Early Stopping Based on Unlabeled Samples in Text Classification

HongSeok Choi[1] **Dongha Choi**[2] **Hyunju Lee**[1,2,*]
[1]School of Electrical Engineering and Computer Science,
[2]Artificial Intelligence Graduate School,
Gwangju Institute of Science and Technology, Gwangju 61005, South Korea
{hongking9,hyunjulee}@gist.ac.kr, dongha528@gm.gist.ac.kr

Abstract

Early stopping, which is widely used to prevent overfitting, is generally based on a separate validation set. However, in low resource settings, validation-based stopping can be risky because a small validation set may not be sufficiently representative, and the reduction in the number of samples by validation split may result in insufficient samples for training. In this study, we propose an early stopping method that uses unlabeled samples. The proposed method is based on confidence and class distribution similarities. To further improve the performance, we present a calibration method to better estimate the class distribution of the unlabeled samples. The proposed method is advantageous because it does not require a separate validation set and provides a better stopping point by using a large unlabeled set. Extensive experiments are conducted on five text classification datasets and several stop-methods are compared. Our results show that the proposed model even performs better than using an *additional* validation set as well as the existing stop-methods, in both balanced and imbalanced data settings. Our code is available at https://github.com/DMCB-GIST/BUS-stop.

1 Introduction

Early stopping, a form of regularization, is a widely used technique to prevent a model from over-fitting (Yao et al., 2007; Zhang et al., 2017). It is generally based on a separate validation set (Goodfellow et al., 2016). While monitoring the validation performance during training, the training process stops when the validation error starts to increase. Validation-based early stopping is advantageous because it is easy to implement and can be interpreted directly (Prechelt, 1998).

In a scenario where sufficient labeled data are available, the use of a validation set is generally preferred (Goodfellow et al., 2016). However, when only a few labeled data exist, a tradeoff problem is encountered (Kann et al., 2019; Choi and Lee, 2021). For example, although the usage of a relatively large validation set enables more reliable estimation, the number of samples for training becomes insufficient. Conversely, if small fractions of the samples are assigned to the validation set, the stopping point becomes ambiguous because the small validation set is not representative enough.

Early stopping is more important in a low resource setting because the prediction accuracy fluctuates highly during training. Such high fluctuations render it challenging when to stop the model. One way to mitigate these fluctuations is to use sufficient training data. In this context, training all the available samples would be more effective, and for this purpose, an appropriate stopping point should be determined without validation split. However, this has not been extensively studied. Duvenaud et al. (2016) and Mahsereci et al. (2017) proposed gradient-based stop-methods and applied statistical inference on the training samples. Lee and Chung (2021) suggested the usage of local intrinsic dimensionality (LID) for early stopping. In addition, some studies treat the stopping epoch as a hyperparameter: the stopping epoch is obtained by grid-search or averaging in cross validation (Choi and Lee, 2021). These methods allow the training of all the labeled samples. However, they do not consider the task-related performance metrics (e.g., accuracy) during training, and the LID and gradient-based stop criteria have not been commonly used in natural language processing (NLP). Furthermore, gradient-based stop-criteria depend on the training samples, the size of which may still be small to be representative.

In this study, we propose an early **stop**ping method **b**ased on **u**nlabeled **s**amples (BUS-stop). We are motivated by the following two considerations: (i) The probabilities of the predicted class label (i.e., the prediction confidences) can serve as

*Corresponding author

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 708 - 718

an indicator for over-fitting or under-fitting. (ii) In a better model, the output class distribution is more likely to be closer to the class distribution of the true labels. To incorporate these two assumptions, two stop criteria are proposed, and combined in the BUS-stop method. Our method monitors the prediction results of unlabeled samples during training and utilizes them for determining the stop-criteria. The first proposed stop-criterion is based on confidence similarity (conf-sim). The model stops when the prediction confidences for the unlabeled samples are most similar to the reference confidences, which are precalculated on the labeled set with cross-validation. Conf-sim is observed to reflect the long-term trend of the loss curve, and thereby assist in preventing over-training. The second stop criterion is based on the class distribution similarity (class-sim). This criterion stops the model when the predicted class distribution on the unlabeled set is most similar to the pre-estimated distribution. To this end, we present a novel estimation method for the true class distribution, which calibrates the predicted distribution by extrapolation such that it is closer to the true distribution. Class-sim is observed to reflect the short-term trend of the accuracy. Our method requires several retraining steps to obtain the reference confidences for conf-sim and the estimated class distribution for class-sim. The BUS-stop method that combines class-sim and conf-sim includes the advantages of both, and thereby performs with better accuracy and loss compared to each (class-sim and conf-sim).

The following characteristics of our method contribute to performance improvement. Our method does not require a separate validation set; hence, all the labeled samples can be trained. Training can stop at a more generalized model, using a large unlabeled set. The proposed stop-criteria, conf-sim and class-sim, consider two performance metrics, namely, the loss and accuracy.

Our contributions are summarized as follows:

- We propose BUS-stop, an early stopping method, based on unlabeled samples. BUS-stop can stop the training at a more generalized model, and the performance is better even than using an *additional* validation set.

- Furthermore, we present a calibration method to better estimate the class distribution. This method calibrates the output class distribution to render it closer to the true distribution, improving the class-sim performance.

- Extensive experiments are conducted on five popular text classification datasets in English. Comparison with several stop-methods demonstrates that the proposed method outperforms these existing stop-methods in both balanced and imbalanced data settings.

2 Related Work

Prechelt (1998) experimented on 14 different validation-based stop criteria. Prechelt (1998) focused on an issue that the validation error during training may represent many local minima prior to a global optimum.

