces our proposed knowledge distillation approach. The teacher model is a multi-task model (MTM) with the word and character input of the sentences. We use bidirectional LSTM (BiLSTM) to process the sequence in both directions [21]. The upper layers, shown in black round rectangle, of the MTM are shared among all the datasets. The bottom layers, shown in red round rectangle, are dataset specific whereas Softmax is used for output labelling. In multi-task learning (MTL) approach shared layers help one task to be learned better with the help of another task. Training jointly on related tasks helps the multi-task model to learn common features among different tasks by using shared layers [2]. The task-specific layers learn features that are more related to the current task. Training related tasks together helps the model to optimize the value of the parameters. The joint learning also lowers the chances to embrace overfitting for any specific task [15]. Therefore, we assume that the student model will also have lower changes to encounter overfitting with the help of knowledge distillation from the MTM. The purpose of our word is to transfer the token level knowledge distillation, therefore, we use softmax function at the output layer. The token level knowledge distillation is not possible with conditional random field (CRF) as it predicts the labels of the whole sequence. The CRF based model labels the sequence globally considering the association between neighboring labels. This limits the distilling knowledge from the teacher models [30].

An alternative training approach was adopted for MTM training phase. Let us suppose we have $D_1, D_2, ..., D_t$ training sets, related to the $T_1, T_2, ..., T_t$ tasks respectively. During the training phase, a training set D_i is selected randomly and both shared layers and the ones specific to the corresponding task T_i are activated. Every task has its own optimizer so during training only the one specific to the task T_i is activated and the loss function related to it is optimized.

The student model is in fact a counterpart single task model (STM) of the MTM. Therefore, the structures of both models are same. In this research we perform knowledge distillation using the teacher (MTM)logits, z_t, which is input to the softmax layer [28]. The logits carry the values that can range $[-\infty, +\infty]$ and therefore, carries more dark information. During the training, student model considers the hard labels as well as the logits (z_t) of the teacher model (MTM). We also have not normalized the logits that means temperature, $T = 1$. We examine losses for both predictions i.e., the loss of the hard labels matching and the loss of the logits matching. The hard targets matching loss, which involves one-hot labels, can be referred as student loss whereas the distillation loss considers the logits loss. The loss function of our student model model is depicted in equation 2. The distillation loss tries to minimize mean-squared-error between the student logits, z_s, and teacher logits, z_t, as depicted in equation 2. The x represents the input, W represents student model's parameters, $\mathcal{H}$ is the cross-entropy loss whereas y is the true hard labels and σ is the softmax function. The logits of student and teacher models are represented as z_s, z_t respectively. The coefficients, α and β, specify the balance between student loss and distillation

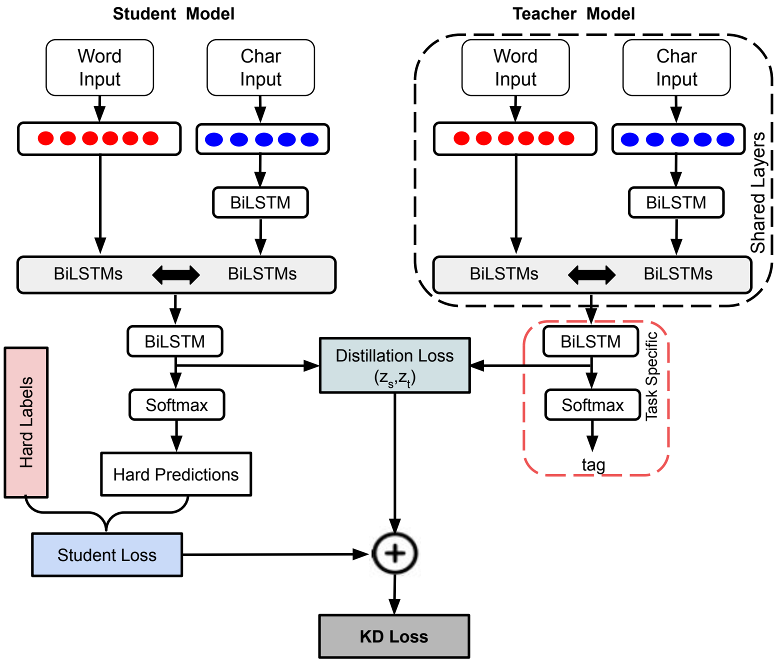

Fig. 1. Proposed Knowledge Distillation Approach (colored circles show embedding)

loss whereas $\beta = 1 - \alpha$.

