"Misc"-Aware Weakly Supervised Aspect Classification

Peiran Li*† Fang Guo*‡ Jingbo Shang†

Abstract

Aspect classification, identifying aspects of text segments, facilitates numerous applications, such as sentiment analysis and review summarization. To alleviate the extensive human effort required by existing aspect classification methods, in this paper, we focus on a weakly supervised setting—the model input only contains domain-specific raw texts and a few seed words per pre-defined aspect. We identify a unique challenge here as to how to classify texts without any pre-defined aspects. The existence of this kind of "*misc*" aspect text segments is very common in review corpora. It is difficult, even for domain experts, to nominate seed words for the "*misc*" aspect, which makes existing seed-driven text classification methods not applicable. Therefore, we propose to jointly model pre-defined aspects and the "*misc*" aspect through a novel framework, ARYA. It enables mutual enhancements between pre-defined aspects and the "*misc*" aspect via iterative classifier training and seed set updating. Specifically, it trains a classifier for pre-defined aspects and then leverages it to induce the supervision for the "*misc*" aspect. The prediction results of the "*misc*" aspect are later utilized to further filter the seed word selections for pre-defined aspects. Experiments in three domains demonstrate the superior performance of our proposed framework, as well as the necessity and importance of properly modeling the "*misc*" aspect.

1 Introduction

Aspect classification is a fundamental task in text understanding, aiming at identifying aspects of text segments [5]. It can facilitate various downstream applications, including sentiment analysis and product review summarization. For instance, understanding aspects of a product's review sentences can help to deliver a holistic summary of this product without missing any important aspect [2].

Following the supervised paradigm to extract aspects requires extensive human effort on annotating massive domain-specific texts because aspects vary across domains. For example, in restaurant reviews, possible aspects include "*food*", "*service*", and "*location*". When it comes to laptop reviews, aspects become "*battery*", "*display*", etc. Therefore, to alleviate such effort, we study the problem of *Weakly-supervised aspect classification*, which only relies on very limited supervision — only a small set (e.g., 5) of seed words per aspect.

The major challenge of this problem lies in how to handle the "*misc*" aspect. The "*misc*" aspect is designed to capture two types of text segments which make it noisy: (1) text segments about some specific aspects out of the pre-defined scope and (2) text segments talking nothing about any specific aspect (e.g., "This is one of my favorite restaurants."). Both cases are quite common in the real world. Due to this noisy nature, it is difficult, even for domain experts, to nominate seed words for the "*misc*" aspect, making existing seed-driven text classification methods [1, 20, 8, 26, 15] not applicable here. In this paper, we aim to jointly model the pre-defined aspects and the "*misc*" aspect.

We make two intuitive yet crucial observations, which shed light on the development of our proposed framework. First, given a text segment, if its distribution over pre-defined aspects is flat, it likely belongs to the "*misc*" aspect. This provides us a chance of inducing "*misc*"-aspect supervision from the classifier trained for pre-defined aspects. Second, if a given word is a strong indicator of the "*misc*" aspect, it is unlikely to be a good seed word of any pre-defined aspect. Excluding such words from the seed sets of pre-defined aspects would give us a more precise representation of these aspects and enhance the accuracy of the classification.

In light of these observations, we propose a novel framework incorporating the "*misc*" aspect in a systematic manner. As shown in Figure 1, it is an iterative framework, alternatively training the classifier for all aspects and updating seed sets of pre-defined aspects. We name it as ARYA.[1] More specifically, by utilizing the words in the seed set of each pre-defined aspect, we are able to learn the aspect representations in the latent space and generate pseudo labels of the training corpus. Then we train a classifier for K pre-defined aspects based on user-provided seed sets. This K-aspect classifier further induces supervision for the "*misc*" aspect based on normalized entropy estimation, enabling a (K+1)-aspect classifier. Finally, we build a seed set updating module via comparative analysis to bridge this (K+1)-aspect classifier and the pseudo label generation. In this module, we create candidate pools for both pre-defined aspects and "*misc*" aspect so that we can expand each pre-defined aspect's seed set to most aspect-indicative words excluding the noisy words

*Equal Contribution.
†University of California, San Diego, {pel047, jshang}@ucsd.edu.
‡University of Illinois at Urbana-Champaign, fangguo1@illinois.edu.

[1] Our framework is named after *Arya Stark* in Game of Thrones, who kills the Night King, bringing an end to the *Others* (i.e., White Walkers, wights, etc.) forever.

