# Don't Miss the Labels: Label-semantic Augmented Meta-Learner for Few-Shot Text Classification

**Qiaoyang Luo, Lingqiao Liu*, Yuhao Lin** and **Wei Zhang**
The Univeristy of Adelaide
{qiaoyang.luo, lingqiao.liu, wei.e.zhang}@adelaide.edu.au
linyuhao0514@gmail.com

## Abstract

Increasing studies leverage pre-trained language models and meta-learning frameworks to solve few-shot text classification problems. Most of the current studies focus on building a meta-learner from the information of input texts but ignore abundant semantic information beneath class labels. In this work, we show that class-label information can be utilized for extracting more discriminative feature representation of the input text from a pre-trained language model like BERT, and can achieve a performance boost when the samples are scarce. Building on top of this discovery, we propose a framework called Label-semantic augmented meta-learner (LaSAML) to make full use of label semantics. We systematically investigate various factors in this framework and show that it can be plugged into the existing few-shot text classification system. Through extensive experiments, we demonstrate that the few-shot text classification system upgraded by LaSAML can lead to significant performance improvement over its original counterparts.

## 1 Introduction

The remarkable capability of quickly learning new concepts from a few training samples is one of the advantages of the human learning system over the current machine learning system. Motivated by this gap, research in few-shot learning has received increasing attention in the past decade. Meta-learning (Vinyals et al., 2016; Snell et al., 2017; Finn et al., 2017), as the dominant methodology in few-shot learning, tackles the problem by learning a mapping function from a few support samples to a classifier through a meta-training dataset. Most existing meta-learning systems (Snell et al., 2017; Sung et al., 2018) were developed or at least evaluated in the field of computer vision. More recently, few-shot learning has been introduced to the NLP field and in particular, text classification (Yu et al., 2018; Geng et al., 2019), as it is the fundamental task in natural language understanding. In parallel to few-shot learning, pre-trained language models (PLMs) (Devlin et al., 2019; Radford et al., 2019) have revolutionized the NLP fields and show strong evidence of being able to perform well in low data regime when transferred to downstream tasks.

**Banking**
Class 1: *can you give me a hand paying my water bill*
Class 2: *tell me the last day I can pay my gas bill*
---
**Travel**
Class 3: *what are some fun activities to do in colorado*
Class 4: *tell me about any travel alerts issued for germany*

Figure 1: The example of fine-grained intent queries from two domains in Clinc150 dataset. Class 1 and Class 2 refer to PAY BILL and BILL DUE. Class 3 and Class 4 represent TRAVEL SUGGESTION and TRAVEL ALERT.

Despite the impressive progress of meta-learning and PLMs, however, most existing few-shot classification systems (Geng et al., 2019; Bao et al., 2020) ignore an important information source — semantic of class labels. When the number of training samples is limited, merely using the input texts per class can lead to ambiguity in interpreting the definition of class. Considering the two groups of examples in Figure 1 which shows four samples belonging to different intent classes, even humans cannot fully understand the semantic meaning of those samples if the definition of labels are not given. For example, it is hard to tell if class 1 and class 2 are about the type of bill — water or gas, or class 3 and class 4 are about the destination of the travel — USA or Germany. However, this ambiguity can be easily resolved if the class definition or simply the class name is provided.

*Corresponding Author



*Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2773–2782

Just as understanding class names can help humans to interpret sentences of a given class, we made an interesting observation that the BERT will extract more discriminative features if we append the class name to the input sentence, and it can boost the classification performance in low-shot scenarios. Motivated by the above observations, this work explores how to better leverage the semantic information beneath class names for few-shot learning. Our key idea is to use meta-learning to further strengthen the guidance of class-label semantics for few-shot classification. Specifically, we use meta-learning to encourage the features extracted from class-name-appended samples to be more class-relevant and compatible to the query features. Moreover, we systematically study the issue of how to extract the label-semantic guided feature representation from the support samples and how to make the query sample features compatible with the meta-learner generated from the support set. Our research leads to a framework that can be plugged into the existing few-shot meta-learner and we call our method Label-semantic Augmented Meta-Learner (LaSAML). To demonstrate the power of LaSAML, we use LaSAML to upgrade the Prototypical Network and creates a new method called LaSAML-PN. By conducting the extensive experimental studies, we show that LaSAML-PN achieves excellent few-shot learning performance and LaSAML upgraded meta-learning obtains superior performance over its original counterpart. Our code has been released at: https://github.com/luoqiaoyang/ACL2021-LaSAML.

## 2 Related work

This section discusses the related work from three aspects: few-shot learning, few-shot text classification, and low-shot learning with label information.

