Label-Driven Denoising Framework for Multi-Label Few-Shot Aspect Category Detection

Fei Zhao1∗ Yuchen Shen2∗ Zhen Wu1† Xinyu Dai1

1National Key Laboratory for Novel Software Technology, Nanjing University
2School of Information and Software Engineering, University of Electronic Science and Technology of China
zhaof@smail.nju.edu.cn, alexchicharitoshen@gmail.com
{wuz, daixinyu}@nju.edu.cn

Abstract

Multi-Label Few-Shot Aspect Category Detection (FS-ACD) is a new sub-task of aspect-based sentiment analysis, which aims to detect aspect categories accurately with limited training instances. Recently, dominant works use the prototypical network to accomplish this task, and employ the attention mechanism to extract keywords of aspect category from the sentences to produce the prototype for each aspect. However, they still suffer from serious noise problems: (1) due to lack of sufficient supervised data, the previous methods easily catch noisy words irrelevant to the current aspect category, which largely affects the quality of the generated prototype; (2) the semantically-close aspect categories usually generate similar prototypes, which are mutually noisy and confuse the classifier seriously. In this paper, we resort to the label information of each aspect to tackle the above problems, along with proposing a novel Label-Driven Denoising Framework (LDF). Extensive experimental results show that our framework achieves better performance than other state-of-the-art methods. Our code and datasets are available at https://github.com/1429904852/LDF.

1 Introduction

Aspect Category Detection (ACD) is an important subtask of fine-grained sentiment analysis (Pontiki et al., 2014), which aims to detect the aspect categories mentioned in a review sentence from a predefined set of aspect categories. For example, given the sentence “The service is good although rooms are pretty expensive.”, the ACD task is to detect two aspect categories from the sentence, respectively service and price. Obviously, the ACD belongs to a multi-label classification problem.

Recently, with the development of deep learning technique, a great number of neural models have been proposed for the ACD task (Zhou et al., 2015; Schouten et al., 2018; Hu et al., 2019). The performance of all these models heavily rely on sufficient labeled data. However, the annotation of aspect categories in ACD is extremely expensive. The limited labeled data restrict the effectiveness of neural models. To alleviate the issue, Hu et al. (2021) refer to few-shot learning (FSL) (Ravi and Larochelle, 2017; Finn et al., 2017; Snell et al., 2017; Gao et al., 2019) and formalize ACD as a few-shot ACD (FS-ACD) problem, learning aspect categories with limited supervised data.

FS-ACD follows the meta-learning paradigm (Vinyals et al., 2016) and builds a collection of N-way K-shot meta-tasks. Table 1 shows a 3-way 2-shot meta-task, which consists of a support set and a query set. The support set samples three classes (i.e., aspect categories), and each class selects two sentences (instances). A meta-task aims to infer the classes of sentences in the query set with the help of the small labeled support set. By sampling different meta-tasks in the training stage, FS-ACD can learn great generalization ability in few-shot scenario and works well in the testing stage. To perform the FS-ACD task, Hu et al. (2021) proposes an

<table>
<thead>
<tr>
<th>Support set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect Category</td>
</tr>
<tr>
<td>(A) food_food_meat_burger</td>
</tr>
<tr>
<td>(B) food_meattype_lunch</td>
</tr>
<tr>
<td>(C) restaurant_location</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Query set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aspect Category</td>
</tr>
<tr>
<td>(B)</td>
</tr>
<tr>
<td>(A) and (C)</td>
</tr>
</tbody>
</table>

Table 1: An example of 3-way 2-shot meta-task. A sentence (instance) may belong to multiple aspects.
attention-based prototypical network Proto-AWATT. It first exploits an attention mechanism (Bahdanau et al., 2015) to extract keywords from the sentences corresponding to aspect category in the support set, and then aggregate them as evidence to generate a prototype for each aspect. Next, the query set utilizes the prototypes to generate corresponding query representations. Finally, the prediction is made by measuring the distance between each prototype representation and corresponding query representation in the embedding space.

Though achieving impressive progress, we find the noise is still a crucial problem for the FS-ACD task. The reason comes from two folds. On the one hand, the previous models easily catch noisy words irrelevant to the current aspect category due to the lack of sufficient supervised data, which largely affects the quality of the generated prototype. As shown in Figure 1, take the prototype of aspect category food_food_meat_burger as an example. We highlight its top-10 words based on attention weights of Proto-AWATT. Because of lacking sufficient supervised data, we observe the model tends to focus on these common but noisy1 words, such as “a”, “the”, “my”. These noisy words fail to produce a representative prototype for each aspect, resulting in the discounted performance. On the other hand, the semantically-close aspect categories usually produce similar prototypes, these close prototypes are mutually noisy and confuse the classifier greatly. According to the statistics, nearly 25% of aspect category pairs in the benchmark dataset have similar semantics, such as food_food_meat_burger and food_meattype_lunch in Table 1. Apparently, the prototypes generated by these semantically-close aspect categories can interfere with each other and confuse the detection results of FS-ACD seriously.

