Progressive Class Semantic Matching for Semi-supervised Text Classification

Anonymous ACL submission

Abstract

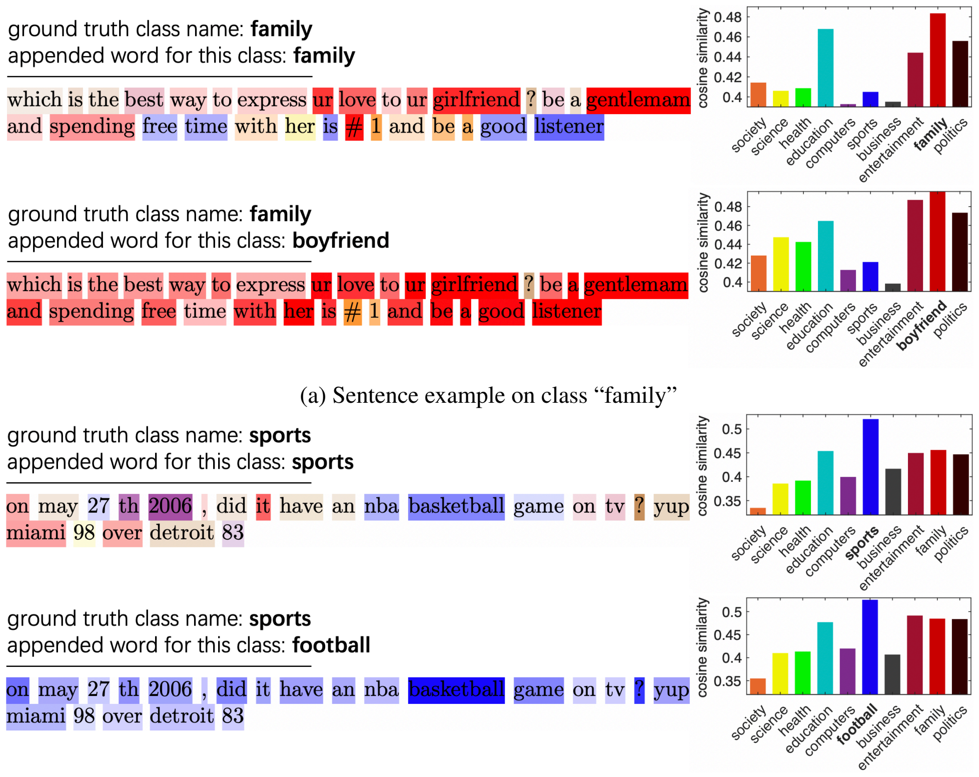
Semi-supervised learning is a promising way to reduce the annotation cost for text-classification. Combining with pre-trained language models (PLMs), e.g., BERT, recent semi-supervised learning methods achieved impressive performance. In this work, we further investigate the marriage between semi-supervised learning and a pre-trained language model. Unlike existing approaches that utilize PLMs only for model parameter initialization, we explore the inherent topic matching capability inside PLMs for building a more powerful semi-supervised learning approach. Specifically, we propose a joint semi-supervised learning process that can progressively build a standard K-way classifier and a matching network for the input text and the Class Semantic Representation (CSR). The CSR will be initialized from the given labeled sentences and progressively updated through the training process. By means of extensive experiments, we show that our method can not only bring remarkable improvement to baselines, but also overall be more stable, and achieves state-of-the-art performance in semi-supervised text classification.

1 Introduction

Text classification is a fundamental task in natural language processing (NLP) and underpins various applications, e.g., spam detection (Jindal and Liu, 2007), sentiment analysis (Pang et al., 2002) and text summarization (Gambhir and Gupta, 2017). Supervised training of text classifiers often demands a large amount of annotation, which can be expensive for many applications. Semi-supervised learning (SSL) provides an economical way for alleviating this burden since it can make use of easy-accessible unlabeled samples to build a reasonably performed classifier with a limited amount of labeled data. Recently, SSL received increasing attention in both image classification (Tarvainen and Valpola, 2017; Berthelot et al., 2019b; Sohn et al., 2020) and text classification (Xie et al., 2019b; Chen et al., 2020; Liu et al., 2021) areas.

Meanwhile, pre-trained language models (PLMs) (Yang et al., 2019a; Devlin et al., 2019; Radford et al., 2019) are developing rapidly and achieve impressive performance in various NLP tasks (Sun et al., 2019; Zhu et al., 2020) including text classification (Garg and Ramakrishnan, 2020). In the context of semi-supervised text classification, many existing methods achieve excellent performance by directly using a PLM as a sentence encoder and further fine-tuning it with a semi-supervised learning process (Xie et al., 2019b; Chen et al., 2020; Bhattacharjee et al., 2020; Sun et al., 2020).