$$\mathcal{L}(x; W) = \alpha * \mathcal{H}(y, \sigma(z_s, z_t)) + \beta * MSE(z_s, z_t) \tag{2}$$

4 Experiments

As a first approach, the MTM model, shown in the right side of Figure 1, is trained separately. This MTM is then used to distill the knowledge to the student model. We perform knowledge distillation from MTM using two approaches. In the first approach, we perform simple knowledge distillation as shown in Figure 1 where MTM's logits are used to train the student model. In the second approach, we use logits from ensemble of MTMs to train the student model. The MTMs used in the ensemble approach have the same architecture, but they are initialized with different seed values which result in different predictions. Although, the structure of all MTMs are same but this gives us five different predictions due to the different seed values. We take the average of the logits from these MTMs, which is then used to train our student model. Furthermore, the F1-score presented in the later section is also based on the average of five runs with different seed values. In the rest of this article, MTM and teacher

MTM will be used interchangeably as the logits of the MTM are used to train the student models.

We perform experiments for different values of α i.e.,$[0, 0.5, 1]$. The hyperparameter tuning is not done for α, instead the values are selected in a simple straight forward way. If $\alpha = 0$ then the student model learns with only distillation loss i.e., $\beta * MSE(z_s, z_t)$, which tries to match logits of the student model and teacher model. Similarly, with $\alpha = 0.5$, both student loss and distillation loss are considered equally. In last, $\alpha = 1$, only allows student model to consider the student loss, $\alpha * \mathcal{H}(y, \sigma(z_s; z_t))$. Furthermore, words are represented with pre-trained domain-specific word embedding. More specifically, we utilize the WikiPubMed-PMC word embedding which is trained on a large set of the PubMedCentral(PMC) articles and PubMed abstracts as well as on English Wikipedia articles [7]. On the other hand, character embedding is initialized randomly which is further processed by BiLSTM. In this paper, we perform experiments on the 15 datasets which are also used by Crichton et al. [6] and Wang et al. [31]. The bio-entities in these datasets are Chemical, Species, Cell, Gene/Protein, Cell Component, and Disease[3]. The description of these entities can be found in [16]. Each dataset contains separate training, development, and test sets. We follow the same experimental setup adopted by Wang et al.[4], which uses both train and development set data for training the model.

5 Results and Discussion

The F1-score comparison of our student model with different α values is shown in Table 1. The MTM is the teacher model as mentioned in the earlier section as well. This MTM is used for distilling knowledge to the student model via its logits. The best results are shown in the bold font while second best score is represented with the Italic style. It can be noticed that our student model has outperformed the MTM approach, except for BioNLP13CG and most of the protein datasets (BioNLP11EPI, BioNLP11ID, BioNLP13GE, and Ex-PTM). We speculate that as BioNLP11EPI, BioNLP11ID, and Ex-PTM are the corpora created for BioNLP 2011 shared task corpus, they might carry similar characteristics. Therefore, we observe a performance decrease for all these three datasets. In particular, the entity mentions in BioNLP11EPI and Ex-PTM were automatically annotated using BANNER named entity tagger [12] which was trained on the GENETAG corpus [27]. We anticipate that the wrong entity classification might have propagated in both datasets due to the annotation from the same named entity tagger. On the other hand, BioNLP13CG contains 16 different classes and some of them have very few examples present in the dataset. These classes represent cancer genetics (CG) and are more correlated with each other. Therefore, our student model might not be able to differentiate among these classes.