**Few-Shot learning** Meta-learning approaches have made substantial progress with few-shot learning (FSL) tasks. The focus of the current meta-learning framework is how to construct the meta-learner. For examples, a meta-learner could be constructed by learning a metric between samples and classes(Koch et al., 2015; Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018), based on a differentiable learning process (Bao et al., 2020), or based on a few-shot gradient update (Mishra et al., 2018; Finn et al., 2017). A complete review of meta-learning is beyond the scope of this paper, and we refer readers to the recent survey (Hospedales et al., 2020).

**Few-shot text classification** Few-shot text classification (FSTC) has also gained increasing attention in recent years. ROBUSTTC-FSL (Yu et al., 2018) uses an adaptive metric learning approach to adaptively select an optimal distance metric for different tasks. Induction Network (Geng et al., 2019) utilizes the dynamic routing algorithm (Sabour et al., 2017) to learn a generalized class-wise representation. Pre-trained language models have also been applied to few-shot text classification. LEOPARD (Bansal et al., 2020) uses BERT (Devlin et al., 2019) with optimization-based meta-learning framework to achieve good performance on diverse NLP classification tasks. More recently, GPT-3 (Brown et al., 2020) shows that the language model itself can be used to perform few-shot text classification without using meta-learning. Meanwhile, another recent work (Bao et al., 2020) points out meta-learning for text classification may have different characteristics to the cases in computer vision. They propose to use distributional signatures to enhance the generalization capability of meta-learner. Our method is still a meta-learning-based few-shot text classification method. The key contribution of our work is the discovery that using label information together with BERT can lead to significantly better generalization performance.

**Using label information for text classification** An increasing number of recent works have realized the value of label semantics. The matching between label information and text can naturally lead to zero-shot learning. For example, CDSSM (Chen et al., 2016) explores zero-shot intent classifications based on class names. Prompt-based strategies (Puri and Catanzaro, 2019; Schick and Schütze, 2020) have been developed to implicitly match text against class names. In the context of few-shot learning, (Hou et al., 2020) incorporate label semantics into the TapNet (Yoon et al., 2019) for few-shot slot tagging tasks. Different from the above works, this paper explores both pre-trained language models and label semantics for few-shot learning. We only require the name of classes rather than manually constructed prompts or templates to convey label semantics. TARS (Halder et al., 2020) also leverages pre-trained language models and label semantics based on binary text classification. However, our method further strengthens generalization ability via meta-learning framework especially in cross-domain and fine-grained cases.

