An Uncertainty-Aware Encoder for Aspect Detection

Thi-Nhung Nguyen[1], **Kiem-Hieu Nguyen**[1], **Young-In Song**[2] and **Tuan-Dung Cao**[1]
[1] School of Information Technology and Communication,
Hanoi University of Science and Technology, Hanoi, Vietnam
[2] NAVER
nguyennhung0799@gmail.com, {hieunk,dungct}@soict.hust.edu.vn,
song.youngin@navercorp.com

Abstract

Aspect detection is a fundamental task in opinion mining. Previous works use seed words either as priors of topic models, as anchors to guide the learning of aspects, or as features of aspect classifiers. This paper presents a novel weakly-supervised method to exploit seed words for aspect detection based on an encoder architecture. The encoder maps segments and aspects into a low-dimensional embedding space. The goal is approximating similarity between segments and aspects in the embedding space and their ground-truth similarity generated from seed words. An objective function is proposed to capture the uncertainty of ground-truth similarity. Our method outperforms previous works on several benchmarks in various domains.

1 Introduction

Aspect detection is essential for downstream tasks in opinion mining such as aspect-based sentiment analysis and opinion summarization (Zhang and Liu, 2014; Angelidis and Lapata, 2018). Given an input review segment, for instance, in the restaurant domain, "*Nevertheless the food itself is pretty good.*", we need to detect its aspect category on which opinions have been expressed (e.g., Location, Drinks, Food, Ambience, and Service). The supervised approach requires a large amount of examples (Zhang et al., 2018; Poria et al., 2016). Its unsupervised counterpart learns aspects using techniques such as topic models and autoencoders. The learned aspects are then manually mapped to golden aspects for prediction. Weakly-supervised methods aim at using minimal supervision in terms of seed words to learn aspect predictors.

The topic modeling approach assumes that review contents are generated from aspect probability distributions. Topic models try to learn these distributions using estimators such as maximum likelihood estimation. Seed words are injected into topic models as prior knowledge to guide the estimation of aspect distributions (Mukherjee and Liu, 2012; Chen et al., 2014). The independence assumption in topic models, i.e., the words in a review segment are generated independently from each other, leads to generating incoherent aspects. This phenomenon is more severe as many review segments only have a few words. Wang et al. (2015) propose using a restricted Boltzmann machine for joint aspect detection and sentiment classification. However, their model requires various linguistic tools and external resources including part-of-speech tagging, tf-idf weighting, SentiWordNet, and aspect and sentiment seed words. The aspect seed words are acquired by learning a Latent Dirichlet Allocation on raw segments and manually mapping the learned topics to golden aspects, while the sentiment seed words are acquired based on SentiWordNet.

To overcome this shortage, the neural approach leverages rich representation from contextual language models to capture semantic similarity between words frequently co-occurring in the same contexts (He et al., 2017). Model parameters are learned in neural frameworks such as autoencoder, joint learning or knowledge distillation. Huang et al. (2020) construct word embeddings and explicit aspect embeddings by jointly learning a skip-gram style language model and maximizing the likelihood of aspects and sentiments given seed words. Their model is further reinforced by knowledge distillation based on pseudo-labels from previously learned aspect embeddings, and later by self-training, both with a Convolutional Neural Network (CNN) classifier. In a recent study, Shi et al. (2020) use a contrastive loss to learn aspect embeddings and manually map them to golden aspects. Their model is further enhanced by knowledge distillation with a contextual language model encoder (Sanh et al., 2019). Karamanolakis et al. (2019) co-train student-teacher classifiers in a knowledge distillation framework. The teacher is

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 797–806
November 7–11, 2021. ©2021 Association for Computational Linguistics

designed as a bag-of-seed-words classifier with the weights updated during iterative co-training. Another direction is directly using pre-trained embeddings of aspect labels as aspect vectors and scoring against segment vectors with cosine similarity for prediction (Tulkens and van Cranenburgh, 2020).