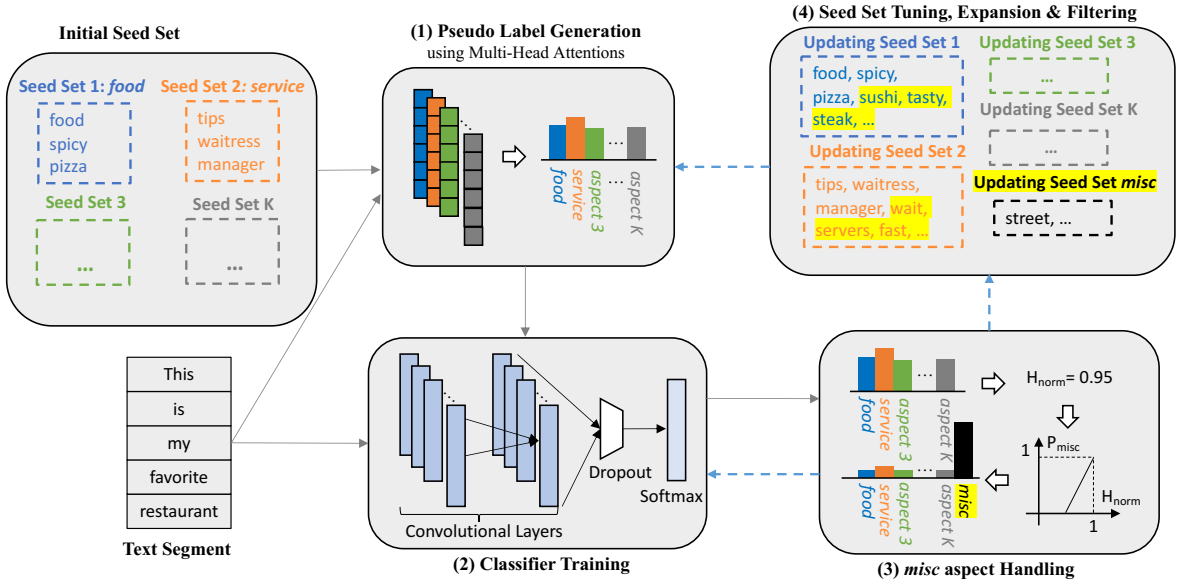


Figure 1: Overview of our proposed framework ARYA. It enables mutual enhancements between the pre-defined aspects and the "*misc*" aspect via iterative classifier training and seed set updating. Pre-defined aspects help to induce supervision for the "*misc*" aspect; The "*misc*" aspect helps to filter out noisy seed words for pre-defined aspects.

that appear in the "*misc*" aspect's candidate pool. This way, ARYA achieves mutual enhancements between pre-defined aspects and the "*misc*" aspect.

To our best knowledge, we are the first to systematically handle the "*misc*" aspect in weakly-supervised aspect classification. Our main contributions are:

- We identify the keystone towards weakly-supervised aspect classification as properly handling the noisy "*misc*" aspect.
- We develop ARYA based on two intuitive observations, making pre-defined aspects and the "*misc*" aspect mutually enhance each other.
- We conduct experiments in three domains, demonstrating the superiority of ARYA and the necessity and importance of considering the "*misc*" aspect systematically.

Reproducibility. We will release our code and datasets in our GitHub repository [2].

2 Overview of ARYA

Problem Formulation. Given a domain-specific corpus $\mathcal{D}$ of n text segments $\{S_1, \ldots, S_n\}$, K pre-defined aspects $\{A_1, \ldots, A_K\}$, and a set of seed words per aspect $\{V_{A_1}, \ldots, V_{A_K}\}$, this paper aims to build an aspect classifier for domain-specific text segments. A domain here refers to a relatively consistent category of products or services, such as *laptop* domain, *hotel* domain, or *restaurant* domain.

In this paper, we assume that there is at most one specific aspect in each text segment. In practice, one can always segment the text in a fine-grained way to ensure that this assumption holds. In other words, for any input text segment S_i, our classifier aims to predict its corresponding aspect label y_i. y_i is either an ID of the pre-defined aspects (between 1 and K) or the number K+1 denoting that S_i focuses on none of the pre-defined aspects.

Algorithm 1: Overall Algorithm

Input: A corpus $\mathbf{D}$ of n text segments $\{S_1, S_2, \ldots, S_n\}$, user-provided seed sets for K pre-defined aspects $\{V_{A_1}, ..., V_{A_K}\}$.
Output: A (K+1)-aspect classifier.
Train word embedding $\mathbf{e}_w$ on $\mathcal{D}$.
while Seed sets are not converged **do**
  Compute aspect embedding $\mathbf{a}_j$ (Eq 3.1)
  Get K-aspect supervision $q_{i,j}$ (Eq 3.4)
  Train K-aspect classifier M_K (Sec 4)
  Get (K+1)-aspect supervision $\hat{q}_{i,j}$ (Eq 5.7)
  Tune, expand, and filter seed sets (Sec 6)
Return The last (K+1)-aspect classifier.

Our Framework. ARYA is an iterative framework as illustrated in Figure 1 and Algorithm 1. It will stop when the seed sets stay unchanged. In each iteration, we apply the following four steps in order.

- **Pseudo Label Generation**. Given seed sets for K aspects, we generate K-aspect pseudo labels for all text segments in the raw corpus.
- **Classifier Training**. We train a K-aspect classifier based on the generated pseudo labels. Our framework is compatible with all text classifiers. As an illustration, we choose to use 1-D CNN in this paper. We will brief its neural architecture for the self-contained purpose.
- **Misc Aspect Handling**. We leverage the predictions of the trained K-aspect classifier to produce pseudo labels for the "*misc*" aspect. After that, we train a new (K+1)-aspect

[2] https://github.com/peiranli/ARYA

classifier, which makes an end-to-end aspect classification.

- **Seed Set Tuning, Expansion, and Filtering**. We conduct a comparative analysis to compare and contrast the text segments projected to different aspects to find new and discriminative seed words for each aspect. The "*misc*" aspect is utilized here to further filter out noisy seed words for pre-defined aspects. This seed set updating module also serves as a bridge between pseudo label generation and misc aspect handling.

Before we discuss the details of the four major components in the following sections, here are some basic notations.

Notations. Each text segment consists of a sequence of tokens, i.e., $S_i = \langle w_1, \ldots, w_{|S_i|} \rangle$, where $|S_i|$ is the number of tokens in S_i. Please note that "token" here includes not only single-word words and punctuation but also multi-word phrases (e.g., "battery life", "chocolate cake") and subword pieces (e.g., "n't"). The tokens are pre-processed from raw texts by applying both tokenization and phrasal segmentation [23].

Let V be the vocabulary set of all tokens. For each token $w \in V$, we denote its d-dimensional embedding vector as $\mathbf{e}_w \in \mathbb{R}^{d \times 1}$. The embedding representation matrix of text segment S_i is then defined as $\mathbf{X}_i = (\mathbf{e}_{w_1}, \ldots, \mathbf{e}_{w_{|S|}}) \in \mathbb{R}^{d \times |S_i|}$ by concatenating each row vector.

