Label Smoothing for Text Mining

Peiyang Liu[1,2,*], **Xiangyu Xi**[3,*], **Wei Ye**[1,†] and **Shikun Zhang**[1]
[1] National Engineering Research Center for Software Engineering, Peking University, Beijing, China,
[2] PX Securities, Shenzhen, China,
[3] Meituan, Beijing, China,
{liupeiyang, wye, zhangsk}@pku.edu.cn, xixiangyu@meituan.com

Abstract

Current text mining models are trained with 0-1 hard label that indicates whether an instance belongs to a class, ignoring rich information of the relevance degree. Soft label, which involved each label of more varying degrees than the hard label, is considered more suitable for describing instances. The process of generating soft labels from hard labels is defined as label smoothing (LS). Classical LS methods focus on universal data mining tasks so that they ignore the valuable text features in text mining tasks. This paper presents a novel keyword-based LS method to automatically generate soft labels from hard labels via exploiting the relevance between labels and text instances. Generated soft labels are then incorporated into existing models as auxiliary targets during the training stage, capable of improving models without adding any extra parameters. Results of extensive experiments on text classification and large-scale text retrieval datasets demonstrate that soft labels generated by our method contain rich knowledge of text features, improving the performance of corresponding models under both balanced and unbalanced settings.

1 Introduction

Instances in most text mining datasets are usually assigned by one label. Such label, called *hard label*, reflects the logical relationship between the label and the instance. In most cases, hard labels are utilized for training models that learn to discriminate between classes with training objectives to maximize the log probability of the correct class. However, there are various real-world text mining tasks where there are not just two possibilities whether an instance belongs to a specific class since instances may be involved with multiple labels of varying degrees. In such scenarios, soft label (Bahri and Jiang, 2021; Hong et al., 2022), which involves the explicit relative importance of each label, is a more reasonable description of an instance. For example, in the sentence S1 shown in the Table 1,[1] though both positive and negative expressions exist in the sentence and sentiment slightly inclines to be positive, annotators must give a solid positive as a hard label. The soft label (0.6,0.4) describes the instance more comprehensively than hard labels.

S1: It was **nice**. The cashier was **chipper**. Wish they would crank the A/C **a little more though**. Restroom was **a little dirty**.	
Hard Label	(1, 0)
Soft Label	(0.6, 0.4)

Table 1: An instance of sentiment classification with hard label and soft label. X denotes the label of being positive, while Y denotes negative in (X, Y). Positive and negative expressions are marked in red and blue, respectively.

Few works are investigating soft labels for text data in the text mining community, although the issue demonstrated in the above example is very common. Existing works all focus on analyzing the effectiveness of classical soft labels in universal data mining tasks (Müller et al., 2019; Wu et al., 2021; Nguyen et al., 2022). However, classical soft labels cannot leverage useful text features in text mining datasets. Meanwhile, in most text datasets, soft labels are not explicitly available. Therefore, automatically generating soft labels from hard labels based on text data characteristics is a fundamental problem worth exploring.

The key to generate informative soft labels for text instances is to exploit the semantic relevance between instances and labels accurately. We are inspired by the fact that a set of highly representative words can represent semantic information of labels and instances (e.g., S1 in Table 1) (Spärck Jones, 1972, 2004), we present a **K**ey**w**ord-based **L**abel **S**moothing method (**KWLS**), which primarily in-

[*] Co-first authors, contributed to this work equally
[†] Corresponding author

[1] A sentence from Yelp Review dataset.

Proceedings of the 29th International Conference on Computational Linguistics, pages 2210–2219
October 12–17, 2022.

volves four steps: (1) Keyword detection for labels, which computes correlations between words and labels; (2) Keyword detection for instances, which calculates the saliency of a word to represent a given instance; (3) Instance-label relevance detection, which determines the correlations between instances and labels; (4) Soft label generation, which generates soft labels for instances based on the instance-label relevance. The generated soft labels can be utilized as complementary targets during the training stage, containing knowledge of label correlation and introducing auxiliary supervision information to improve models.

To verify the effectiveness of our method, we conduct extensive experiments on the two most typical application scenarios in text mining: text classification and large-scale text retrieval. Experimental results demonstrate that models equipped with KWLS gain significant improvements over the original models, especially in the highly unbalanced large-scale text retrieval task (Liu et al., 2021c). To further analyze the ability of KWLS to deal with unbalanced problems, we construct various unbalanced datasets to simulate multifarious unbalanced problems in text mining. Experimental results on different unbalanced settings show that KWLS may bring extra supervised signals to facilitate model learning.

Our main contributions are listed as follows:

1. Previous studies focus on leveraging label smoothing on universal data mining tasks so that they ignore the valuable text features in text mining tasks. We propose a novel keyword-based LS method (KWLS) that automatically generates soft labels with rich knowledge of label correlation from hard labels in text data.