In this paper, we propose a novel weakly-supervised method to exploit seed words for aspect detection. Our motivation is from the success of using uncertainty in matrix factorization-based collaborative filtering for movie recommendation from implicit feedback (Hu et al., 2008). The authors show that by adding a confidence score for each user-item pair, their model could learn better from both positive and negative pairs. In our work, we define a confidence score so that ambivalent segments will have a low score. The proposed model is now aware of this ambivalence in learning via a specific designed objective function. Our contributions are as follows:

- A simple and effective encoder architecture is proposed for aspect detection. The goal is to represent segments and aspects in a common latent space. The encoder strives to learn a mapping function that approximates similarity in the latent space and ground-truth similarity generated from the given seed words.
- Inspired by collaborative filtering from implicit feedback (Hu et al., 2008), an uncertainty-aware objective function is proposed to effectively exploit seed words for weakly-supervised learning.
- A selective mechanism is proposed to learn a particularly challenging aspect, namely *General*, based on its seed words.
- The proposed model achieves state-of-the-art performance on several benchmark datasets in various domains.

2 Unsupervised and Weakly-supervised Neural Aspect Detection

Due to its independence assumption, topic models could generate incoherent aspects. He et al. (2017) propose an autoencoder that aims at learning coherent aspects by leveraging word co-occurrence in neural word embeddings. Following this line of research, many works have investigated neural networks for unsupervised aspect detection (Angelidis and Lapata, 2018; Luo et al., 2019; Shi et al., 2020). However, their unsupervised nature requires additional human effort for manual aspect mapping. Weakly-supervised methods have exploited seed words to overcome this shortage and to enhance aspect learning (Karamanolakis et al., 2019; Tulkens and van Cranenburgh, 2020; Huang et al., 2020). In this section, we discuss the key ingredients of unsupervised and weakly-supervised neural models, focusing on representation, aspect mapping, seed words, and the General aspect.

2.1 Segment and aspect representation

Segments are input data for learning autoencoders and aspect classifiers. In the series of autoencoders models, the segments are represented as the weighted sum of word embeddings with the weights estimated from an attention mechanism (He et al., 2017; Angelidis and Lapata, 2018; Shi et al., 2020). In these models, the word embeddings are loaded from pre-trained in-domain word2vec and are fixed during training. In aspect classifiers, the segments are represented using various encoding paradigms, such as the mean of word embeddings (Huang et al., 2020), word embeddings with attention (Tulkens and van Cranenburgh, 2020), CNNs (Huang et al., 2020), BERT (Devlin et al., 2019; Karamanolakis et al., 2019; Shi et al., 2020), or bag-of-words (Karamanolakis et al., 2019).

Previous works represent aspects as explicit parameterized vectors and learn these vectors during training (He et al., 2017; Angelidis and Lapata, 2018; Shi et al., 2020; Huang et al., 2020). In another direction, the embeddings of aspect labels (i.e., 'food' or 'ambience') could be used to represent aspects (Tulkens and van Cranenburgh, 2020).

2.2 Exploiting seed words

Unsupervised methods require human effort to manually map learned aspects to golden aspects using a many-to-one mapping (He et al., 2017) or its recent variant (Shi et al., 2020). By leveraging a few seed words, weakly-supervised methods directly learn golden aspects and require no manual mapping (Angelidis and Lapata, 2018; Huang et al., 2020). Karamanolakis et al. (2019) propose a knowledge distillation framework in which the teacher is a bag-of-seed-words classifier.

2.3 The General aspect

Based on its content, a segment could be classified into a homogeneous *typical* aspect (e.g., Food or

I purchased this case.
My wife is trying to convince me to return it.
We'll just keep it.
And that's the most important feature in a case.
I am going to contact Kensington about a refund.

Figure 1: Five examples of General in the *Laptop Bags* domain demonstrating the variety of this aspect.

Ambience) or a more heterogeneous General aspect. General is an aspect of which contents largely vary. A segment in this aspect could express an overall review of a product, background information, or even irrelevant contents (see Figure 1 for examples). Therefore, it is challenging to detect the segments belonging to this type of aspect. Previous works simply treat General equally to the typical aspects (He et al., 2017; Angelidis and Lapata, 2018; Shi et al., 2020). In some cases, General is ignored in the evaluation (He et al., 2017; Huang et al., 2020). In (Karamanolakis et al., 2019), a segment is classified as General if it does not contain any seed word of the typical aspects.

3 Method

The *Aspect Detection* problem is defined as assigning a review segment to one of the K pre-defined aspect categories. In the unsupervised settings, a corpus of review segments is given. In the weakly-supervised setting as in this work, the segment corpus and sets of seed words for aspect categories are given. As in the literature, we assume that the seed words have already been acquired manually or been extracted automatically from a small number of labeled examples.