3 Pseudo Label Generation

We generate pseudo labels following a multi-head attention mechanism, where each attention head focuses on a specific aspect. It helps our model focus on aspect indicative words and ignore irrelevant ones, and derive aspect-oriented representation. The outputs from all attention heads are finally aggregated to derive the prominent aspect of the text segment.

First, we assume that the user-provided seed sets can characterize the aspect's semantics. So we compute $\mathbf{a}_j$, the aspect representation of A_j, by averaging embedding of its seed words.

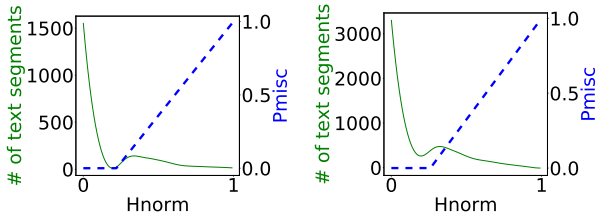
$$\mathbf{a}_j = \frac{\sum_{w \in V_{A_j}} \mathbf{e}_w}{|V_{A_j}|} \tag{3.1}$$

A higher embedding similarity between a word and an aspect implies that the word is more closely related to the aspect, and it should be paid greater attention to. Therefore, given a word w, its attention weight is defined as its maximum similarity over K aspects.

$$\beta_w = \max_{j=1}^{K} \{\mathbf{a}_j^T \mathbf{e}_w\} \tag{3.2}$$

Since text segments are usually short, we use the average of its tokens following the attention weights as its aspect-oriented representation $\mathbf{z}_i$.

$$\mathbf{z}_i = \frac{\sum_{w \in S_i} \beta_w \cdot \mathbf{e}_w}{\sum_{w \in S_i} \beta_w} \tag{3.3}$$



(a) Restaurant Dataset (b) Bags&Cases Dataset

Figure 2: H_{norm} Distribution and P_{misc} Visualization.

Based on the similarity between text segment representation $\mathbf{z}_i$ and aspect representation $\mathbf{a}_j$, we derive the pseudo label assignments as

$$q_{i,j} \propto \exp(\mathbf{a}_j^T \mathbf{z}_i) \tag{3.4}$$

We normalize $q_{i,*}$ into a label distribution over all K aspects.

4 Aspect Classifier Training

Our framework is generally compatible with any text classifiers. In this paper, we choose to use a 1D-CNN model because the multi-head attention mechanism in our pseudo label generation can be viewed as applying a few corresponding convolutional filters. Specifically, every aspect representation $\mathbf{a}_i$ is equivalent to a convolutional filter of window size one.

As mentioned before, we have $\mathbf{X}_i$ as the embedding representation matrix of text segment S_i. We feed $\mathbf{X}_i$ to our 1D-CNN model, as illustrated in Figure 1. Specifically, we employ various filters of window sizes two, three, and four, corresponding to bi-grams, tri-grams, and four-grams. We apply these filters on the input matrix and then add a dropout layer after convolutional layers to alleviate overfitting. Finally, we use a softmax layer to transform the output to probabilities as $P_{i,j}$, denoting the probability of S_i belonging to aspect A_j. The pseudo label distribution q_i generated in the previous step serves as supervision here, using the KL-divergence loss as below.

$$\mathcal{L} = KL(q_i, P_i) = -\sum_{j=1}^{K} q_{i,j} \log \frac{P_{i,j}}{q_{i,j}} \tag{4.5}$$

The same classification logic applies to the training of both K-aspect and (K+1)-aspect classifiers.

5 Misc Aspect Handling

There are two types of text segments belong to the "*misc*" aspect: (1) text segments about some specific aspects different from the K pre-defined aspects; and (2) text segments talking nothing about any specific aspects. These text segments are expected to have a relatively flat distribution in the predictions of the K-aspect classifier. Therefore, it is intuitive to leverage normalized entropy H_{norm}, which measures how chaotic the

distribution is, to estimate the likelihood of S_i belonging to the "*misc*" aspect, i.e., $P_{i,misc}$. Specifically,

$$H_{norm} = -\frac{1}{\log K}\sum_{j=1}^{K} P_{i,j}\log(P_{i,j}) \tag{5.6}$$

As shown in Figure 2, we plot the distribution of H_{norm} for all text segments on the Restaurant and Bags&Cases datasets. One can easily observe that a large volume of the text segments have low H_{norm}, indicating that they belong to some pre-defined aspects. At the same time, those "*misc*" aspect text segments follow a long-tail distribution over large H_{norm} values. Ideally, we want to (1) classify text segments with low enough H_{norm} values to be a non-"*misc*" and (2) assign a higher $P_{i,misc}$ if the H_{norm} is higher. Therefore, we propose to leverage a ReLU-like function to quantify $P_{i,misc}$:

$$P_{i,misc} = \begin{cases} (H_{norm}-\gamma)/(1-\gamma) & H_{norm} \geq \gamma \\ 0 & H_{norm} < \gamma \end{cases}$$

We choose the value of γ as the 3rd quantile of the H_{norm} scores of all documents because based on Figure 2, the 3rd quantile will give a suitable pivot point. Specifically, in Figure 2, the γ values on the Restaurant and Bags&Cases datasets are 0.22 and 0.26, respectively.

After getting this, we combine $P_{i,misc}$ and $P_{i,j}$ to obtain pseudo labels $\hat{q}_{i,j}$ for all aspects, including the "*misc*" aspect.

$$\hat{q}_{i,j} = \begin{cases} (1-P_{i,misc})P_{i,j} & j \leq K \\ P_{i,misc} & j = K+1 \end{cases} \tag{5.7}$$

We then train a (K+1)-aspect 1D-CNN classifier.