2. Generated soft labels can be incorporated as complementary targets, introducing auxiliary supervision information, capable of improving models without adding any extra parameters.

3.Experimental results show that the knowledge of label correlation characterized by KWLS is practical under balanced and unbalanced settings and more suitable for text mining tasks than classical label smoothing methods.

2 Related Work

Label smoothing, a form of output distribution regularization, prevents overfitting of a neural network by softening the ground-truth labels to penalize overconfident outputs (Li et al., 2020), has made tremendous achievements in many data mining fields. For example:

Müller et al. (2019) reveal that label smoothing improves model calibration and summarize several behaviors observed while training deep neural networks with label smoothing.

Chelombiev et al. (2019) propose an improved mutual information estimator based on binning and show the correlation between compression of softmax layer representations and generalization, which may explain why networks trained with label smoothing generalize so well.

The above works analyze the effectiveness of classical label smoothing methods in various fields. The two most classical label smoothing methods are follows:

Szegedy et al. (2016) first introduced label smoothing that improves accuracy by computing cross-entropy not with the "hard" targets from the dataset, but with a weighted mixture of these targets with the uniform distribution, and many state-of-the-art image classification models have incorporated label smoothing into training procedures ever since.

Zhang and Sabuncu (2020) regard self-distillation as a label smoothing method and propose a novel instance-specific label smoothing technique that promotes predictive diversity without the need for a separately trained teacher model.

However, these two methods mentioned above are proposed for universal data mining tasks so that they cannot leverage the valuable text features in text mining datasets. This paper proposes a KWLS method explicitly designed for text mining, which incorporates semantic relevance between labels and instances into soft labels. Experimental results demonstrate that our method is more suitable for text data.

It is worth noting that the primary purpose of LS is incorporating the possibility (or uncertainty) into the original hard label to facilitate model performances rather than generating the ground truth soft labels.

3 Task Description and Formulation

Given a training set $D = \{(x_i, \boldsymbol{l_i}) | 1 \leq i \leq N\}$ with N instances, x_i is a sequence of words $x_i = \{w_1, w_2, ..., w_n\}$ where w_1, w_2, and w_n are words in the sentence and n is the length of the sequence. The hard label vector of x_i is denoted by $\boldsymbol{l_i} = (l_i^1, l_i^2, ..., l_i^p)$, where $l_i^j \in \{0, 1\}$ denotes whether label l^j describes x_i, where p is the total number of labels. Our task is to generate the soft label $\boldsymbol{d_i}$ of x_i where $\boldsymbol{d_i} = (d_i^1, d_i^2, ..., d_i^p)$ is a distribution vector. The training set D is thus transformed into $\mathcal{E} = \{(x_i, \boldsymbol{d_i}) | 1 \leq i \leq N\}$ which is available for further exploration.

The proposed KWLS method is introduced in the following section.

4 Keyword-Based Label Smoothing

To represent labels and instances by a set of highly representative words, we propose to first determine keywords for instances and labels then calculate correlations between them. Our method mainly involves the following four components.

4.1 Keyword Detection for Labels

This section illustrates how to detect keywords for each label. We propose to use **Word-Label Relevance (WLR)** based on TF-IDF (Term Frequency–Inverse Document Frequency) (Spärck Jones, 1972, 2004) and BM25 (Robertson et al., 2009) to estimate the importance of a word to a label, which is decided by two factors: *Word-Label Weight* and *Word-Label Correlation*.

Word-Label Correlation (WLC) based on TF (Term Frequency) reflects the saliency of a word to represent a given label. In traditional TF, if a word frequently appears in instances of a given label, the word will own a larger WLC to the label. However, if a label contains 200 occurrences of "COLING", is it twice as relevant as a label containing 100 occurrences? We could argue that if "COLING" occurs a large enough number of times, say 100, the label is almost certainly relevant, and any further mentions do not increase the likelihood of relevance. According to the observation by Robertson et al. (2009), when the word frequency becomes larger, the WLC will grow slower. We define WLC as follows:

$$
\begin{aligned}
&WLC_{w_j,m} = \frac{f_{w_j,X_m} * k_1}{f_{w_j,X_m} + K}, w_j \in C, \\
&K = k_1 * (1 - b + b * \frac{S_m}{S_{avg}}) \\
&S_m = \sum_{x_i \in X_m} |x_i| \qquad (1) \\
&S_{avg} = \frac{1}{p} * \sum_{i}^{p} S_i \\
&X_m = \{x_i | x_i \in X, l_i^m = 1\}
\end{aligned}
$$