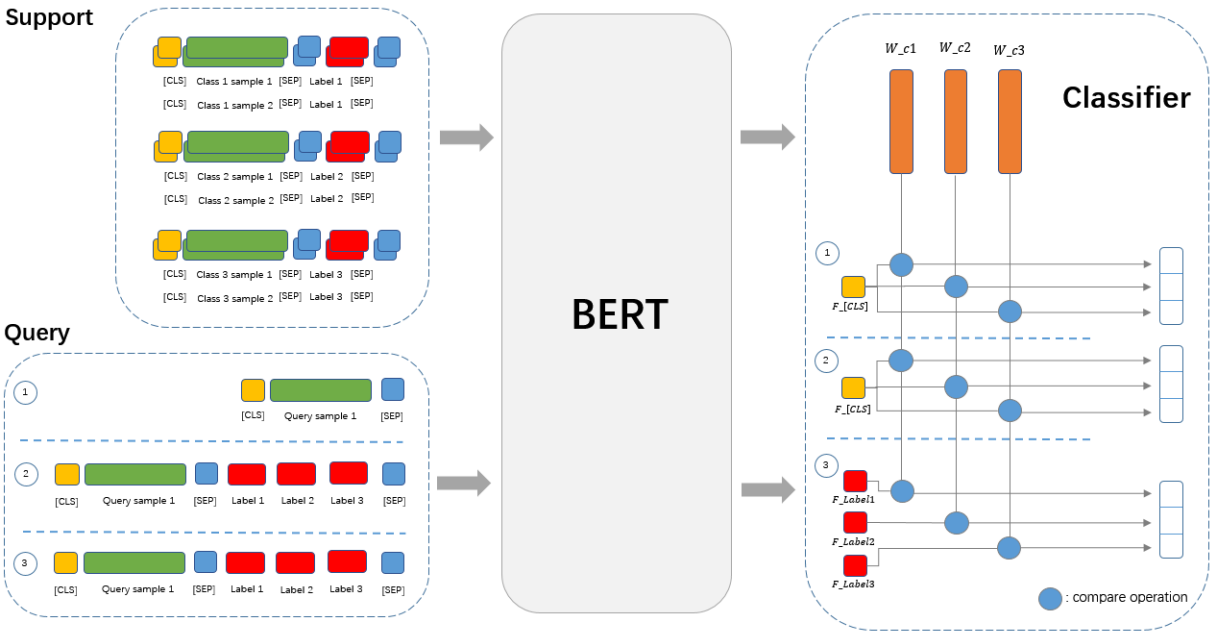


Figure 2: The main framework of LaSAML.This graph gives an example on how LaSAML process a 3-way 2-shot few-shot task. All support samples append with the corresponding class name to format as "[CLS] + Sentence + [SEP] + Label + [SEP]". The query data can be one of three forms: ① remains original form while ② and ③ append all three class names. ① and ② use BERT emebeddings from [CLS] token, ③ uses all label embeddings.

| Appending Word | Number of training samples per class | |
|---|---|---|
| | 5 | 10 |
| None | 0.6425 | 0.7324 |
| class name | 0.7437 | 0.7925 |

Table 1: Results of fine-tuning BERT with only 5 or 10 labeled data per class on the AGNews dataset. None: the standard input format for the BERT classifier, class name: appending the respective class names for each training sample.

## 3 Our method

### 3.1 The value of label information in the low data regime

As described in the introduction, label information is essential for human to accurately interpret the meaning conveyed in the limited number of training samples. In this section, we demonstrate that label information is also useful for extracting discriminative features from a pre-trained language model [1]. More specifically, we consider the following modification to input of BERT for text classification: we append the corresponding class name after each training sentence (for which we know the ground-truth classes) and a [SEP] token. In other words, we use the following input format "[CLS] sentence [SEP] class name [SEP]" rather than "[CLS] sentence [SEP]" as the common practice of using BERT for text classification. Then we extract embeddings from the [CLS] token to train a linear classifier. The reason of appending "class name [SEP]" is to mimic the scenario of the next sentence prediction (NSP) task for training BERT. To perform well in the NSP task, BERT needs to extract information that are most predictive for the next sentence from the first sentence. In our case, we replace the next sentence with the class name and consequently, we expect that BERT can extract information that is relevant to the class name from the input sentence. We call this method label-semantic augmented feature extraction hereafter.

From the experimental results are shown in Table 1, we can clearly see that the classifier trained from label-semantic augmented feature extraction achieves better performance than the baseline approach. When only five samples are used per class, the improvement can be as significant as 10%.

From this motivating experiment, we clearly see the potential of incorporating class-label information. To further strengthen BERT's capability of leveraging class-label semantic information, we incorporate the above idea into the meta-learning framework, lending itself to a new meta-learning framework termed label-semantic augmented meta-

[1]Throughout our experiments, we use BERT (Devlin et al., 2019) as the PLM unless specified otherwise.

learner (LaSAML). We expect that through fine-tuning a PLM by the meta-learning process, the network can find an optimal way of building meta-classifiers with the guidance of class-labels.

### 3.2 The general framework of the label-semantic augmented meta-leaner

We first present the proposed LaSAML in its general form and then dive into more details of this framework. Formally, we consider the following problem setting. Our aim is to build a meta-learner which can convert a set of support samples, denoted as $\mathcal{X}_s = \{x_s, y_s, t_s\}$, into a classifier $\phi(\cdot; \mathcal{X}_s)$, where $x_s$, $y_s$ and $t_s$ denote the input text, class label, and the lexical definition of the class, i.e., the class name, respectively [2] Applying $\phi(\cdot; \mathcal{X}_s)$ to test data, i.e., query data $x_q$, we could obtain the predicted class $\hat{y}_q$ through $\phi(x_q; \mathcal{X}_s)$. The meta-learner is trained from the meta-training set, from which one can randomly construct a support set $\mathcal{X}_s^C = \{x_s^c, y_s^c, t_s^c\}$ and a query set with ground-truth class name, $\mathcal{X}_q^C = \{x_q^c, y_q^c\}$ for C-Way K-shot settings, where $c \in C$. Therefore, the performance of $\phi(\cdot; \mathcal{X}_s^C)$, classifier generated from the meta-learner, can be evaluated by comparing the predicted class against the ground-truth query label $\{y_q^c\}$. The key difference of traditional meta-learner and the proposed LaSAML is that the lexical definition of class name $\{t_s^c\}$ will be used for building the meta-learner.