To tackle the above issues, we propose a novel Label-Driven Denoising Framework (LDF) for the FS-ACD task. Specifically, for the first issue, the label text of aspect category contains rich semantics describing the concept and scope of aspect, such as the text “restaurant location” for the aspect restaurant_location, which intuitively help the attention capture label-relevant words better. Therefore, we propose a label-guided attention strategy to filter noisy words and guide LDF to yield better aspect prototypes. Given the second issue, we propose an effective label-weighted contrastive loss, which incorporates inter-class relationships of support set into a contrastive objective function, thereby enlarging the distance among similar prototypes.

Our contributions are summarized as follows:

- To the best of our knowledge, we are the first to exploit the label information of each aspect to address noise problems in the FS-ACD task.
- We propose a novel Label-Driven Denoising Framework (LDF), which contains a label-guided attention strategy to filter noisy words and generate a representative prototype for each aspect, and a label-weighted contrastive loss to avoid generating similar prototypes for semantically-close aspect categories.
- The LDF framework has good compatibility and can be easily extended to existing models. In this work, we apply it to two latest FS-ACD models, Proto-HATT (Gao et al., 2019) and Proto-AWATT (Hu et al., 2021). Experimental results on three benchmark datasets prove the superiority of our framework.

2 Notations and Background

In this section, we first present the task formalization of FS-ACD and then give brief introductions to the background.

2.1 Task Formalization

The FS-ACD task follows the meta-learning paradigm (Vinyals et al., 2016). Specifically, given labeled instances from a set of classes (i.e., aspect categories) C_train, the goal is to acquire knowledge from C_train, and use the knowledge to recognize...
novel classes, which have only a few labeled instances. These novel classes belong to a set of classes $C_{\text{test}}$ and disjoint from $C_{\text{train}}$.

To emulate the few-shot scenario, meta-learning algorithms learn from a group of $N$-way $K$-shot meta-tasks sampled from $C_{\text{train}}$. Within each meta-task, we randomly select $N$ classes ($N$-way) from $C_{\text{train}}$, each with $K$ instances ($K$-shot) to form a support set $S = \{s^n_k | k = 1, ..., K\}^N_n$. Meanwhile, $M$ instances are sampled from the remaining data of the $N$ classes to construct a query set $Q = \{(q_i, y_i) | y_i \in \mathbb{R}^N\}_i^M$, where $y_i$ is a binary label vector whose $n$-th bit is set to 1 if $q_i$ belongs to the $n$-th class (i.e., aspect category), 0 otherwise. A meta-task aims to infer the class(es) of query instance $q_i$ in $Q$ according to a small labeled support set $S$. By sampling different meta-tasks in the training stage, FS-ACD can learn great generalization ability. During the testing stage, we apply the same manner to test whether our model can adapt quickly to novel classes within $C_{\text{test}}$.

2.2 Background

In this work, we abstract a general attention architecture based on the Proto-AWATT (Hu et al., 2021) and Proto-HATT (Gao et al., 2019) models, which both achieve satisfying performance and thus are chosen as the foundations of our work.

Given an instance $x^n_k = \{w_1, w_2, ..., w_I\}$ consisting of $I$ words, we first map it into an word sequence $e^n_k = \{e_1, e_2, ..., e_I\}$ by looking up an embedding table. And then, we apply a convolutional neural network (CNN) (Zeng et al., 2014; Gao et al., 2019) to encode the word sequence into a contextual representation $H^n_k$. Next, an attention layer assigns a weight $\beta$ to each word in the instance. The final instance representation is given by:

$$\beta = \text{ATT}_W(H^n_k),$$

$$r^n_k = \beta H^n_k,$$  

where $H^n_k$ is the $k$-th instance representation of the class $n$ in the support set $S$, $\text{ATT}_W(\cdot)$ denotes an attention mechanism. After that, we aggregate all instance representations for the class $n$ to produce the prototype:

$$r^n = \text{Aggregation}(r^n_1, ..., r^n_K),$$

where $\text{Aggregation}(\cdot)$ denotes the attention mechanism or average pooling operation. After processing all classes in the support set $S$, we obtain $N$ prototypes $\{r^1, r^2, ..., r^n, ..., r^N\}$.