where f_{w_j,X_m} equals the number of times w_j appears in X_m, X_m is the set where $x_i \in X$ and $l_i^m = 1$, and C is the corpus. k_1 is a positive hyperparameter, which is used to scale and control the f_{w_j,X_m}. If k_1 is set to close to 0, then f_{w_j,X_m} will be ignored. If k_1 is set to a large value, then the $WLC_{w_j,m}$ is equal to f_{w_j,X_m}. $|x_i|$ is the number of words in x_i, and S_m is the number of words in X_m. S_{avg} is the average number of words in all labels. b is another hyperparameter to control S_m to normalize word frequency, and $0 \leq b \leq 1$. When b is set to 1, the f_{w_j,X_m} will be fully scaled based on S_m. When b is set to 0, S_m is not taken into account in normalization. $WLC_{w_j,m}$ is the Word-Label Correlation between w_j and the m-th label.

Word-Label Weight (WLW) based on IDF (Inverse Document Frequency) reflects the ability in which a word can be used to discriminate different labels. If a word occurs in every label, the word will have a low WLW. In traditional IDF, if a word occurs once in every label, the WLW of the word will be set to 0. However, words can easily occur in every label at least once, which will make most of the words' WLW be set to 0 unreasonably. So that we modified the traditional IDF, and our WLW can be calculated as follows:

$$
\begin{aligned}
&WLW_{w_j} = \log \frac{p}{sum(\mathbf{F_{w_j,L}})} + 1 \\
&\mathbf{F_{w_j,L}} = \{F_{w_j,l^1}, ..., F_{w_j,l^m}, ..., F_{w_j,l^p}\} \qquad (2) \\
&F_{w_j,l^m} = \begin{cases} 1, f_{w_j,X_m} \geq 1 \\ 0, f_{w_j,X_m} < 1 \end{cases}
\end{aligned}
$$

where $L = \{l^1, l^2, ..., l^p\}$ denote the finite set of labels and p is the number of possible labels. WLW_{w_j} is the weight of w_j to labels. $\mathbf{F_{w_j,L}}$ is a vector to represent whether w_j occurs in labels, where $F_{w_j,l^m} = 1$ means the word w_j occurs in the m-th label. Finally, WLR is computed as follows:

$$
WLR_{w_j,m} = WLW_{w_j} * WLC_{w_j,m} \qquad (3)
$$

where $WLR_{w_j,m}$ is the Word-Label Relevance between w_j and the m-th label.

4.2 Keyword Detection for Instances

This section illustrates how to detect keywords for each instance. We propose to use **Word-Instance Relevance (WIR)** to estimate the importance of a word to an instance, which is decided by two factors: *Word-Instance Weight* and *Word-Instance Correlation*.

Word-Instance Correlation (WIC) based on TF reflects the saliency of a word to represent a given instance. Similar to WLC, if a word frequently appears in a given instance, the word will own a larger WIC to the instance, and the correlation between word frequency and WIC is also nonlinear. We define WIC as follows:

$$WIC_{w_j,i} = \frac{f_{w_j,x_i} * k_2}{f_{w_j,x_i} + k_2}, w_j \in x_i \tag{4}$$

where f_{w_j,x_i} is the number of times w_j appears in x_i. k_2 is a positive hyperparameter, which is used to scale and control the word frequency of x_i. $WIC_{w_j,i}$ is the Word-Instance Correlation between w_j and the x_i.

Word-Instance Weight (WIW) based on IDF reflects the ability in which a word can be used to discriminate different instances. Similar to WLW, WIW can be calculated as follows:

$$\begin{aligned} WIW_{w_j} &= \log \frac{N}{sum(\mathbf{F_{w_j,X}})} + 1 \\ \mathbf{F_{w_j,X}} &= \{F_{w_j,x_1}, ..., F_{w_j,x_i}, ..., F_{w_j,x_N}\} \\ F_{w_j,x_i} &= \begin{cases} 1, f_{w_j,x_i} \geq 1 \\ \\ 0, f_{w_j,x_i} < 1 \end{cases} \end{aligned} \tag{5}$$

where $X = \{x_1, x_2, ..., x_N\}$ denote the set of instances and N is the size of X. WIW_{w_j} is the weight of w_j to instances. $\mathbf{F_{w_j,X}}$ is a vector to represent whether w_j occurs in instances, where $F_{w_j,x_i} = 1$ means the word w_j occurs in the i-th instance. Finally, WIR is computed as follows:

$$WIR_{w_j,i} = WIW_{w_j} * WIC_{w_j,i} \tag{6}$$

where $WIR_{w_j,i}$ is the Word-Instance Relevance between w_j and the i-th instance.