In this paper, we further explore the usage of PLMs for SSL. We go beyond the strategy of using PLMs for encoder initialization and make full use of inner knowledge of PLMs. Concretely, we identify that some PLMs, e.g., BERT, have an inherent matching capability between sentence and class-related words thanks to its pre-training pretext task (Devlin et al., 2019) (as the examples shown in Fig. 1). We further propose to strengthen this capability through SSL on labeled and unlabeled data. Specifically, we develop a joint training process to update three components progressively, that is, a classifier that performs the standard K-way classification, a class semantic representation (CSR) that represents the semantic of each category, and a matching classifier that matches the input sentence against the CSR. Those three components can help each other during the training process, i.e., the K-way classifier will receive more accurate pseudo-labels by jointly generating pseudo-labels with the matching classifier; the matching classifier will also upgrade its matching capability with the guidance of the K-way classifier. The CSR will become more accurate and comprehensive with the improvement of the K-way classifier and matching classifier. This joint process leads to a more pow-



(a) Sentence example on class "family"

(b) Sentence example on class "sports"

Figure 1: Visualization of the inherent matching capability of BERT on examples from Yahoo! Answers. We append class semantic-related words (CSW) of all classes at the end of input sentence . Different colors denote different classes. The color on each token of input sentence represents the category of its most attended CSW (with color brightness indicating the attention value, please see Sec.3 for more details). The histograms on the right demonstrate the cosine similarity between the average features of sentence and features of each CSW.

erful semi-supervised learning algorithm for the text classification task. Throughout our experimental evaluation, we demonstrate that the proposed method achieves the state-of-the-art performance on text data, especially when the number of labeled sentences becomes extremely low, i.e., 3 or 5.

2 Related work

In this section, we briefly review the relevant research works.

2.1 General Semi-Supervised Learning

Semi-supervised learning is a longstanding research topic in machine learning. Existing methods adopt different ways of utilizing unlabeled samples, e.g., "transductive" models (Joachims, 2003; Gammerman et al., 2013), multi-view style approaches (Blum and Mitchell, 1998; Zhou and Li, 2005) and generative model-based methods (Kingma et al., 2014; Springenberg, 2016). With the renaissance of the deep neural network, consistency-regularization-based deep SSL approaches (Laine and Aila, 2017; Tarvainen and Valpola, 2017; Miyato et al., 2018) have achieved impressive performance on various tasks, and our work largely builds upon the method in this category. The key idea of these methods is to constrain the model to be consistent in the neighborhood of each sample in the input space. Specifically, Π-Model (Laine and Aila, 2017) and UDA (Xie et al., 2019b) and FixMatch (Sohn et al., 2020) directly add various perturbations to the input data, Mean-teacher (Tarvainen and Valpola, 2017) uses a teacher model to simulate sample perturbation, and Virtual Adversarial Training (Miyato et al., 2018) skillfully constructs an adversarial sample. More recently, mixup (Zhang et al., 2018) method proposed another kind of consistency constraint that requires the input and output of the model to satisfy an identical linear relationship. Based on this technique, many state-of-the-art methods are published, e.g., ICT (Verma et al., 2019b), MixMatch (Berthelot et al., 2019b) and ReMixMatch (Berthelot et al., 2019a).

2.2 Semi-Supervised Text Classification

Semi-supervised learning has gained a lot of attention in the field of text classification. Many recent semi-supervised text classification methods focus on how to adapt the existing SSL methodologies to the sentence input. (Miyato et al., 2017) applied perturbations to word embeddings for constructing adversarial and virtual adversarial training. (Clark et al., 2018) designed auxiliary prediction modules with restricted views of the input to encourage consistency across views. With the development of PLMs, (Jo and Cinarel, 2019) performed self-training between two sets of classifiers which are initialized differently, one with pre-trained word embeddings and random values for the other. Both (Xie et al., 2019b) and (Chen et al., 2020) took the pre-trained BERT to initialize the sentence feature extractor, where the former conducted consistency-regularization between the original sentence and its back-translation generated one, and the latter further introduced the manifold mixup (Verma et al., 2019a) into text classification. Although these methods may achieve decent performances, we believe that they haven't fully explored the inherent knowledge in a PLM. Our work takes a step further in this direction.