Similarly, for a query instance $q_i$, we first encode $q_i$ to obtain its contextual representation, and then exploit an attention mechanism to produce $N$ prototype-specific query representations $r^n_i$ based on the $N$ prototypes. After that, we compute the Euclidean distance (ED) between each prototype and the corresponding prototype-specific query representation. Finally, we normalize the negative Euclidean distances to obtain the ranking of prototypes and use a threshold to select the positive predictions (i.e., aspect categories).

$$\hat{y}_i = \text{softmax}(-\text{ED}(r^n, r^n_i)), n \in [1, N]$$

The training objective is the mean square error (MSE) loss as follows:

$$L_{\text{mse}} = \sum_{i=1}^M (\hat{y}_i - y_i)^2$$

3 Label-Driven Denoising Framework

Figure 2 shows the overall architecture of LDF, which contains two components: Label-guided Attention Strategy and Label-weighted Contrastive Loss. With the aid of label information, the former can focus on the class-relevant words better, thus producing a more accurate prototype for each class, the latter utilizes the inter-class relationships of support set to avoid generating similar prototypes.

3.1 Label-guided Attention Strategy

Due to lack of sufficient supervised data, the attention weights $\beta$ in Equation 1 usually focus on some noisy words irrelevant to the current class (i.e., aspect category), resulting in the prototype in Equation 3 becoming unrepresentative.

Intuitively, the label text of each class contains rich semantics, which can provide guidance for capturing class-relevant words. Thus, we leverage label information to tackle the above problem and propose a Label-guided Attention Strategy.

Specifically, we first locate the keywords of each class by calculating the semantic similarity between the label and each word in the instance:

$$\alpha = \cos(L^n, e^n_k),$$

where $L^n$ is the label embedding of class $n$ in the support set and calculated by averaging the multiple word embeddings of each class (e.g., $\text{food}_1\text{food}_2\text{meat}_3\text{burger}_4$), $e^n_k$ is the word embedding of instance $s^n_k$, $\cos(\cdot)$ is the cosine function.
Under the constraints of label information, the similarity weight $\alpha$ tends to focus on the limited words\(^3\) highly relevant to the label text and may neglect other informative words. Thus, we take it as the complementary information of the attention weights $\beta$ to generate a more comprehensive and accurate attention weight $\theta$. Formally,

$$\theta = W_\theta [\alpha; \beta] + b_\theta \quad (7)$$

where $W_\theta$ and $b_\theta$ are weight matrices and bias, $[\cdot; \cdot]$ denotes the concatenation operation.

Then, to regain the probabilistic attention distribution, the attention weight $\theta$ is re-normalized:

$$\tilde{\theta} = \text{softmax}(\theta) \quad (8)$$

Finally, we replace $\beta$ in Equation 1 with the new attention vector $\tilde{\theta}$ to obtain a representative prototype for each class in the support set.

### 3.2 Label-weighted Contrastive Loss

As mentioned before, the semantically-close aspect categories often generate similar prototypes in the support set, which are mutually noisy and confuse the classifier seriously.

Intuitively, a feasible and natural approach is to leverage supervised contrastive learning (CL)\(^3\) (Khosla et al., 2020), which can push the prototype of different classes away as follows:

$$\mathcal{L}_{scl} = \sum_{(n,k) \in \mathcal{P}(n,k)} \frac{-1}{|\mathcal{P}(n,k)|} \sum_{r_p^m \in \mathcal{P}(n,k)} \log \frac{\exp(r_p^m \cdot r_p^m / \tau)}{\sum_{r_p^m \in \mathcal{P}(n,k) \setminus (n,k)} \exp(r_p^m \cdot r_p^m / \tau)} \quad (9)$$

where $\mathcal{P}(n,k)$ is the positive set of $r_p^m$ in Equation 2, which contains all the other samples (e.g., $r_p^m$) of the same class with $r_p^m$ in the support set. The rest of the $(N-1) \times K$ samples in the support set belong to the negative set, where $r_p^m$ is one negative sample from class $m$. $\tau$ is a temperature parameter.

However, the supervised CL does not well-resolve our problem since it treats different prototypes equally in the negative set, while our goal is to encourage the more similar prototypes to be farther apart. For example, “food_food_meat_burger” is semantically closer to “food_mealtype_lunch” than “room_bed”. Thus, “food_food_meat_burger” should be farther from “food_mealtype_lunch” than “room_bed” in the negative set.

