MaskLoss: A Regularizing Loss by Masking Similar Labels

Sunan Liu[1*] **Changsong Dai**[2*] **Ranran Zhen**[2*] **Qingliang Meng**[3] **Tian Li**[2]
Shumei AI Research Institute, Beijing, China
[1]liu_sn@yeah.net
[2]{daichangsong, zhenranran, litian}@ishumei.com
[3]mengqingliang9485@163.com

Abstract

All of the researches on neural networks have the problem of the wrong prediction, and most of them are similar labels. However, most parts of the current studies focus on how to make model structure deeper and more complex to obtain richer textual information, but few of the methods they raised are purely based on loss of neural networks. In this work, we propose a novel and effective method, namely MaskLoss, which employs a mask mechanism on losses with similar labels without any other training cost. To explore this method more comprehensively, we study the MaskLoss in two aspects, label-based and element-based, which are manual and automatic respectively. We systematically do lots of experiments on those different sub-methods on four public datasets and results show that both of them are very effective ways that can lead to significant performance improvement over their counterparts. In addition, we fully analyze the important factor, the rate of random, and find that it is a key role that affects the effectiveness of MaskLoss. Finally, case analysis proves that MaskLoss is effective in solving the issue of similar label misprediction[1].

1 Introduction

In neural networks (NN), it is very common for labels to be predicted incorrectly, which exists in almost all tasks in natural language processing, such as sentiment analysis, syntactic analysis, question answering, to name but a few. And these mispredicted labels have a high probability of being similar labels to the correct labels. It is worth noting that addressing this issue is the most needed and studies recently.

There are several existing works to enhance the discriminative ability of the model by changing the model structure or adding a new model. Some studies (Zhang et al., 2018; Yang et al., 2018; Tsai and Lee, 2020; Zhang et al., 2021a) aim to explore label structure and label semantics to obtain its correlations. Hu et al. (2018) strengthen the distinction between similar labels by adding an attention neural network for each label. For model structure, almost all main methods use advanced natural networks to capture the context representation to classify the label (like Graph Convolution Network (GCN) (Ma et al., 2021)) to find a relationship between labels. Although they can solve the problem of label relevance to a certain extent, their methods are highly complex and have a deeper model structure. Meanwhile, there is a problem with the previous methods. For example in AAPD dataset (Yang et al., 2018), `cs.it` is the gold label, all labels (including `cs.ai` which is similar to `cs.it`) participate in loss calculation when training. It is unfair, because `cs.ai` and `cs.it` have common features.

To figure out this problem, we propose an easy and effective method without any other training cost, label-based MaskLoss (MaskLoss$_{\texttt{label}}$) approach that employs a mask mechanism to make relevant labels do not participate in loss calculation. All labels are divided into groups in a manual way (e.g., the surface meaning of label), and all labels in the same group are similar to each other. But this grouping way is a coarse-grained method and has the limitations of manual existence, we need to consider an automatic, smart, and fine-grained approach. Furthermore, to handle this issue, we employ an element-based MaskLoss (MaskLoss$_{\texttt{elem}}$) approach which dynamically divides groups based on the world in each sentence.

We conduct lots of experiments based on traditional methods, MaskLoss$_{\texttt{label}}$ and MaskLoss$_{\texttt{elem}}$, respectively. We find that both of our methods have achieved promising results in many datasets, and MaskLoss$_{\texttt{elem}}$ is even

[*]Equal contribution.
[1]This work is still in progress and has yet to be added in terms of experiments, writing, etc.

better than MaskLoss$_{\texttt{label}}$ totally. In addition, we analyze the important factor, the rate of random which is very effective for our methods. Finally, during the case analysis, our method can greatly alleviate the problem of misprediction of similar labels.