6 Seed Set Tuning, Expansion, and Filtering

Merely relying on user-provided seed sets might not be optimal, since there are usually undiscovered but strongly aspect-indicative words in the raw corpus. Also, it is possible that the user-provided ones may sometimes bring noise. Thus it is wiser to update the seed set for each pre-defined aspect: expand to more indicative and discriminative words and also prune the noisy ones if there is any.

Seed Set Tuning. Not every word could be a candidate seed word (e.g., stopwords). Specifically, we try to replace each word by the special UNK token and compute the KL divergence between the K-aspect prediction results before and after. Given a word, if there exists one text segment where this word leads to a KL divergence difference of more than 0.05, the word becomes a candidate. The intuition here is we want to prepare a candidate pool with high recall and reasonable precision. Also, as further ranking and filtering will be applied, this threshold is fairly easy to decide.

Seed Set Expansion. We expand each seed set by ranking and adding words from the candidate pool. Given an aspect A_j and a word w, we mainly consider two measurements:

Table 1: Dataset Statistics.

Dataset	Unlabeled Segments	Test Segments
Restaurant	16,061	1,166
Laptop	14,683	780
Bags&Cases	42,632	624

- **Indicative**. As our pseudo label generation can be viewed as a soft version of string matching using embedding, we want to select words that are popular enough within a certain aspect to ensure the coverage. Mathematically, we want to select the word w, if it has a high probability $P(w|A_j)$. $P(w|A_j)$ means that the word w is popular in the text segments belongs to the aspect A_j. Therefore, we define the indicative measure:

$$\text{Indicative}(A_j, w) = \frac{f_{A_j,w}}{f_{A_j}} \tag{6.8}$$

  where $f_{A_j,w}$ is the frequency of the word w appeared in text segments of the aspect A_j, and f_{A_j} refers to the total text segments of the aspect A_j. The frequency is calculated based on the prediction results on the training set.
- **Distinctive**. Ideally, a seed word should be only frequent in its own aspect. Therefore, we propose a distinctive measure to capture this, which is inspired by $P(A_j|w)$.

$$\text{Distinctive}(A_j, w) = \frac{f_{A_j,w}}{\max_{k\neq j} f_{w,A_k}} \tag{6.9}$$

Since these two scores are of different scales, we aggregate them using the geometric mean following other comparative analyses [26]. Ranking by the aggregated score, we update the seed set of the aspect A_j by the top words here.

Seed Set Filtering. It is worth noting that the same ranking heuristic can be applied to the "*misc*" aspect as well. We observe that highly ranked words in the "*misc*" aspect are mostly general words or some ambiguous words that are related to multiple pre-defined aspects. For instance, it is intuitive for users to select "place" as a seed word for the "*location*" aspect. However, "place" is such a general word in restaurant reviews that it appears frequently in text segments like "this restaurant is such a great place." and "this place serves the best pizza in town.", which are not related to the "*location*" aspect. Therefore, when updating the seed sets, we propose to maintain a new pool of noisy words following the ranking in the "*misc*" aspect and exclude top words in this pool from seed sets in pre-defined aspects.

7 Experiments

In this section, we empirically evaluate our proposed framework ARYA against many compared methods.

Table 2: User-Provided Seed Sets for the Restaurant Dataset. By default, we randomly sample 5 seed words for each aspect as the initial seed sets and run experiments.

Aspect	Seed Word List
Location	street, convenient, block, avenue, river, subway, neighborhood, downtown, bus
Drinks	drinks, beverage, wines, margaritas, sake, beer, wine list, cocktail, vodka, soft drinks
Food	food, spicy, sushi, pizza, tasty, steak, delicious, bbq, seafood, noodle
Ambience	romantic, atmosphere, room, seating, small, spacious, dark, cozy, quaint, music
Service	tips, manager, wait, waitress, servers, fast, prompt, friendly, courteous, attentive

7.1 Datasets We have prepared three review datasets in the restaurant, laptop, and bags&cases domains for evaluation. Table 1 presents you some statistics. These three datasets can be found in our repo[3].

- **Restaurant.** There are 5 aspects in our Restaurant dataset: "*food*", "*service*", "*ambience*", "*drinks*", and "*location*". For training, we have collected 16,061 unlabeled restaurant reviews from the Yelp Dataset Challenge data[4].
- **Laptop.** There are 7 aspects in our Laptop dataset: "*support*", "*display*", "*battery*", "*software*", "*keyboard*", "*os*", "*mouse*". For training, we are using 14,683 unlabeled Amazon reviews on laptop, collected by [12, 6].
- **Bags&Cases.** There are 8 aspects in our Bags&Cases dataset: "*compartments*", "*customer service*", "*handles*", "*looks*", "*price*", "*protection*", "*quality*", "*size fit*". For training, we are using 42,632 unlabeled Amazon reviews on bags&cases, collected by [2]

User-Provided Seed Sets. For all datasets, three annotators are asked to scan the unlabeled corpus and write down 10 seed words for each aspect. Then the annotators need to discuss their picks on each aspect and reach an agreement on their selections. Table 2 shows the finalized seed word list for the Restaurant dataset. By default, for each aspect, we will randomly choose 5 words from the seed word list as the initial seed words in the seed set to train all the models, including both ours and baselines. We report the average of these test results. For one tricky aspect, the "*keyboard*" aspect of the Laptop dataset, we have only collected 3 seed words.

Pre-processing. We pre-process the corpus using the spaCy[5]. Special characters such as "*", "#" and redundant punctuations are removed. We learn word embedding on the unlabeled training corpus.