3 Inherent matching capability of a PLM

In this section, we will demonstrate the inherent topic matching capability of BERT which motivates our method. Utilizing PLMs for a downstream task has become common since it often brings a significant performance boost (Zhu et al., 2020; Chen et al., 2020). In the context of semi-supervised learning, a PLM is usually employed for initializing the network before performing semi-supervised training. However, the value of a PLM can go beyond a good initial model or feature extractor. In particular, a PLM like BERT has already learned certain topic matching capabilities thanks to its pretext tasks. For example, BERT uses the next sentence prediction (NSP) as one of its pretext tasks. In this task, the network is asked to discern if two input sentences are two successive sentences in the original corpus. After training on this task, BERT can implicitly acquire topic matching capability since two successive sentences in a paragraph usually share the same topic.

Fig. 1 shows a concrete investigation of the inherent matching capability of BERT. Following the NSP task, we concatenate the sentence and class semantic-related words C_k, e.g., "sports", via the format: "[CLS] sentence [SEP] $C_1 \cdots C_k \cdots C_K$ [SEP]". Then we pass the input sequence to a pre-trained BERT and calculate the attention value of each token with respect to each class name. Specifically, this attention value is calculated by averaging the last layer self-attention values across all heads between a token and the appended word C_k . For better visualization, we use different color to show the class that leads to the largest attention value (indicated by the color brightness).

From the visualization, we can see that BERT can automatically match keywords corresponding to the respective class names. Moreover, we find that if we replace the class names with words under the same topic, i.e., family → boyfriend, sports → football, the words related to the ground-truth class can still be attended, as shown in Fig. 1a and 1b.

Finally, we extract BERT last-layer's feature corresponding to each class word C_k and average features align with sentence tokens, and compare the cosine similarity between them. As histograms shown in Fig. 1a and 1b, we can find that the correct class leads to the highest matching score, although not always by a large margin.

4 Progressive Class-semantic Matching

To further strengthen the above topic matching capability and use it for classification, we propose to progressively build a sentence-class matching model through the framework of semi-supervised learning. Formally, we aim to build a classifier from a few annotated samples $\mathcal{L} = \{x_1, x_2, \cdots, x_{n_l}\}$, whose labels are $\mathcal{Y} = \{y_1, y_2, \cdots, y_{n_l}\}, y_i \in \{1, \cdots, k, \cdots, K\}$, and a large amount of unlabeled samples $\mathcal{U} = \{x_1, x_2, \cdots, x_{n_u}\}$ (where $n_l \ll n_u$).

The idea is to construct a process that can jointly update three components: (1) a standard K-way classifier (2) a matching classifier which matches texts against class semantic representation (3) the class semantic representation (CSR) itself. The update of each component will help other components and thus can iteratively bootstrap classification performance. We call our method as Progressive Class-semantic Matching (PCM).

4.1 Three components of PCM

Fig. 2 shows how we realize the three components. Similar to the example in Section 3, we construct the input to the BERT by concatenat-

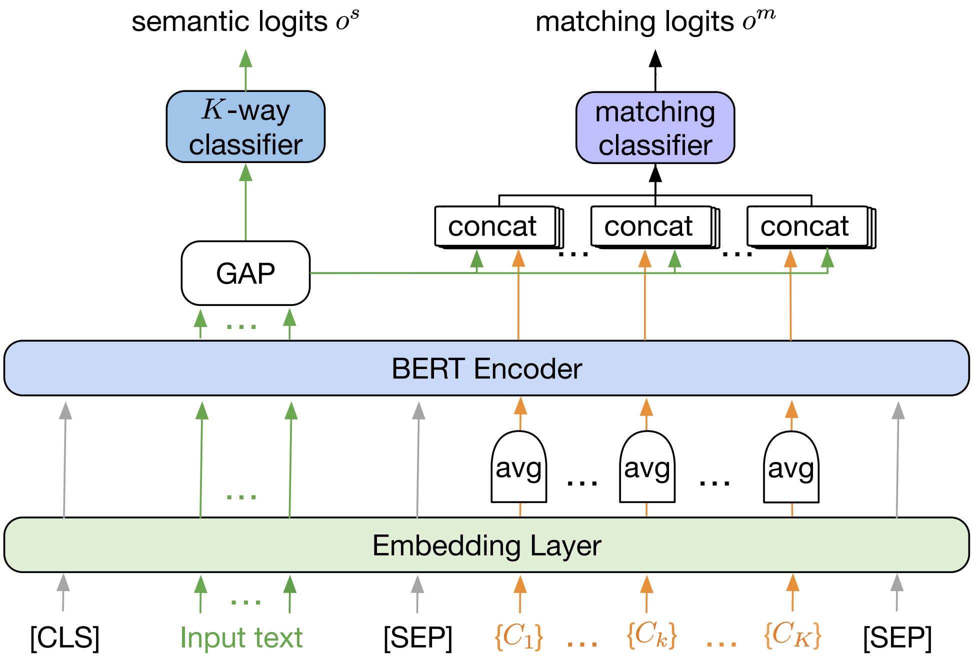


Figure 2: Overview of the proposed PCM model. Lines in the same color indicate how the information travels in our model. $\{C_k\}$ denotes the set of class semantic-related words. "avg" means the average of word embeddings within the same class. "GAP" represents the global average pooling of the input text features. "concat" is a feature concatenate operation. We clarify the details of initializing and updating of $\{C_k\}$ in Secs. 4.2 and 4.3.