2 Related Work

Similar Label Problem There are some studies (Hu et al., 2018; You et al., 2019; Xiao et al., 2019; Du et al., 2019) use attention mechanism to get representation for each label to alleviate the problem of misprediction of similar labels. Yang et al. (2018) use seq2seq approach to make predict similar labels more accurate. Xu et al. (2020) build a label graph by TF-IDF and extract distinguishable information through graph distillation operator. Ma et al. (2021) get representations for each label to make differences between similar labels, and then used dual GCN co-occurrence graph and rebuild it by relearning. Zhang et al. (2021b) enhance label correlation learning by pairwise and conditional label co-occurrence prediction.

Dropout The dropout strategy proposed by Srivastava et al. (2014) is very widely used in neural networks nowadays. In current use, almost all of this operation is used in the forward or back propagation process. However, our method MaskLoss drops out the losses of some similar labels, to prevent them from participating in the back propagation.

3 Method

In this section, we carry out our work in standard classification (single-label and multi-label task). We will explain baseline, and our proposed methods MaskLoss$_{\texttt{label}}$ and MaskLoss$_{\texttt{elem}}$ in detail. Appendix A.1 shows an overview of our methods.

Notation. We use $\texttt{C}$ for the number of categories and $\texttt{N}$ for the number of training examples. We denote the training data by $\mathcal{D} = \{(x^1, y^1), ..., (x^N, y^N)\}$, where $x^i = [x_1^i, ..., x_m^i]$ is the i^{th} sample which length is $\texttt{m}$ and $y^i = [y_1^i, ..., y_c^i] \in y \subseteq \{0, 1\}^{\texttt{C}}$ is the gold label. For a given example $\texttt{i}$ and category $\texttt{c}$, $y_c^i = 1$ (resp. 0) means the category is true (resp. false).

3.1 Basic Model

We exploit BERT which is a stack of transformer encoder pre-trained on the objective of the masked language model (Devlin et al., 2018), to obtain the semantic information. The formula for the prediction is as follows:

$$\begin{aligned} \boldsymbol{h}_{cls}^i &= \text{BERT}(\boldsymbol{x}^i) \\ p_1^i, ..., p_C^i &= \boldsymbol{\sigma}(\text{W}(\boldsymbol{h}_{cls}^i)) \end{aligned} \tag{1}$$

where $\boldsymbol{h}_{cls}^i \in \mathbb{R}^{1\times\texttt{S}}$ is the hidden of [CLS] token in last layer of BERT, $\texttt{S}$ is the hidden size of BERT, a metric $\text{W} \in \mathbb{R}^{\texttt{S}\times\texttt{C}}$, σ is Sigmoid activation function, and p_j^i is the probability of j^{th} label of $\boldsymbol{x}^i$. *Even with single-label classification, we still use Sigmoid instead of Softmax because Sigmoid can control each label, and Softmax can not.* We use binary cross-entropy (BCE) to be the loss function for the model training. And for each sentence, it can be described as:

$$\begin{aligned} \mathcal{L} &= \frac{1}{\text{C}} \sum_{k\in\texttt{C}} \text{BCE}(y_k, p_k) \\ \text{BCE}(y, p) &= -(y \cdot \log(p) + (1-y) \cdot \log(1-p)) \end{aligned} \tag{2}$$

3.2 Label-based MaskLoss

We divide labels manually into some groups (G) (detail in Appendix A.2). Labels in the same group all have some similar features, so when y_i is the gold label, other labels in this group should not be suppressed by the back propagation of loss. However, because there are differences in labels within the same group, it is necessary to participate in the normal loss calculation to distinguish between each other. Therefore, due to empirical knowledge, we balance the two choices through a sampling method, which is also to prevent labels in a group from underfitting or overfitting. The integrated loss is following:

$$\begin{aligned} \mathcal{L}_m = \frac{1}{\text{C}}(\lambda \sum_{k\in\texttt{C}} \text{BCE}(y_k, p_k) \\ +(1-\lambda) \sum_{k\in\texttt{C}\&k\notin G(k)} \text{BCE}(y_k, p_k)) \end{aligned} \tag{3}$$

where $\lambda \sim Bernoulli(num, \alpha)$ distribution and num is the total steps of model training, α is the sampling rate. The function $G(k)$ returns all labels that are not gold labels in the group where y_k exists.