In particular, we consider the meta-learner that can be written in the following form:

$$\{\mathbf{w}_c\} = \psi(f(\{x_s^c\}), \{y_s^c\}) \;\; c = 1, \cdots, C$$
$$q_c = m(g(x_q); \mathbf{w}_c), \;\; \hat{y} = \underset{c}{argmax}\, q_c, \quad (1)$$

where $f$ and $g$ denotes the feature extractors which convert the input text to a feature vector. For many meta-learning approaches, $f = g$. $\psi$ is a mapping function to map the support set data to a set of class vectors, one for each class. Then the classifier is defined by a function $m(\cdot, \cdot)$ that measures the compatibility, $q_c$, between a query sample $x_q$ and the class vector $\mathbf{w}_c$. The class with highest $q_c$ is the predicted class.

The above formulation encompasses a wide range of meta-learning approaches. For example, for Prototypical network (Snell et al., 2017), $\mathbf{w}_c$ is the $c$-th class mean vector calculated from the

[2] In our following discussion, we slightly relax the distinction between "class label", "class name" and "class tag" and use them exchangeably when no confusion is caused.

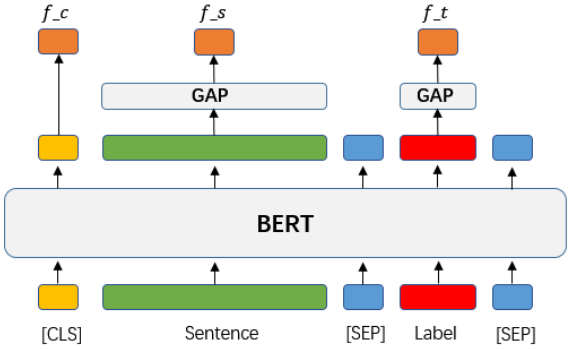


Figure 3: This graph shows how we extract support sample features from various positions. GAP refers to the global average pooling operation.

feature extractor and $m(\cdot, \cdot)$ is simply a negative Euclidean distance between $g(x_q)$ and $\mathbf{w}_c$.

The proposed LaSAML introduces label-semantic guidance to $f(\cdot)$ and $g(\cdot)$. In other words, the feature extractors $f$ and $g$ may take the class name as an additional input. Due to that the availability of class name information will be different for support set samples and query set samples, i.e., we know the ground-truth class name for support set samples but not for query samples, we may choose different ways of incorporating label information into the feature extractor, resulting in different implementation of $f$ and $g$.

### 3.3 Incorporating label information into feature extractors

This subsection discusses various options of incorporating label information into $f$ and $g$. We show the possible configurations in Figure 3. For the feature extractor of the support set, $f$, we append the corresponding ground-truth class name to each sentence. Then we have different options of extracting sentence feature representations. In our study, we consider extracting sentence features from [CLS] token, the global average pooling (GAP) of embeddings of sentence tokens, GAP of embeddings of the class name (since a class name may contain multiple tokens), and the average of them.

For the feature extractor of the query set, $g$. We consider three cases. First, the most straightforward way is not appending anything since we do not know the ground-truth class name for query samples. Second, we can append all class names, as shown in Figure 2. Finally, we can append all class names but extract $C$ features from the corresponding class name, one for each class. Then the $c$-th feature will be compared against the $c$-th

class vector and calculate the matching score with $m$. The class corresponding to the highest matching score will be the prediction. This scheme is visualized in Figure 2 option 3 for query. Formally, this process can be written as:

$$\hat{y}_q = \underset{c}{argmax}\, m(g(x_q, t_c), w_c), \tag{2}$$

where $g(x_q, t_c)$ denote the feature extracted from class name token $t_c$.

We will leave the detailed comparison results and discussion of those schemes to Section 4.2 and Section 4.3. Here, we report our major discovery. (1) For supporting set samples, extracting sentence features from different positions leads to similar performance. Extracting features from "[CLS]" and "[CLS]+Tag" in general is slightly better than other options. (2) For query samples, without appending all class-label names leads to the overall best performance for our best performed method. In the following, we by default consider the setting of extracting features from "[CLS]" and not appending class names to a query sample unless otherwise specified.

### 3.4 Upgrade existing meta-learner with LaSAML

The proposed LaSAML can be incorporated into a variety of existing meta-learning frameworks. In our study, we mainly consider Prototypical Network (Snell et al., 2017) as the meta-learning framework and upgrade it with LaSAML.

The Prototypical Network (Snell et al., 2017) is a metric-based meta-learning framework, which calculates the class vector by averaging the same-class features extracted from the support set. In its LaSAML-upgraded version (denoted as LaSAML-PN), we calculate the class vector $\mathbf{w}_c$ by

$$\mathbf{w}_c = \frac{1}{|\mathcal{X}_s^c|} \sum_{(x_s^c, t_s^c) \in \mathcal{X}_s^c} f(x_s^c, t_s^c), \tag{3}$$

where $f(x_s^c, t_s^c)$ indicates the feature extracted by incorporating class name information.