To achieve this goal, we again leverage the label information and propose to incorporate inter-class relationships into the supervised CL to adaptively distinguish similar prototypes in the negative set:

$$\mathcal{L}_{ldt} = \sum_{(n,k) \in \mathcal{P}(n,k)} \frac{-1}{|\mathcal{P}(n,k)|} \sum_{r_p^m \in \mathcal{P}(n,k)} \log \frac{\exp(r_p^m \cdot r_p^m / \tau)}{\sum_{r_p^m \in \mathcal{P}(n,k) \setminus (n,k)} \exp(r_p^m \cdot r_p^m / \tau) w_{mn} \cdot \exp(r_p^m \cdot r_p^m / \tau)} \quad (10)$$

where $w_{mn}$ denotes the cos similarity between different classes in the negative set and is computed.
as follows:

\[ w_{mn} = \cos(L^m, L^n), \]  

(11)

where \( L^m \) and \( L^n \) are the label embedding of the class \( m \) and \( n \). The final loss is formulated as:

\[ \mathcal{L} = \mathcal{L}_{\text{inse}} + \lambda \mathcal{L}_{\text{lcl}} \]  

(12)

where \( \lambda \) is a hyper-parameter that measures the importance of \( \mathcal{L}_{\text{lcl}} \) and can be adjusted.

4 Experimental Settings

4.1 Datasets and Implementation Details

To evaluate the effect of our framework, we carry out experiments on three datasets FewAsp(single), FewAsp(multi), and FewAsp from (Hu et al., 2021), which share the same 100 aspects, with 64 aspects for training, 16 aspects for validation and 20 aspects for testing. It is notable that a sentence may belong to a single aspect or multiple aspects. FewAsp(single), FewAsp(multi), and FewAsp are composed of single-aspect, multi-aspect, and both types of sentences, respectively. General information for three datasets is presented in Table 2.

In each dataset, we construct four FS-ACD tasks, where \( N = 5, 10 \) and \( K = 5, 10 \). And the number of query instances per class is 5. All the models are implemented by the TensorFlow framework with an NVIDIA Tesla V100 GPU. The hyperparameters and training details are given in Appendix A.1.

4.2 Evaluation Metrics

Following (Hu et al., 2021), we use Macro-F1 and AUC scores as our evaluation metrics, and the thresholds in the 5-way setting and 10-way setting are set to \{0.3, 0.2\}, respectively. Besides, the paired \( t \)-test is conducted to test the significance of different approaches. Finally, we report the average performance and standard deviation over 5 runs, where the seeds are set to \{5, 10, 15, 20, 25\}, as with the previous study (Hu et al., 2021).

4.3 Compared Methods

Following (Hu et al., 2021), we chose some frequently-used baselines: Matching Network (Vinyals et al., 2016), Prototypical Network (Snell et al., 2017), Relation Network (Sung et al., 2018), Graph Network (Satorras and Estrach, 2018), IMP (Allen et al., 2019), Proto-HATT (Gao et al., 2019) and Proto-AWATT (Hu et al., 2021).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#cls.</th>
<th>#inst./cls.</th>
<th>#inst.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FewAsp(single)</td>
<td>100</td>
<td>200</td>
<td>20000</td>
</tr>
<tr>
<td>FewAsp(multi)</td>
<td>100</td>
<td>400</td>
<td>40000</td>
</tr>
<tr>
<td>FewAsp</td>
<td>100</td>
<td>630</td>
<td>63000</td>
</tr>
</tbody>
</table>

Table 2: Statistics of three datasets. #cls. is the number of classes. #inst. is the total number of instances. #inst./cls. is the number of instances per class.

To verify the superiority of the LDF framework, we chose two dominant models with the best performance as the foundations of our work, i.e., Proto-HATT and Proto-AWATT. Finally, we integrate LDF into Proto-HATT and Proto-AWATT to obtain the model LDF-HATT and LDF-AWATT.

5 Results and Discussion

5.1 Main Results

The main experiment results are shown in Table 3. From this table, we can see that: (1) LDF-HATT and LDF-AWATT consistently outperform their base models on three datasets. It is worth mentioning that LDF-HATT at most obtains 5.62% and 1.32% improvements in Macro-F1 and AUC scores. In contrast, LDF-AWATT outperforms Proto-AWATT by 3.17% and 1.30% at most. These results reveal that our framework has good compatibility; (2) It is a fact that the Macro-F1 of LDF-AWATT is improved by about 2% in most settings, while that of LDF-HATT is improved by about 3% on average. This is consistent with our expectations since the original Proto-AWATT has a more powerful performance; (3) LDF-HATT and LDF-AWATT perform better on the FewAsp(multi) dataset than on the FewAsp(single) dataset. A possible reason is that each class in the FewAsp(multi) dataset contains more instances, which allows LDF-HATT and LDF-AWATT to generate a more accurate prototype in multi-label classification.