3.3 Element-based MaskLoss

The above method on obtaining groups is based on human knowledge and non-automatic, which is limited, so we propose a more fine-grained word-level smart way namely MaskLoss$_{\texttt{elem}}$ approach.

Dataset	Train	Dev	Test	#Label	Type
SST-5	8544	1101	2210	5	S
TREC	4978	474	500	47	S
WOS	31757	3460	11768	134	S
AAPD	53840	1000	1000	54	M

Table 1: Dataset name, size, label and type of datasets we used to be our benchmark. TREC used is its 50 classification version. WOS used its WOS46985 version. #Label, S, and M denote the number of label sets, single and multi-label.

It consists of three main steps: 1) Obtaining a new sentence by deleting one word in an incorrectly predicted sentence at a time, which is defined as $\mathcal{D}$. 2) The trained basic model (Section 3.1) decodes $\mathcal{D}$. If the difference between the predicted label score and the pervious label score of the sentence before the word is removed is greater than a certain threshold (details of threshold in Appendix A.6 and A.4), we define these labels are the dependent labels of the word. 3) We get the group for each original sentence by the dependent labels of its words. More details of all processes in Appendix A.3.

4 Experimental Setup

In this section, we evaluate the proposed methods on four common datasets (three datasets for single-label, and one dataset for multi-label) by comparing with state-of-the-art models in terms of widely used metrics. For single label task, we choose several symbolic datasets: SST-5 (Socher et al., 2013) with small scale labels and TREC-50 (Li and Roth, 2002; Hovy et al., 2001) with medium scale labels, and WOS46985 (Kowsari et al., 2017) with large scale labels. For multi-label task, we select widely used dataset: AAPD (Yang et al., 2018) with 54 labels. We summarize the statistics of the datasets used in our study in Table 1.

5 Experiment Results

We report the accuracy scores for single-label datasets in SST-5, TREC-50, and WOS46985 and the micro precision, recall, and F1 scores for the multi-label dataset in AAPD, comparing with a series of strong baselines in Table 2 and Table 3 respectively.

Looking at those tables as a whole, MaskLoss$_{\texttt{elem}}$ method obtains a promising result and achieves the best performance. The MaskLoss$_{\texttt{label}}$ method which group collection

Model	Accuracy
SST-5	
BERT-Base♣	53.2
RoBERTa-Base♠	53.5
BERT-Base♢	53.7
XLNet-Base♠	53.8
SelfExplain-RoBERTa-Base♠	54.3
MaskLoss$_{\texttt{label}}$	54.8
MaskLoss$_{\texttt{elem}}$	**54.9**
TREC-50	
XLNet-Base♠	82.8
SelfExplain-XLNet♠	83.0
RoBERTa-Base♠	89.0
SelfExplain-RoBERTa-Base♠	89.4
BERT-Base♢	90.8
MaskLoss$_{\texttt{label}}$	91.4
MaskLoss$_{\texttt{elem}}$	**92.2**
WOS46985	
BERT-Base♢	82.4
MaskLoss$_{\texttt{label}}$	83.0
MaskLoss$_{\texttt{elem}}$	**83.1**

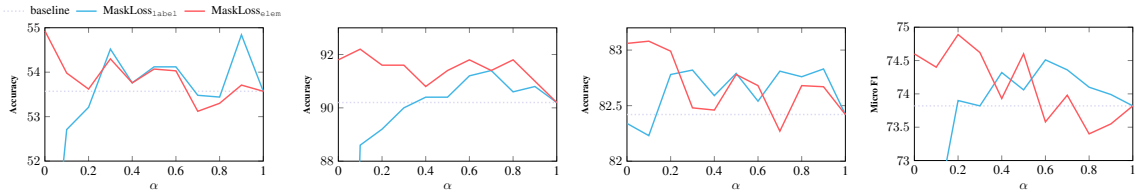
Table 2: A performance demonstration of single classification task in accuracy measure on SST-5, TREC-50, and WOS46985 test datasets. Best values are bolded. ♣: results from Munikar et al. (2019), ♠: results from Rajagopal et al. (2021), and ♢: our implementation of BERT.