[3] https://github.com/peiranli/ARYA
[4] https://www.yelp.com/dataset/challenge
[5] https://spacy.io/

7.2 Compared Models We compare our model with a wide range of baseline models, described as follows.

- **CosSim** assigns the most similar aspect to each text segment according to the cosine similarity between the average word embedding of the text segment and that of the seed set for each aspect.
- **Dataless** [24] accepts aspect names as supervision and leverages Wikipedia and Explicit Semantic Analysis (ESA) to derive vector representation of both aspects and documents. The class is assigned based on the vector similarity between aspects and documents.
- **ABAE** [5] is an unsupervised neural topic model. We extend the ABAE by utilizing user-provided seed set for each aspect to align its topics to pre-defined aspects.
- **MATE** [2] is an extended version of ABAE, which accepts seed information for guidance and replaces ABAE's aspect dictionary with seed matrices.
- **WeSTClass** [15] is a strong weakly supervised text classification model, which accepts seed words as supervision.
- **BERT** [4] is a powerful contextualized representation learning technique. We use seed words matching and majority voting to generate sentence labels and then fine-tune the BERT for classification.

Most of these models do not take care of the "*misc*" aspect systematically. Therefore, we fine-tune the best compared method using our proposed "*misc*"-aspect handling, referred to as **Best+OurMisc**.

We denote our model as **ARYA**. Besides, we have a few ablated versions as follows. **ARYA-NoIter** uses our proposed "*misc*" aspect handling technique to generate the probability of "*misc*" aspect based on K-aspect classifier, however, without any further steps. **ARYA-NoTuning** refers to the version of our model without the seed set tuning technique, i.e., no KL divergence threshold for seed word candidates. **ARYA-NoFilter** is our model without the seed set filtering technique, i.e., no noisy seed words removal in pre-defined aspects based on "*misc*" aspect information.

7.3 Experiment Setup Default Parameters. We set the word embedding dimension $d = 200$. For the classifier training, we fix the number of epoch as 5 since it converges quickly. The KL divergence threshold for seed set tuning is 0.05. This value is set based on some human effort by eyeballing the words with a KL divergence difference less than 0.05. We set the maximum number of seed words per each aspect as 10 on the 3 datasets. We use macro-weighted average precision, recall, and F_1 scores.

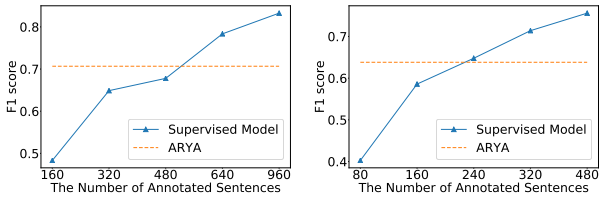
7.4 Experiment Results We present the evaluation results on the Restaurant, Laptop, and Bags&Cases datasets in Table 3. It is clear that our proposed method ARYA outperforms all other methods with significant margins on all datasets because none of these models considers the

Table 3: Evaluation Results on the Restaurant, Laptop, and Bags&Cases Datasets. All precision, recall, and F_1 scores are averaged in the macro-weighted manner. Underlines highlight the best compared models.

Method	Restaurant			Laptop			Bags&Cases		
	Precision	Recall	F_1	Precision	Recall	F_1	Precision	Recall	F_1
CosSim	0.5455	0.4782	0.4985	0.6055	0.5437	0.5083	0.4070	0.3547	0.3858
ABAE	0.5494	0.4904	0.5112	0.6127	0.6168	0.5950	0.4431	0.3736	0.4094
MATE	0.5613	0.5127	0.5177	0.6418	0.6550	0.6474	0.4620	0.4098	0.4335
WeSTClass	0.6153	0.5259	<u>0.5461</u>	0.6688	0.6848	<u>0.6523</u>	0.5301	0.4691	0.4936
Dataless	0.5225	0.4467	0.4265	0.5601	0.5693	0.5569	0.3907	0.3420	0.3786
BERT	0.5955	0.5285	0.5404	0.5949	0.5672	0.5632	0.5656	0.5021	<u>0.5317</u>
Best+OurMisc	0.6288	0.5358	0.5586	0.6724	0.6996	0.6685	0.5810	0.5247	0.5518
ARYA	**0.7410**	**0.6913**	**0.7067**	**0.7849**	**0.7321**	**0.7447**	**0.6565**	**0.6362**	**0.6381**
ARYA-NoIter	0.6934	0.6740	0.6749	0.7508	0.7037	0.7027	0.6463	0.6153	0.6203
ARYA-NoTuning	0.7019	0.6620	0.6729	0.7349	0.6874	0.6822	0.6390	0.6010	0.6094
ARYA-NoFilter	0.7145	0.6706	0.6836	0.7619	0.7158	0.7306	0.6513	0.6218	0.6265

Table 4: Evaluations of different models on no-"*misc*" test sets (i.e., the subset with pre-defined aspects).

Method	Restaurant	Laptop	Bags&Cases
	F_1	F_1	F_1
WeSTClass	0.6567	0.7345	0.6084
BERT	0.6025	0.6257	0.6421
ARYA	**0.8037**	**0.7858**	**0.6806**



(a) Restaurant Dataset (b) Bags&Cases Dataset

Figure 3: Comparison with Supervised Model (i.e., BERT). BERT requires hundreds of labeled sentences to achieve similar result as our ARYA.

"*misc*" aspect systematically. Even compared with the fine-tuned second best models Best+OurMisc, ARYA results in 0.15, 0.08, and 0.09 in absolute improvements over it on the Restaurant, Laptop, and Bags&Cases datasets. It is worth noting that ARYA-NoIter significantly outperforms all compared methods. All these observations show the importance of properly handling the "*misc*" aspect.

