STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **16**, December 2026, pp 6096–6105.
Published online in International Academic Press (www.IAPress.org)

An Empirical Study of Label Smoothing in Transformer-Based Text Classification

ALI ALSHAHRANI[1,*], XIUWEN LIU[2], KHALID ALHARTHI[1], AHMED ABDELGHANY[1,*], SAAD ALSHAHRANI[3]

[1]*Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Bisha 61922, P.O. Box 551, Saudi Arabia*
[2]*Department of Computer Science, Florida State University, Tallahassee, USA*
[3]*Department of English Language and Literature, University of Bisha, Bisha 61922, Saudi Arabia*

Abstract Regularization methods play an important role in improving the generalization performance of deep neural networks. Label smoothing is an output-level regularization technique that replaces one-hot targets with softened class distributions and can reduce excessive confidence in model predictions. This study examines label smoothing in Transformer-based text classification across the Optimism/Pessimism Twitter (OPT), TweetEval, and FinancialPhraseBank benchmarks. BERT, RoBERTa, and ELECTRA are evaluated under hard-label and label-smoothing training conditions. The study also examines the sensitivity of performance to the smoothing coefficient and examines representation geometry and self-attention behavior as complementary analyses. A validation-based screening of $\alpha \in \{0.00, 0.05, 0.10, 0.15\}$ was conducted, and $\alpha = 0.05$ was selected for the main experimental protocol based on aggregate validation performance.

Keywords BERT; transformer; fine-tuning; label smoothing; regularization; attention

DOI: 10.19139/soic-2310-5070-3645

1. Introduction

Deep neural networks training often includes minimizing the cross-entropy loss function concerning hard target labels expressed as one-hot encoding vectors. Unfortunately, this method may result in overfitting, which is a substantial problem because of the poor generalization capabilities of the deep learning algorithms. To prevent it, regularization methods like dropout [1], weight penalties (e.g., L1-norm and L2-norm), and data augmentation [2] were developed; however, they operate on the network weights or activations rather than the output layer itself.

Label smoothing (LS) [3] is an output-level regularization technique which became a popular choice for improving the quality of models as well as reducing overconfidence. It uses softening of the target labels through their replacement by an interpolation of the one-hot vector and uniform distribution over all the classes; it was proved that it affects not only generalization but also confidence of the model predictions [4, 5]. Recent works investigated label smoothing on various text sentiment datasets and Transformer models (BERT and RoBERTa) with different smoothing coefficients used per dataset/model [6]. Despite its success, the mechanisms of label smoothing are not completely understood yet [7].

Studies have considered the impact of label smoothing as well as its role in making the model generalize well by avoiding the overconfidence in the training prediction. Various methods have been developed to enhance label smoothing. “Fake Label” [8] is an approach where part of the probability mass is shifted to a fake class. This

*Correspondence to: ALI ALSHAHRANI (Email: afmohamad@ub.edu.sa) and AHMED ABDELGHANY (Email: ahmedaee@outlook.com). Department of Computer Science and Artificial Intelligence, College of Computing and Information Technology, University of Bisha, Bisha 61922, P.O. Box 551, Saudi Arabia.

ISSN 2310-5070 (online) ISSN 2311-004X (print)
Copyright © 2026 International Academic Press

approach has made considerable advancements in the task of text classification as well as cross-lingual transfer learning. Another enhancement that is made with respect to label smoothing is a dynamic soft pseudo label method that creates dynamic soft labels using prediction history in the training process. This method optimizes model parameters based on the minimization of the distance between dynamic soft labels and prediction while applying Mix-up data augmentation in order to avoid overfitting in the dataset, which results in considerable advances in the task of text classification [9]. Other enhancements include discrimination-aware label smoothing [10] and AMPLIFY [11]. Label smoothing has been studied in relation to domain generalization, with special focus on its connection with overfitting and inter-class dependencies [5]. Concerning sentiment analysis, label distribution based supervision has been studied in the context of predictive and calibrating performance [12].

This study explores how label smoothing works empirically for Transformer-based text classification on several datasets and different Transformer models. The experiments explore the effect of label smoothing on OPT, TweetEval, and FinancialPhraseBank, as well as the performance of BERT, RoBERTa, and ELECTRA under both hard-label and label-smoothed learning modes.