4.3 Instance-Label Relevance Detection

We propose **Instance-Label Relevance (ILR)** to represent correlations between instances and labels, which is calculated as follows:

$$\begin{aligned} ILR_{i,m} &= \frac{1}{X_{avg}} \sum_{w_j \in x_i} WLR_{w_j,m} * WIR_{w_j,i} \\ X_{avg} &= \frac{1}{N} \sum_{x_i \in X} |x_i| \end{aligned} \tag{7}$$

where X_{avg} is the average length of all instances. $ILR_{i,m}$ is the relevance between the i-th instance and the m-th label.

4.4 Soft Label Generation

After the ILR is computed, we generate the soft label by the softmax function (Elfadel and Jr., 1993):

$$\boldsymbol{d_i} = Softmax(\boldsymbol{ILR_i}) \tag{8}$$

4.5 Incorporating Soft Label

After soft labels are generated for each instance, we incorporate soft labels as auxiliary fitting targets, and the loss function can be defined as:

$$Loss = -\frac{\sum_{i=1}^{N}\sum_{j=1}^{p}[l_i^j \log(o_i^j) - \frac{\alpha}{p}(d_i^j - o_i^j)^2]}{2N(1+\alpha)^2} \tag{9}$$

where o_i^j is j-th dimension of i-th instance's output probability predicted by the model. α denotes the loss weight of soft label during the training stage. We use Mean Squared Error (MSE) as the loss function for soft labels, where α denotes the loss weight of soft label during the training stage. When α is set to 0, the model degenerates into the standard classifier.

5 Experiments and Results

5.1 Datasets

For text classification, the following four real-world datasets are used in the experiment.

AG News consists of news articles from the AG's corpus of news articles on the web about the four largest classes (Technology, Sports, Business, and World). The dataset contains 30,000 training and 1,900 testing examples for each class (Gulli, 2005; Del Corso et al., 2005).

DBpedia is a project aiming to extract structured content from the information created in the Wikipedia project (Lehmann et al., 2015). The DBpedia ontology dataset contains 560,000 training samples and 70,000 testing samples for each of 14 non-overlapping classes from DBpedia (Lehmann et al., 2015).

IMDb is a binary sentiment analysis dataset consisting of 50,000 reviews from the Internet Movie Database (IMDb) labeled positive or negative. The dataset contains an even number of positive and negative reviews (Maas et al., 2011).

Yelp is a subset of businesses, reviews, and user data for use in personal, educational, and academic purposes (Zhang et al., 2015).

DataSet	Model	Accuracy	Model	Accuracy	Model	Accuracy
AG News	BERT	94.08	TextCNN	88.88	LSTM	87.02
	+ LS-Classic	94.17	+ LS-Classic	89.37	+ LS-Classic	87.93
	+ LS-Distill	94.37	+ LS-Distill	89.56	+ LS-Distill	88.17
	+ LS-TFIDF	94.42	+ LS-TFIDF	90.11	+ LS-TFIDF	90.13
	+ KWLS	**95.13**	+ KWLS	**91.07**	+ KWLS	**90.85**
DBpedia	BERT	99.30	TextCNN	98.48	LSTM	98.71
	+ LS-Classic	99.35	+ LS-Classic	98.52	+ LS-Classic	98.77
	+ LS-Distill	99.40	+ LS-Distill	98.63	+ LS-Distill	98.80
	+ LS-TFIDF	99.41	+ LS-TFIDF	98.86	+ LS-TFIDF	98.84
	+ KWLS	**99.65**	+ KWLS	**99.12**	+ KWLS	**99.13**
IMDb	BERT	93.21	TextCNN	91.62	LSTM	91.03
	+ LS-Classic	93.29	+ LS-Classic	91.70	+ LS-Classic	91.26
	+ LS-Distill	93.47	+ LS-Distill	91.76	+ LS-Distill	91.36
	+ LS-TFIDF	93.65	+ LS-TFIDF	92.10	+ LS-TFIDF	91.73
	+ KWLS	**94.26**	+ KWLS	**92.43**	+ KWLS	**92.55**
Yelp	BERT	69.54	TextCNN	64.38	LSTM	64.03
	+ LS-Classic	69.68	+ LS-Classic	64.53	+ LS-Classic	64.12
	+ LS-Distill	69.70	+ LS-Distill	64.76	+ LS-Distill	64.27
	+ LS-TFIDF	69.81	+ LS-TFIDF	65.02	+ LS-TFIDF	64.92
	+ KWLS	**70.08**	+ KWLS	**65.89**	+ KWLS	**65.68**

Table 2: Experimental results of models with different targets in text classification datasets.