Our model is depicted in Figure 2: Firstly, the encoder maps an input segment and the aspects into the same embedding space. For General, the encoder takes all its seed word embeddings as an embedding matrix. Otherwise, the segment and the typical aspects are encoded as mean of their (seed) word embeddings. A similarity function in the embedding space is defined as the dot product of a segment vector and an aspect vector. Finally, the objective function approximates this similarity and the ground-truth similarity generated from the seed words. For the General aspect, the objective function performs a global max pooling over the similarities between the segment and the seed words of General to select the best seed word for updating.

The encoder is an embedding-lookup table and is identical to the word embeddings matrix $\boldsymbol{W} \in \mathbb{R}^{V \times d}$ where V is the vocabulary size and d is the dimension of word vectors. $\boldsymbol{W}$ is initialized by pre-trained in-domain word embeddings. For this task, we used the Skip-gram model (Mikolov et al., 2013). We are going into the details of our model in the subsequent sections. Section 3.4 is dedicated to the generation of ground-truth similarity from seed words.

3.1 Segment and typical aspect embeddings

A segment is encoded as mean of its word embeddings:

$$\boldsymbol{x} = mean(\boldsymbol{w}_1, \boldsymbol{w}_2, .., \boldsymbol{w}_n), \tag{1}$$

in which $\boldsymbol{w}_i$ is a d-dimensional vector of the i^{th} word of the segment and n is the segment length.

Similarly, a typical aspect $\boldsymbol{a}_i$ is mean of its seed word vectors:

$$\boldsymbol{a}_i = mean(\boldsymbol{w}_{i,1}^{(a)}, \boldsymbol{w}_{i,2}^{(a)}, .., \boldsymbol{w}_{i,l_i}^{(a)}), \tag{2}$$

in which $\boldsymbol{w}_{ij}^{(a)}$ is the vector of the j^{th} seed word of the i^{th} aspect, and l_i is the number of seed words in the i^{th} aspect. Our assumption is that an aspect tends to form a cluster in the embedding space. The aspect could then be represented as the centroid of seed words. Those seed words, in turn, will pull the segments belonging to the aspect closer during learning, and will make the cluster more coherent.

3.2 The General aspect embeddings

For General, the encoder takes all its seed words to form an aspect matrix $\boldsymbol{G}$:

$$\boldsymbol{G} = [\boldsymbol{w}_1^{(g)} \boldsymbol{w}_2^{(g)} .. \boldsymbol{w}_{l_g}^{(g)}] \tag{3}$$

where $\boldsymbol{w}_j^{(g)}$ is the j$^{\text{th}}$ seed word of General and l_g is the number of seed words for this aspect. Among the seed words, the closest to the segment is selected, and only the embeddings of this one will be updated during back propagation (the second term of the objective function, as shown in Equation 4).

Our intuition: The segments not belonging to a typical aspect could express anything, either an overall review of the object, background information, or even irrelevant contents. As we group them into an *aspect* with an umbrella term General, it is challenging to define this aspect. Its seed words, typically acquired by manual inspection or by automatic extraction from a small set of labeled examples, tend to be relevant but incoherent. We,

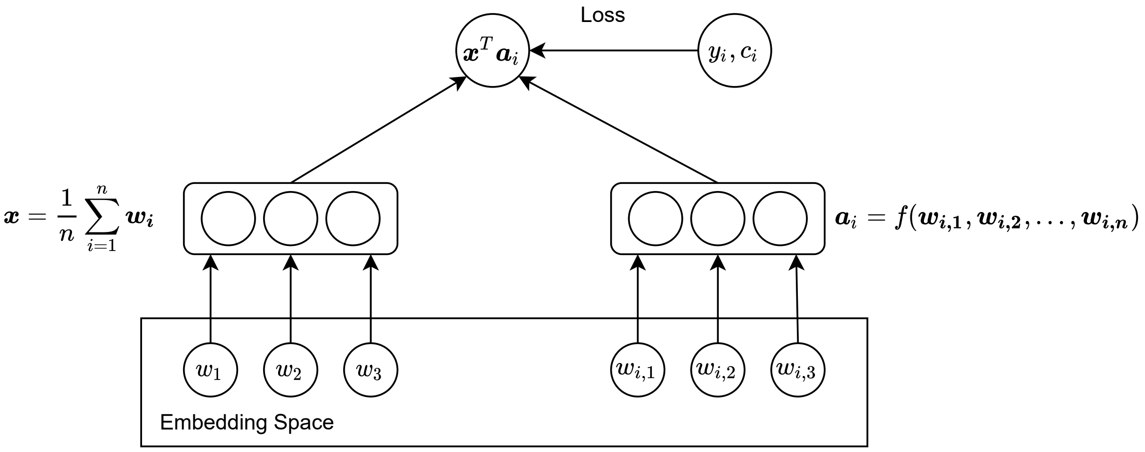
Loss
$\boldsymbol{x}^T\boldsymbol{a}_i$
y_i, c_i
$\boldsymbol{x} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{w_i}$
$\boldsymbol{a}_i = f(\boldsymbol{w_{i,1}}, \boldsymbol{w_{i,2}}, \dots, \boldsymbol{w_{i,n}})$
w_1
w_2
w_3
$w_{i,1}$
$w_{i,2}$
$w_{i,3}$
Embedding Space