Existing non-validation stop-criteria are generally based on statistical inference. Duvenaud et al. (2016) interpreted stochastic gradient descent in terms of the variational inference and proposed an estimation method for the marginal likelihood of the posterior, which was applied as an early stopping criterion. However, this method requires considerable computation for the Hessian, which is not practical in large models. Mahsereci et al. (2017) also proposed a gradient-related stopping method referred to as evidence-based stopping (EB). The EB-criterion is based on the fast-to-compute local statistics of the computed gradients. The criterion represents whether the gradients of the training samples lie within the expected range. Intrinsic dimensionality (ID), which refers to the minimum number of parameters required to represent a dataset, has been used for analyzing the training or redundancy of neural networks (Amsaleg et al., 2015). LID is a version of ID that estimates the subspace dimensions of the local regions. Lee and Chung (2021) found that LID works well as a stopping-criterion in several few-shot image classification datasets. Moreover, LID can be applied to unlabeled samples. Another method involves the pre-estimation of the the number of training epochs by training the model multiple times, such as cross validation (Choi and Lee, 2021); the model can stop at the pre-estimated (PE) stop-epoch when training all the labeled samples.

However, these methods have not been commonly studied for NLP tasks and do not consider the performance metrics during training. Furthermore, comparisons among the non-validation stop-methods have not been reported. In this study, we compare our method with the EB, LID, PE, and validation-based stopping methods on five text classification datasets. The method proposed by

Algorithm 1 Preliminary stage for BUS-stop

Input: Labeled set D_l, Unlabeled set D_u
Output: Sorted output probabilities $\vec{P_l}$, Calibrated class distribution $\vec{C_u}$

```
Let Count[1 ··· n_l] = 0
Let P_l[1 ··· n_l] = 0
for t ∈ {1, ··· , T} do
    Initialize a model, M
    Split D_l into D_train and D_val at a ratio of r
    Train the M with (D_train, D_val)
    M ← load the M that was the best on D_val
    for x_i ∈ D_val do
        p_i ← M(x_i)
        P_l[i] = P_l[i] + p_i
        Count[i] = Count[i] + 1
    end for
    Ĉ_u ← M(D_u)
    Ĉ_val, Acc_val ← M(D_val)
    C⃗_u^t = Calibration(Ĉ_u, Ĉ_val, Acc_val)
end for
for x_i ∈ D_l do
    P_l[i] = P_l[i]/Count[i]
end for
P⃗_l ← sort P_l in ascending (or descending) order
C⃗_u = Σ_{t=1}^{T} C⃗_u^t / T
return P⃗_l, C⃗_u
```

Duvenaud et al. (2016) was not compared because it involves considerable computational cost.

3 Method

In this section, we describe the proposed method in detail. The main notations used are as follows: $D_l = \{(x_i, y_i)\}_{i=1}^{n_l}$ and $D_u = \{(x_i)\}_{i=1}^{n_u}$ denote the labeled and unlabeled sets, respectively. x_i and y_i are the i-th sample and its true label, respectively, and n_l and n_u are the numbers of labeled and unlabeled samples, respectively. p_{ij} denotes the prediction probability of the j-th class on the i-th sample. Let C be the true class distribution of the samples. The output probability (i.e., confidence) p_i associated with the predicted label on sample x_i and the predicted (i.e., output) class distribution $\hat{C}$ of the samples are defined as follows:

$$p_i = \max_j(p_{ij})$$
$$\hat{C}[j] = \sum_{i=1}^{n_{data}} p_{ij}/n_{data}$$

where $\forall j \in \{1, \cdots, n_c\}$; n_c is the number of classes.

3.1 Preliminary Stage

The pseudocode for the preliminary stage is summarized in Alg. 1. In the preliminary stage, the prediction confidences $\vec{P_l}$ for the labeled samples in D_l and the estimated class distribution $\vec{C_u}$ of the unlabeled set D_u are calculated. Using D_l, the model is reinitialized-and-retrained T-times using a resampling method such as cross-validation. In low-resource settings, such retraining enables more reliable predictions by averaging the results. Each sample in P_l is evaluated when the validation loss is the lowest. Each sample should be validated at least once; the prediction confidences are averaged for each sample. P_l (and P_u in Alg.2 as well) is sorted in order of size for confidence comparison between two different sample sets, D_l and D_u, in the main stage; we denoted it as $\vec{P_l}$ ($\vec{P_u}$ for P_u). When retraining T-times, the output class distributions of the unlabeled set D_u are obtained and calibrated (this calibration is defined in Section 3.3). Then, the T calibrated class distributions are averaged, resulting in $\vec{C_u}$. After this stage, $\vec{P_l}$ and $\vec{C_u}$ are used to calculate the similarities for the two stop criteria, conf-sim and class-sim, respectively.

3.2 Main Stage Applying BUS-stop

After the preliminary stage, we train all the labeled samples and refer to this stage as the main stage. The combined BUS-stop method applied in the main stage is summarized in Alg. 2. The unlabeled set is predicted at every epoch during training.

Conf-sim The first proposed stop criterion conf-sim S_{conf} represents the similarity of the prediction confidences $\vec{P_u}$ for the unlabeled samples with the reference confidences $\vec{P_l}$. To calculate the similarity between $\vec{P_u}$ and $\vec{P_l}$, their dimensions must be the same. We sample $\vec{P_u}$ at regular intervals $\frac{n_u}{n_l}$ such that it is the same size as $\vec{P_l}$ and denoted it as $\dddot{P}_u$. We use the Euclidean distance to calculate the similarity, resulting in S_{conf}. Then, the first stop criterion is when S_{conf} has the lowest value, i.e., $\dddot{P}_u$ is most similar to $\vec{P_l}$. There is a natural concern that $\dddot{P}_u$ is likely to produce higher (thus dissimilar) confidences than $\vec{P_l}$ because $\dddot{P}_u$ is obtained by training all the labeled samples, unlike $\vec{P_l}$. However, the fact that the confidence for each sample in $\vec{P_l}$ is obtained when the validation error is the lowest can alleviate this concern. Thereby, S_{conf} can be a rough criterion for avoiding under- and overfitting, and can reflect the trend of the loss, based on comparison with the reference confidences.