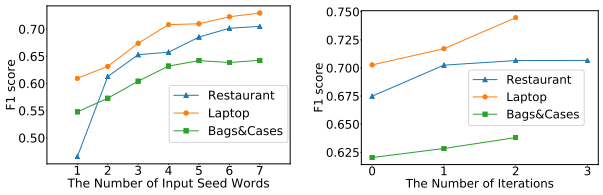
Among all compared methods, MATE is arguably the second-best method. It utilizes the multi-head attention mechanism, which is the same as our pseudo label generation. This implies that attention mechanism is very important for aspect classification. ARYA generalizes attentions to more convolutional filters, thus being more powerful.

The advantage of ARYA over ARYA-NoIter demonstrates the importance of progressively refine the model by updating seed sets at every iteration. Comparing ARYA-NoTuning and ARYA-NoIter, one can see that if we do not limit the scope of seed word candidates, noisy seed words could be added which will lead to even worse performance (e.g., on the Laptop dataset). The improvement of ARYA over ARYA-NoFilter reveals the effectiveness of filtering the seed words in pre-defined aspects by the "*misc*" aspect.

7.5 Performance on test set w/o misc label As "*misc*" label is much harder to be detected than the pre-defined aspects, we conduct an experiment to understand how the model performance changes when there is no "*misc*" label in the test data. Thus, in the test set, we remove those text segments whose corresponding aspect is "misc". We compare ARYA with WeSTClass and BERT due to their robust performance in our previous experiment.

From Table 4, we can observe that even without "*misc*" aspect in the test data, ARYA still achieves the best classification result. This show ARYA is not only good at detecting "*misc*" aspect but also the equipped multi-head attention based Pseudo Label Generation module and CNN-based Classifier training module is effective in classifying pre-defined aspects.

7.6 Comparison with Supervised Model In order to know the gap between full and weakly-supervised methods, we fine-tune BERT [4], a popular and strong supervised model, on different percentage of the training data. For each dataset, we split the labeled texts into 80% for training and 20% for testing. From Figure 3, we can see that on the Restaurant dataset, the supervised model requires more than 50% of the training data to outperform ARYA. Similarly, on the Bags&Cases dataset, almost half of the labeled data is required for BERT to surpass ARYA. Through this comparison, we can conclude that ARYA, as a weakly-supervised model that requires much less human effort, is able to achieve



(a) F1 vs # Input Seed Words (b) F1 vs # Iterations

Figure 4: Effects of the Number of Input Seed Words and Iterations. ARYA stops automatically when the seed sets remain unchanged—the F_1 scores keep improving.

Table 5: Evaluations on different domains by treating one pre-defined aspect as parts of the "*misc*" aspect.

Dataset	Precision	Recall	F_1
Restaurant w/o Ambience	0.6875	0.6921	0.6858
Restaurant w/o Location	0.6893	0.6861	0.6825
Bags&Cases w/o Size_Fit	0.7047	0.6426	0.6596
Bags&Cases w/o Looks	0.6668	0.6330	0.6429

comparable performance with the supervised model.

7.7 Number of Seed Words in Initial Seed Set Because the initial user-provided seed sets are the only supervision, we conduct an experiment to discover how the number of initial seed words affects the performance of ARYA. We use the list of 10 seed words per aspect from the annotators on all the datasets, randomly pick n seed words from each list, run our ARYA for 10 times and report the average macro F_1 scores. We vary n from 1 to 7 and plot the results in Figure 4(a).

Figure 4(a) shows that the F_1 score increases w.r.t. the number of input seed words on all datasets. And the performance becomes stable after the number of input seed words is greater than or equal to 5. In other words, 5 input seed words can provide enough supervision and guidance. This is the reason why we choose 5 input seed words for our model training, which is also affordable for most users.

7.8 Effects of Seed Set Evolution ARYA keeps iterating until the words in the seed set can not be updated any more. So the number of iterations in ARYA is decided automatically. Figure 4(b) shows that the F_1 score increases w.r.t. iterations on all datasets. This suggests that our framework truly enables mutual enhancements between pre-defined aspects and the "*misc*" aspect over iterations.

7.9 Performance by leaving out one pre-defined aspect Table 5 presents the evaluations on difference domains by leaving out one pre-defined aspect. We remove one certain pre-defined aspect from the input and train our model. We can observe that the performances are very closed to the evaluations on the original datasets. This shows that ARYA can capture the "*misc*" aspect given different inputs because the H_{norm} threshold is determined by 3rd quantile.

Table 6: Seed Set Updating Examples. The 0-th iteration indicates the user-provided seed sets.

Dataset	Aspect	Iter	Seed Set
Restaurant	food	0	spicy, pizza, sushi, food, tasty
		1	pizza, spicy, variety, tasty, tuna, sushi, portion, food, specials, bland
	location	0	avenue, convenient, river, street, block
		1	convenient, street, view, river, block, avenue, located
		2	convenient, view, river, block, avenue, located
		3	convenient, view, watching, river, block, avenue, located
Laptop	keyboard	0	keyboard, key, space
		1	keyboard, keys, key
		2	keys, keyboard, numeric, volume, palm, key, layout, keyboards
	os	0	system, os, ios, windows, mac
		1	system, os, ios, operating, mac, windows, lion, interface
Bags&Cases	quality	0	quality, standard, durable, materials, leather
		1	quality, durable, stitching, materials, sharp, substandard, material, leather
		2	quality, durable, stitching, materials, sharp, substandard, material
		3	zippers, quality, durable, stitching, materials, sharp, substandard, material
	protection	0	denting, secure, scratches, protected, cushioning
		1	scratching, cushioning, denting, scratches, protect, protected, protection, secure, protecting

7.10 Case Studies: Seed Words in Different Iterations Table 6 presents the seed set of each aspect w.r.t. different iterations on all datasets. We can observe that the seed words become much better after the seed expansion than the initial seed words.