ing sentence with class semantic-related words $\{C_i\}, i \in \{1, \cdots, k, \ldots, K\}$. Considering the size of $\{C_i\}$ may vary and the computation cost may increase heavily when the number of classes grows, we calculate an average of embeddings of all words belonging to the same class before passing them to the pre-trained BERT encoder. This average embedding is called **class semantic representation (CSR)**.

The last layer output features corresponding to tokens in the input text are averaged and treat as the sentence representation. On top of the sentence representation, we build a **standard K-way classifier**. We implement it by a two-layer MLP and it will output a set of logits $\{o_i^s\}$ called semantic logits and posterior probabilities $\{p_i^s\}$ after applying Softmax to $\{o_i^s\}$.

In addition to the K-way classifier, we also build **a class-sentence matching classifier** which is realized by another MLP applying to the concatenation between the sentence representation and the output features corresponding to each CSR. The output of this matching classifier is called matching logits $\{o_i^m\}$ and Sigmoid function is applied to convert it into the probabilistic form, denoted as $\{p_i^m\}$. Note that the matching classifier is realized in a multi-label formulation, that is, the summation of $\{p_i^m\}$ over all classes is not necessarily equal to 1. It allows the scenario that a sentence matches more than one class and the case that a sentence does not match any class. This design avoids the case that achieving high matching probability for one class merely because its matching score is higher than those of other classes (but it actually with low absolute matching logits for all classes). We empirically find that using this mechanism is helpful for the matching classifier (but not necessarily for the K-way predictor as discussed in Section 5.2).

4.2 Initialization of CSR

The proposed PCM model requires an initial CSR, i.e., the average word embedding of a set of class semantic-related words, to start the iteration. Although manually choosing a list of seed words (e.g., class names) can be an ideal way for the CSR initialization, it may suffer from leveraging prior knowledge and leads to an unfair comparison to existing SSL algorithms. An alternative approach is to automatically identify a set of class semantic-related words. This might be useful for the case that class names in some corpora do not carry a clear semantic meaning, e.g., the rating of reviews.

In this paper, we use the following method to automatically collect the class semantic-related words: we start by fine-tuning a pre-trained BERT classifier on the labeled set. Then passing each labeled text into the fine-tuned model and calculate attention values for each token. The attention value of a token is calculated by averaging all the attention received for this token [1]. After removing stop words, we retain the top-j e.g., $j = 75$, attended words for each class to calculate the initial CSR.

4.3 Update of three components

The three components are progressively updated by seamlessly incorporating them into an SSL framework. In particular, our method is built upon UDA (Xie et al., 2019b), one of the state-of-the-art approaches in semi-supervised text classification. The idea is to first construct an augmented version of unlabeled data by back translation (Edunov et al., 2018) and then enforce the prediction to be consistent through a consistency-regularization loss for unlabeled data. The following describes the detailed updating process:

Update of the standard K-way classifier and the class-sentence matching classifier: The update is performed on labeled and unlabeled data at the same time. For labeled data, both classifiers are updated by performing stochastic gradient descent with the following objective function.

$$\mathcal{L}_l = \frac{1}{n_l}\sum_{j=1}^{n_l}\sum_{i=1}^{K} \underbrace{-\mathbb{I}_i^j \log p_i^s(x_j)}_{\text{cross entropy (CE)}} + \underbrace{-\mathbb{I}_i^j \log p_i^m(x_j) - (1-\mathbb{I}_i^j)\log\big(1-p_i^m(x_j)\big)}_{\text{binary cross entropy (BCE)}}, \tag{1}$$

where $p_i^s(x_j)$ and $p_i^m(x_j)$ are the probabilities of x_j belonging to class i from the view of the K-way classifier and the matching classifier, respectively. Since the matching classifier is designed in a multi-label style, we use binary cross-entropy loss for it. $\mathbb{I}_i^j$ is an indicator whose value equals to 1 if $y_j = i$, and 0 otherwise.