1.1. Early Regularization Techniques

The first regularization method that was developed and gained popularity was Dropout, which was invented by Geoffrey Hinton and his team [1]. Dropout consists of randomly dropping out some neurons during the model training and forces the model to learn more general representations and prevents feature co-adaptation. L1 and L2 regularization techniques for constraining model parameters were also used. Batch Normalization was another important approach suggested by Sergey Ioffe and Christian Szegedy and consisted of normalizing activations of each layer, which helped to accelerate and improve the process of training [13]. Moreover, techniques such as random cropping, flipping, and rotation of images can be used for enlarging the training set in computer vision applications. While all the aforementioned methods affect weights or hidden activations of the model, none of them touches the output distribution of the model. The problem of over-confidence in predictions of the model is still relevant, especially for large models trained on noisy and small datasets.

1.2. Label Smoothing as Output Regularization

To resolve this problem, Szegedy et al. [3] have developed label smoothing which is used as an output regularizer. Label smoothing replaces the one-hot vector used as a target with a mixture of ground truth and uniform distribution over classes. It makes the model predict with lower confidence in the result, thus preventing overconfidence from happening. At first, label smoothing has been proposed as a method for image classification using Inception-v4 and has shown stable improvements. Later on, the research showed that label smoothing is not limited to improving classification performance but also affects calibration and distribution of the predicted probabilities. Current research in the field of sentiment analysis has highlighted the necessity to take both the accuracy and reliability of calibration into account when training models using label distribution as supervision [12]. Muller et al. [7] have extensively studied the effect of label smoothing and concluded that the representations produced by label smoothing models in penultimate layers were denser and more structured compared to the hard-label trained ones.

1.3. Label Smoothing in Transformer-based Architectures

Though the technique of label smoothing came from computer vision, the application of label smoothing in natural language processing was made possible by the success of Transformer architectures [14]. Language models using transformers such as Bidirectional Encoder Representations from Transformers (BERT) and XLNet rely extensively on their pre-training and fine-tuning phases. Overfitting is an issue during the fine-tuning phase, especially for small downstream data sets. Recent research in text sentiment classification has found that the impacts of label smoothing must be considered based on the particular dataset and model architecture used and not automatically advantageous [6]. For short-text sentiment analysis, recent BERT-based approaches have incorporated attention mechanisms to better capture sentiment-related information in short and heterogeneous texts [15]. Label smoothing was applied in the classification parts of the Transformer architecture to improve

generalization performance, better calibrate the model, and stabilize the training dynamics. For instance, Clark et al. [16] studied attention mechanisms in BERT and found out that each head is responsible for some syntactic tasks. Further research shows that label smoothing helps Transformer to cope with noisy labels [17], deal with unbalanced datasets, and work under low resource conditions, where hard targets encourage memorization of patterns instead of learning generalized features.

The contribution of this study is empirical rather than methodological. Specifically, the study (i) evaluates label smoothing across multiple text-classification benchmarks and Transformer architectures, (ii) examines the sensitivity of performance to the smoothing coefficient through a validation-based ablation, (iii) reports paired uncertainty and significance analyses, and (iv) uses qualitative error analysis, representation geometry, and self-attention observations to characterize how the effect varies across settings.

The rest of this study is organized as follows. Section 2 describes the mathematical formulation of label smoothing, the training protocol, and the benchmark datasets. Section 3 presents the validation-based coefficient selection, classification results, exploratory statistical analysis, and qualitative error analysis. Section 4 presents the complementary attention analysis and its limitations. Finally, Section 5 concludes the paper.

2. METHODS AND MATERIALS

2.1. Label Smoothing Technique

This part presents a mathematical formulation of the label smoothing technique. Let the prediction generated by the neural network be defined as a function of the activations in its penultimate layer:

$$p_c = \text{softmax}(w_c^T x + b_c) \tag{1}$$

where p_c represents the probability assigned by the model to the cth class, w_c and b_c denote the corresponding weight and bias parameters, respectively, and x is the vector of activations in the penultimate layer.

For models trained using hard labels, where the correct class is represented by "1" and all others by "0," the objective is to minimize the cross-entropy between the true labels y_c and the predicted probabilities p_c. The cross-entropy loss is defined as:

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C} y_c \log p_c, \tag{2}$$

where y_c denotes the one-hot encoded target for class c, and C is the total number of classes.