Algorithm 2 BUS-stop in main stage

Input: D_l, D_u, $\vec{P_l}$, $\vec{C_u}$
Output: Expected best model M_{best}
  Let $Queue[1 \cdots n_{que}] = 0$
  Let $B_{conf} = \text{inf}$, and $n_{pat} = 0$
  Initialize a model, M
  for $epoch \in \{1, 2, 3, \cdots\}$ **do**
    Train the M one epoch on D_l
    $P_u, \hat{C}_u \leftarrow M(D_u)$
    $\vec{P_u} \leftarrow$ sort P_u in ascending (or descending) order
    $\dddot{P}_u \leftarrow$ sampling $\vec{P_u}$ at regular intervals $\frac{n_u}{n_l}$
    $S_{conf} = Euclidian\text{-}distance(\dddot{P}_u, \vec{P_l})$
    $S_{class} = Cosine\text{-}similarity(\hat{C}_u, \vec{C_u})$
    if $S_{conf} < B_{conf}$ **then**
      $n_{pat} = 0$ and $Queue[1 \cdots n_{que}] = 0$
      $B_{conf} = S_{conf}$
    else
      $n_{pat} = n_{pat} + 1$
    end if
    if $n_{pat} < n_{que}$ **then**
      if $S_{class} > \max(Queue)$ **then**
        $M_{best} \leftarrow$ save the current M
      end if
      $Queue \xleftarrow[\text{enqueue}]{\text{dequeue \&}} S_{class}$
    else
      End training
    end if
  end for
  return M_{best}

Class-sim The second proposed stop criterion is class-sim, S_{class}. The predicted class distribution $\hat{C}_u$ on the unlabeled set is compared with the estimated class distribution $\vec{C_u}$ from the preliminary stage. The assumption is that a well-trained model can also predict the class distribution more accurately. Therefore, estimation of the true class distribution is crucial. A calibration method that facilitates better estimation of the class distribution is presented in Section 3.3. We use the cosine similarity to calculate the similarity between $\hat{C}_u$ and $\vec{C_u}$, and obtain S_{class}. The second stop criterion is when S_{class} has the highest value, i.e., $\hat{C}_u$ is most similar to $\vec{C_u}$. Thereby, S_{class} can reflect the short-term trend of the accuracy because it is more likely that the outputs of a higher accuracy model are closer to the true class distribution.

BUS-stop Finally, we combine the two stop-criteria, conf-sim and class-sim, to form the BUS-stop method, as depicted in Alg. 2. A simple product of the two stop criteria can be an ineffective stop criterion because the sizes of S_{conf} and S_{class} are relative. Our combined stop-criterion is to save the model with the highest S_{class} among of the epochs from the lowest S_{conf} to the subsequent $(n_{que}-1)$-th epoch. This technique enables fine-stopping by considering both S_{conf} and S_{class}, which reflect the long-term and short-term performances, respectively. It is to be noted that early stopping methods should be operated as an ongoing process, and not as a type of post-hoc method. To this end, we use a fixed-size queue $Queue$, and its size n_{que} as a hyperparameter, as shown in Alg. 2.

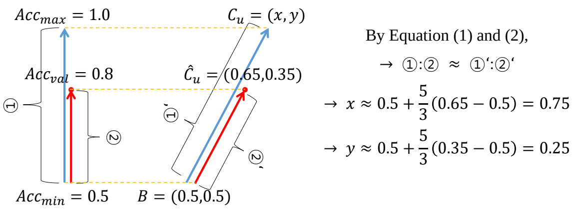

Figure 1: Calibration example in binary classification.

3.3 Calibration of Class Distribution

In this section, we describe the calibration of the predicted class distribution. The calibration method aims to better estimate the true class distribution of the unlabeled set, thereby improving the performance of class-sim, particularly for imbalanced classification.

Trained neural networks often involve sampling biases. For example, in binary classification, the prediction results of a model trained with a class ratio a:b tend to follow the distribution of a:b. Thus, when the class distributions are different in the test and training sets, the model performance can deteriorate. Let us suppose the following somewhat ideal and naive situations. Let C_u be the true class distribution of the unlabeled set. If the model is perfectly trained with an accuracy of 1.0, the output class distribution will be equal to C_u. On the other hand, if the model fails to learn any inference knowledge from training, the model will output the predictions only by its sampling bias; i.e., when the accuracy is the same as the random expectation (denoted as Acc_{min}, e.g., 0.5 in binary classification), the output class distribution will be equal to the sampling bias B. Thus, the model accuracy can reflect whether the output class distribution is closer to the sampling bias or the true distribution. In the preliminary stage, we obtained the models' proxy accuracy and output class distribution as Acc_{val}

Data	Class	Train	Test	Len
SST-2	2	6.9K	1.8K	19
IMDB	2	25K	25K	231
Elec	2	25K	25K	107
AG-news	4	120K	7.6K	38
DBpedia	14	560K	70K	49

Table 1: Statistics for datasets. **Len** denotes the average number of words per sample.

and $\hat{C}_u$, respectively. Assuming that there is an approximate linear relationship, we can define a proportional expression as follows:

$$\begin{aligned}(1 - Acc_{min}) : (Acc_{val} - Acc_{min}) \\ \approx (C_u - B) : (\hat{C}_u - B)\end{aligned} \tag{1}$$

We rearrange the above expression in terms of C_u:

$$C_u \approx B + \frac{(1 - Acc_{min})}{(Acc_{val} - Acc_{min})}(\hat{C}_u - B) \tag{2}$$

Then, we denote the approximation of C_u as $\vec{C}_u$. Considering the class distribution as a vector, Eq. (2) is a type of extrapolation. B can be defined as the class distribution of D_{train} or the predicted distribution in the validation set, $\hat{C}_{val}$, of the preliminary stage. In addition, the Acc can be replaced with F1-score. Fig. 1 illustrates an example of our calibration method.

4 Experimental

4.1 Datasets

We conducted extensive experiments using five text classification datasets. The statistics are summarized in Table 1. These datasets have been extensively used in NLP research, and are publicly available. The SST-2 (Socher et al., 2013), IMDB (Maas et al., 2011), and Elec (McAuley and Leskovec, 2013) datasets are used for sentiment analysis. SST-2 and IMDB include movie reviews, and Elec includes reviews on Amazon electronics. AG-news (Zhang et al., 2015) and DBpedia (Zhang et al., 2015) are topic classification tasks for Wikipedia and news articles, respectively. For each dataset, we sampled K labeled samples per class from the training set. K was set to 50 for low-resource settings; we also experimented by varying $K \in \{50, 100, 200, 400, 800, 1600\}$. We used the test samples as the unlabeled set for each dataset, which is referred to as transductive setting in few-shot classification (Liu et al., 2019).