Then, we make the decision by comparing the feature extracted from a query sample against $\{\mathbf{w}_c\}$:

$$P(c|x_q) = \frac{\exp(-d(g(x_q^c), \mathbf{w}_c))}{\sum_{c' \in C} \exp(-d(g(x_q^c), \mathbf{w}_{c'}))} \tag{4}$$

where $d(\cdot, \cdot)$ is the squared Euclidean distance.

## 4 Experimental results

In this section, we conduct experiments to evaluate the performance of LaSAML. We first introduce our experimental setting. Then, we present the main results by comparing LaSAML against various existing few-shot text classification approaches. Finally, we provide ablation studies to investigate multiple factors in the proposed method.

### 4.1 Experimental Setting

#### 4.1.1 Datasets

Three text classification datasets are used in our experiment.

**HuffPost** is a dataset including a wide range of news topics. The dataset consists of 36900 news headline samples and 900 samples for each class. Following the settings from (Bao et al., 2020), we use the same 20/5/16 classes for training, validation, and testing, respectively, for a fair comparison. Due to the limited number of classes, we only consider the 5-way 1-shot and 5-way 5-shot text classification tasks in this dataset.

**Banking77** published by (Casanueva et al., 2020) is a dataset for intent classification tasks. The dataset covers 13,083 fine-grained intents from 77 classes in the banking domain. We construct the few-shot tasks in 10-way 1-shot, 10-way 5-shot, 15-way 1-shot, and 15-way 5-short. The dataset is partitioned into a training, a validation, and a testing dataset. 30, 15, and 32 classes are sampled for each partition[3].

**Clinc150** is a cross-domain intent classification dataset which was originally proposed in (Larson et al., 2019). It provides 22,500 in-scope queries and 150 intent classes from 10 domains. Each domain contains 15 intent classes, and there is no overlap between those classes. We use this dataset to evaluate the performance of meta-learner under domain shift. We split the datasets into 4/1/5 domains for training, validation, and testing.

#### 4.1.2 Comparing methods

We compare the proposed method against several commonly used few-shot learning approaches, which have shown promising results in both computer vision and natural language processing fields. For all the compared methods expect distributional signature (Bao et al., 2020) which shows better performance without BERT, we re-implement them

[3]We released the partition of Banking77 and Clinc150 along with our code.

| Models | HuffPost | | Banking77 | | | | Clinc150 (cross domain) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 5-way | | 10-way | | 15-way | | 10-way | | 15-way | |
| | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| BERT+PN* | 0.4059 | 0.5348 | 0.6305 | 0.7860 | 0.5918 | 0.7412 | 0.5743 | 0.7290 | 0.5231 | 0.6606 |
| BERT+PN | 0.4611 | 0.6556 | 0.7622 | **0.8883** | 0.7028 | **0.8582** | 0.7130 | 0.8798 | 0.6303 | 0.8163 |
| BERT+RN | 0.4080 | 0.5187 | 0.6388 | 0.7348 | 0.5629 | 0.6457 | 0.5465 | 0.6009 | 0.4654 | 0.5883 |
| BERT+IN | 0.3996 | 0.5079 | 0.4872 | 0.6432 | 0.4945 | 0.5527 | 0.4652 | 0.5765 | 0.4172 | 0.4998 |
| BERT+RRML | 0.4078 | 0.6198 | 0.7045 | 0.8780 | 0.6346 | 0.8565 | 0.6272 | 0.8713 | 0.5761 | 0.8076 |
| DS+RRML | 0.4134 | 0.6248 | 0.5933 | 0.8371 | 0.5337 | 0.7896 | 0.5556 | 0.7876 | 0.5341 | 0.7969 |
| LaSAML-PN | **0.6216** | **0.7011** | **0.8278** | 0.8806 | **0.7877** | 0.8443 | **0.7760** | **0.8831** | **0.7248** | **0.8489** |

Table 2: Experiment results of 5-way 1-shot and 5-way 5-shot on HuffPost Dataset, 10-way 1-shot, 10-way 5shot, 15-way 1-shot and 15-way 5-shot on Banking77 and Clinc150 (cross domain) datasets. The model BERT+PN* contains MLP in PN to process BERT embeddings before applying the distance metric.