As mentioned before, even domain experts feel challenging to provide seed words for the "*keyboard*" aspect. Only 3 seed words, "keyboard", "key", and "space" are given at the very beginning. After a few rounds of seed set tuning, expansion, and filtering, some interesting words are added to its seed set, such as "layout", "numeric", and "palm", which make sense for the keyboard aspect. For example, "palm" describes the how comfortable the palms are when typing on a keyboard or how big the keyboard is compared with palms. It is interesting to see that our model can automatically discover these words beyond typical examples come up by experts.

Our model is also capable of pruning the undesirable seed words from seed sets. For example, for the "*location*" aspect in the restaurant dataset, the seed word "street" is initially chosen by annotators but it could be misleading. A counter-example is "Saul is the best restaurant on Smith Street and in Brooklyn ." This segment only states that Saul is the best on Smith Street but does not indicate any comment about the location of the restaurant. That is why this segment is of "*misc*" aspect instead of "*location*" aspect. This shows that our model can automatically detect the undesirable words provided by experts.

We also observe that the seed sets of popular aspects converge faster than infrequent aspects. For example, on the Restaurant dataset, the "*food*", "*ambience*", and "*service*" aspects converge after the 1st iteration, and the "*drinks*" and "*location*" aspects requires 2 and 3 iterations, respectively. The first three have significantly more text segments than the latter two.

Another observation is that the tricky aspects converge slower than the other aspects. For example, on the Laptop dataset, the "*keyboard*" aspect converges much slower than the other aspects, because it is very counter-intuitive to come up with the seed words such as "palm" and "numeric". On the contrary, the "*os*" aspect is relatively easy to be expanded.

7.11 Case Studies: "*Misc*" Text Segments We present two successfully classified text segments of the different types of "*misc*" aspect.

The first example is from the Restaurant dataset, "There is nothing more pleasant than that.", without any specific aspect. ARYA detects that the word "pleasant" as a noisy seed word, because it can refer to "*service*" or "*ambience*". Therefore, it is filtered for these two aspects. Eventually, ARYA predicts the probabilities of this segment belong to "*misc*", "*service*", and "*ambience*" as 0.38, 0.29, and 0.25 respectively. Therefore "*misc*" wins in the end.

The second example is from the Laptop dataset: "the only problem is that i had to add 1 gb RAM, the computer was kinda slow.", about the out-of-pre-defined "*hardware*" aspect. ARYA predicts it as "*misc*" and "*os*" with chances 0.47 and 0.19 respectively, mainly because the word "slow" is widely used to complain about OS.

8 Related Works

Aspect classification was originated at a document-level task, instead of working on text segments. Rule-based methods [7, 10, 34, 21, 31, 19] are the pioneers along this direction. Several unsupervised learning methods based on the LDA topic model and its variants [27, 33, 3, 18, 30, 22] treat extracted topics as aspects. More recently, a neural model ExtRA [11] is proposed to further improve the aspect classification at the document level. However, since our problem focuses on text segments, directly applying these document-level methods leads to some unsatisfactory results. There are also some other recent works that focus on hierarchical text classification [14, 32, 16] and weak supervision [13, 28].

There are several recent unsupervised attempts on aspect classification for text segments. ABAE [5] employs an attention module to learn embedding for text segments and an auto-encoder framework to build aspect dictionaries. However, it requires users to first set the number of topics as a much larger number than the number of desired aspects, and then manually merge and map the extracted topics back to the aspects. Building upon ABAE, [2] further proposed a multi-seed aspect extractor MATE using seed aspect words as guidance. This model keeps the human effort at a minimal degree and fits our problem setting well. However, even with its multi-task counterpart, the reconstruction objective in the MATE model is not able to provide adequate training signals. Our proposed method leverages the seed set tuning and expansion to overcome this issue, thus outperforming MATE significantly in the extensive experiments.

Our problem shares certain similarities with the weakly-supervised text classification problem. Existing methods can build document classifiers by taking either hundreds of labeled training documents [25, 17, 29], class/category names [24, 9], or user-provided seed sets [15] as the source of weak supervision. However, all these methods assume that users can always provide seed sets for all classes, while overlooking the noisy "*misc*" aspect in our problem. We incorporate the "*misc*" aspect systematically into our framework.

9 Conclusions and Future Work

In this paper, we explore to build an aspect classification model for text segments using only a few user-provided seed words per aspect. We identify the key challenge lies in how to properly handle the "*misc*" aspect, for which even domain experts cannot easily design seed words. We propose a novel framework, ARYA, which incorporates the "*misc*" aspect systematically. In our framework, we induce supervision for the "*misc*" aspect using seed words of pre-defined aspects. At the same time, we utilize the "*misc*" aspect information to filter out the noisy words from the seed set of pre-defined aspects. Extensive experiments have demonstrated the effectiveness of ARYA and verified the necessity of modeling the "*misc*" aspect.

In the future, we would like to integrate the extracted aspect information with downstream tasks, such as sentiment analysis and opinion summarization. We also want to explore the use of contextualized representation in weakly supervised aspect classification, further disambiguating words based on contexts. In addition, we are interested in extending our work to document classifications with multiple labels.

Acknowledgements

We thank anonymous reviewers and program chairs for their valuable and insightful feedback. The research was sponsored in part by National Science Foundation Convergence Accelerator OIA-2040727. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and should not be interpreted as necessarily representing the views, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for government purposes notwithstanding any copyright annotation hereon.

References

[1] E. AGICHTEIN AND L. GRAVANO, *Snowball: Extracting relations from large plain-text collections*, in Proceedings of the fifth ACM conference on Digital libraries, ACM, 2000, pp. 85–94.

[2] S. ANGELIDIS AND M. LAPATA, *Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised*, arXiv:1808.08858, (2018).

[3] S. BRODY AND N. ELHADAD, *An unsupervised aspect-sentiment model for online reviews*, in NAACL, 2010, pp. 804–812.