4.2 Methods for Comparison

In this section, we describe the various stop-criteria for comparison with our method.

EB The EB (Mahsereci et al., 2017) is a criterion based on gradients of training samples. The EB-criterion stops when the following condition is met:

$$1 - \frac{|\mathcal{S}|}{D}\sum_{k=1}^{D}[\frac{(\nabla L_{\mathcal{S},k})^2}{\hat{\Sigma}_k}] > 0 \tag{3}$$

where $\mathcal{S}$ represents a sample set, D is the number of parameters, ∇L is the gradients of loss, and subscript k indicates the k-th weight of the total parameters. $\hat{\Sigma}$ is the variance estimator, which is calculated as follows:

$$\hat{\Sigma}_k = \frac{1}{(|\mathcal{S}| - 1)}\sum_{x \in \mathcal{S}}(\nabla l_k(x) - \nabla L_{\mathcal{S},k})^2 \tag{4}$$

where $\nabla l(x)$ is the loss gradient on sample x. Note that $L_{\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{x \in \mathcal{S}} l(x)$. For further details, refer Mahsereci et al. (2017).

LID Lee and Chung (2021) approximated LID as follows:

$$LID = -\sum_{x \in D_u}\left[\frac{1}{m}\sum_{i=1}^{m}\ln\frac{d_i(\vec{z}(x))}{d_m(\vec{z}(x))}\right]^{-1} \tag{5}$$

where $\vec{z}(x)$ is the representation vector of sample x, and d_i is the Euclidean distance of $\vec{z}(x)$ and its i-th nearest neighbor. m is a hyperparameter, which denotes the number of nearest neighbors. The lowest LID is the stop criterion.

Val-stop$_{split(x)}$ and Val-stop$_{add(x)}$ Val-stop denotes validation-based stopping. Val-stop$_{split(x)}$ indicates that x validation samples per class are taken from the labeled set. Therefore, $K-x$ samples are trained and x samples are validated for each class. Val-stop$_{add(x)}$ indicates that x additional samples per class are used for validation; i.e., Val-stop$_{add(x)}$ uses a total of $K+x$ labeled samples per class. Val-stop$_{add(x)}$ has an unfair advantage because it uses additional labeled samples.

PE-stop-epoch The stopping epoch is considered a hyperparameter, which is **p**re-**e**stimated with cross-validation, as described in Section 2. We use four-fold cross-validation.

Conf-sim and **class-sim** can also be used as a single stop-criterion, as mentioned before. We compare the single criteria with the combined BUS-stop criterion. Conf-sim stops when S_{conf} is the lowest, and class-sim stops when S_{class} is the highest.

Dataset	SST-2		IMDB		Elec		AG-news		DBpedia		Average	
Method	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss	Acc	Loss
Val-stop$_{split(25)}$	0.775	0.516	0.746	0.572	0.781	0.507	0.846	0.477	0.982	0.085	0.826	0.431
EB	0.826$^{\simeq}$	0.565	**0.833**$^{\simeq}$	0.551	0.843$^{\simeq}$	0.534	0.861	0.491	0.986$^{\simeq}$	0.103	0.869	0.449
LID	0.794	0.602	0.761	0.571	0.815	0.494	0.859	0.515	0.971	0.765	0.840	0.589
PE-stop-epoch	0.816	0.628	0.826$^{\simeq}$	0.585	0.837	0.524	0.859	0.487	0.985	0.079	0.865	0.460
Conf-sim (ours)	0.807	**0.442**$^{\simeq}$	0.793	0.484$^{\simeq}$	0.823	0.433$^{\simeq}$	0.863$^{\simeq}$	**0.421**	0.985$^{\simeq}$	0.077$^{\simeq}$	0.854	0.371
Class-sim (ours)	0.795	0.570	0.789	0.560	0.793	0.531	0.857	0.561	**0.986**$^{\simeq}$	0.078	0.844	0.460
BUS-stop (ours)	**0.831**	0.455	0.828	**0.456**	**0.848**	**0.417**	**0.865**	0.432	**0.986**	**0.074**	**0.872**	**0.367**
*Val-stop$_{add(25)}$	0.819	0.431	0.824$^{\simeq}$	0.447$^{\simeq}$	0.842$^{\simeq}$	0.407$^{\simeq}$	0.867	0.415	0.986$^{\simeq}$	0.075$^{\simeq}$	0.868	0.355

Table 2: Performance comparison of different stop-criteria in balanced classification. We used 50 labeled samples per class for all stop-criteria except for Val-stop$_{add(25)}$. *Note that the Val-stop$_{add(25)}$ has an unfair advantage: for each class, it used 25 additional labeled samples for validation while using 50 labeled samples for training. The best performances, except for the Val-stop$_{add(25)}$, are denoted in bold. '$\simeq$' denotes that the performance is statistically similar to the BUS-stop (i.e., p-value over 0.05).

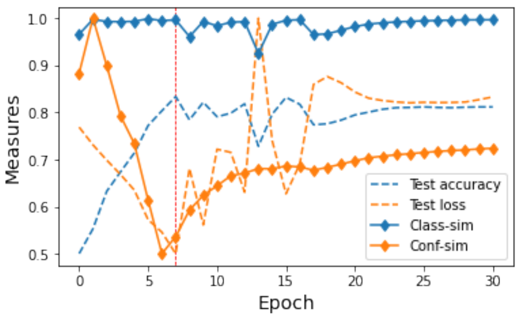

Figure 2: Example of the accuracy and loss curves with SST-2 dataset. The loss and conf-sim were scaled between 0.5-1.0. The red vertical line denotes the best model selected by the BUS-stop method.