When label smoothing is applied, the training objective instead minimizes the cross-entropy between the smoothed targets y_c^{LS} and the network predictions p_c, where the smoothed targets are defined as:

$$y_c^{LS} = (1-\alpha)y_c + \frac{\alpha}{C}, \tag{3}$$

where α is the label smoothing coefficient, C is the total number of classes, and y_c is the hard target for class c (0 or 1). The corresponding training objective is defined as:

$$\mathcal{L}_{LS} = -\sum_{c=1}^{C} y_c^{LS} \log p_c. \tag{4}$$

This formulation effectively softens the target distribution, reducing overconfidence by assigning a small portion of the probability mass to the other classes.

2.2. Dataset Characteristics

Three benchmark datasets are used in this study: OPT, TweetEval, and FinancialPhraseBank. Table 1 summarizes the fixed benchmark splits used in the experiments.

OPT (Optimism/Pessimism Twitter) is a sentiment dataset compiled by [18]. The original annotation was conducted using Amazon Mechanical Turk, with several annotators rating sentiment on a continuous scale. In this study, the fixed benchmark split is used, and binary classification is obtained using the thresholding technique proposed by [19]. The benchmark contains 7,475 tweets categorized as optimistic or pessimistic.

TweetEval [20] is used as a three-class sentiment benchmark. The standardized configuration and fixed train, validation, and test splits are retained in the revised experiments.

FinancialPhraseBank [21] is used as a non-social-media benchmark for financial sentiment classification. It provides a structured text classification setting that complements the informal social-media content represented by OPT and TweetEval.

The dataset sizes used in the experiments are summarized in Table 2. OPT contains 5,980 training, 747 validation, and 748 test instances; TweetEval contains 45,615 training, 2,000 validation, and 12,284 test instances; and FinancialPhraseBank contains 3,100 training, 776 validation, and 970 test instances. There are also differences in average text length across the datasets: the mean training-set text lengths are approximately 13.7 words for OPT, 19.2 words for TweetEval, and 23.1 words for FinancialPhraseBank. These differences provide a basis for examining whether the observed behavior of label smoothing varies across informal social-media text and more structured text classification.

Table 1. Dataset sizes used in the experiments.

Dataset	Train	Validation	Test
OPT	5,980	747	748
TweetEval	45,615	2,000	12,284
FinancialPhraseBank	3,100	776	970

2.3. Training Process

The training framework fine-tunes three base-scale Transformer encoders: BERT-base [22], RoBERTa-base, and ELECTRA-base. For each architecture, a classification head is added to the pretrained encoder. The final hidden representation used for classification is obtained from the model-specific sequence representation. The same training objective and evaluation protocol are applied to the hard-label and label-smoothing conditions so that the principal comparison isolates the effect of the target distribution.

The fixed benchmark splits are retained, and the main experiments use paired runs across hard-label and label-smoothing conditions. The inferential analysis uses corresponding paired runs for the evaluated models and covers BERT, RoBERTa, and ELECTRA. Inputs are truncated or padded to a maximum length of 128 tokens. Fine-tuning uses AdamW [23], a learning rate of 2×10^{-5}, and a warm-up ratio of 0.10. The main comparison uses zero weight decay so that the effect of label smoothing is not confounded with optimizer regularization. The selected label-smoothing coefficient is $\alpha = 0.05$, based on the completed validation-only alpha screening described below.

2.4. Label-Smoothing Alpha Selection

The smoothing coefficient was selected through a validation-only ablation on the two designated validation benchmarks, FinancialPhraseBank and TweetEval, to reduce the computational time required for coefficient screening. OPT was retained for the main benchmark evaluation. Four values, $\alpha \in \{0.00, 0.05, 0.10, 0.15\}$, were evaluated using multiple replicated validation runs for each coefficient and dataset. The dataset-specific results are reported in Table 2, and the corresponding validation curves are shown in Figure 1. The aggregate mean validation accuracy and macro-F1, obtained by averaging the dataset-level means with equal weight across the two validation datasets, were highest at $\alpha = 0.05$ (79.769% and 77.822%, respectively). This was an aggregate selection criterion; $\alpha = 0.05$ was not the best value on every individual dataset. Accordingly, $\alpha = 0.05$ was fixed before the main test-set evaluation and is used as the common smoothing coefficient in the main experiment. This selection is based exclusively on validation data and does not use test-set performance.