5.2 Ablation Study

Without loss of generality, we choose LDF-AWATT model for the ablation study to investigate the effects of different components in LDF.\(^3\)

Effect of Label-Driven Denoising Framework. We study the two main components of LDF: Label-guided Attention Strategy (LAS) and Label-weighted Contrastive Loss (LCL). Based on the

\(^3\)Due to space limitations, we report the ablation results of LDF-HATT in Appendix A.3.
## 5.3 Discussion

### Effect of Encoder

We also conduct experiments (shown in Table 5) using the pre-trained BERT model (Devlin et al., 2019). Concretely, we replace the Supervised Contrastive Loss (SCL) to see the contribution of label. It can be seen from Table 4 that: (1) Proto-AWATT+SCL performs slightly better than Proto-AWATT on FewAsp dataset, but their results are much lower than Proto-AWATT+LCL. These results further highlight the effectiveness of LCL; (2) After integrating inter-class relationships into Proto-AWATT+SCL, Proto-AWATT+LCL achieve better performance, which indicates that the inter-class relationships play a crucial role in distinguishing similar prototypes.

### Analysis of Label in Contrastive Loss

We compare Label-weighted Contrastive Loss (LCL) with the Supervised Contrastive Loss (SCL) to see the contribution of label. It can be seen from Table 4 that: (1) Proto-AWATT+SCL performs slightly better than Proto-AWATT on FewAsp dataset, but their results are much lower than Proto-AWATT+LCL. These results further highlight the effectiveness of LCL; (2) After integrating inter-class relationships into Proto-AWATT+SCL, Proto-AWATT+LCL achieve better performance, which indicates that the inter-class relationships play a crucial role in distinguishing similar prototypes.
the Glove+CNN encoder with BERT and keep the other components the same as our original model. It’s clear that LDF-AWATT and LDF-HATT perform remarkably well than the base model Proto-AWATT and Proto-HATT on all encoders, which proves that our framework has good scalability.

Effect of Label Similarity Weight \( \alpha \). To illustrate the role of the similarity weight \( \alpha \), we directly replace the attention weight \( \beta \) in Equation 1 with the similarity weight \( \alpha \) in Equation 6, and name this method as Proto-AWATT(LSW). From the results in Table 6, we can see that the performance of Proto-AWATT(LSW) is far inferior to Proto-AWATT, which implies that the similarity weight only plays a supporting role to the attention weight, and cannot be treated independently for the FS-ACD task.

Effect of hyper-parameter \( \lambda \). We tune the hyper-parameter \( \lambda \) on the development set of each dataset, and then evaluate the performance of LDF-AWATT on the test set. Specifically, we conduct experiments for values set at 0.1 intervals in the range (0, 1). Figure 3 shows the performance of LDF-AWATT with different \( \lambda \) on three dataset. Actually, as \( \lambda \) increases, the performance of LDF-AWATT has an initial upward trend, and then flattens out or begins to fall. In the upward part, the Label-weighted Contrastive Loss (LCL) is useful guidance to help the LDF-AWATT distinguish similar prototypes more accurately, thus improving the performance. However, once the weight \( \lambda \) exceeds 0.2, the LCL begins to dominate and performs poorly. The reason behind this may be that the bigger \( \lambda \) has a negative effect on the MSE loss of the model. Therefore, we set \( \lambda \) to be 0.2 on three datasets. In addition, we find that the best results of the development set and test set are basically consistent, which indicates that our framework has good robustness.

5.4 Case Study
To better understand the advantage of our framework, we select some samples from FewAsp dataset for a case study. Specifically, we randomly sample 5 classes and then sample 50 times of 5-way 5-shot meta-tasks for the five classes. Finally for each class, we obtain 50 prototype vectors.

Table 6: The effect of label similarity weight \( \alpha \) in the 10-way 5-shot scenario on FewAsp dataset.

<table>
<thead>
<tr>
<th>Models</th>
<th>GloVe + CNN</th>
<th>BERT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
<td>AUC</td>
</tr>
<tr>
<td>Proto-HATT</td>
<td>57.26</td>
<td>90.63</td>
</tr>
<tr>
<td>LDF-HATT</td>
<td>60.68±0.92</td>
<td>91.22±0.53</td>
</tr>
<tr>
<td></td>
<td>63.72±0.27</td>
<td>91.99±0.12</td>
</tr>
<tr>
<td>Proto-AWATT</td>
<td>65.65</td>
<td>92.06</td>
</tr>
<tr>
<td>LDF-AWATT</td>
<td>67.13±0.41</td>
<td>92.74±0.12</td>
</tr>
<tr>
<td></td>
<td>72.76±0.29</td>
<td>95.31±0.19</td>
</tr>
</tbody>
</table>

Table 5: The effect of different encoders in the 10-way 5-shot scenario on FewAsp dataset. The results with symbol \( ♣ \) are retrieved from (Hu et al., 2021).