Model	Micro P	Micro R	Micro F1
SGM♣	74.8	67.5	71.0
BERT-Base♠	-	-	73.4
BERT-Base♢	77.0	70.9	73.8
MaskLoss$_{\texttt{label}}$	76.8	**72.3**	74.5
MaskLoss$_{\texttt{elem}}$	**78.1**	72.0	**74.9**

Table 3: A performance demonstration of multi-label classification task on AAPD in test datasets with micro precision (P), micro recall (R), and micro F1 measure. Best values are bolded. ♣: results from Yang et al. (2018), ♠: results from Rajagopal et al. (2021), and ♢ is our implementation of BERT.

is easier than MaskLoss$_{\texttt{elem}}$, and also has a nice result over all baselines.

In all of the specific value comparisons, MaskLoss$_{\texttt{elem}}$ is higher than the best baseline SelfExplain-RoBERTa-Base 0.6%, MaskLoss$_{\texttt{label}}$ is 0.5% in SST-5, MaskLoss$_{\texttt{elem}}$ approach is only slightly higher than MaskLoss$_{\texttt{label}}$. And in TREC-50, MaskLoss$_{\texttt{elem}}$ is higher than the best baseline, our own BERT-Base 1.4% and 0.6% in MaskLoss$_{\texttt{label}}$. The result of WOS46985 is similar to SST-5, it maybe that fewer or much more labels will not create a gap between these two methods. As for AAPD, our implement baseline fine-tuned BERT-Base is a litter more than the one in previous work. And base on it, the



(a) SST-5 (b) TREC-50 (c) WOS46985 (d) AAPD

Figure 1: An overview of the influence of random rate α in all datasets, comparing with its baseline, MaskLoss$_{label}$, and MaskLoss$_{elem}$ approach.

Gold	Baseline	MaskLoss$_{label}$	MaskLoss$_{elem}$
NUM_speed	NUM_dist	NUM_speed	NUM_speed
NUM_money	NUM_perc	NUM_perc	NUM_money

Table 4: There are some cases of different models. Those cases are from TREC-50. Color red and green denote wrong and right labels, respectively.

MaskLoss$_{label}$ and MaskLoss$_{elem}$ are better than any other methods overall, with 0.7% and 1.1% points increase in micro F1 measure on the test dataset, respectively. Above all, it can be seen that both methods are very effective.

6 Analysis

In this section, we analyze the influence of random rate α and have a case study in detail.

6.1 Influence of Random Rate α

We listed the rate of random α performance presentation from 0 to 1 at 0.1 interval on all datasets, as shown in Figure 1.

In terms of the overall trend, MaskLoss$_{label}$ (blue line) method is roughly convex object with the increase of α, while MaskLoss$_{elem}$ (red line) is different from MaskLoss$_{label}$, decreasing with the α. Most rates of that two methods are both stronger than the baseline (dotted line), and what exceeded expectations is the values on TREC-50 dataset all perform better (Figure 1(b)).

In particular, α = 0 means that the random is not adopted; α = 1 means the baseline.

6.2 Case Study

We select some labels (Figure 2) and typical cases (Table 4) from TREC-50, to show the effect of MaskLoss$_{label}$ and MaskLoss$_{elem}$ in detail.