[3] The datasets can be found at https://github.com/cambridgeltl/MTL-Bioinformatics-2016

[4] https://github.com/yuzhimanhua/Multi-BioNER

Student model, with $\alpha = 0$, has shown a performance gain for 6 datasets compared to the MTM (teacher). The student model, trained with $\alpha = 0.5$, achieves an increase in performance for 9 and 8 datasets compared to the MTM and student ($\alpha = 0$) model, respectively. Similarly, student model ($\alpha = 1$) improves results for 11 datasets against MTM whereas it yields best performance for 10 and 11 datasets compared to the student with ($\alpha = 0$) and ($\alpha = 0.5$), respectively.

We further analyse the performance of the student models considering the STM which is also depicted in Table 1. It can be noticed that our student model has outperformed many datasets, except BC4CHEMD and CRAFT. We analyzed the performance of our teacher model (MTM) for BC4CHEMD and CRAFT datasets, and found a performance drop upto F1-score of 3% for these two datasets compared to STM. Therefore, we assume that teacher MTM model could not able to perform much knowledge distillation for these two datasets. The student model ($\alpha = 0$) obtained best performance for 13 datasets compared to the STM. Likewise, student ($\alpha = 0.5$) obtained a performance gain for 12 datasets whereas student ($\alpha = 1$) attains performance for 13 datasets compared to STM.

We also use the second approach to train our student model where logits from an ensemble of MTMs is used to train the student model. Instead of using the teacher model with a different architecture, we use the same MTM teacher model but these teacher models are initialized with different seed values. For this reason, all the 5 teacher models produce different predictions. We average their logits and train each single student model on such logits.

Table 2 represents the results comparison of our second approach. We can notice the remarkable improvement in results for the student models using ensemble approach. We notice that for two protein datasets (BioNLP13GE and Ex-PTM), our student models are unable to show an increase in results compared to the teacher (MTM). However, the student models are able to show a performance gain for other protein datasets; for which our previous approach of student model does not show increase in performance. We observed that the student model with distillation loss ($\alpha = 0$) shows performance gain for 11 datasets against teacher model (MTM). Similarly, considering both the losses ($\alpha = 0.5$) i.e., student loss and distillation loss, the student model is able to leverage the results for 13 and 6 datasets compared to teacher model and student model ($\alpha - 0$), respectively. On the other hand, student model trained with only student loss ($\alpha = 1$) achieves performance gain for 13 datasets compared to the teacher model (MTM). Whereas, it is able to enhance the results for 6 datasets compared to both student models with $\alpha = 0$ and $\alpha = 0.5$. Comparing the results with STM, we can notice that all the student models have shown performance gain for all 15 datasets compared to STM. We see our student models, trained with the logits from ensemble of MTMs, produce better results. This is because ensemble predictions are more accurate than a single prediction, and therefore our student models perform better with the ensemble approach.

Datasets	**MTM**	**STM**	**Student** $\alpha = 0$	**Student** $\alpha = 0.5$	**Student** $\alpha = 1$
AnatEM	86.78	86.53	*87.56*	87.55	**87.63**
BC2GM	79.68	81.07	*81.25*	81.04	**81.29**
BC4CHEMD	86.80	**90.24**	89.45	89.50	*89.58*
BC5CDR	87.49	88.09	**88.33**	88.30	*88.32*
BioNLP09	88.40	87.37	88.70	**88.82**	*88.74*
BioNLP11EPI	**84.56**	82.58	*84.45*	84.44	**84.56**
BioNLP11ID	**87.26**	85.58	*86.98*	86.77	86.91
BioNLP13CG	**83.83**	82.11	83.27	*83.39*	83.35
BioNLP13GE	**80.06**	75.38	77.64	*78.08*	77.84
BioNLP13PC	*88.17*	87.26	88.05	88.09	**88.22**
CRAFT	81.96	**84.27**	*83.98*	*83.98*	83.81
ExPTM	**80.69**	73.06	76.11	76.39	*76.71*
JNLPBA	70.40	70.86	*72.14*	**72.20**	72.02
linnaeus	88.32	87.88	88.49	*88.58*	**88.91**
NCBI	84.50	83.98	**84.88**	84.72	*84.67*
Average	83.93	83.08	84.08	84.12	**84.17**
Average Variance	0.17	0.27	**0.15**	0.21	0.24

Table 1. Results comparison of the proposed student models. The Average represents the average F1-score of all datasets. The Average Variance represents the average variance of all datasets.