For unlabeled data, we follow UDA to use a student-teacher alike training strategy, that is, we first use the original sentence input $x_j \in \mathcal{U}$ to obtain the prediction target (similar to a pseudo label) and then enforce the prediction of the back-translated version x_j^a of x_j being close to the prediction target. Formally, if the prediction of one unlabeled sample satisfies all the following rules, the prediction target will be generated:

$$\begin{cases} \max_i\big(p_i^s(x_j)\big) >= \text{confid1} \\ \max_i\big(p_i^m(x_j)\big) >= \text{confid2} \\ \text{argmax}_i\big(p_i^s(x_j)\big) == \text{argmax}_i\big(p_i^m(x_j)\big) \end{cases} \tag{2}$$

where confid1 and confid2 are two pre-defined confidence thresholds and we empirically find confid1 $= 0.95$ and confid2 $= 0.7$ performs well in our experiments. For the K-way classifier, the pseudo prediction target is a sharpened posterior probability, i.e., $\hat{p}^s = \text{Softmax}(o^s/T)$ with $T \leq 1$. For the matching classifier, we directly generate a pseudo-label by $\hat{y}_i = \text{argmax}_i\ p_i^m$. The loss function for the unlabeled data is

$$\mathcal{L}_u = \frac{1}{n_u}\sum_{j=1}^{n_u}\Big(\text{KL}\big(p^s(x_j^a), \hat{p}^s(x_j)\big) + \text{BCE}\big(p^m(x_j^a), \hat{y}_i(x_j)\big)\Big) \tag{3}$$

where $\text{KL}(\cdot,\cdot)$ denotes the KL divergence.

Update of CSR: The initialized CSR might not be accurate or comprehensive enough to represent the class semantics. Similar to the approach proposed in Section 4.2, we use the newly updated model to collect a better CSR. The collection process on labeled sentences is still as described in Section 4.2. While the same extraction operation is performed on unlabeled texts only when they satisfy the conditions in Eq. 2. We update the CSR whenever the number of validation set[2] samples meeting conditions in Eq. 2 increases. Generally, during the course of semi-supervised learning, the classifiers become stronger and the selected class-related words tend to become more accurate. Table 3 gives an example to show the difference of most attended words between initialization and after training.

5 Experimental results

In this section, we perform the experimental study of the PCM method on four text datasets.

Datasets Following MixText (Chen et al., 2020), we use four datasets, namely, AG News (Zhang et al., 2015), DBpedia (Lehmann et al., 2015), Yahoo! Answers (Chang et al., 2008), and IMDB (Maas et al., 2011) for our experiments. We use the same data splits as in MixText (Chen et al.,

[1] Magnitude of the attention value indicates the importance of this token.

[2] Please note that we do not use any label information here.

Dataset	**Label Type**	**# Classes**	**# Unlabeled**	**# Test**
AG News	News Topic	4	20,000	7,600
DBpedia	Wikipedia Topic	14	70,000	70,000
Yahoo! Answers	QA Topic	10	50,000	60,000
IMDB	Review Sentiment	2	10,000	25,000

Table 1: Statistics of four text datasets.

Dataset	Method	Label Number Per Class				
		3	**5**	**10**	**20**	**50**
AG News	BERT-FT	76.70±4.72	79.90±2.34	83.46±2.73	84.97±1.73	87.35±0.56
	UDA	78.25±7.61	82.97±2.87	86.75±0.88	86.77±0.10	88.23±0.49
	MixText	81.60±9.04	85.84±1.32	85.56±2.95	87.60±0.48	88.14±0.75
	PCM(ours)	**84.85±0.86**	**87.20±0.42**	**88.31±0.47**	**88.34±0.27**	**88.85±0.27**
DBpedia	BERT-FT	86.68±2.59	91.86±2.46	96.60±0.46	97.84±0.23	98.59±0.22
	UDA	93.51±2.23	95.88±2.78	97.26±1.50	98.59±0.04	98.93±0.06
	MixText	93.25±0.68	96.93±0.41	98.39±0.09	98.64±0.18	98.84±0.05
	PCM(ours)	**94.37±0.49**	**97.04±0.68**	**98.70±0.04**	**98.80±0.06**	**99.07±0.05**
Yahoo!	BERT-FT	45.93±3.67	50.75±4.32	61.84±2.37	63.89±0.94	67.29±0.68
	UDA	48.30±11.09	57.09±5.69	65.15±1.54	67.76±0.60	69.38±0.78
	MixText	60.27±4.29	65.77±1.78	67.23±1.97	68.19±1.33	69.11±0.73
	PCM(ours)	**63.52±2.63**	**67.09±0.54**	**68.34±1.03**	**69.21±0.42**	**70.28±0.47**
IMDB	BERT-FT	60.11±2.41	65.17±8.39	73.20±2.97	78.70±6.75†	83.91±1.13
	UDA	63.01±1.07	71.90±10.80	89.05±1.70	90.20±0.54‡	90.41±0.45
	MixText	56.27±3.46	71.89±4.89	83.38±3.35	86.27±1.36	88.30±1.24
	PCM(ours)	**73.86±1.04**	**86.06±0.74**	**89.94±0.44**	**91.10±0.28**	**91.15±0.15**

† Single run accuracy (81.6%) is reported in UDA (Xie et al., 2019b) for a reference. ‡ This number is reasonable on one GPU card with 11GB memory. See experimental tutorial (Xie et al., 2019a) for details.