Figure 2 illustrates the performance of similar labels of NUM (number) and $ENTY$ (entity) in TREC-50 with the comparison of three models: baseline, MaskLoss$_{label}$, and MaskLoss$_{elem}$, respectively. It is worth pointing out that

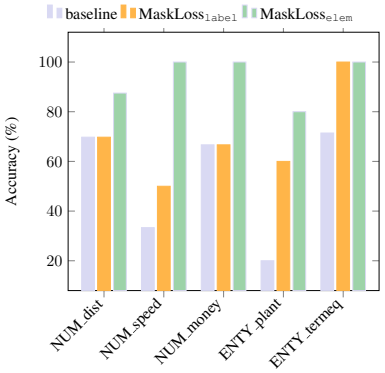


Figure 2: An overview of the performance of different labels in TREC-50 datasets, comparing with its baseline, MaskLoss$_{label}$, and MaskLoss$_{elem}$ approach.

MaskLoss$_{label}$ and MaskLoss$_{elem}$ approach perform better than baseline, in general, and the latter is stronger than the former.

In fact, most of wrong cases are predicted on its similar labels, as shown in Table 4, like NUM_speed to NUM_dist, NUM_money to NUM_perc. Note that the labels with the same prefix are defined as similar labels. MaskLoss$_{label}$ can solve this problem to some extent, while MaskLoss$_{elem}$ is more effective, and this conclusion is consistent with the trend in Figure 2.

7 Conclusion

We present a very easy and effective method MaskLoss for similar labels misprediction by masking the loss mechanism without any other training cost. There are two approaches of MaskLoss: manual MaskLoss$_{label}$ and automatic MaskLoss$_{elem}$, respectively. Experiments and analysis were performed based on four widely used datasets. Results showed that those two methods are both effective, and further find that MaskLoss$_{elem}$ performs better than the MaskLoss$_{label}$. And in addition, we investigated an important factor, the rate of random, which plays a critical role. Finally, similar labels have a significantly improved performance

through the case study, and it proves that MaskLoss is suitable to solve the problem with similar labels.

References

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*.

Cunxiao Du, Zhaozheng Chen, Fuli Feng, Lei Zhu, Tian Gan, and Liqiang Nie. 2019. Explicit interaction model towards text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6359–6366.

Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. 2001. Toward semantics-based answer pinpointing. In *Proceedings of the First International Conference on Human Language Technology Research*.

Zikun Hu, Xiang Li, Cunchao Tu, Zhiyuan Liu, and Maosong Sun. 2018. Few-shot charge prediction with discriminative legal attributes. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 487–498.

Kamran Kowsari, Donald E Brown, Mojtaba Heidarysafa, Kiana Jafari Meimandi, Matthew S Gerber, and Laura E Barnes. 2017. Hdltex: Hierarchical deep learning for text classification. In *2017 16th IEEE international conference on machine learning and applications (ICMLA)*, pages 364–371. IEEE.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *COLING 2002: The 19th International Conference on Computational Linguistics*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Qianwen Ma, Chunyuan Yuan, Wei Zhou, and Songlin Hu. 2021. Label-specific dual graph neural network for multi-label text classification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3855–3864.

Manish Munikar, Sushil Shakya, and Aakash Shrestha. 2019. Fine-grained sentiment classification using bert. In *2019 Artificial Intelligence for Transforming Business and Society (AITB)*, volume 1, pages 1–5. IEEE.

Dheeraj Rajagopal, Vidhisha Balachandran, Eduard Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. *arXiv preprint arXiv:2103.12279*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y 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.

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.

Che-Ping Tsai and Hung-Yi Lee. 2020. Order-free learning alleviating exposure bias in multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6038–6045.

Lin Xiao, Xin Huang, Boli Chen, and Liping Jing. 2019. Label-specific document representation for multi-label text classification. 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 466–475.

Nuo Xu, Pinghui Wang, Long Chen, Li Pan, Xiaoyan Wang, and Junzhou Zhao. 2020. Distinguish confusing law articles for legal judgment prediction. *arXiv preprint arXiv:2004.02557*.

Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. Sgm: sequence generation model for multi-label classification. *arXiv preprint arXiv:1806.04822*.