We also compare our results with state-of-the-art models. Table 3 compares the results of our proposed student model with Wang et al. [31] and Crichton et al. [6] models. We use their published results instead of regenerating them. Wang et al. and Crichton et al. have used the MTL approach and used the same 15 datasets to train their MTM. Our MTM structure resembles with the proposed model of Wang et al. but we use a task specific BiLSTM layer, and we use Softmax instead of CRF. In the given table, we can notice that our proposed approach shows substantial increase in F1-score compare to the model proposed by Crichton et al., while model proposed by Wang et al. shows performance gain for 5 datasets. The student model, $\alpha = 1$, shows the best results against the benchmark results. The comparison of our second approach of student models, trained with ensemble of MTMs, is depicted in Table 4. We see that our second approach again outperformed against Crichton et al., while shows absolute gain for most of the datasets compared to the Wang et al. except for BioNLP13GE and Ex-PTM.

We also performed a statistical analysis of our results using the Friedman test [34], shown in Figure 2. We are interested to see if the difference in the results among different models is statistically significant or not. We observe that the student models trained with single teacher logits (our first approach) do not produce statistically significant results with respect to the teacher model.

Datasets	**MTM**	**STM**	**Student*** $\alpha = 0$	**Student*** $\alpha = 0.5$	**Student*** $\alpha = 1$
AnatEM	86.78	86.53	*87.97*	*87.97*	**88.04**
BC2GM	79.68	81.07	**81.96**	81.78	*81.89*
BC4CHEMD	86.80	90.24	**90.48**	**90.47**	*90.45*
BC5CDR	87.49	88.09	**88.76**	88.68	*88.71*
BioNLP09	88.40	87.37	89.05	**89.12**	*89.08*
BioNLP11EPI	84.56	82.58	*84.73*	84.72	**84.89**
BioNLP11ID	*87.26*	85.58	87.05	**87.52**	87.37
BioNLP13CG	83.83	82.11	83.80	*83.88*	**84.00**
BioNLP13GE	**80.06**	75.38	*78.61*	78.60	78.60
BioNLP13PC	88.17	87.26	*88.72*	**88.76**	88.52
CRAFT	81.96	84.27	***85.15***	*84.89*	*84.89*
ExPTM	**80.69**	73.06	76.93	77.17	*77.33*
JNLPBA	70.40	70.86	*72.51*	**72.54**	72.50
linnaeus	88.32	87.88	**89.44**	*89.05*	88.84
NCBI	84.50	83.98	**86.12**	85.70	*85.66*
Average	83.93	83.08	**84.75**	84.72	84.72
Average Variance	0.17	0.27	**0.09**	0.21	0.11

Table 2. Results comparison of proposed student models. The Average represents the average F1-score of all datasets. The Average Variance represents the average variance of all datasets.(★ The student model trained with ensemble of MTMs.)

This is understandable as the student model is unable to show performance gain for most of the datasets against the teacher model (Table 1). However, the results produced by that student model (our first approach) are statistically significant, considering the results of STM. On the other hand, results of our second approach of student model (trained with an ensemble of MTMs' logits), represented as Ens_MTM, are statistically significant compared to both teacher (MTM) and STM. We also see that our student models' approaches, with and without ensemble approach, produce results statistically significant with each other. We also see that the student models trained without ensemble of MTM's logits are not significantly different among themselves. The same behavior can be noticed for our second approach of student models trained with ensemble MTM's logits.