Table 2: Test accuracy (%) of all comparing methods on four datasets. Models are trained with 3/5/10/20/50 labeled data per class. ± denotes the Standard Error of the Mean (S.E.M.) over three random sampled label sets. Best results are indicated as bold.

2020). The detailed statistics of the four datasets are presented in Table 1.

Implementation details Same as MixText [3], we use back-translation to perform data augmentation. Two languages, German and Russian, are chosen as the intermediate language. The back-translation texts on Yahoo! Answers are provided by MixText, and we directly use them. For the other three datasets, we generate the back-translation data by ourselves (with Fairseq toolkit (Ott et al., 2019).[4]

We use the input format "[CLS] Sentence [SEP]" for all the baseline methods. We empirically find this format leads to the overall best performance. Meanwhile, this format actually brings performance improvement to both UDA and MixText methods. So we are comparing against stronger baselines in our paper.

Due to BERT's length limit, we only kept the last 256 tokens for IMDB and the first 256 tokens for the other datasets during training. We use the same learning-rate setting for all methods: 5e-6 for the BERT encoder and 5e-4 for the classifier (i.e., a two-layer MLP with a 128 hidden size and $tanh$ as its activation function). All our experiments were run on a GeForce RTX 2080 Ti GPU and each experiment takes around 5 hours.

Comparing methods We compare the proposed PCM method with three baselines: (1) fine-tuning the pre-trained BERT-based-uncased model on the labeled texts directly, denote as **BERT-FT**. (2) Unsupervised data augmentation method (UDA) (Xie et al., 2019b) and (3) the recently proposed MixText method (Chen et al., 2020). To make a fair

[3] `https://github.com/GT-SALT/MixText` (2-clause BSD License)

[4] Our code will be released after the anonymity period.

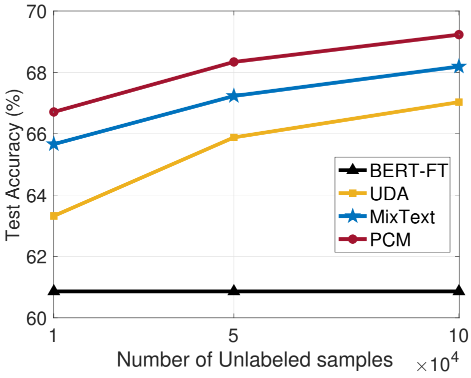


Figure 3: Accuracy on varying number of unlab. data.

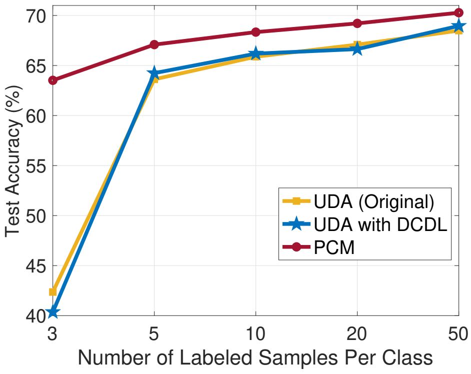


Figure 4: Ablation study on the DCDL strategy in PCM.

comparison, we conduct all experiments based on the same codebase released by the authors of MixText (Chen et al., 2020).

5.1 Main results

Table 2 presents the performance comparison of the proposed PCM method and other baselines on different datasets. From that, we can have the following observations. (1) By using BERT, all methods achieve reasonable performance. Even the BERT fine-tune baseline achieves good performance when there are ten samples per class. However, BERT fine-tune is still inferior to the semi-supervised approaches, especially when the number of training samples becomes smaller or the classification task becomes more challenging. (2) As expected, the MixText method excels UDA in most cases, but performs similarly when the number of labeled samples becomes large (e.g., 50 labels/class). Since the proposed method could also be incorporated into MixText, it might be able to boost its performance. (3) the proposed PCM methods achieves significant performance improvement over UDA approaches. Please note that PCM is built on top of the UDA method and this performance gain indicates the effectiveness of using the proposed progressive training process. (4) It is clear that PCM can not only **always outperform other baselines** and achieve state-of-the-art text classification performance on all four datasets, but also have **smaller standard error and be more stable**. PCM performs especially well when the number of labeled samples becomes small. A much larger performance gain is observed when only three labeled samples are available.