Ronghui You, Zihan Zhang, Ziye Wang, Suyang Dai, Hiroshi Mamitsuka, and Shanfeng Zhu. 2019. Attentionxml: Label tree-based attention-aware deep model for high-performance extreme multi-label text classification. *Advances in Neural Information Processing Systems*, 32:5820–5830.

Qian-Wen Zhang, Ximing Zhang, Zhao Yan, Ruifang Liu, Yunbo Cao, and Min-Ling Zhang. 2021a. Correlation-guided representation for multi-label text classification.

Wenjie Zhang, Junchi Yan, Xiangfeng Wang, and Hongyuan Zha. 2018. Deep extreme multi-label learning. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 100–107.

Ximing Zhang, Qian-Wen Zhang, Zhao Yan, Ruifang Liu, and Yunbo Cao. 2021b. Enhancing label correlation feedback in multi-label text classification via multi-task learning. *arXiv preprint arXiv:2106.03103*.

A Appendix

A.1 Our Model

An overview of models with traditional method, MaskLoss$_{\texttt{label}}$, and MaskLoss$_{\texttt{elem}}$ as shown in Table 3.

A.2 Manual Group Construction

The construction of manual groups is used in the vanilla MaskLoss method. Table 6 shows the specific groups of all datasets. The following is the construction source of each dataset:

- **SST-5**: According to the sentiment polarity of the label, the original fine categories are divided into three groups.
- **TREC-50**: Since the fine-grained 50 classifications of TREC-50 originally derived from the coarse-grained 6 classifications, it can be obtained easily into 6 groups.
- **WOS46985**: Following the official website[2], we called the area of the same domain are in the same group.
- **AAPD**: We split each label with '.', and then make them a group with the same prefix part.

A.3 Detailed Group Construction of Element-Base MaskLoss

(1) We get the decoded training dataset $\mathcal{D}_1 = \{x; y; p_1\}$, where p denotes the probability scores for each incorrect predicted sentence, by the trained model $\texttt{M}_{basic}$ in Section 3.1.

(2) For each sample $x^i \in x$ in $\mathcal{D}_1$, we select x^i when all $y_j^i = S_j^i$ ($y_j^i \in y^i, S_j^i \in S^i$) and $S_j^i = 1$ when $p_j^i > 0.5$ else 0, and named this new dataset $\mathcal{D}_2 = \{x; y; p_1\}$.

(3) For each sample $x^i = [x_1^i, ..., x_n^i]$ in $\mathcal{D}_2$, we split it into n new samples with the original labels by remove each word x_j^i $\texttt{n}$ times, and we call it $\mathcal{D}_3 = \{x; y; p_1\}$.

(4) We obtain the decoded $\mathcal{D}_4 = \{x; y; p_1; p_4\}$ dataset by $\texttt{M}_{basic}$ model, and collect keywords set. For each probability $p_{4j}^i \in p_4^i$, if $\texttt{count}(p_{1j}^i > p_{4j}^i + \texttt{t})$ ($p_{1j}^i \in p_1^i$ and $p_{4j}^i \in p_4^i$), where $\texttt{t} \in (0, 1)$ is threshold. We get its index set $I^i = \{I_0^i, ..., I_k^i\}$ and the removed word w_i of original text x^i together and finally we get the keyword and its related label pair $(w_i; I^i)$, and for each sample x^i, its group G^i is the union set of I of all keywords in x^i.

[2]https://data.mendeley.com/datasets/9rw3vkcfy4/2

A.4 Threshold of Element-Based MaskLoss

We tried experiments from 0 to 1 at 0.1 intervals to obtain the best optimal threshold $\texttt{t}$ for each dataset, it can be shown in Table 5.

Dataset	Threshold $\texttt{t}$
SST-5	0.6
TREC-50	0.5
WOS46985	0.6
AAPD	0.5

Table 5: Threshold $\texttt{t}$ selection of each dataset for MaskLoss$_{\texttt{elem}}$ approach.