In Figure 3, the models are shown according to their best statistical ranks, decreasing from left to right. The arrows show that a difference in results between models is statistically significant with $p < 0.001$. The group of student models trained with an ensemble of MTMs, shown in black dashed rectangle, are statistically better than the rest of the other models. In particular, the student model (St_Ens_$\alpha = 0$) is statistically better than those of the other models. This shows that our second approach learns much better with only distillation loss. We also consider our first group of student training (trained without ensemble

Datasets	**Wang**	**Crichton**	**Student**	**Student**	**Student**
	et al. [31]	**et al.** [6]	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$
AnatEM	86.04	82.21	*87.56*	87.55	**87.63**
BC2GM	78.86	73.17	*81.25*	81.04	**81.29**
BC4CHEMD	88.83	83.02	89.45	*89.50*	**89.58**
BC5CDR	88.14	83.90	**88.33**	88.30	*88.32*
BioNLP09	88.08	84.20	88.70	**88.82**	*88.74*
BioNLP11EPI	83.18	78.86	*84.45*	84.44	**84.56**
BioNLP11ID	83.26	81.73	**86.98**	86.77	*86.91*
BioNLP13CG	82.48	78.90	83.27	**83.39**	83.35
BioNLP13GE	**79.87**	78.58	77.64	*78.08*	77.84
BioNLP13PC	**88.46**	81.92	88.05	88.09	*88.22*
CRAFT	82.89	79.56	**83.98**	**83.98**	*83.81*
ExPTM	**80.19**	74.90	76.11	76.39	*76.71*
JNLPBA	**72.21**	70.09	72.14	*72.20*	72.02
linnaeus	*88.88*	84.04	88.49	88.58	**88.91**
NCBI	**85.54**	80.37	84.88	*84.72*	84.67
Average	83.79	79.70	84.08	84.12	**84.17**
Average Variance	—	—	**0.15**	0.21	0.24

Table 3. Results comparison of proposed student models with state-of-the-art results

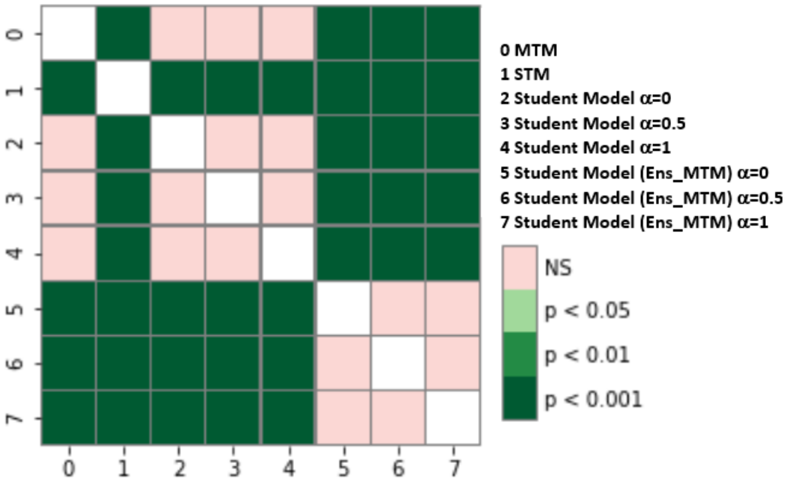

Fig. 2. Posthoc Pairwise Analysis with Conover Friedman Test

approach), as shown in the blue dashed rectangle. We find the student model (St_$\alpha = 1$), trained with student loss, is statistically better than the rest of the models shown on its right.