[4] J. DEVLIN, M.-W. CHANG, K. LEE, AND K. TOUTANOVA, *Bert: Pre-training of deep bidirectional transformers for language understanding*, arXiv:1810.04805, (2018).

[5] R. HE, W. S. LEE, H. T. NG, AND D. DAHLMEIER, *An unsupervised neural attention model for aspect extraction*, in ACL, 2017, pp. 388–397.

[6] R. HE AND J. MCAULEY, *Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering*, in WWW, 2016.

[7] M. HU AND B. LIU, *Mining and summarizing customer reviews*, in SIGKDD, 2004, pp. 168–177.

[8] B. J. KUIPERS, P. BEESON, J. MODAYIL, AND J. PROVOST, *Bootstrap learning of foundational representations*, Connection Science, 18 (2006), pp. 145–158.

[9] K. LI, H. ZHA, Y. SU, AND X. YAN, *Unsupervised neural categorization for scientific publications*, in SIAM Data Mining, SIAM, 2018, pp. 37–45.

[10] B. LIU, M. HU, AND J. CHENG, *Opinion observer: analyzing and comparing opinions on the web*, in WWW, 2005, pp. 342–351.

[11] Z. LUO, S. HUANG, F. F. XU, B. Y. LIN, H. SHI, AND K. ZHU, *Extra: Extracting prominent review aspects from customer feedback*, in EMNLP, 2018, pp. 3477–3486.

[12] J. MCAULEY, C. TARGETT, Q. SHI, AND A. VAN DEN HENGEL, *Image-based recommendations on styles and substitutes*, in SIGIR, 2015.

[13] D. MEKALA AND J. SHANG, *Contextualized weak supervision for text classification*, in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 323–333.

[14] D. MEKALA, X. ZHANG, AND J. SHANG, *Meta: Metadata-empowered weak supervision for text classification*, in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2020, pp. 8351–8361.

[15] Y. MENG, J. SHEN, C. ZHANG, AND J. HAN, *Weakly-supervised neural text classification*, in CIKM, 2018, pp. 983–992.

[16] Y. MENG, J. SHEN, C. ZHANG, AND J. HAN, *Weakly-supervised hierarchical text classification*, in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 6826–6833.

[17] T. MIYATO, A. M. DAI, AND I. GOODFELLOW, *Adversarial training methods for semi-supervised text classification*, arXiv:1605.07725, (2016).

[18] A. MUKHERJEE AND B. LIU, *Aspect extraction through semi-supervised modeling*, in ACL, 2012, pp. 339–348.

[19] G. QIU, B. LIU, J. BU, AND C. CHEN, *Opinion word expansion and target extraction through double propagation*, Computational linguistics, 37 (2011), pp. 9–27.

[20] E. RILOFF, J. WIEBE, AND T. WILSON, *Learning subjective nouns using extraction pattern bootstrapping*, in Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4, Association for Computational Linguistics, 2003, pp. 25–32.

[21] C. SCAFFIDI, K. BIERHOFF, E. CHANG, M. FELKER, H. NG, AND C. JIN, *Red opal: product-feature scoring from reviews*, in Proceedings of the 8th ACM conference on Electronic commerce, 2007, pp. 182–191.

[22] M. SHAMS AND A. BARAANI-DASTJERDI, *Enriched lda (elda): combination of latent dirichlet allocation with word co-occurrence analysis for aspect extraction*, Expert Systems with Applications, 80 (2017), pp. 136–146.

[23] J. SHANG, J. LIU, M. JIANG, X. REN, C. R. VOSS, AND J. HAN, *Automated phrase mining from massive text corpora*, TKDE, 30 (2018), pp. 1825–1837.

[24] Y. SONG AND D. ROTH, *On dataless hierarchical text classification*, in AAAI, 2014.

[25] J. TANG, M. QU, AND Q. MEI, *Pte: Predictive text embedding through large-scale heterogeneous text networks*, in SIGKDD, 2015, pp. 1165–1174.

[26] F. TAO, C. ZHANG, X. CHEN, M. JIANG, T. HANRATTY, L. KAPLAN, AND J. HAN, *Doc2cube: Automated document allocation to text cube via dimension-aware joint embedding*, Dimension, 2016 (2015), p. 2017.

[27] I. TITOV AND R. MCDONALD, *Modeling online reviews with multi-grain topic models*, in WWW, ACM, 2008, pp. 111–120.

[28] Z. WANG, D. MEKALA, AND J. SHANG, *X-class: Text classification with extremely weak supervision*, arXiv preprint arXiv:2010.12794, (2020).

[29] W. XU, H. SUN, C. DENG, AND Y. TAN, *Variational autoencoder for semi-supervised text classification*, in AAAI, 2017.

[30] C. ZHANG, H. WANG, L. CAO, W. WANG, AND F. XU, *A hybrid term–term relations analysis approach for topic detection*, Knowledge-Based Systems, 93 (2016), pp. 109–120.

[31] L. ZHANG, B. LIU, S. H. LIM, AND E. O'BRIEN-STRAIN, *Extracting and ranking product features in opinion documents*, in Proceedings of the 23rd international conference on computational linguistics: Posters, Association for Computational Linguistics, 2010, pp. 1462–1470.

[32] Y. ZHANG, X. CHEN, Y. MENG, AND J. HAN, *Hierarchical metadata-aware document categorization under weak supervision*, arXiv preprint arXiv:2010.13556, (2020).

[33] W. X. ZHAO, J. JIANG, H. YAN, AND X. LI, *Jointly modeling aspects and opinions with a maxent-lda hybrid*, in EMNLP, 2010, pp. 56–65.

[34] L. ZHUANG, F. JING, AND X.-Y. ZHU, *Movie review mining and summarization*, in CIKM, 2006, pp. 43–50.