4.3 Implementation

BERT-base (Devlin et al., 2019) was adopted as our text encoder. The Adam optimizer (Kingma and Ba, 2015) was applied for categorical cross-entropy loss (i.e., $-\sum y_i \log p_i$), and its learning rate was set to $3e$-5. The dropout (Srivastava et al., 2014) was set to 0.2, and the batch size was 16. All the stop-criteria were evaluated simultaneously for each run to reduce the variance of the estimation. We averaged 10 results in all the experiments. In EB, 64 random training samples were used for S in Eq. (3). In LID, the final vector of the [CLS] token in the BERT model was assigned to $\vec{z}(x)$ in Eq. (5), and the best m was selected from $\{5, 10, 20, 50, 100\}$. In BUS-stop, n_{que} in Alg. 2 was set to five. Note that K is the number of training samples per class. When K was set to 50, T and r in the preliminary stage (see Alg. 1) were set to 5 and 1:1, respectively. When K was set above 50, T and r were set to 4 and 3:1, respectively. In our calibration method, we used $\hat{C}_{val}$ as B and macro F1-score as the Acc_{val}.

	SST-2	IMDB	Elec	AG-news	Avg.
Val-stop$_{split(25)}$	0.052	0.070	0.049	0.020	0.048
EB	0.119	0.123	0.117	0.074	0.109
LID	0.088	0.076	0.058	0.052	0.069
PE-stop-epoch	0.131	0.122	0.107	0.069	0.107
Conf-sim (ours)	**0.036**	**0.064**	**0.040**	**0.011**	**0.038**
Class-sim (ours)	0.079	0.069	0.064	0.059	0.068
BUS-stop (ours)	0.072	0.071	0.061	0.039	0.061
Val-stop$_{add(25)}$	0.035	0.056	0.045	0.021	0.039

Table 3: Over-confidence error (OE) of different stop-criteria. In DBpedia, all the OEs were close to zero.

5 Results

5.1 Balanced Classification

Table 2 shows the results when K=50 for training. It is noted that the original test sets have a balanced class distribution. We also report the loss measure as well as accuracy because loss can imply over-training. As shown in Table 2, our BUS-stop method exhibits the best performance on an average, and the accuracy is better even than Val-stop$_{add(25)}$, which uses a larger numbers of labeled samples. Note that Val-stop$_{add(25)}$ uses a total of 75 labeled samples per class. The performance of Val-stop$_{split(25)}$ indicates that splitting data for validation can result in poor performance in low-resource settings. LID underperforms compared to the PE-stop-epoch that does not require unlabeled samples. Conf-sim shows the second-best loss on an average. Class-sim underperforms as a stop criterion by itself. However, the BUS-stop method, which combines these two methods, shows better performance than each one on an average. Figure 2 displays the results of conf-sim and class-sim over the epochs. More examples are presented in Appendix A. In Fig. 2, the conf-sim curve is similar to the long-term trend of the loss; however, it does

Dataset	SST-2			IMDB			Elec			Average		
Method	Acc	F1	Loss	Acc	F1	Loss	Acc	F1	Loss	Acc	F1	Loss
Val-stop$_{split(25)}$	0.788	0.719	0.499	0.732	0.674	0.589	0.783	0.724	0.507	0.768	0.706	0.532
EB	0.846$^{\simeq}$	0.786$^{\simeq}$	0.504	0.810	0.749	0.568	0.839	0.789	0.541	0.832	0.775	0.537
LID	0.750	0.698	0.632	0.712	0.668	0.678	0.780	0.728	0.574	0.747	0.698	0.628
PE-stop-epoch	0.843	0.779	0.527	0.821	0.763	0.589	0.843	0.789	0.521	0.836	0.777	0.545
Conf-sim (ours)	0.816	0.754	0.427	0.813	0.750	0.432$^{\simeq}$	0.835	0.775	0.398	0.821	0.760	0.419
Class-sim (ours)	**0.862**$^{\simeq}$	**0.797**$^{\simeq}$	0.489	0.844$^{\simeq}$	0.779$^{\simeq}$	0.510	0.873$^{\simeq}$	0.807$^{\simeq}$	0.409	0.860	0.794	0.469
BUS-stop (ours)	0.860	0.792	**0.379**	**0.849**	**0.787**	**0.406**	**0.876**	**0.815**	**0.343**	**0.861**	**0.798**	**0.376**
Val-stop$_{add(25)}$	0.823	0.767	0.412	0.820	0.767	0.457	0.837	0.784	0.407	0.827	0.773	0.426

Table 4: Performance comparison in an imbalanced setting of binary classification tasks. We used 50 labeled samples per class for training (i.e., K=50), and the class distributions of the test sets were adjusted to 2:8 (negative:positive). '$\simeq$' denotes that the performance is statistically similar to the BUS-stop (i.e., p-value over 0.05).

Train	Test	2:8	4:6	6:4	8:2
2:8	EB	**0.845**	**0.732**	0.643	0.511
	BUS-stop (ours)	0.828	0.719	**0.669**	0.521
	Val-stop$_{add(25)}$	0.679	0.660	0.621	**0.634**
4:6	EB	0.860	0.820	0.790	0.728
	BUS-stop (ours)	**0.864**	**0.825**	**0.815**	**0.808**
	Val-stop$_{add(25)}$	0.820	0.808	0.801	0.794
6:4	EB	0.790	0.816	0.825	0.845
	BUS-stop (ours)	**0.845**	**0.826**	**0.833**	**0.864**
	Val-stop$_{add(25)}$	0.826	0.824	0.823	0.824
8:2	EB	0.611	0.696	0.774	**0.870**
	BUS-stop (ours)	**0.682**	**0.714**	**0.793**	0.865
	Val-stop$_{add(25)}$	0.667	0.707	0.733	0.782

Avg.: EB=0.760, BUS-stop=**0.779**, Val-stop$_{add(25)}$=0.750

Table 5: Accuracy comparison in various imbalanced settings (negative:positive) of the SST-2. The bold denotes the best performance of the three stop-criteria.

not accurately reflect the short-term fluctuation of the performance from epochs 7–16. On the other hand, class-sim is observed to be well responsive to the short-term fluctuation of the accuracy, but does not reflect the long-term trend. BUS-stop, which is a combination of these two methods, takes advantage of the short- as well as long-term methods, and thereby facilitates fine stopping. The EB-criterion shows the statistically similar accuracy to the BUS-stop method in most datasets. In the EB-criterion and PE-stop-epoch, the average loss is not good enough compared to the high accuracy. The accuracy and loss show somewhat conflicting results. That was due to over-confidence on the misclassified samples, caused by over-training. Note that $Loss = -\sum y_i \log p_i$. Overconfidence on the wrong label makes p_i close to zero on its true label y_i. Thus, excessively low p_i can increase the loss drastically. Table 3 lists the over-confidence error (OE); the equation for OE is presented in Thulasidasan et al. (2019). This confidence error can be detrimental in various applications, as described by Guo et al. (2017).