<table>
<thead>
<tr>
<th>Models</th>
<th>10-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>F1</td>
</tr>
<tr>
<td>Proto-AWATT</td>
<td>65.65</td>
</tr>
<tr>
<td>Proto-AWATT (LSW)</td>
<td>57.84±0.49</td>
</tr>
</tbody>
</table>
concentrated than those by Proto-AWATT. Besides, in contrast to Proto-AWATT, Proto-AWATT+LAS can focus on class-relevant words better (shown in Figure 1 and Figure 4). These observations suggest that Proto-AWATT+LAS can indeed generate a more accurate prototype for each class.

Proto-AWATT+LAS vs. LDF-AWATT. As depicted in Figure 5(b) and 5(c), after incorporating LCL into Proto-AWATT+LAS, the prototype representation of “food_mealtype_lunch” and “food_food_meat_burger” learned by LDF-AWATT are more separable than those by Proto-AWATT+LAS. This reveals that LCL can indeed distinguish similar prototypes.

5.5 Error Analysis

We present the error analysis in the Appendix A.4.

6 Related Work

Aspect Category Detection. Previous works formulate ACD in a data-driven scenario, and can be generally divided into two kinds: one is unsupervised approach, which detects aspect categories by exploiting semantic association (Su et al., 2006) or co-occurrence frequency (Hai et al., 2011; Schouten et al., 2018); the other is supervised approach, which uses hand-crafted features (Kiritchenko et al., 2014), learns useful representations automatically (Zhou et al., 2015), adopts a multi-task learning strategy (Hu et al., 2019), or utilizes a topic-attention model (Movahedi et al., 2019) to address the ACD task. However, the above methods heavily rely on large-scale training data, which is time-consuming to annotate.

Multi-Label Few-Shot Learning. In comparison with single-label FSL, multi-label FSL is more difficult and less explored, as it aims to identify multiple labels for an instance. Rios and Kavuluru (2018) propose few-shot learning methods for multi-label text classification over a structured label space. Further research on multi-label FSL are developed on image synthesis (Alfassy et al., 2019), signal processing (Cheng et al., 2019), and intent detection (Hou et al., 2021). Recently, Hu et al. (2021) formalize aspect category detection in a multi-label few-shot scenario to alleviate the dependency on large-scale labeled data. However, Hu et al. (2021) ignore the label information of each class, which is crucial for generating a representative prototype in the FS-ACD task.

Contrastive Learning. Contrastive Learning is a representation learning technique and has proven its effectiveness in the field of natural language processing (Gunel et al., 2021; Kim et al., 2021; Ye et al., 2021). With the help of label information, Khosla et al. (2020) propose supervised contrastive learning, which aims to improve the quality of learnt representations in a supervised setting. Different from their work, we do not treat label information equally and propose a label-weighted contrastive loss to distinguish similar prototypes.

7 Conclusion

In this paper, we propose a novel Label-Driven Denoising Framework (LDF) to alleviate the noise problems for the FS-ACD task. Specifically, we design two reasonable components: Label-guided Attention Strategy and Label-weighted Contrastive Loss, which aim to produce a better prototype for each class and distinguish similar prototypes. Results from numerous experiments indicate that our framework LDF achieves better performance than other state-of-the-art methods.
Limitations

We consider two major limitations in the FS-ACD task that need to be addressed in current research and related fields: (1) Existing studies for few-shot learning (FSL) require both the training and testing data have the same number of classes (denoted as $N$-way) and the same number of instances in each class (denoted as $K$-shot) in the support set. However, little investigation has been done towards inconsistent classes and inconsistent instances per class during training and testing. As far as we know, inconsistent FSL is more realistic and meaningful, which may be extremely helpful in low-resource scenarios; (2) The FS-ACD models usually give incorrect predictions when a sentence belongs to more than four aspect categories. A possible reason is that these sentences account for a small proportion of the dataset. Thus, it is also important to find effective methods to tackle the long-tail problem in multi-label classification. In general, the above limitations are of practical meaning and need us to do further research and exploration.

Acknowledgements

We would like to thank the anonymous reviewers for their constructive comments. This work was supported by the National Natural Science Foundation of China (No. 61936012 and 61976114).