Table 2. Validation-based alpha ablation used to select the smoothing coefficient. Values are mean (SD) across replicated validation runs for each dataset and coefficient. The aggregate rows are the equal-weight means of the two dataset-level means.

Dataset	α	Accuracy (%)	Macro-F1 (%)
FinancialPhraseBank	0.00	85.052 (0.387)	82.817 (0.727)
	0.05	**85.138 (0.581)**	**82.991 (0.797)**
	0.10	85.138 (0.324)	82.978 (0.561)
	0.15	84.708 (0.710)	82.566 (0.843)
TweetEval	0.00	74.417 (0.029)	72.764 (0.110)
	0.05	74.400 (0.132)	72.653 (0.107)
	0.10	74.267 (0.104)	72.455 (0.260)
	0.15	74.300 (0.482)	72.508 (0.575)
Aggregate mean	0.00	79.734	77.790
	0.05	**79.769**	**77.822**
	0.10	79.702	77.717
	0.15	79.504	77.537

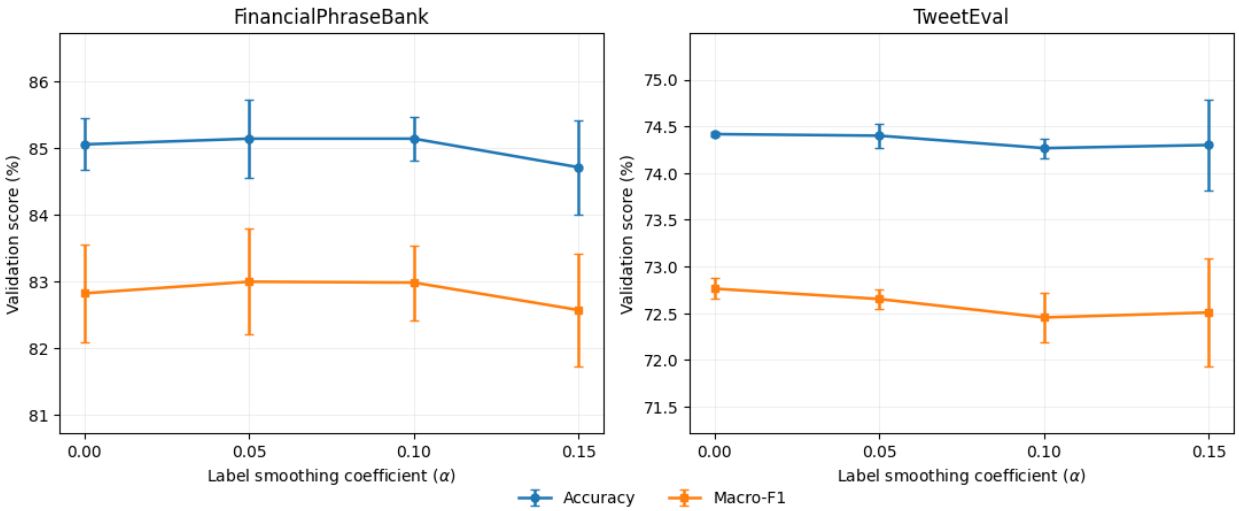


Figure 1. Performance of the validation procedure as a function of label smoothing coefficient in BERT with FinancialPhraseBank and TweetEval. The markers indicate the mean performance of the validation procedure through repeated validation procedures, with the error bars indicating the standard deviation.

2.5. Computational Environment

All experiments were conducted using an NVIDIA Tesla T4 GPU with 14.56 GB of GPU memory and 15 GB of system RAM. The experiments were implemented in Python using PyTorch 2.11.0 with CUDA 12.8. The computational environment was used for model fine-tuning, validation-based alpha selection, and test-set evaluation.