Metric	BERT					TextCNN					LSTM				
	O	C	D	T	K	O	C	D	T	K	O	C	D	T	K
R@1	35.27	37.46	37.62	41.03	**43.87**	33.12	33.18	34.97	36.42	**37.25**	32.43	33.12	34.13	35.41	**36.96**
R@10	48.48	52.62	55.89	61.29	**62.32**	41.99	43.50	43.87	47.81	**52.62**	40.27	41.06	41.28	45.89	**50.63**
R@50	68.21	71.11	72.54	78.45	**79.72**	62.56	62.74	67.46	71.61	**76.56**	60.42	63.09	64.44	69.35	**74.15**
R@100	78.85	79.52	80.50	83.89	**85.28**	71.41	72.01	73.71	76.50	**79.13**	70.26	72.05	72.91	75.32	**77.56**

Table 3: Recall@k on the large-scale retrieval dataset. Numbers are in percentage (%). O, C, D, T, and K represent the original model, LS-Classic, LS-Distill, LS-TFIDF, and KWLS separately.

For large-scale retrieval, we consider the Retrieval Question-Answering (ReQA) benchmark proposed by Ahmad et al. (2019). The dataset we selected is SQuAD, which is a reading comprehension dataset, consisting of questions posed by crowd workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable (Rajpurkar et al., 2016). Each entry of this dataset is a tuple (q, a, p), where q is the question, a is the answer span, and p is the evidence passage containing a. Following Ahmad et al. (2019), we split a passage into sentences $p = s_1 s_2 ... s_n$. For a query q, we need to retrieve the correct sentence from a candidate set consisting of sentences of all passages. A query-sentence pair (q, s) is labeled as 1 if s is the sentence containing the corresponding answer span and labeled as 0 otherwise. This problem is more challenging than retrieving the evidence passage only since the larger number of candidates to be retrieved.

For each dataset, we randomly split the train sets into train/dev sets at the ratio of 3:1. The test sets remain unchanged.[2] We apply four-fold cross-validation to do significance tests.

5.2 Baselines

Considering current text mining models can be split into three types, RNN-, CNN-, and Transformer-based models. We incorporate our method with the following three models, which are existing representative models widely used.

LSTM is the most widely used RNN-based deep neural network in text mining tasks (Liu et al., 2016).

TextCNN is the most famous CNN-based text mining baseline proposed by Kim (2014).

BERT is the most representative Transformer-based model in the text mining community (Devlin et al., 2018).

We also compare our methods with three alternative label smoothing methods.

LS-TFIDF is a variant of our method in which the soft label is generated based on TF-IDF.

LS-Classic is the most classical label smoothing method proposed by Szegedy et al. (2016), widely

[2] Note that all LS methods are only used in the training set.

DataSet	Model	Accuracy	Model	Accuracy	Model	Accuracy
AG News *	BERT	76.34	TextCNN	72.62	LSTM	72.06
	+ LS-Classic	76.82	+ LS-Classic	72.76	+ LS-Classic	72.38
	+ LS-Distill	76.92	+ LS-Distill	72.84	+ LS-Distill	72.81
	+ LS-TFIDF	78.12	+ LS-TFIDF	73.61	+ LS-TFIDF	73.87
	+ KWLS	**79.94**	+ KWLS	**74.45**	+ KWLS	**74.04**
DBpedia *	BERT	97.27	TextCNN	95.30	LSTM	95.28
	+ LS-Classic	97.60	+ LS-Classic	95.85	+ LS-Classic	95.44
	+ LS-Distill	98.14	+ LS-Distill	96.04	+ LS-Distill	95.53
	+ LS-TFIDF	98.25	+ LS-TFIDF	97.09	+ LS-TFIDF	96.85
	+ KWLS	**98.41**	+ KWLS	**97.21**	+ KWLS	**97.35**
IMDb *	BERT	50.02	TextCNN	48.73	LSTM	48.31
	+ LS-Classic	54.10	+ LS-Classic	50.59	+ LS-Classic	50.89
	+ LS-Distill	54.60	+ LS-Distill	52.86	+ LS-Distill	52.92
	+ LS-TFIDF	56.31	+ LS-TFIDF	55.49	+ LS-TFIDF	55.14
	+ KWLS	**60.62**	+ KWLS	**57.93**	+ KWLS	**57.81**
Yelp *	BERT	51.69	TextCNN	49.04	LSTM	48.19
	+ LS-Classic	52.40	+ LS-Classic	52.18	+ LS-Classic	51.36
	+ LS-Distill	54.52	+ LS-Distill	53.07	+ LS-Distill	52.48
	+ LS-TFIDF	56.54	+ LS-TFIDF	54.10	+ LS-TFIDF	53.14
	+ KWLS	**58.81**	+ KWLS	**56.91**	+ KWLS	**56.43**

Table 4: Experimental results of models with different targets in unbalanced text classification datasets. "*" denotes the under-sampled thus unbalanced datasets.

used in computer vision and natural language processing.

LS-Distill is another widely used label smoothing method in which the soft label set as predicting scores of the original model (Zhang and Sabuncu, 2020). This method is similar to self-distillation process in born-again networks (Furlanello et al., 2018) and widely used in knowledge distillation (Hinton et al., 2015; Liu et al., 2021b).