| Support Features | LaSAML-PN | | | | LaSAML-RRML | | | |
|---|---|---|---|---|---|---|---|---|
| | HuffPost | | Clinc150 | | HuffPost | | Clinc150 | |
| | 5-Way | | 15-Way | | 5-Way | | 15-Way | |
| | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| CLS | 0.6216 | **0.7011** | 0.7248 | **0.8489** | 0.6260 | 0.6808 | 0.6201 | 0.8051 |
| Sent | 0.6223 | 0.6713 | 0.6834 | 0.8359 | 0.6095 | 0.6781 | 0.5811 | 0.7342 |
| Tag | 0.6264 | 0.6758 | 0.6936 | 0.8199 | 0.5960 | 0.6358 | 0.6470 | 0.7604 |
| CLS + Sent | 0.6250 | 0.6998 | 0.6955 | 0.8303 | **0.6286** | **0.6927** | 0.6160 | **0.8064** |
| CLS + Tag | **0.6325** | 0.6946 | **0.7348** | 0.8275 | 0.6055 | 0.627 | 0.6485 | 0.7952 |
| Sent + Tag | 0.6289 | 0.6964 | 0.6974 | 0.8364 | 0.609 | 0.6488 | 0.6532 | 0.7738 |
| Weighted All | 0.6211 | 0.7003 | 0.6996 | 0.8245 | 0.6022 | 0.6448 | **0.6599** | 0.7938 |

Table 3: Ablation study results of extracting support data features from varies positions and its combinations on HuffPost and Clinc150 (cross domain). According to the results on another ablation study in Table 5, we pick up different query settings ① and ② in Figure 2 for LaSAML-PN and LaSAML-RRML individually.

with the BERT encoder as the feature extractor. Note that this might lead to different (in most cases higher) performance than the one originally reported.

**Prototypical Network** (Snell et al., 2017) is a metric-based few-shot learning method. Our LaSAML-PN is an upgraded version of it. We use two implementations of PN. One extracts features from the [CLS] token, and another applies a multi-layer-perceptron (MLP) for the embedding of [CLS] token. The latter was used in a recent study (Bao et al., 2020). We denote the original implementation of Prototypical Network as PN and the implementation with an MLP as PN*.

**Relation Network** (Sung et al., 2018) (RN) does not directly compare class vector against the query feature but compare through a relation module learned during meta-training.

**Induction Network** (Geng et al., 2019) (IN) integrates the dynamic routing algorithm into the relation network to learn the class vectors during the meta-learning process. The original induction network uses LSTMs as the encoder. To make a fair comparison, we replace the encoder with the BERT encoder in our experiment.

**Ridge Regression Meta-Learner** (Bertinetto et al., 2019) (RRML) calculates the class vector by solving a Ridge regression problem on the support set. The solution has a closed-form, and thus it is possible to directly back-propagate training error to the feature extractor used in the optimization problem.

**Distributional Signature** (Bao et al., 2020) (DS) learns how to use the statistic pattern of tokens to selectively attend key information of the input text and build a meta-learner with better generalization performance. The method in (Bao et al., 2020) can be applied to a wide variety of meta-learning methods. In (Bao et al., 2020), the combination with RRML shows the best performance (DS+RRML).

#### 4.1.3 Implementation Details

In our methods, $\text{BERT}_{\text{BASE}}$ is employed as the feature encoder and meta-learner. We construct 100, 100, and 1000 random sampled tasks for each training, validation, and testing epoch individually.

Moreover, we use the Adam algorithm (Kingma and Ba, 2015) as the optimizer. For better training performance, we set different learning rates for the BERT encoder and the other modules, that is, $2e{-}5$ for the BERT encoder and $1e{-}3$ for other modules. Both Relation Network and Induction Network consist of a relation module, and we set the dense hidden layer dimension to 50 for the relation module. We follow other settings of Induction Network in the original paper (Geng et al., 2019).

### 4.2 Main results of LaSAML

The main experiment results are displayed in Table 1. From the result, we make the following observations: (1) The proposed LaSAML-PN achieves significant performance improvement over the original PN, especially on the one-shot classification setting: on average, the improvement is around 6% to 16%. With more training data, the gap between LaSAML-PN and PN becomes smaller: on Banking77, LaSAML-PN and PN become comparable; but we can still see 3-5% improvement on Clinc150 and HuffPost. This is understandable because, with more samples, the class-related text patterns become more pronounced. However, this might be data-dependent. In general, if the difference between classes is more subtle, i.e., fine-grained classes, more samples might be needed and consequently, the guidance from class name/definition will be more beneficial. (2) We find that the original implementation of the Prototypical network performs much better than the one used in (Bao et al., 2020) which employs an additional MLP. The former achieves even higher performance than the method proposed in (Bao et al., 2020) (which is DS+R2D2. Our re-implementation achieves almost the same performance in (Bao et al., 2020)). (3) Another surprising finding is that the Relation Network and Induction network do not perform better than the traditional Prototypical network. From the above observations, we may conclude that using modules without prior information of a language model, e.g., the MLP whose parameters are randomly initialized rather than pre-trained as in the PLM, leads to poor generalization performance. In contrast, methods directly fine-tuning parameters inside a PLM, e.g., PN, RRML, and our methods, tend to perform better. This observation can somehow be supported by the argument in (Bao et al., 2020). In (Bao et al., 2020), it points out that in NLP, "the lexical features highly informative for one task may be insignificant for another. " Thus, the weight learned from those randomly initialized modules may overfit the meta-training set and cannot generalize well to the target task. In (Bao et al., 2020), the authors suggest a solution by building a meta-learner with the generalizable statistics of words. In our study, we find that this solution might not be stable in all cases. For example, DS+RRML does not perform well in Banking77 and Clinc150. Instead, our results suggest an alternative solution: building the meta-learner by not introducing additional parameters to BERT parameters, since the latter is pre-trained from a large corpus and tends to generalize better across tasks.