2.6. Feature Extraction and Representation Geometry

In this part, the output from the last layer is analyzed to understand the representation process of the model. The main goal is to find out how well the clusters formed by the neural network when the training is done with and without label smoothing. In this case, the BERT base model was fine-tuned and the hidden states related to [CLS]

token from the last layer were selected as input representations. From these input representations, the Silhouette score was computed, which can be described as:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \tag{5}$$

Here, $a(i)$ and $b(i)$ are obtained as follows. The value of $a(i)$ represents the mean distance from instance i to all other data points within the same cluster:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, j \neq i} d(i, j) \tag{6}$$

In this equation, $d(i, j)$ denotes the distance between data points i and j. The term $b(i)$ refers to the mean distance from i to all points in the nearest neighboring cluster C_k, where $C_k \neq C_i$, and is calculated as:

$$b(i) = \min_{k \neq C_i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i, j) \tag{7}$$

The Silhouette value measures how similar each data point is to those in the same cluster and how dissimilar it is to points in other clusters. The Silhouette metric is used as a descriptive measure of representation geometry when comparing the available hard-label and label-smoothing representations.

For BERT on FinancialPhraseBank, the final-layer [CLS] representations were evaluated on the 970 test instances across the paired runs. The descriptive representation-geometry results are reported in Table 3 and Figure 2. The mean Silhouette coefficient was 0.0921 for hard-label training and 0.0883 for label smoothing, while the corresponding mean Davies–Bouldin indices were 2.7307 and 2.8014, respectively. These measures provide a complementary view of representation geometry alongside the classification results.

Table 3. Mean descriptive representation-geometry results for BERT on FinancialPhraseBank using the final-layer [CLS] representation.

Training Condition	Silhouette	Davies–Bouldin
Hard labels	0.0921	2.7307
Label smoothing	0.0883	2.8014

3. EXPERIMENTAL RESULTS AND COMPARISONS

The experiments described in this section begin with the selection of the smoothing coefficient through validation. Then the main experiment compares the use of hard labels versus label smoothing through training according to the defined protocol.

Selection of the smoothing coefficient is done by evaluating the four different candidate values based only on the validation dataset. As shown in Table 2 and Figure 1, alpha = 0.05 provides the highest aggregate validation accuracy and macro-F1 among the candidate values. Dataset-level plots show that the selected coefficient is not necessarily optimal for each individual dataset. The coefficient was pre-defined before testing on the main dataset. The main comparison between hard labels and label smoothing is made with paired runs.

3.1. Main Classification Results

The test accuracy results of the three selected Transformer architectures are presented in Table 4. The figures are provided as averages of paired experiments, while the standard deviations are in parentheses.

The results in Table 4 show variation between the hard-label and label-smoothing conditions across the evaluated datasets and Transformer architectures. The mean paired change (label smoothing minus hard labels) was -0.095

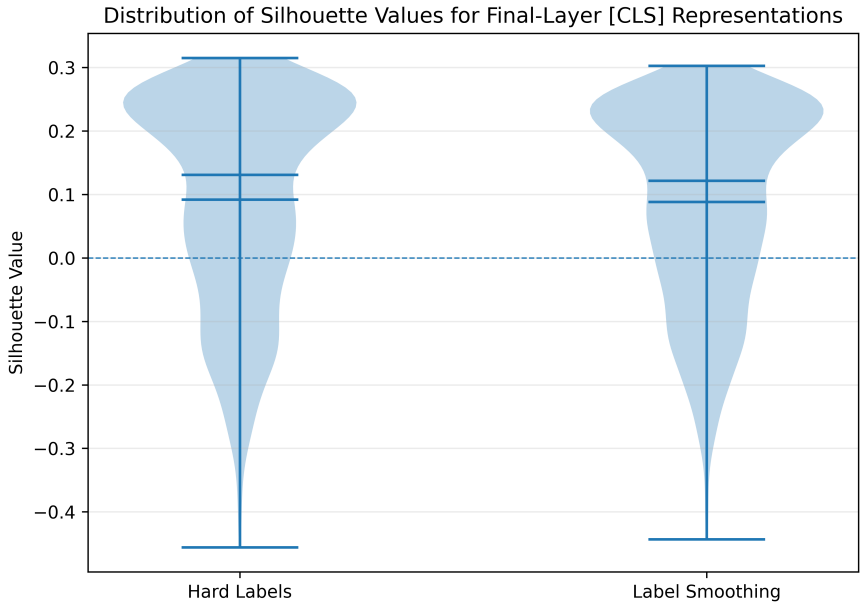


Figure 2. Silhouette coefficient distribution for the [CLS] representations produced by BERT on FinancialPhraseBank using hard label and label smoothing training. The violin plot shows the distribution between paired experiments, with the mean and median shown within each distribution.