A.5 Settings

The implementation of models of our experiments is Pytorch with version 1.8 and GPU device with V100(32G). There are several hyper-parameters contained in our model. We set the max word length to 300, batch size to 32, and total epoch to 20 (max length to 512 and batch size to 16 for AAPD).

To get a more convincing baseline, we use BERT (Devlin et al., 2018) (uncased version)[3] with AdamW optimizer (Loshchilov and Hutter, 2017), learning rate 2e-5, and weight decay 0.01 empirically. We evaluate model each half epoch and obtain the final test performance from the best development score via early stop strategy. As for prediction, the threshold of multi-label task is 0.5 and single-label is argmax.

We use the evaluate tool is sklearn tool.

A.6 Influence of Threshold $\texttt{t}$

The threshold of group selection in MaskLoss$_{\texttt{elem}}$ method from 0.1 to 1 at 0.1 interval on all datasets, as shown in figure 4. On the whole, most of the thresholds are higher than baseline. All the optimal values are concentrated in the range of 0.5-0.7, and it can be concluded that at this range of influence of a word on labels, there is a strong correlation between the labels and the final performance. And we are glad to see that all AAPD dataset (deep blue line) performances are better than baseline.

In particular, threshold = 1 means the baseline method.

[3]https://github.com/google-research/bert

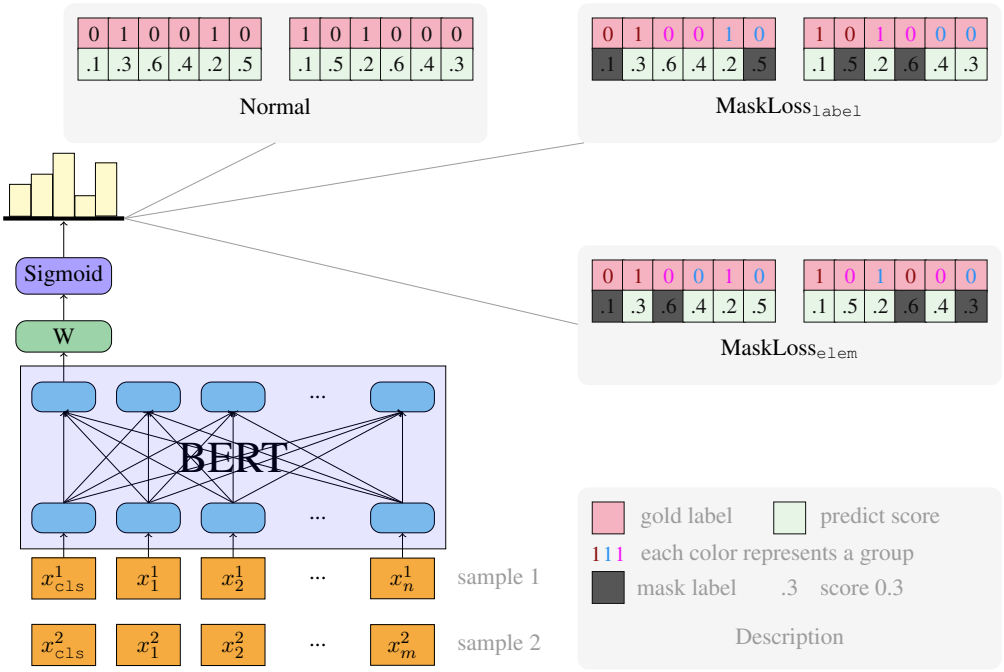


Figure 3: An overview of normal, MaskLoss$_{\texttt{label}}$ and MaskLoss$_{\texttt{elem}}$ approach.