| Model | HuffPost | | Clinc150 | |
|---|---|---|---|---|
| | 5-way | | 15-way | |
| | 1-shot | 5-shot | 1-shot | 5-shot |
| BERT+PN | 0.4611 | 0.6556 | 0.6303 | 0.8163 |
| LaSAML-PN | **0.6216** | **0.7011** | **0.7248** | **0.8489** |
| BERT+RRML | 0.4078 | 0.6198 | 0.5761 | **0.8076** |
| LaSAML-RRML | **0.6260** | **0.6808** | **0.6201** | 0.8051 |

Table 4: Ablation study results for integrating LaSAML with RRML and testing performance on HuffPost and Clinc150 (cross domain).

### 4.3 Ablation Study

In this section, We investigate LaSAML in depth by answering three questions. First, whether LaSAML is applicable to other meta-learning frameworks. Second, what is the impact of different ways of extracting features from BERT for support set samples? Third, how to leverage class name information for query samples? We conduct a serial of experiments on HuffPost and Clinc150 datasets to answer those questions.

**LaSAML with other meta-learning framework** To further explore the potential of LaSAML, we incorporate it into the Ridge Regression Meta-learner (RRML) (Bertinetto et al., 2019), which is achieved by simply replacing the feature extractor $f$ and $g$ with the feature extractors used in LaSAML-PN. The results are shown in Table4. As seen, using LaSAML leads to significant improvement in one-shot cases for the RRML. For five-shot cases, the improvement gain becomes smaller for Clinic-150 but still significant in HuffPost. This experiment result suggests that the proposed LaSAML has a great potential to upgrade a wide variety of meta-learning approaches.

**Comparing support set feature extraction strategies** In LaSAML, the input format of a sup-

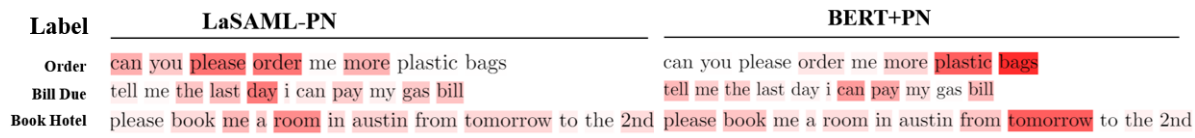


Figure 4: Visualization of BERT attention maps for LaSAML-PN and PN. The darker red color refers to higher attention weight.

port set sample is "[CLS] sentence [SEP] class name [SEP]". As mentioned in Section 3.3, we may extract sentence features from the last layer embedding of the [CLS] token, the average embedding of sentence tokens, the embedding (average embedding) of the class name, and various combination of them. Table 3 provides the comparison of the results. From the results, we can see that no single strategy achieves consistently better performance than the others. Their performance, in most cases, is also similar. Therefore, we extract the sentence feature from the last layer embedding of the [CLS] token for simplicity.

**Comparing query feature extraction strategies** In this section, we further investigate the impact of the input format for query samples. Three configurations, not appending class names and extracting features from [CLS] (L: None F: CLS), appending all class names but extracting features from [CLS] (L: All F: CLS), appending all class names but extracting features from the respective class (L: All F: Tag), and make a prediction by using Eq. 2. We also make our comparison with the LaSAML upgraded PN, or LaSAML-RRML. The experiment results are shown in Table 5. From the results, we can see that the best strategy seems to be method dependent. Appending all class names leads to better performance for LaSAML-RRML, but for LaSAML-PN, the best strategy is not appending any class names. Another observation is that extracting features from the respective class tag and comparing them against the respective class vector may lead to worse performance. However, extracting features from the respective class tag is capable of achieving better performance (or comparable performance on 5-shot classification in Clinc150 dataset) than previous state-of-the-art methods.