Table 4. Test-set accuracy and macro-F1 across datasets and Transformer architectures. Values are mean (SD) across runs.

Dataset	Model	Accuracy (%)		Macro-F1 (%)	
		Hard-label	Label-smoothing	Hard-label	Label-smoothing
FinancialPhraseBank	BERT	83.35 (1.39)	83.14 (1.24)	81.02 (1.77)	80.77 (2.01)
FinancialPhraseBank	ELECTRA	83.97 (0.66)	84.07 (0.80)	81.95 (1.16)	82.04 (1.48)
FinancialPhraseBank	RoBERTa	85.05 (0.44)	85.00 (0.36)	84.52 (0.22)	84.09 (0.27)
OPT	BERT	84.63 (0.95)	84.56 (1.42)	83.77 (0.91)	83.67 (1.51)
OPT	ELECTRA	84.63 (0.76)	84.76 (0.57)	83.79 (0.75)	83.80 (0.62)
OPT	RoBERTa	85.03 (0.95)	84.49 (1.13)	84.15 (1.02)	83.55 (1.19)
TweetEval	BERT	70.38 (0.19)	70.33 (0.27)	70.24 (0.10)	70.20 (0.16)
TweetEval	ELECTRA	71.60 (0.52)	71.52 (0.37)	71.59 (0.50)	71.47 (0.36)
TweetEval	RoBERTa	71.52 (0.34)	71.41 (0.06)	71.57 (0.34)	71.48 (0.07)

percentage points for accuracy and -0.171 percentage points for macro-F1. The paired analysis yielded $p = 0.179$ for accuracy and $p = 0.048$ for macro-F1. These results indicate that the magnitude and direction of the observed differences vary across dataset–architecture combinations and evaluation metrics.

Label smoothing modifies the target distribution by assigning a portion of the target probability mass to the other classes. Accordingly, its effect can be examined in relation to the characteristics of the dataset, the Transformer architecture, and the evaluation metric. The present analysis therefore considers label smoothing as an experimental training condition and examines the resulting differences across the evaluated settings without assuming a uniform effect.

To complement accuracy, the analysis also considers test cross-entropy (CE), which reflects the quality of the predicted probability distribution rather than classification correctness alone. Lower CE indicates a closer match between the predicted distribution and the observed target distribution. The corresponding CE results are reported separately in Table 5.

For the inferential analysis, the findings under the hard-label and label-smoothing conditions for each dataset and architecture were paired within their corresponding runs, after which the run-level effects were averaged for each dataset–architecture combination. The process generated paired effects across the three datasets and three Transformer architectures in the evaluation set. The mean paired change (LS minus Hard) was -0.095 percentage points for accuracy (SD $= 0.194$, 95% CI $[-0.245, 0.054]$, paired t-test $p = 0.179$, Cohen's $d_z = -0.491$) and -0.171 percentage points for macro-F1 (SD $= 0.220$, 95% CI $[-0.340, -0.002]$, paired t-test $p = 0.048$, Cohen's $d_z = -0.777$). The corresponding Wilcoxon signed-rank tests gave $p = 0.250$ for accuracy and $p = 0.027$ for macro-F1.

The results therefore support a context-dependent interpretation of label smoothing in the present benchmark: its effect varies with dataset, architecture, and evaluation metric.

Table 5. Test cross-entropy (CE) for hard-label and label-smoothing training across the three datasets and retained Transformer architectures. Values are mean (SD) across replicated runs; lower values indicate a lower test cross-entropy.

Dataset	Model	Hard-label CE	Label-smoothing CE
FinancialPhraseBank	BERT	0.438 (0.029)	0.446 (0.021)
FinancialPhraseBank	ELECTRA	0.499 (0.022)	0.494 (0.025)
FinancialPhraseBank	RoBERTa	0.360 (0.003)	0.364 (0.005)
OPT	BERT	0.379 (0.017)	0.367 (0.021)
OPT	ELECTRA	0.371 (0.010)	0.375 (0.020)
OPT	RoBERTa	0.357 (0.008)	0.353 (0.005)
TweetEval	BERT	0.685 (0.0003)	0.671 (0.0001)
TweetEval	ELECTRA	0.666 (0.001)	0.650 (0.002)
TweetEval	RoBERTa	0.660 (0.003)	0.646 (0.004)

The CE results complement the accuracy analysis by showing that the two training objectives can differ in the quality of their predicted probability distributions even when accuracy differences are small. For example, label smoothing has lower mean CE than hard labels for BERT and RoBERTa on OPT and for all three retained architectures on TweetEval, whereas the pattern on FinancialPhraseBank varies by architecture. Accordingly, the analysis considers both classification correctness and predictive-distribution behavior rather than reducing the comparison to a single performance measure.