References


A Appendices

A.1 Implementation Details

Hyperparameters. We initialize word embedding with 50-dimension Glove vectors. All other parameters are initialized by sampling from a normal distribution \( \mathcal{N}(0, 0.1) \). The hyper-parameter \( \lambda \) is set to 0.2 on three datasets. The dimension of the hidden state is set to 50. The convolutional window size is set as 3. The optimizer is Adam with a learning rate \( 10^{-3} \) and the temperature \( \tau \) is set to 0.1. In each dataset, we construct four FS-ACD tasks, where \( N = 5, 10 \) and \( K = 5, 10 \). And the number of query instances per class is 5. For example, in a 5-way 10-shot meta-task, there are \( 5 \times 10 = 50 \) instances in the support set and \( 5 \times 5 = 25 \) instances in the query set.

Training Details. During training, we train each model for a fixed 30 epochs, and then select the model with the best AUC score on the development set. Finally, we evaluate its performance on the test set. In every epoch, we randomly sample 800 meta-tasks for training. The number of meta-tasks during validation and testing are both set as 600. Besides, we employ an early stop strategy if the AUC score of the validation set is not improved in 3 epochs. For all baselines and our model, we report the average testing results from 5 runs, where the seeds are set to \([5, 10, 15, 20, 25]\). All the models are implemented with the best AUC score on the development set. Finally, we evaluate its performance on the test set.

We construct four FS-ACD tasks, where \( N = 5, 10 \) and \( K = 5, 10 \). And the number of query instances per class is 5. For example, in a 5-way 10-shot meta-task, there are \( 5 \times 10 = 50 \) instances in the support set and \( 5 \times 5 = 25 \) instances in the query set.

We construct four FS-ACD tasks, where \( N = 5, 10 \) and \( K = 5, 10 \). And the number of query instances per class is 5. For example, in a 5-way 10-shot meta-task, there are \( 5 \times 10 = 50 \) instances in the support set and \( 5 \times 5 = 25 \) instances in the query set.

![Table](https://example.com/table.png)

Table 7: Test Macro-F1 and AUC score on the FewAsp, FewAsp(single), and FewAsp(multi) datasets (%) . The baseline results are retrieved from (Hu et al., 2021). We report the average performance and standard deviation over 5 runs, the thresholds in the 5-way setting and 10-way setting are set to \([0.3, 0.2]\). Best results are in bold. The marker \( \dagger \) refers to significant test p-value \(< 0.05 \) when comparing with Proto-HATT and Proto-AWATT.

![Diagram](https://example.com/diagram.png)
A.2 Main Result

As shown in Table 7, we list all the frequently-used baselines and our enhanced version. It is clear that Proto-HATT and Proto-AWATT consistently outperform other baselines, thus we chose them as the foundation of our work. Besides, we observe that our framework achieves better performance compared to all the baselines.

A.3 Ablation Study

In Table 8 and 9, we present the ablation results of LDF-HATT and LDF-AWATT in details.

<table>
<thead>
<tr>
<th>Models</th>
<th>5-way 5-shot F1</th>
<th>AUC</th>
<th>5-way 10-shot F1</th>
<th>AUC</th>
<th>10-way 5-shot F1</th>
<th>AUC</th>
<th>10-way 10-shot F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proto-HATT</td>
<td>70.26</td>
<td>91.54</td>
<td>75.24</td>
<td>93.43</td>
<td>57.26</td>
<td>90.63</td>
<td>61.51</td>
<td>92.86</td>
</tr>
<tr>
<td>Proto-HATT+LAS</td>
<td>73.02±0.69</td>
<td>92.56±0.37</td>
<td>78.09±0.90</td>
<td>94.16±0.29</td>
<td>60.10±1.24</td>
<td>90.95±0.62</td>
<td>65.95±1.39</td>
<td>93.88±0.52</td>
</tr>
<tr>
<td>Proto-HATT+LCL</td>
<td>72.77±0.75</td>
<td>92.43±0.88</td>
<td>77.42±0.46</td>
<td>94.04±0.20</td>
<td>59.85±1.37</td>
<td>90.89±0.74</td>
<td>65.05±0.70</td>
<td>93.48±0.14</td>
</tr>
<tr>
<td>Proto-HATT+SCL</td>
<td>71.56±1.07</td>
<td>91.85±0.53</td>
<td>76.20±0.42</td>
<td>93.63±0.18</td>
<td>58.51±0.56</td>
<td>90.85±0.45</td>
<td>62.86±0.68</td>
<td>93.27±0.28</td>
</tr>
</tbody>
</table>

Table 8: Ablation study over two main components of LDF-HATT. Besides, we also report the ablation result of Proto-HATT+LCL. We report the average performance and standard deviation over 5 runs.