Datasets	**Wang** **et al.** [31]	**Crichton** **et al.** [6]	**Student**★ $\alpha = 0$	**Student**★ $\alpha = 0.5$	**Student**★ $\alpha = 1$
AnatEM	86.04	82.21	*87.97*	*87.97*	**88.04**
BC2GM	78.86	73.17	**81.96**	81.78	*81.89*
BC4CHEMD	88.83	83.02	**90.48**	*90.47*	90.45
BC5CDR	88.14	83.90	**88.76**	88.68	*88.71*
BioNLP09	88.08	84.20	89.05	**89.12**	*89.08*
BioNLP11EPI	83.18	78.86	*84.73*	84.72	**84.89**
BioNLP11ID	83.26	81.73	87.05	**87.52**	*87.37*
BioNLP13CG	82.48	78.90	83.80	*83.88*	**84.00**
BioNLP13GE	**79.87**	78.58	*78.61*	78.60	78.60
BioNLP13PC	88.46	81.92	*88.72*	**88.76**	88.52
CRAFT	82.89	79.56	***85.15***	*84.89*	*84.89*
ExPTM	**80.19**	74.90	76.93	77.17	*77.33*
JNLPBA	72.21	70.09	*72.51*	**72.54**	72.50
linnaeus	88.88	84.04	**89.44**	*89.05*	88.84
NCBI	85.54	80.37	**86.12**	85.70	*85.66*
Average	83.79	79.70	**84.75**	84.72	84.72
Average Variance	—	—	**0.09**	0.21	0.11

Table 4. Results comparison of proposed student models with state-of-the-art results(★ The student model trained with ensemble of MTMs.)

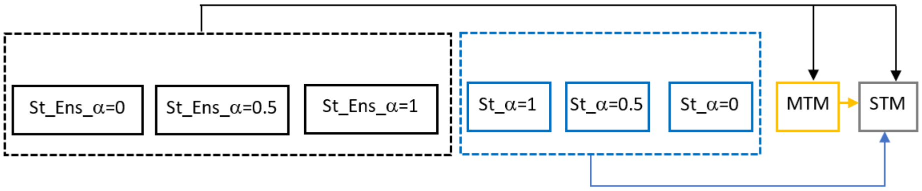

Fig. 3. Statistical Comparison of Our Models. The arrows show that models are statistically significant to another model with $p < 0.001$. St_Ens represents Student model trained with ensemble of MTMs.

6 Conclusions

In this research, we introduced knowledge distillation to increase the performance of the BioNER task. We use MTM as our teacher model because of the advantages MTM has over STM. We further use ensemble MTMs in our proposed knowledge distillation approach. The knowledge distillation is done by using MTM's logits. By analyzing the F1-score and statistical test, we found our approach better than teacher MTM and STM. We found that using the ensemble of MTMs as a teacher model is more beneficial than using a single MTM. In future work, we will use the probability distributions of the softmax prediction for student models. Furthermore, different teacher models' architecture will also be used in an ensemble approach to supervising the student model.