5.2 Imbalanced Classification

We experimented with an imbalanced setting in binary classification tasks. For testing, we sampled 1,000 instances in the SST-2 test set, and 10,000 instances each in the IMDB and Elec test sets, with a class distribution of 2:8 (negative:positive). The macro F1-score is also reported. Table 4 shows the results when K was set to 50 for training. In most cases, BUS-stop exhibits the best performance with respect to the accuracy as well as loss. In addition, it is noted that BUS-stop outperforms the other methods with a greater margin in an imbalanced setting than in a balanced one (Table 2). It is observed that ratios marked with '$\simeq$' are fewer in the imbalanced setting. Class-sim shows the best or second-best accuracy among the datasets. It is observed that the output class distribution can be an important indicator for a better model.

Table 5 shows the results in various imbalanced settings of the SST-2 (both the training and test sets are imbalanced). The number of training samples was fixed to 100 for the different class-distribution settings. In general, when the class distributions of the training and test sets are similar, the results shows better performance for all the three methods, EB, BUS-stop, and Val-stop$_{add(25)}$. In most cases, BUS-stop consistently outperforms Val-stop$_{add(25)}$ and EB, and the margin is greater when the class distributions are more different between the training and test sets. This result indicates that BUS-stop is robust to imbalanced classification.

6 Discussion

Impact of the training size Figure 3 indicates the accuracy curve with respect to the training size, using the IMDB dataset. The x values of Val-stop$_{add(x)}$ and Val-stop$_{split(x)}$ were set to 25, 25, 50, 100, 200, and 400, according to the increase

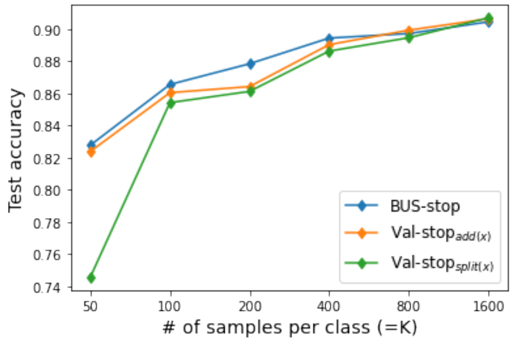

Figure 3: Accuracy by different training sizes in IMDB.

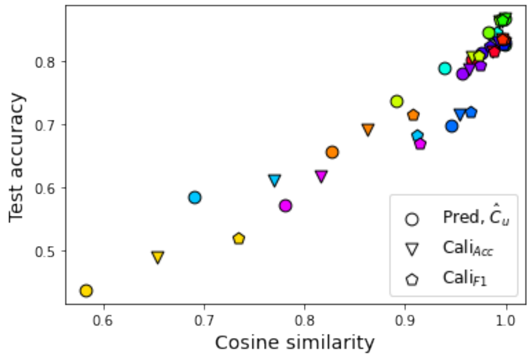

Figure 4: BUS-stop accuracy for different class distribution estimators in the 16 imbalanced settings depicted in Table 6.

Train	Test	2:8	4:6	6:4	8:2
2:8	Pred, $\hat{C}_u$	**0.999**	0.946	0.781	0.583
	Cali$_{Acc}$	0.999	0.954	0.816	0.653
	Cali$_{F1}$	0.997	**0.965**	**0.915**	**0.734**
4:6	Pred, $\hat{C}_u$	0.986	**0.999**	0.966	0.892
	Cali$_{Acc}$	0.997	0.998	0.987	0.966
	Cali$_{F1}$	**0.998**	0.998	**0.989**	**0.973**
6:4	Pred, $\hat{C}_u$	0.939	0.976	**0.998**	0.983
	Cali$_{Acc}$	0.989	**0.992**	0.997	0.994
	Cali$_{F1}$	**0.991**	0.984	0.997	**0.994**
8:2	Pred, $\hat{C}_u$	0.691	0.827	0.957	**0.999**
	Cali$_{Acc}$	0.770	0.863	0.964	0.999
	Cali$_{F1}$	**0.912**	**0.908**	**0.975**	0.996

Avg.: $\hat{C}_u$=0.908, Cali$_{Acc}$=0.934, Cali$_{F1}$=**0.958**

Table 6: Cosine similarity between the class distribution of the test set and the estimated distribution in various imbalanced settings of the SST-2 dataset.

Method	Time complexity	Measured time SST-2 ($n_u = 1.8$k)	Measured time DBpedia ($n_u = 70$k)
EB	$g(n_l) + \alpha$	0.32 m	0.49 m
LID	$g(n_l) + p(n_u)$	0.12 m	**5.02** m
PE-stop-epoch	$(T+1) * g(n_l)$	**0.43** m	1.14 m
BUS-stop	$(T+1) * g(n_l) + p(n_u)$	**0.47** m	**5.97** m
Val-stop$_{add(25)}$	$g(n_l)$	0.07 m	0.19 m

Table 7: Running time comparison for different stop-criteria. The two longest times are denoted in bold.

in K. It can be observed that the performance of BUS-stop is good in the sufficient-data regime as well. However, the performances of the three stop-criteria converge almost similarly with the increase in the training size. The impact of splitting the samples for validation does not deteriorate the performance when K is greater than 400. Rather, Val-stop$_{split(x)}$ performs slightly better when K is 1600. This result suggests that when sufficient labeled data are available, validation-based stopping can be a better choice.