Furthermore, we compare PCM to baselines with 10 labeled data per class and varying number of unlabeled ones on Yahoo! Answers dataset (range from 10,000 to 100,000 unlabeled samples). Fig. 3 shows that PCM continuously benefits from more unlabeled data and can be consistently superior than other methods.

5.2 Ablation studies

PCM model consists of several components. In this section, we perform ablation studies to examine their impact. Most of these studies are performed on Yahoo! Answers dataset with one identical labeled set, unless otherwise specified.

1. The importance of using two classifiers in PCM. The proposed PCM model contains a K-way classifier (i.e., p^s) and a matching classifier (i.e., p^m), and they are jointly trained in the proposed process. We investigate the role of them by constructing a variant of PCM by only using either one of them. As the results shown in Table 4, without using the K-way classifier, the method totally fails to a random guess. In contrast, only keeping the K-way classifier can obtain reasonable results. More interestingly, this variant actually performs better than UDA on 3 and 5 label cases (see the Table 2). The difference between this variant and UDA is that the former appends CSR to the input sequence. Its good performance shows that merely appending CSR can be helpful for semi-supervised text classification. Finally, we can see that using both classifiers can lead to the best performance. This clearly validates the necessity of the proposed joint learning process.

2. If using the dual-classifier-dual-loss is the key to success? In our method, we utilize a slightly unconventional dual-classifier-dual-loss strategy (DCDL): the pseudo-labels are generated by checking the agreement of the two classifiers, and two losses, i.e., BCE and CE, are used for training those two classifiers. One may suspect that our good per-

Initial	bush, car, bomb, killed, chancellor, black, moscow, inter, leftlist, putin, story, presidential, texas, president, campaign, documents, ap, unearthed, caracas, ...
Final	iraq, president, iraqi, government, baghdad, military, palestinian, security, nuclear, prime, minister, country, israeli, leader, war, peace, gaza, iran, israel, troops, ...

Table 3: The class semantic-related word lists on class "world" of AG News dataset. The top row is the initial class semantic-related words obtained from fine-tuned BERT, while the bottom one is the final class semantic-related words after PCM training with upper initial words. All models are trained on the 3 labels per class case.

p^s	p^m	Label Number Per Class				
		3	**5**	**10**	**20**	**50**
✗	✓	10.01	10.45	10.01	10.21	10.05
✓	✗	49.51	65.32	65.70	67.88	68.43
✓	✓	**63.52**	**67.09**	**68.34**	**69.21**	**70.28**

Table 4: Ablation study on the importance of two classifiers of the proposed PCM model.

update CSR	Label Number Per Class				
	3	**5**	**10**	**20**	**50**
✗	39.49	66.04	66.41	67.09	68.85
✓	**63.52**	**67.09**	**68.34**	**69.21**	**70.28**

Table 5: Ablation study on the importance of updating the CSR during training of PCM.

formance actually stems from this DCDL scheme rather than leveraging BERT's matching capability. To investigate this problem, we conduct an ablation study by modifying UDA with this strategy. Specifically, we use two classifiers, one trained from the BCE loss and the other one trained from the CE loss. The pseudo-prediction targets are generated by following the same strategy as in PCM. The result is shown in Fig. 4. As seen, simply incorporating this training strategy does not necessarily bring better classification accuracy. This result provides evidence that the PCM's good performance can not be simply attributed to the DCDL strategy.

3. The prediction quality of the K-way classifier and the matching classifier. In our PCM model, the K-way classifier is chosen for the final testing phase. We further validate the quality of the matching classifier. As the results presented in Fig. 5, the matching classifier gains comparable performance to the K-way one. This proves that the collaborative training of two classifiers bootstraps each other to have good prediction capability.

4. The impact of updating CSR. Our PCM method dynamically updates the CSR through the training process. In this part, we investigate the impact of this updating process. Table 5 compares the results obtained by updating or not updating CSR. As seen, updating CSR leads to overall better performance. The difference becomes quite significant when only three labeled samples are used. For example, PCM may fail when the class semantic representation is fixed in the 3-label case.

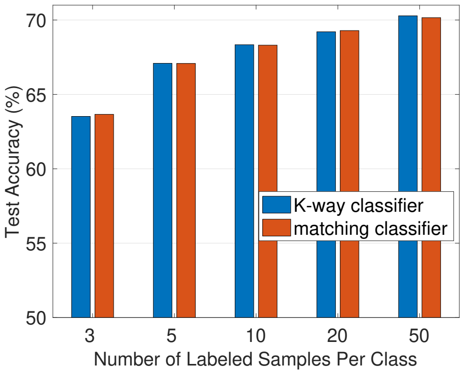


Figure 5: Ablation study on classifier quality of PCM.