References

1. Alam, F., Corazza, A., Lavelli, A., Zanoli, R.: A knowledge-poor approach to chemical-disease relation extraction. Database J. Biol. Databases Curation **2016** (2016), https://doi.org/10.1093/database/baw071
2. Bansal, T., Belanger, D., McCallum, A.: Ask the GRU: Multi-task learning for deep text recommendations. In: Sen, S., Geyer, W., Freyne, J., Castells, P. (eds.) Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016. pp. 107–114. ACM (2016), https://doi.org/10.1145/2959100.2959180
3. Bingel, J., Søgaard, A.: Identifying beneficial task relations for multi-task learning in deep neural networks. In: Lapata, M., Blunsom, P., Koller, A. (eds.) Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers. pp. 164–169. Association for Computational Linguistics (2017), https://doi.org/10.18653/v1/e17-2026
4. Chowdhury, M.F.M., Lavelli, A.: Disease mention recognition with specific features. In: Cohen, K.B., Demner-Fushman, D., Ananiadou, S., Pestian, J., Tsujii, J., Webber, B.L. (eds.) Proceedings of the 2010 Workshop on Biomedical Natural Language Processing, BioNLP@ACL 2010, Uppsala, Sweden, July 15, 2010. pp. 83–90. Association for Computational Linguistics (2010), https://www.aclweb.org/anthology/W10-1911/
5. Clark, K., Luong, M., Khandelwal, U., Manning, C.D., Le, Q.V.: Bam! born-again multi-task networks for natural language understanding. In: Korhonen, A., Traum, D.R., Màrquez, L. (eds.) Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers. pp. 5931–5937. Association for Computational Linguistics (2019), https://doi.org/10.18653/v1/p19-1595
6. Crichton, G.K.O., Pyysalo, S., Chiu, B., Korhonen, A.: A neural network multi-task learning approach to biomedical named entity recognition. BMC Bioinform. **18**(1), 368:1–368:14 (2017), https://doi.org/10.1186/s12859-017-1776-8
7. Giorgi, J.M., Bader, G.D.: Transfer learning for biomedical named entity recognition with neural networks. Bioinformatics **34**(23), 4087–4094 (2018), https://doi.org/10.1093/bioinformatics/bty449
8. Gridach, M.: Character-level neural network for biomedical named entity recognition. J. Biomed. Informatics **70**, 85–91 (2017), https://doi.org/10.1016/j.jbi.2017.05.002
9. Hinton, G.E., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. CoRR **abs/1503.02531** (2015), http://arxiv.org/abs/1503.02531
10. Jung, H., Ju, J., Jung, M., Kim, J.: Less-forgetting learning in deep neural networks. CoRR **abs/1607.00122** (2016), http://arxiv.org/abs/1607.00122
11. Kim, Y., Rush, A.M.: Sequence-level knowledge distillation. In: Su, J., Carreras, X., Duh, K. (eds.) Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016. pp. 1317–1327. The Association for Computational Linguistics (2016), https://doi.org/10.18653/v1/d16-1139

12. Leaman, R., Gonzalez, G.: BANNER: an executable survey of advances in biomedical named entity recognition. In: Altman, R.B., Dunker, A.K., Hunter, L., Murray, T., Klein, T.E. (eds.) Biocomputing 2008, Proceedings of the Pacific Symposium, Kohala Coast, Hawaii, USA, 4-8 January 2008. pp. 652–663. World Scientific (2008), http://psb.stanford.edu/psb-online/proceedings/psb08/leaman.pdf
13. Liu, P., Qiu, X., Huang, X.: Adversarial multi-task learning for text classification. In: Barzilay, R., Kan, M. (eds.) Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers. pp. 1–10. Association for Computational Linguistics (2017), https://doi.org/10.18653/v1/P17-1001
14. Liu, X., He, P., Chen, W., Gao, J.: Improving multi-task deep neural networks via knowledge distillation for natural language understanding. CoRR **abs/1904.09482** (2019), http://arxiv.org/abs/1904.09482
15. Lu, P., Bai, T., Langlais, P.: SC-LSTM: learning task-specific representations in multi-task learning for sequence labeling. In: Burstein, J., Doran, C., Solorio, T. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers). pp. 2396–2406. Association for Computational Linguistics (2019), https://doi.org/10.18653/v1/n19-1249
16. Mehmood, T., Gerevini, A., Lavelli, A., Serina, I.: Leveraging multi-task learning for biomedical named entity recognition. In: Alviano, M., Greco, G., Scarcello, F. (eds.) AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11946, pp. 431–444. Springer (2019), https://doi.org/10.1007/978-3-030-35166-3_31
17. Mehmood, T., Gerevini, A., Lavelli, A., Serina, I.: Multi-task learning applied to biomedical named entity recognition task. In: Bernardi, R., Navigli, R., Semeraro, G. (eds.) Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019. CEUR Workshop Proceedings, vol. 2481. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2481/paper47.pdf
18. Mehmood, T., Gerevini, A.E., Lavelli, A., Serina, I.: Combining multi-task learning with transfer learning for biomedical named entity recognition. Procedia Computer Science **176**, 848–857 (2020)
19. Mishra, A.K., Marr, D.: Apprentice: Using knowledge distillation techniques to improve low-precision network accuracy. In: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings. OpenReview.net (2018), https://openreview.net/forum?id=B1ae1lZRb
20. Putelli, L., Gerevini, A., Lavelli, A., Serina, I.: Applying self-interaction attention for extracting drug-drug interactions. In: Alviano, M., Greco, G., Scarcello, F. (eds.) AI*IA 2019 - Advances in Artificial Intelligence - XVIIIth International Conference of the Italian Association for Artificial Intelligence, Rende, Italy, November 19-22, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11946, pp. 445–460. Springer (2019), https://doi.org/10.1007/978-3-030-35166-3_32
21. Putelli, L., Gerevini, A.E., Lavelli, A., Serina, I.: The impact of self-interaction attention on the extraction of drug-drug interactions. In: Bernardi, R., Navigli, R., Semeraro, G. (eds.) Proceedings of the Sixth Italian Conference on Computational Linguistics, Bari, Italy, November 13-15, 2019. CEUR Workshop Proceedings, vol. 2481. CEUR-WS.org (2019), http://ceur-ws.org/Vol-2481/paper61.pdf

22. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), http://arxiv.org/abs/1412.6550
23. Serrà, J., Suris, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: Dy, J.G., Krause, A. (eds.) Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018. Proceedings of Machine Learning Research, vol. 80, pp. 4555–4564. PMLR (2018), http://proceedings.mlr.press/v80/serra18a.html
24. Sun, S., Cheng, Y., Gan, Z., Liu, J.: Patient knowledge distillation for BERT model compression. In: Inui, K., Jiang, J., Ng, V., Wan, X. (eds.) Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019. pp. 4322–4331. Association for Computational Linguistics (2019), https://doi.org/10.18653/v1/D19-1441
25. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N.D., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada. pp. 3104–3112 (2014), http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks
26. Tan, X., Ren, Y., He, D., Qin, T., Zhao, Z., Liu, T.: Multilingual neural machine translation with knowledge distillation. In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), https://openreview.net/forum?id=S1gUsoR9YX
27. Tanabe, L.K., Xie, N., Thom, L.H., Matten, W., Wilbur, W.J.: GENETAG: a tagged corpus for gene/protein named entity recognition. BMC Bioinform. **6**(S-1) (2005), https://doi.org/10.1186/1471-2105-6-S1-S3
28. Tang, R., Lu, Y., Liu, L., Mou, L., Vechtomova, O., Lin, J.: Distilling task-specific knowledge from BERT into simple neural networks. CoRR **abs/1903.12136** (2019), http://arxiv.org/abs/1903.12136
29. Wang, L., Yoon, K.: Knowledge distillation and student-teacher learning for visual intelligence: A review and new outlooks. CoRR **abs/2004.05937** (2020), https://arxiv.org/abs/2004.05937
30. Wang, X., Jiang, Y., Bach, N., Wang, T., Huang, F., Tu, K.: Structure-level knowledge distillation for multilingual sequence labeling. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020. pp. 3317–3330. Association for Computational Linguistics (2020), https://www.aclweb.org/anthology/2020.acl-main.304/
31. Wang, X., Zhang, Y., Ren, X., Zhang, Y., Zitnik, M., Shang, J., Langlotz, C., Han, J.: Cross-type biomedical named entity recognition with deep multi-task learning. Bioinformatics **35**(10), 1745–1752 (2019), https://doi.org/10.1093/bioinformatics/bty869
32. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. CoRR **abs/1706.00384** (2017), http://arxiv.org/abs/1706.00384
33. Zhou, J., Cao, Y., Wang, X., Li, P., Xu, W.: Deep recurrent models with fast-forward connections for neural machine translation. Trans. Assoc. Comput. Linguistics **4**, 371–383 (2016), https://transacl.org/ojs/index.php/tacl/article/view/863

34. Zubani, M., Sigalini, L., Serina, I., Gerevini, A.E.: Evaluating different natural language understanding services in a real business case for the italian language. Procedia Computer Science **176**, 995–1004 (2020)