Calibration performance In the BUS-stop method, accurate estimation of the class distribution plays a crucial role. The cosine similarity between the class distribution of the test set and the estimated distributions by various estimators are shown in Table 6, where the uncalibrated output distributions ($\hat{C}_u$) and the estimated distributions by the calibration methods, based on the Acc-score (Cali$_{Acc}$) and macro F1-score (Cali$_{F1}$), were compared. When the class distributions are similar between the test and training sets, the performance of $\hat{C}_u$ is slightly better than those of the other estimators. However, the estimation by calibration based on the F1-score (Cali$_{F1}$) is better on an average, and particularly when the class distributions of the test and training sets are different. Figure 4 indicates the BUS-stop accuracies when each model stops based on the estimated class distribution in Table 6 (the same color corresponds to one cell in Table 6). For example, the yellow colors correspond to the settings in which the class distribution is 2:8 and 8:2 in the training and test sets, respectively. As shown in Fig. 4, the better the class distribution is estimated, the higher is the accuracy of BUS-stop. Such high correlation indicates the importance of the class distribution estimator. This result is consistent with our assumption that the output class distribution of better models will be closer to the true distribution.

Running time The running times are not directly comparable owing to the different hyperparameter settings for each method. For example, the BUS-stop and PE-stop-epoch require a separate preliminary stage that consumes additional time. We add up both the times taken in the preliminary stage and main stage. We denote the average running time per epoch as $g(n_l)$ for training the labeled samples and $p(n_u)$ for predicting the unlabeled samples. The time complexity and the measured time are shown in Table 7. Note that T is

Method	Val-stop$_{split(25)}$		Val-stop$_{add(25)}$		BUS-stop
Selection	local	global	local	global	local
Balanced classification					
SST-2	0.775	0.785	0.819	**0.840**	0.831
IMDB	0.746	0.786	0.824	**0.838**	0.828
Elec	0.781	0.805	0.842	**0.852**	0.848
AG-news	0.846	0.857	0.867	**0.871**	0.865
Imbalanced classification					
SST-2	0.788	0.807	0.823	0.832	**0.860**
IMDB	0.732	0.757	0.820	0.834	**0.849**
Elec	0.783	0.820	0.837	0.853	**0.876**

Table 8: Accuracy by global selection in Val-stop.

the number of retrainings in the preliminary stage, which was set to five. The experimental settings are the same as in Section 5.1. The time measurement was conducted on a PC with an Intel Core i7 CPU, 64-GB RAM and an NVIDIA Titan X Pascal GPU. As shown in the expression of time complexity, the running time depends on the numbers of labeled and unlabeled samples, n_l and n_u, respectively. In DBpedia, which has a large number of unlabeled samples, n_u, the LID and BUS-stop methods take the two longest running times. On the other hand, in SST-2, the PE-stop-epoch and BUS-stop methods show the two longest running times, because the n_u is relatively small such that the $g(n_l)$ is more dominant than the $p(n_u)$. The BUS-stop requires a longer running time than other methods due to the T-times retraining and the continual prediction on the unlabeled set. To reduce the time, we can adjust the T value or sample a smaller amount of data from the unlabeled set.

Limitations The proposed BUS-stop method was designed for classification tasks, and thereby can be applied when the model can output confidences. Regression tasks as well can be addressed by converting into classification problems. The continuous values normalized between 0-1 can be represented as confidences in a binary classification. However, it may be difficult to apply to other more complex tasks (e.g., text summarization). This study is limited to classification tasks. Another limitation is that the BUS-stop, which is a non-validation stop-method, cannot make direct comparisons between two models with different runs. Early stopping can be seen as selecting the best resulting model over the epochs. In a similar way, it is also possible to select the best model among multiple runs. We refer to the former as local selection and the latter as global selection. In validation-based stopping, the global selection is simply to select the model with the lowest validation loss over multiple runs. However, the non-validation methods have no clear criterion for this purpose. We repeated training five runs for each and selected the best model among the runs based on validation loss. Other experimental settings are the same as in Section 5. As shown in Table 8, the global selection in validation-based stopping improves performance across the datasets in both balanced and imbalanced settings. However, in the imbalanced setting, the BUS-stop still results in better performance. Note that Val-stop$_{add(25)}$ uses additional labeled samples. We also report that the global selections that are based on the S_{conf}, S_{class}, and LID did not show significant performance improvement in our experiment. The development of non-validation global selection methods is left for future work.

7 Conclusion and Future Work

Validation-based early stopping can be detrimental in low-resource settings because the reduction in the number of samples by validation split may result in insufficient samples for training. In this study, we proposed an early stopping method called BUS-stop, based on unlabeled samples. Moreover, we proposed a calibration method to better estimate the true class distribution, which was used in the BUS-stop method to improve the performance. We conducted experiments on five popular text classification datasets. The results indicated that BUS-stop outperformed the existing stop-criteria in both balanced and imbalanced settings. In particular, BUS-stop showed robustness to imbalanced classification. The proposed BUS-stop method enables the training of all the available samples and presents a better stopping point using large unlabeled samples. In future, we plan to better exploit the unlabeled samples in self-training schemes. We can also combine BUS-stop and self-training methods. BUS-stop can be used to improve the performance of the initial model, which plays an important role in the final self-training performance. Additionally, we consider applying the BUS-stop to domain adaptation tasks in the future.

Acknowledgements

This research was supported by the Bio-Synergy Research Project (NRF-2016M3A9C4939665) of the Ministry of Science and ICT through the National Research Foundation of Korea (NRF) and the NRF grant funded by the Korean government (Ministry of Science and ICT) (NRF-

2018M3C7A1054932), and partly supported by Institute for Information and communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) [No. 2019-0-01842, Artificial Intelligence Graduate School Program (GIST)].

References

Laurent Amsaleg, Oussama Chelly, Teddy Furon, Stéphane Girard, Michael E. Houle, Ken-ichi Kawarabayashi, and Michael Nett. 2015. Estimating local intrinsic dimensionality. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 29–38, New York, NY, USA. Association for Computing Machinery.