3.2. Qualitative Error Analysis on TweetEval

To examine the error structure of short social-media text, we analyzed the class-transition patterns on the TweetEval sentiment test set for the paired BERT, ELECTRA, and RoBERTa comparisons. The dominant errors were confusions between neutral and the two sentiment classes. For BERT, the hard-label model produced 1,058 neutral-to-negative, 979 negative-to-neutral, 817 neutral-to-positive, and 599 positive-to-neutral errors; the corresponding LS model produced 1,041, 981, 837, and 592 errors. ELECTRA and RoBERTa showed the same general pattern, with neutral-to-negative and neutral-to-positive transitions accounting for the largest error groups, whereas direct negative-to-positive or positive-to-negative confusions were much less frequent. These counts provide a descriptive view of the observed error patterns. The small differences between the two training conditions are interpreted cautiously, without attributing them to specific linguistic phenomena.

These patterns are consistent with the difficulty of separating semantically adjacent classes in short, noisy text. They do not establish that label smoothing specifically reduces sarcasm, slang, or any other particular linguistic phenomenon; rather, they provide a concrete description of the error structure underlying the small observed performance changes.

The clustering discussion is presented as a descriptive analysis of the representation geometry. For short-text classification, limited context, lexical overlap, and locally mixed class boundaries may make changes in cluster compactness less pronounced than changes in classification behavior; this is offered as an interpretation of the observed geometry, not as a tested causal explanation.

4. ATTENTION ANALYSIS

In this section, we perform a qualitative attention-based analysis between the models that are trained using label smoothing technique and the models that use hard target labels. The key aspect of the BERT architecture is the self-attention layer which allows each token from the input sequence to attend to all other tokens from the sequence depending on the context. This allows constructing representations with respect to the context where there are complicated dependencies between the positions from the sequence. In many cases, attention heads in different Transformer layers specialize into specific roles associated with particular linguistic relations [16].

Prior work has examined whether attention heads concentrate on tokens carrying salient semantic or linguistic information [24].

The analysis of attention was extended from individual examples to an aggregate examination of negation-related and sentiment-related tokens across the 970 FinancialPhraseBank test instances for the paired runs. Table 6 reports the mean attention values for each training condition. The paired LS-minus-Hard differences were -0.000031 for negation-related tokens and -0.000023 for sentiment-related tokens when averaged across the paired runs. These results show only minimal differences in the aggregate attention values of the considered token groups. The cross-entropy results provide information about the predicted probability distributions, while the aggregate attention analysis shows very similar attention values for the examined token groups under the two training conditions.

Table 6. Aggregate attention analysis for negation-related and sentiment-related tokens in BERT on FinancialPhraseBank. Values are means across the paired runs.

Token group	Hard labels	Label smoothing	LS–Hard
Negation-related	0.001345	0.001314	-0.000031
Sentiment-related	0.004623	0.004600	-0.000023

4.1. Discussion of the Observed Variability

The paired results indicate that the effect of label smoothing is not uniform across Transformer text classification settings. The observed differences are generally small and vary across datasets, architectures, and evaluation metrics. This variability highlights the context-dependent behavior of label smoothing across different experimental settings. In particular, the short and informal nature of social-media text may make the effect sensitive to the encoder architecture and to the distribution of lexical and contextual cues. The inclusion of TweetEval and FinancialPhraseBank broadens the empirical scope beyond OPT and provides evidence from both social-media and structured text classification settings.