## 5 What has been learned in LaSAML

In this section, we demonstrate what has been learned by LaSAML-PN. We use an example in Figure 4 to highlight the difference between LaSAML-PN and the standard prototypical network. By investigating the attention weight with respect to the [CLS] token (we average the attention value across all heads in the last layer of BERT), we can see that the prototypical network fails to attend the words relevant to the class. In contrast, LaSAML-PN successfully attends the relevant keywords.

| | | | HuffPost | | Clinc150 | |
|---|---|---|---|---|---|---|
| | | | 5-way 1-shot | 5-way 5-shot | 15-way 1-shot | 15-way 5-shot |
| **M** | **L** | **F** | | | | |
| LaSAML-PN | None | CLS | 0.6216 | **0.7011** | **0.7248** | **0.8489** |
| LaSAML-PN | All | CLS | 0.6291 | 0.6726 | 0.6962 | 0.8423 |
| LaSAML-PN | All | Tag | **0.6365** | 0.6560 | 0.6680 | 0.7877 |
| LaSAML-RRML | None | CLS | **0.6284** | 0.6631 | 0.5751 | 0.7834 |
| LaSAML-RRML | All | CLS | 0.6260 | **0.6808** | **0.6201** | **0.8051** |
| LaSAML-RRML | All | Tag | 0.5814 | 0.6782 | 0.5599 | 0.7897 |

Table 5: Ablation study results for process query data in three different ways. M refers to the models including: prototypical network, relation network and ridge regression classifier. L refers to whether query samples append all class names or none. F refers to the features used. Here, support data append related class name and use [CLS] token features.

## 6 Conclusion

In this paper, we systematically study the potential of using class name information for few-shot text classification tasks. We identify that appending the class name to the sentence as the input to a BERT encoder can lead to more discriminative sentence features. By adopting this scheme to meta-training, we propose a new meta-learning framework called LaSAML. Implementing this framework with the Prototypical network (Snell et al., 2017), we achieve significant improvement over the existing few-shot text classification methods.

## References

Trapit Bansal, Rishikesh Jha, and Andrew McCallum. 2020. Learning to few-shot learn across diverse natural language classification tasks. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5108–5123, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yujia Bao, Menghua Wu, Shiyu Chang, and Regina Barzilay. 2020. Few-shot text classification with distributional signatures. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. 2019. Meta-learning with differentiable closed-form solvers. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*, pages 38–45, Online. Association for Computational Linguistics.

Yun-Nung Chen, Dilek Hakkani-Tür, and Xiaodong He. 2016. Zero-shot learning of intent embeddings for expansion by convolutional deep structured semantic models. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016, Shanghai, China, March 20-25, 2016*, pages 6045–6049. IEEE.

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.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135. PMLR.

Ruiying Geng, Binhua Li, Yongbin Li, Xiaodan Zhu, Ping Jian, and Jian Sun. 2019. Induction networks for few-shot 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 3904–3913, Hong Kong, China. Association for Computational Linguistics.

Kishaloy Halder, Alan Akbik, Josip Krapac, and Roland Vollgraf. 2020. Task-aware representation of sentences for generic text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3202–3213.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

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

Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. 2015. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille.

Stefan Larson, Anish Mahendran, Joseph J. Peper, Christopher Clarke, Andrew Lee, Parker Hill, Jonathan K. Kummerfeld, Kevin Leach, Michael A. Laurenzano, Lingjia Tang, and Jason Mars. 2019. An evaluation dataset for intent classification and out-of-scope prediction. 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 1311–1316, Hong Kong, China. Association for Computational Linguistics.

Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. 2018. A simple neural attentive meta-learner. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Raul Puri and Bryan Catanzaro. 2019. Zero-shot text classification with generative language models. *arXiv preprint arXiv:1912.10165*.

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.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. 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 3856–3866.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *arXiv preprint arXiv:2001.07676*.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. 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 4077–4087.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 1199–1208. IEEE Computer Society.

Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 3630–3638.

Sung Whan Yoon, Jun Seo, and Jaekyun Moon. 2019. Tapnet: Neural network augmented with task-adaptive projection for few-shot learning. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7115–7123. PMLR.

Mo Yu, Xiaoxiao Guo, Jinfeng Yi, Shiyu Chang, Saloni Potdar, Yu Cheng, Gerald Tesauro, Haoyu Wang, and Bowen Zhou. 2018. Diverse few-shot text classification with multiple metrics. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1206–1215, New Orleans, Louisiana. Association for Computational Linguistics.