5.3 Effectiveness in Text Classification

To explore the effectiveness of our LS method, we conduct experiments to explore whether our method can improve the CNN-based model TextCNN, RNN-based model LSTM, and Transformer-based model BERT. For all of the above text classification tasks, we report the classification accuracy over the test set. Table 2 demonstrates the results,[3] from which we have following five observations:

1. All CNN-, RNN-, and Transformer-based models incorporating soft labels generated by all LS methods as auxiliary targets gain improvements over the original models in all tasks, which verifies the intuition of label smoothing.

2. LS-Classic only gains minor improvements over original models. The reason is that LS-Classic generates soft labels from the hard label in a brute way, which ignores rich knowledge from the instances.

DataSet	Business	Technology	Sports	World
c=0	30000	30000	30000	30000
c=1	30000	30000	30000	300
c=2	30000	30000	300	300
c=3	30000	300	300	300
m=300	30000	300	300	300
m=200	30000	200	200	200
m=100	30000	100	100	100
m=50	30000	50	50	50
m=10	30000	10	10	10
m=1	30000	1	1	1
Test Set	1900	1900	1900	1900

Table 5: The experimental setting for investigation on effects of different degrees of unbalance.

3. The improvement of LS-TFIDF over original models shows that TF-IDF weights serve as beneficial prior knowledge to characterize soft labels.

4. LS-Distill also achieves notable enhancements. This observation is consistent with other knowledge distillation works (Hinton et al., 2015). The self-distillation process brings valuable "dark" knowledge (Furlanello et al., 2018) via the generated soft predicting scores even without utilizing term weights information.

5. Our KWLS has clear superiority over LS-Distill and LS-TFIDF among all datasets. Rather than predicting relevance score directly as LS-Distill, KWLS incorporates improved TF-IDF information into supervised signals. Therefore, the final generated soft label integrates explicit prior term weight knowledge, and some "dark" knowledge is produced during training. We believe that is the main reason behind this superiority.

[3] The experiment results in this paper are statistically significant with $p < 0.05$.

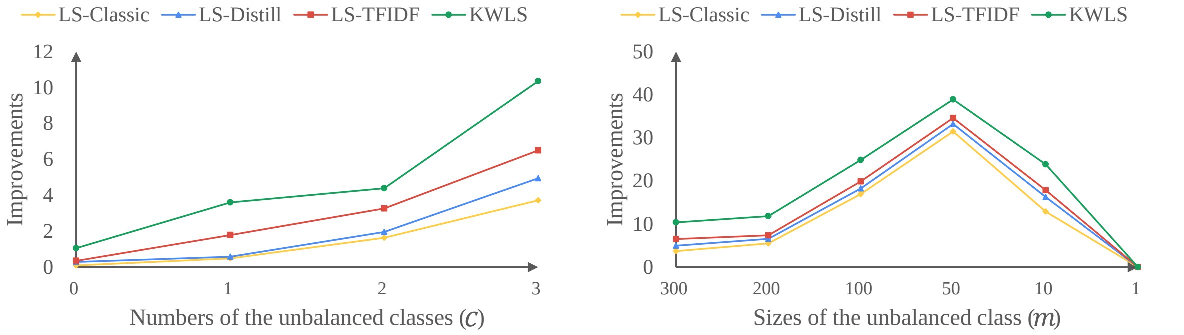


Figure 1: Experimental results of the investigation on the effect of different degrees of unbalance. The x-axis denotes the different settings of unbalance degrees. The y-axis denotes improvements (Accuracy) LS methods gain over the original model.

5.4 Effectiveness in Large-Scale Retrieval

As described in the above section, all LS methods improve the original model on common text classification tasks. To further explore the ability of LS to deal with text datasets, we evaluate LS methods on the more challenging large-scale retrieval task. Since the goal of retrieval is to capture the positives in the top-k results (Liu et al., 2021a), we select **Recall@k** as the evaluation metric.

The experimental results are shown in Table 3, from which we can observe that all LS methods perform exceptionally well for the large-scale retrieval task (especially the LS-TFIDF and KWLS). We can quickly guess the following two reasons for this phenomenon:

1. Note that the data collection process and human annotations of SQuAD are biased towards question-answer pairs with overlapping tokens (Rajpurkar et al., 2016). We can naturally expect that the generated soft label could better characterize query-document relevance degree in the SQuAD dataset due to the capability of term weighting LS methods to identify overlapped highly-representative tokens.

2. Another straightforward guess is that the large-scale retrieval task is highly unbalanced (Retrieve one result in large-scale candidates). For a class without adequate training instances, soft labels generated by LS methods will provide auxiliary supervision information from other categories, which may help the model identify the specific class better.