5. CONCLUSION

This paper provides an empirical assessment of label smoothing across OPT, TweetEval, and FinancialPhraseBank using three Transformer models. Validation-only coefficient screening across FinancialPhraseBank and TweetEval selected $\alpha = 0.05$ on the basis of aggregate validation performance among the tested values. The paired test-set analysis shows heterogeneous behavior: the relationship between hard-label and label-smoothing training varies across dataset–architecture combinations and depends on the evaluation metric. Across the paired effects, the mean LS-minus-Hard change was -0.095 percentage points in accuracy and -0.171 percentage points in macro-F1. Thus, the present evidence supports an analytical, context-dependent interpretation of label smoothing rather than treating it as a uniformly beneficial training modification.

The attention and representation-geometry analyses provide complementary descriptive perspectives on the effects of label smoothing. Overall, the study contributes empirical evidence on label smoothing across diverse Transformer classification settings. Future research can examine broader architectures and datasets, adaptive smoothing strategies, and the interaction of label smoothing with attention patterns, representation geometry, and other regularization approaches.

Acknowledgment

The authors extend their appreciation to Deanship of Graduate Studies and Scientific Research at University of Bisha, Saudi Arabia for funding this research work through the Promising Program under Grant Number (UB-Promising-42-1445).

REFERENCES

1. Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
2. Patrice Y. Simard, Yann A. LeCun, John S. Denker, and Bernard Victorri. Transformation invariance in pattern recognition: Tangent distance and tangent propagation. In *Neural Networks: Tricks of the Trade*, pages 239–274. Springer, Berlin, Germany, 1998.
3. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2016.
4. Wei Li, Gautam Dasarathy, and Visar Berisha. Regularization via structural label smoothing. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1453–1463, 2020.
5. Yanmei Wang, Xin Wu, XiYao Liu, Fupeng Chu, Huan Liu, and Zhi Han. Label smoothing regularization-based no hyperparameter domain generalization. *Knowledge-Based Systems*, 309:112877, 2025.
6. Shijing Si, Yijie Gao, Haixia Sun, Yugui Zhang, and Hua Luo. Revisiting the role of label smoothing in enhanced text sentiment classification. *Electronics*, 15(10):1984, 2026.
7. Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4694–4703, 2019.
8. Zhen Luo, Yicheng Xi, and Xianling Mao. Smoothing with fake label. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 3303–3307, 2021.
9. Jingjing Wang, Haoran Xie, Fu Lee Wang, and Lap-Kei Lee. Improving text classification via a soft dynamical label strategy. *International Journal of Machine Learning and Cybernetics*, 14(7):2395–2405, 2023.
10. Han Ren, Yajie Zhao, Yong Zhang, and Wei Sun. Learning label smoothing for text classification. *PeerJ Computer Science*, 10:e2005, 2024.
11. Leixin Yang and Yu Xiang. Amplify: attention-based mixup for performance improvement and label smoothing in transformer. *PeerJ Computer Science*, 10:e2011, 2024.
12. Zhenze Liu, Shijing Si, and Jiawen Gu. Calibrating sentiment analysis: A unimodal-weighted label distribution learning approach. *IEEE Access*, 13:148816–148826, 2025.
13. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 448–456, 2015.
14. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.
15. Zhenyu Li and Zongfeng Zou. Punctuation and lexicon aid representation: A hybrid model for short text sentiment analysis on social media platform. *Journal of King Saud University - Computer and Information Sciences*, 36(3):102010, 2024.
16. Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. What does BERT look at? an analysis of BERT's attention. *arXiv preprint arXiv:1906.04341*, 2019.
17. Mohammed I Salih, Salim M Mohammed, Asaad Kh Ibrahim, Omar M Ahmed, and Lailan M Haji. Fine-tuning bert for automated news classification. *Engineering, Technology & Applied Science Research*, 15(3):22953–22959, 2025.
18. Xianzhi Ruan, Steven Wilson, and Rada Mihalcea. Finding optimists and pessimists on twitter. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 320–325, 2016.
19. Cornelia Caragea, Liviu P. Dinu, and Bogdan Dumitru. Exploring optimism and pessimism in twitter using deep learning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 652–658, 2018.
20. Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, 2020.
21. Pekka Malo, Ankur Sinha, Pekka Korhonen, Jyrki Wallenius, and Pyry Takala. Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4):782–796, 2014.
22. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
23. Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101*, 2017.
24. Abdullah Alshahrani, Mahboobeh Ghaffari, Kamran Amirizirtol, and Xiaodan Liu. Identifying optimism and pessimism in twitter messages using XLNet and deep consensus. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020.