6 Limitations and Potential Risks

One underlying assumption about our findings is that we mainly consider BERT-style pre-trained language models for semi-supervised text classification. The utilization of inherent knowledge of other language models (e.g., GPT (Radford et al., 2018) and XLNet (Yang et al., 2019b)) are not explored in this paper and is left for future work.

PCM algorithm has been verified to be effectiveness on texts in English, whether other languages can achieve the same performance improvement is at risk and will be explored in the future.

7 Conclusion

In this paper, we proposed a semi-supervised text classification approach by leveraging the inherent topic matching capability in pre-trained language models. The method progressively updates three components, a K-way classifier, the class semantic representation, and a matching classifier that matches input text against the class semantic representation. We show that the updating of the three components can benefit each other and achieve superior semi-supervised learning performance.

References

David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. 2019a. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*.

David Berthelot, Nicholas Carlini, Ian J. Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019b. Mixmatch: A holistic approach to semi-supervised learning. 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 5050–5060.

Kasturi Bhattacharjee, Miguel Ballesteros, Rishita Anubhai, Smaranda Muresan, Jie Ma, Faisal Ladhak, and Yaser Al-Onaizan. 2020. To BERT or not to BERT: Comparing task-specific and task-agnostic semi-supervised approaches for sequence tagging. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7927–7934, Online. Association for Computational Linguistics.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

Ming-Wei Chang, Lev-Arie Ratinov, Dan Roth, and Vivek Srikumar. 2008. Importance of semantic representation: Dataless classification. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence*, volume 2, pages 830–835.

Jiaao Chen, Zichao Yang, and Diyi Yang. 2020. MixText: Linguistically-informed interpolation of hidden space for semi-supervised text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2147–2157, Online. Association for Computational Linguistics.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1914–1925, Brussels, Belgium. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Alex Gammerman, Volodya Vovk, and Vladimir Vapnik. 2013. Learning by transduction. *arXiv preprint arXiv:1301.7375*.

Siddhant Garg and Goutham Ramakrishnan. 2020. BAE: BERT-based adversarial examples for text classification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6174–6181, Online. Association for Computational Linguistics.

Nitin Jindal and Bing Liu. 2007. Review spam detection. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007*, pages 1189–1190. ACM.

Hwiyeol Jo and Ceyda Cinarel. 2019. Delta-training: Simple semi-supervised text classification using pretrained word embeddings. 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 3458–3463, Hong Kong, China. Association for Computational Linguistics.

Thorsten Joachims. 2003. Transductive learning via spectral graph partitioning. In *Machine Learning, Proceedings of the Twentieth International Conference (ICML 2003), August 21-24, 2003, Washington, DC, USA*, pages 290–297. AAAI Press.

Diederik P. Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. 2014. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3581–3589.

Samuli Laine and Timo Aila. 2017. Temporal ensembling for semi-supervised learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

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.

Chen Liu, Mengchao Zhang, Zhibin Fu, Pan Hou, and Yu Li. 2021. Flitext: A faster and lighter semi-supervised text classification with convolution networks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*

(EMNLP), Online. Association for Computational Linguistics.

Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.

Takeru Miyato, Andrew M. Dai, and Ian J. Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 79–86. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *Advances in Neural Information Processing Systems*, volume 33, pages 596–608. Curran Associates, Inc.

Jost Tobias Springenberg. 2016. Unsupervised and semi-supervised learning with categorical generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385, Minneapolis, Minnesota. Association for Computational Linguistics.

Zijun Sun, Chun Fan, Xiaofei Sun, Yuxian Meng, Fei Wu, and Jiwei Li. 2020. Neural semi-supervised learning for text classification under large-scale pretraining. *arXiv preprint arXiv:2011.08626*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 1195–1204.

Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. 2019a. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning*, pages 6438–6447. PMLR.

Vikas Verma, Alex Lamb, Juho Kannala, Yoshua Bengio, and David Lopez-Paz. 2019b. Interpolation consistency training for semi-supervised learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3635–3641. ijcai.org.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019a. Uda experimental tutorial on text classification. `https://github.com/google-research/uda#text-classifiation`. Accessed: 2022-01-15.

Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019b. Unsupervised data augmentation for consistency training. *arXiv preprint arXiv:1904.12848*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019a. Xlnet: Generalized autoregressive pretraining for language understanding. 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 5754–5764.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *arXiv preprint arXiv:1509.01626*.

Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on knowledge and Data Engineering*, 17(11):1529–1541.

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tie-Yan Liu. 2020. Incorporating BERT into neural machine translation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.