5.5 Effectiveness on unbalanced Datasets

As shown in the above section, LS methods perform exceptionally well in highly unbalanced large-scale retrieval tasks. To evaluate the effectiveness of LS methods on unbalanced datasets, for each task of text classification mentioned above, we manually remove some samples to simulate the unbalanced scenario. More specially, we undersample each class in the training set with the ratio of 1:100 and keep other classes unchanged. Then we can get n unbalanced training sets where n is the number of classes in the task. The test set remains unchanged (balanced). The average scores across the test set of models trained on n unbalanced training sets are used to evaluate each task. From the evaluation results in Table 4, we can see:

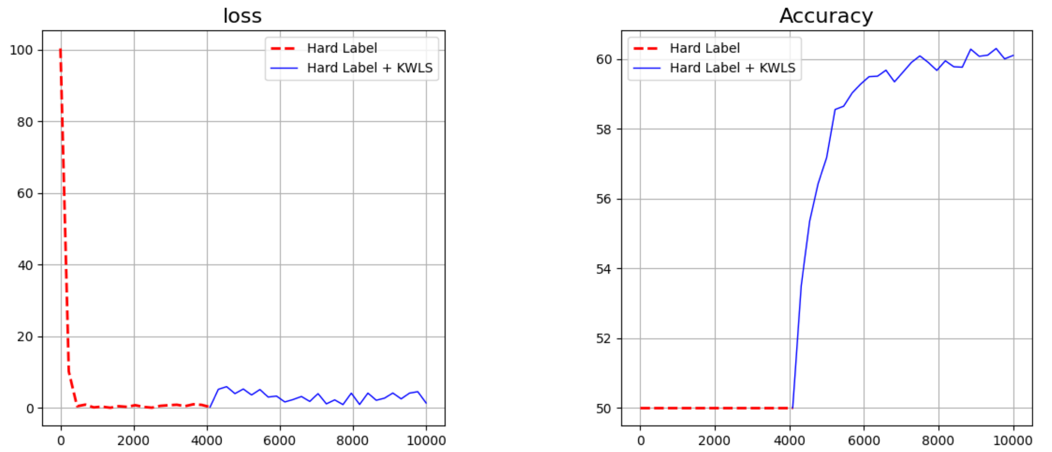
1. The inadequacy and unbalance of training data will significantly hinder the performance, especially for the IMDb dataset with binary labels.

2. Models with LS methods gain significant improvements over the original model on unbalanced datasets. For example, for the unbalanced IMDb dataset, the BERT with KWLS achieves **10.60** improvements in terms of accuracy.

Compared with balanced datasets, LS methods achieve more performance enhancement on unbalanced datasets, which verifies our assumption. For a data-lacking category, soft labels may provide auxiliary supervision information from other categories.

5.6 Effects of Different Degrees of Unbalance

To further explore the ability of LS methods to deal with various unbalance scenarios, we evaluate BERT equipped with LS methods on news classification datasets with multiple unbalanced settings. AG News, composed of 4 classes (Technology, Sports, Business, and World), is selected as the dataset for the experiment. As shown in Table 5, we set the training set unbalanced in two ways as follows:



(a) Loss during the training step

(b) Accuracy in validate set during the training step

Figure 2: Loss and Accuracy during the training step in unbalanced IMDb* dataset.

1. To explore the effect of the number of unbalanced classes, we keep the instances of Business news unchanged and under-sample other c classes into the ratio of 1:100, where c is set to 0, 1, 2, and 3 respectively.

2. To explore the effect of the sizes of unbalanced classes, we keep the instances of Business news unchanged and under-sample sizes of other classes in training set to m, which is set to 300, 200, 100, 50, 10, and 1 respectively.

From the evaluation results in Figure 1, we can see that with the increment of c, the models with LS methods gain more improvements over the original models. Since bigger c means a more unbalanced training set, it is not strange for the improvements increment. The trends of relative improvement reveal that soft labels generated by LS methods play a more critical role in a more unbalanced situation. The reason is that the knowledge of label correlation in soft labels helps to discriminate classes with fewer instances.

Based on the most unbalanced setting ($c = 3$), we decrease m to further intensify the degree of unbalance. We can see that the relative improvement in performance brought by LS gradually increases as m becomes smaller until m is less than 50. The reason is that although LS methods can provide auxiliary supervision information from other classes to help models identify the data-lacking class, the supervised signal from the original class is still important, which may degrade dramatically in an extreme case with very few instances.