Figure 2: Our Uncertainty-Aware Encoder: Segments are encoded as mean of word embeddings. Aspects are encoded as a function of seed words depending on the aspect type (i.e. a typical aspect or General). An objective function is designed to capture the uncertainty of ground-truth similarity generated from the given seed words.

therefore, assume that the aspect contains several sub-clusters, and more importantly, the number of sub-clusters is unknown beforehand. In this way, using the best seed word as representative is a reasonable solution, with a condition that the seed words also scatter over the sub-clusters. Taking the best seed word has another advantage in parameter learning: For a typical aspect, all its seed words will be updated during back propagation. For General, only the best seed word will be updated while the other seed words in the same sub-cluster (if any) and in the other sub-clusters will not be affected. We will later demonstrate in our empirical experiments that this selection plays an important role in the model.

3.3 Objective function

The goal is to approximate similarity in the embedding space and ground-truth similarity. Minimizing mean squared error is a typical choice for this approximation. Inspired by weighted matrix factorization for collaborative filtering from implicit feedback (Hu et al., 2008), given the confidence of ground-truth, our objective function is defined as a mean weighted squared error loss as follows:

$$\begin{aligned} L = \frac{1}{|D|}\sum_{\boldsymbol{x}\in D}(&\sum_{i=1}^{k} c_i(y_i - \boldsymbol{x}^T\boldsymbol{a}_i)^2 \\ &+c^{(g)}(y^{(g)} - \max_{1\le j\le l_g}\boldsymbol{x}^T\boldsymbol{w}_j^{(g)})^2), \end{aligned} \tag{4}$$

where D is the training corpus, y_i and c_i are the ground-truth similarity and confidence of the i^{th} aspect, and $y^{(g)}$ and $c^{(g)}$ are the ground-truth similarity and confidence of General.

For convenience, let's consider General as the (k+1)th aspect: $\boldsymbol{a}_{k+1} = \boldsymbol{w}_{j*}^{(g)}$, where $j* = arg\max_{1\le j\le l_g}(\boldsymbol{x}^T\boldsymbol{w}_j^{(g)})$, $y_{k+1} = y^{(g)}$ and $c_{k+1} = c^{(g)}$. The objective function could be shortened as follows:

$$L = \frac{1}{|D|}\sum_{\boldsymbol{x}\in D}\sum_{i=1}^{k+1} c_i(y_i - \boldsymbol{x}^T\boldsymbol{a}_i)^2 \tag{5}$$

3.4 Generating ground-truth similarity

Generating ground-truth similarity is basically identical to predicting an unseen segment. However, instead of using an optimized encoder, we use a vanilla version of our encoder of which parameters are set-up by pre-trained word embeddings and no optimization is involved. The steps are straightforward: At first, the encoder maps an input segment and the aspects into a d-dimensional space. The similarity between the segment and an individual aspect is then calculated using the dot product function:

$$s_i = \boldsymbol{x}^T\boldsymbol{a}_i, 1 \le i \le k+1. \tag{6}$$

The ground-truth similarity is finally binarized as the following:

$$y_i = \begin{cases} 1 & i = \underset{1\le i\le k+1}{argmax}(s_i) \\ 0 & otherwise \end{cases} \tag{7}$$

Estimating the confidence of ground-truth binary similarity takes more steps. The similarity