<table>
<thead>
<tr>
<th>Models</th>
<th>5-way 5-shot F1</th>
<th>AUC</th>
<th>5-way 10-shot F1</th>
<th>AUC</th>
<th>10-way 5-shot F1</th>
<th>AUC</th>
<th>10-way 10-shot F1</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>FewAsp</td>
<td>83.33</td>
<td>96.45</td>
<td>86.71</td>
<td>97.62</td>
<td>73.42</td>
<td>95.71</td>
<td>77.65</td>
<td>97.00</td>
</tr>
<tr>
<td>FewAsp(single)</td>
<td>83.33</td>
<td>96.45</td>
<td>86.71</td>
<td>97.62</td>
<td>73.42</td>
<td>95.71</td>
<td>77.65</td>
<td>97.00</td>
</tr>
<tr>
<td>Proto-HATT+LAS</td>
<td>83.96±0.23</td>
<td>96.92±0.27</td>
<td>87.80±0.12</td>
<td>98.12±0.30</td>
<td>75.82±0.49</td>
<td>96.15±0.14</td>
<td>79.90±1.04</td>
<td>97.24±0.24</td>
</tr>
<tr>
<td>Proto-HATT+LCL</td>
<td>83.89±0.81</td>
<td>96.88±0.27</td>
<td>87.54±1.19</td>
<td>97.86±0.36</td>
<td>75.48±0.61</td>
<td>96.12±0.14</td>
<td>79.66±1.05</td>
<td>97.16±0.55</td>
</tr>
<tr>
<td>Proto-HATT+SCL</td>
<td>83.35±0.70</td>
<td>96.80±0.23</td>
<td>86.96±0.79</td>
<td>97.67±0.44</td>
<td>74.60±0.47</td>
<td>96.00±0.16</td>
<td>78.55±1.06</td>
<td>97.16±0.18</td>
</tr>
<tr>
<td>FewAsp(multi)</td>
<td>83.33</td>
<td>96.45</td>
<td>86.71</td>
<td>97.62</td>
<td>73.42</td>
<td>95.71</td>
<td>77.65</td>
<td>97.00</td>
</tr>
<tr>
<td>FewAsp(multi)</td>
<td>83.33</td>
<td>96.45</td>
<td>86.71</td>
<td>97.62</td>
<td>73.42</td>
<td>95.71</td>
<td>77.65</td>
<td>97.00</td>
</tr>
<tr>
<td>Proto-HATT+LAS</td>
<td>69.15</td>
<td>91.10</td>
<td>73.91</td>
<td>93.03</td>
<td>55.34</td>
<td>90.44</td>
<td>60.21</td>
<td>92.38</td>
</tr>
<tr>
<td>Proto-HATT+LCL</td>
<td>71.15±0.38</td>
<td>91.59±0.10</td>
<td>75.86±0.49</td>
<td>93.47±0.58</td>
<td>57.90±0.96</td>
<td>90.50±0.45</td>
<td>64.65±0.72</td>
<td>92.57±0.47</td>
</tr>
<tr>
<td>Proto-HATT+SCL</td>
<td>70.37±0.39</td>
<td>91.41±0.15</td>
<td>74.82±0.88</td>
<td>93.32±0.50</td>
<td>56.72±0.91</td>
<td>90.49±0.40</td>
<td>62.77±0.96</td>
<td>92.40±0.37</td>
</tr>
</tbody>
</table>

Table 9: Ablation study over two main components of LDF-AWATT. Besides, we also report the ablation result of Proto-AWATT+LCL. We report the average performance and standard deviation over 5 runs.
Table 10: Error analysis of LDF-AWATT.

A.4 Error Analysis

To analyze the limitations of our framework, we randomly sample 100 error cases by LDF-AWATT from the FewAsp dataset, and roughly classify them into two categories. Table 10 shows the proportions and some representative examples for each category. The primary category is Complex, which includes examples that require deep comprehension to be understood. As shown in example (1), the word fragment “Chandler downtown Serrano” related to restaurant_location appears no more than five times in the dataset, the low frequency of those expressions makes it hard for our model to capture their patterns, so it is really challenging to give a right prediction. The second category is no obvious clues, which includes examples with insufficient information. As shown in example (2), the sentence is very short and unable to provide abundant information to predict the true label.

Through the error analysis, we can conclude that although current models have achieved appealing progress, there are still some complicated sentences beyond their capabilities. There ought to be more advanced natural language processing techniques to further address them.