5.7 Effectiveness of KWLS

To further explore why KWLS can improve models' performances in unbalanced scenarios, we record losses of the hard label and our KWLS during the training step in the unbalanced IMDb* dataset, which is a binary sentiment classification task (positive or negative). We under-sample the positive instances in the training set with the ratio of 1:100 and keep negative instances unchanged. As shown in Figure 2, we first train our model only with the hard label. The loss decreases dramatically since there are few instances in the data-lacking category, and models will easy to fitting supervisory signals of instances in a mini-batch and predict all instances as positive. In the 4000-th step, we incorporate KWLS into the model training. We can see that models with KWLS will obtain extra supervisory signals from other categories, which will help models identify data-lacking categories.

6 Conclusion

We have presented our **K**ey**w**ord-based **L**abel **S**moothing method (**KWLS**) for text mining tasks and demonstrated it's usage and effect on model training. Unlike previous works that focus on universal data mining tasks, our method is explicitly designed for text mining, which incorporates semantic relevance between labels and instances into soft labels. Like other widely-used tricks of text mining, the technical design of KWLS is simple yet effective, making it extremely easy to be applied in different text mining tasks, as shown in experiments. Despite its simplicity, using soft labels generated by KWLS as an auxiliary training target shows significant superiority in improving model performance, whether the data is balanced or unbalanced. Our codes are released on the Github.[4]

Ackowlegement

Thanks for the support of Xi Wang, Yunlai Hao, and Huihui Bai. We hope they will have a healthy, safe, and happy life. This research was supported by the National Key Research and Development Program of China (No. 2019YFB1405802).

References

Amin Ahmad, Noah Constant, Yinfei Yang, and Daniel Cer. 2019. Reqa: An evaluation for end-to-end answer retrieval models. In *MRQA@EMNLP*, pages 137–146.

Dara Bahri and Heinrich Jiang. 2021. Locally adaptive label smoothing improves predictive churn. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 532–542. PMLR.

Ivan Chelombiev, Conor J. Houghton, and Cian O'Donnell. 2019. Adaptive estimators show information compression in deep neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Gianna M Del Corso, Antonio Gulli, and Francesco Romani. 2005. Ranking a stream of news. In *WWW*, pages 97–106. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ibrahim M. Elfadel and John L. Wyatt Jr. 1993. The softmax nonlinearity: Derivation using statistical mechanics and useful properties as a multiterminal analog circuit element. In *Advances in Neural Information Processing Systems 6, [7th NIPS Conference, Denver, Colorado, USA, 1993]*, pages 882–887. Morgan Kaufmann.

Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-again neural networks. In *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 1602–1611. PMLR.

Antonio Gulli. 2005. The anatomy of a news search engine. In *Special interest tracks and posters of the 14th international conference on World Wide Web*, pages 880–881. ACM.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *stat*, 1050:9.

Seungbum Hong, Jihun Yoon, Min-Kook Choi, and Junmo Kim. 2022. Self-supervised knowledge transfer via loosely supervised auxiliary tasks. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 2947–2956. IEEE.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1746–1751. ACL.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.

Weizhi Li, Gautam Dasarathy, and Visar Berisha. 2020. Regularization via structural label smoothing. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 1453–1463. PMLR.

Peiyang Liu, Sen Wang, Xi Wang, Wei Ye, and Shikun Zhang. 2021a. QuadrupletBERT: An efficient model for embedding-based large-scale retrieval. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3734–3739, Online. Association for Computational Linguistics.

Peiyang Liu, Xi Wang, Lin Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021b. Distilling knowledge from BERT into simple fully connected neural networks for efficient vertical retrieval. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*, pages 3965–3975. ACM.

Peiyang Liu, Xi Wang, Sen Wang, Wei Ye, Xiangyu Xi, and Shikun Zhang. 2021c. Improving embedding-based large-scale retrieval via label enhancement. In *Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 16-20 November, 2021*, pages 133–142. Association for Computational Linguistics.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2873–2879. IJCAI/AAAI Press.

[4] https://github.com/PeiYangLiu/KWLS

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 *ACL*, pages 142–150. Association for Computational Linguistics.

Rafael Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4696–4705.

Chuong H. Nguyen, Thuy C. Nguyen, Tuan N. Tang, and Nam L. H. Phan. 2022. Improving object detection by label assignment distillation. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, Waikoloa, HI, USA, January 3-8, 2022*, pages 1322–1331. IEEE.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100, 000+ questions for machine comprehension of text. In *EMNLP*, pages 2383–2392.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Karen Spärck Jones. 1972. A statistical interpretation of term specificity and its application in retrieval. *Journal of documentation*, 28(1):11–21.

Karen Spärck Jones. 2004. Idf term weighting and ir research lessons. *Journal of documentation*, 60(5):521–523.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society.

Yichen Wu, Jun Shu, Qi Xie, Qian Zhao, and Deyu Meng. 2021. Learning to purify noisy labels via meta soft label corrector. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 10388–10396. AAAI Press.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 649–657.

Zhilu Zhang and Mert R. Sabuncu. 2020. Self-distillation as instance-specific label smoothing. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.