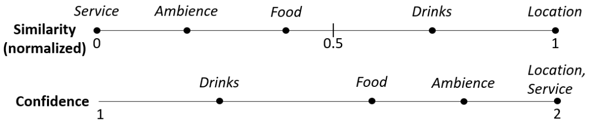

Figure 3: An illustration of the confidence of ground-truth similarity.

in Equation 6 is first scaled to the range of [0,1] using min-max normalization, i.e. the minimum and maximum scaled similarities will be 0 and 1 in that order:

$$\bar{s}_i = \frac{s_i - s_{min}}{s_{max} - s_{min}} \tag{8}$$

Later, an evidence term is defined so that: the most ($y = 1$) and the least similar ($y = 0$) aspects have an absolute evidence value ($e = 1$); The other aspects with $y = 0$ and a similarity value $\bar{s} \leq 0.5$ will have a high evidence value; The rest aspects with $y = 0$ and a similarity value $\bar{s} > 0.5$ will have a low evidence value.

$$e_i = \begin{cases} 1(= \bar{s}_i) & y_i = 1 \\ 1 - \bar{s}_i & otherwise \end{cases} \tag{9}$$

Let's explain this intuition by an example (Figure 3): Suppose that we have to assign a review segment "*Nevertheless the food itself is pretty good.*" to one of the aspects {(L)ocation, (D)rinks, (F)ood, (A)mbience, (S)ervice}[1]. Suppose that $\bar{s}_L = 1 > \bar{s}_D > 0.5 > \bar{s}_F > \bar{s}_A > \bar{s}_S = 0$, the evidence of assigning to Location is $e_L = 1$. The evidence of not assigning to Service is equally $e_S = 1$. For Food and Ambience, the evidence of not assigning to these two aspects should be high since $\bar{s}_F < 0.5$ and $\bar{s}_A < 0.5$. In the end, not assigning to Drinks should have a low evidence value as $\bar{s}_D > 0.5$.

We finally add a constant term to provide a minimal confidence value for each y_i[2]:

$$c_i = 1 + e_i. \tag{10}$$

4 Experiments

In this part, we first describe the datasets used in our experiments in Section 4.1, following by the experimental settings (Section 4.2). The methods selected for comparison are introduced in Section 4.3. The evaluation results are finally discussed in Section 4.4. In all the experiments, our model is referred to as UCE, which stands for UnCertainty-aware Encoder.

4.1 Datasets

We evaluated our method on the following datasets (see Table 1 for the statistics of the datasets):

OPOSUM: The dataset was first introduced in (Angelidis and Lapata, 2018). It contains Amazon product reviews across six domains: *Laptop Bags (Bags), Bluetooth Headsets (B/T), Boots, Keyboards (KBs), Televisions (TVs), Vacuums (VCs).* The dataset was already divided into train/dev/test sets. Like previous works, we used the dev sets to extract seed words. On this dataset, General is a major aspect, its proportion is in the range of $48 - 57\%$ across the six domains.

Restaurant/Laptop: We used the same training and test data as in (Huang et al., 2020). Following previous works, we only evaluated on subsets of aspects. To extract seed words, we used Semeval-2016 (Pontiki et al., 2016) training sets. Only the examples belonging to an aspect of interest were taken. For Restaurant, as the number of such examples is quite large, we randomly selected a subset of 1/6 data to be compatible with the other dev sets (the Dev column in Table 1). Note that previous works ignored the General aspect on these datasets. For a robust evaluation, we followed this setting in our experiments.

Dataset	Train	Dev	Test
Restaurant	17,027	792	643
Laptop	14,683	301	307
Bags	584,332	598	641
B/T	1,419,812	661	656
Boots	957,309	548	611
KBs	603,379	675	681
TVs	1,422,192	699	748
VCs	1,453,651	729	725

Table 1: Statistics of the datasets.

4.2 Experiment settings

For pre-processing, seed word extraction, hyper-parameters settings, and evaluation metrics, we followed previous works for a fair and robust evaluation.

[1] For convenience, we will use L, D, F, A, S to denote Location, Drinks, Food, Ambience and Service, respectively in the similarity and evidence values in this example.

[2] In (Hu et al., 2008), the confidence is defined as $c_i = 1 + \alpha e_i$, in which α is a hyper-parameter. In in-house experiments, we found that the choice of α did not have a significant effect on the model. We thus omitted this hyper-parameter in our confidence.