Dataset	Group	#Num
SST-5	weakly positive, strongly positive	2
	neutral	1
	strongly negative, weakly negative	2
TREC-50	ENTY_animal, ENTY_body, ENTY_color, ENTY_cremat, ENTY_currency, ...	22
	DESC_def, DESC_desc, DESC_manner, DESC_reason	4
	HUM_gr, HUM_ind, HUM_title, HUM_desc	4
	LOC_city, LOC_country, LOC_mount, LOC_other, LOC_state	5
	NUM_code, NUM_count, NUM_date, NUM_dist, NUM_money, NUM_ord, NUM_other, ...	13
	ABBR_abb, ABBR_exp	2
WOS46985	Computer Science: Symbolic computation, Computer graphics, ...	17
	Medical Sciences: Alzheimer's Disease, Sprains and Strains, Cancer, Sports Injuries, ...	53
	Civil Engineering: Green Building, Water Pollution, Stealth Technology, ...	11
	Electrical Engineering: Electric motor, Satellite radio, Digital control, Microcontroller, ...	16
	biochemistry: Molecular biology, Enzymology, Southern blotting, Human Metabolism, ...	9
	Mechanical Engineering: Fluid mechanics, Hydraulics, computer-aided design, ...	9
	Psychology: Prenatal development, Attention, Eating disorders, ...	19
AAPD	cs.it, cs.lg, cs.ai, cs.ds, cs.si, cs.dm, cs.lo, cs.cc, cs.ni, cs.cv, cs.cl, cs.cr, cs.sy, cs.dc, cs.ne, ...	33
	stat.ml, stat.th, stat.ap, stat.me	4
	physics.soc-ph, physics.data-an	2
	nlin.ao	1
	cond-mat.stat-mech, cond-mat.dis-nn	2
	q-bio.nc, q-bio.qm	1
	math.it, math.co, math.oc, math.pr, math.na, math.st, math.lo, math.nt	8
	quant-ph	1
	cmp-lg	1

Table 6: An overview of each manual group of each dataset. #Num denotes the number of label in each group.

Text	Gold	Baseline	MaskLoss	ElementWise
optimization methods are at the core of many ...	cs.lg, math.oc, cs.cv, cs.na	math.oc	cs.lg, math.oc	cs.lg, math.oc, cs.na
the dominant cost in solving least square ...	cs.lg, cs.na	cs.lg, stat.ml, cs.na	cs.na, math.na	cs.lg, cs.na

Table 7: There are some cases of different models. That case is from AAPD. Color red and green denote wrong and right labels, respectively.

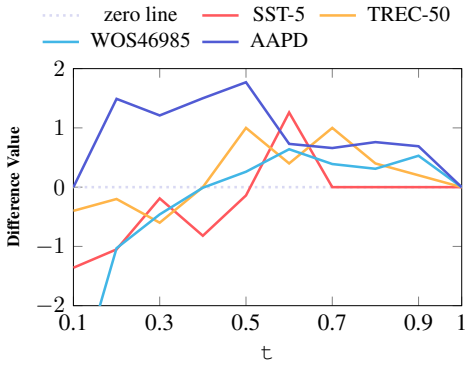


Figure 4: Influence of threshold in MaskLoss$_{\texttt{elem}}$ approach in all datasets.

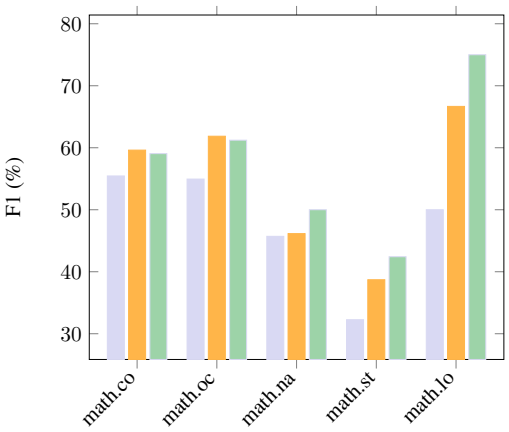


Figure 5: An overview of the performance of math-related labels in AAPD datasets, comparing with its baseline, MaskLoss$_{\texttt{label}}$, and MaskLoss$_{\texttt{elem}}$ approach.