HongSeok Choi and Hyunju Lee. 2021. Exploiting all samples in low-resource sentence classification: early stopping and initialization parameters. *arXiv preprint arXiv:2111.06971*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

David Duvenaud, Dougal Maclaurin, and Ryan Adams. 2016. Early stopping as nonparametric variational inference. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, pages 1070–1077, Cadiz, Spain. PMLR.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1321—1330. JMLR.org.

Katharina Kann, Kyunghyun Cho, and Samuel R. Bowman. 2019. Towards realistic practices in low-resource natural language processing: The development set. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3342–3349, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Dong Hoon Lee and Sae-Young Chung. 2021. Unsupervised embedding adaptation via early-stage feature reconstruction for few-shot classification. In *Proceedings of the 38th International Conference on Machine Learning*, pages 6098–6108. PMLR.

Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sungju Hwang, and Yi Yang. 2019. Learning to propagate labels: Transductive propagation network for few-shot learning. In *International Conference on Learning Representations*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Maren Mahsereci, Lukas Balles, Christoph Lassner, and Philipp Hennig. 2017. Early stopping without a validation set. *arXiv preprint arXiv:1703.09580*.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: Understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, page 165–172, New York, NY, USA. Association for Computing Machinery.

Lutz Prechelt. 1998. *Early Stopping - But When?*, pages 55–69. Springer Berlin Heidelberg, Berlin, Heidelberg.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2017. Understanding deep learning requires rethinking generalization.

In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 2017, Conference Track Proceedings*. OpenReview.net.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

A Appendix

Fig. 5 provides several examples of the learning curves and the stop-criteria measurements over the epochs.

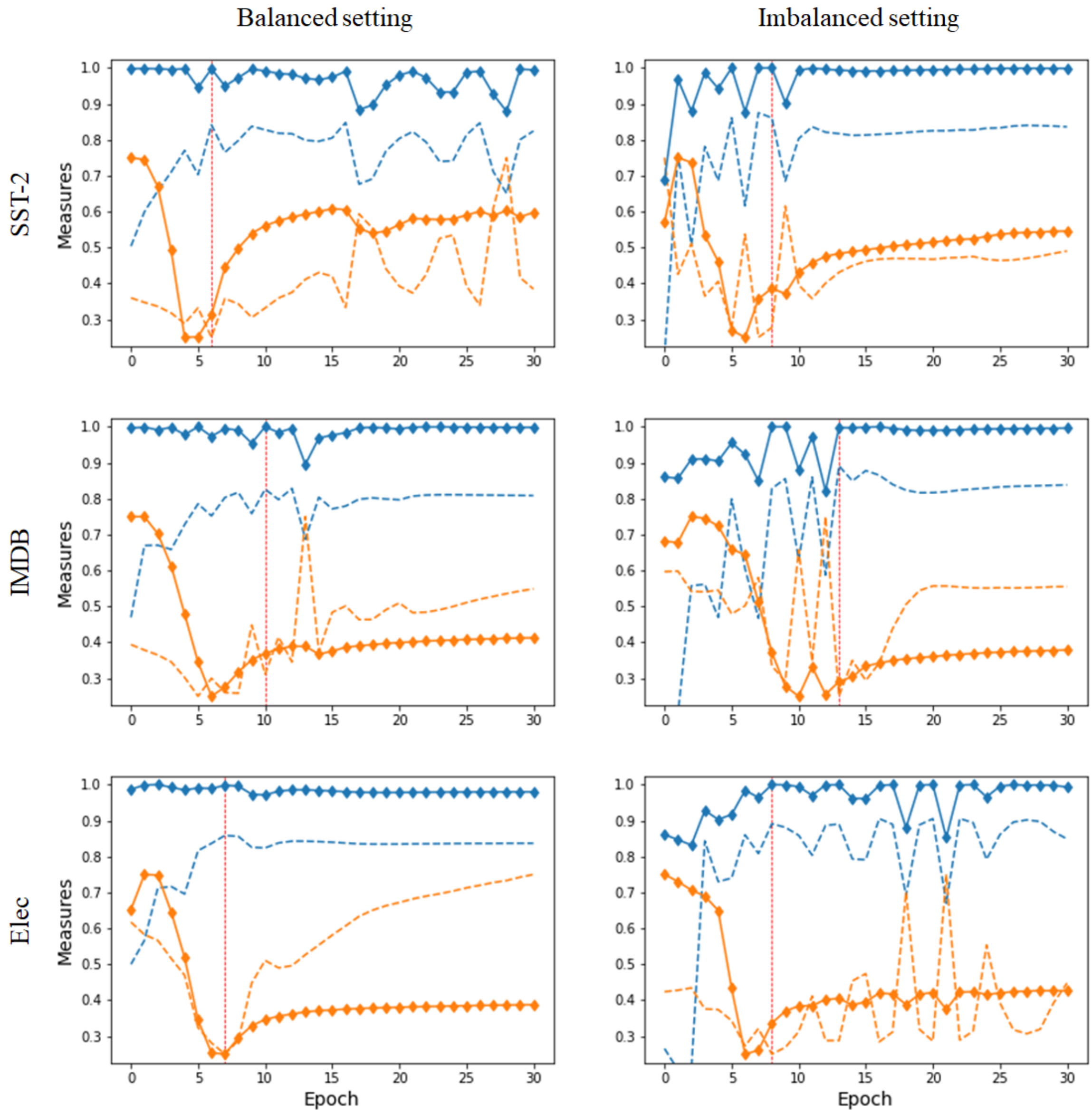

Figure 5: Examples in balanced and imbalanced settings of the SST-2, IMDB, and Elec datasets. and denotes conf-sim and class-sim, respectively; and denotes the test loss and accuracy, respectively. The red vertical line denotes the best model selected by the BUS-stop method. The balanced and imbalanced settings are the same as the settings in Section 5.1 and 5.2, respectively. The loss and conf-sim were scaled between 0.25-0.75 for easy comparison. The BUS-stop enables fine-stopping. As shown in these figures, our method skillfully avoids the points where the performance is decreased by fluctuations.