Pre-processing: OPOSUM and Restaurant/Laptop were pre-processed similarly to (Shi et al., 2020) and (Huang et al., 2020), respectively. Like previous methods, we only focused on sentence-level segments. We fixed the sentence length at 30. Longer segments were truncated, shorter segments were padded.

Seed words: Unless stated otherwise, seed words were extracted from the dev sets using the same extraction method as described in (Angelidis and Lapata, 2018). Given a small number of labeled examples, the method returns ranked lists of terms that are the most representative of aspects. For term scoring, they use a *clarity* function that measures how likely an individual term is observed in an aspect (Cronen-Townsend et al., 2002).

Hyper-parameters: The best hyper-parameters and parameters were selected using the dev sets. Gensim[3] was used to train Skip-gram with the following hyper-parameters: the embedding size to 200, the window size to 10, and the negative sample size to 5. Skip-gram was learned on the training data (the Train column in Table 1). To learn our models, we used the Adam optimizer (Kingma and Ba, 2017) with a learning rate of $1e-5$, a batch size of 512, and a weight decay of $1e-5$. Our best models used five seed words for typical aspects and 30 seed words for General. Section 5.1 discusses the number of seed words in more detail.

Evaluation metrics: The average performance over five runs with different random seeds was reported. For comparison with the previous works, we used micro-averaged F1 for OPOSUM and Accuracy, Precision, Recall and macro-F1 for Restaurant/Laptop.

4.3 Model comparison

For a robust assessment, we compared our method with seven models and baselines on both data sets.

Skip-gram baseline is a variant of our model without parameter learning. It uses the word embeddings from pre-trained Skip-gram to encode segments and aspects as mean of (seed) words. For all the aspects, we used five seed words[4]. **Skip-gram + Max** uses the *maximum* selective mechanism for the General aspect.

ABAE (He et al., 2017) is an autoencoder that learns aspect embeddings by exploiting pre-trained word2vec. The learned topics were manually mapped to golden aspects. **MATE** (Angelidis and Lapata, 2018) improves ABAE by using seed words to learn an aspect matrix. **ISWD** (Karamanolakis et al., 2019) is a weakly-supervised student-teacher co-training framework. The teacher is a bag-of-seed-words classifier. The student is a neural classifier that uses word2vec/BERT to encode segments. **CAt** (Tulkens and van Cranenburgh, 2020) is a heuristic model that consists of a contrastive attention mechanism based on RBF kernels and that uses cosine similarity to assign aspects. **JASen** (Huang et al., 2020) jointly learns word embeddings and aspect embeddings using manually collected aspect and sentiment seed words. It utilizes a CNN with pseudo labels to learn to classify aspects. The CNN classifier is further strengthened by knowledge distillation. **SSCL** (Shi et al., 2020) extends the idea of ABAE to learn aspect embeddings and uses the so-called High-Resolution-Selective-Mapping (HRSMap) for aspect mapping. Similar to ISWD and JASen, their model is further strengthened via a BERT encoder and knowledge distillation.

4.4 Evaluation results

Overall results on all aspects on OPOSUM[5] are reported in Table 2. ABAE reports the lowest F1. Interestingly, when equipped with HRSMap, it is dramatically improved. MATE falls behind the other weakly-supervised methods. Our guess is that its performance is affected by treating General equally to the other aspects. ISWD takes a significant step forward by co-training and specific treatment of General.

The closest to ours is SSCL, which is only 1% to our best model. Its idea is similar to ABAE, using a regressive autoencoder. However, it requires manual aspect mapping. Its best performed version is achieved by using BERT for encoding input segments. It can be seen from Table 2 that no individual method performs best across all the six domains. On average, our model reports the state-of-the-art on the dataset.

In Table 4 and Table 5, we report the performance of UCE, Skip-gram and ISWD on typical aspects and General for OPOSUM. The results for typical aspects are calculated using weighted-F1. It can be seen that UCE outperforms both Skip-gram

[3] https://radimrehurek.com/gensim/

[4] Our in-house experiments showed that using 30 seed words for General in Skip-gram resulted in a poor performance.

[5] The standard deviations of UCE on the F1 metric over the 5 runs with different random seeds are less than 0.3% in all the six domains of OPOSUM.

Method	Bags	B/T	Boots	KBs	TVs	VCs	AVG
ABAE	38.1	37.6	35.2	38.6	39.5	38.1	37.9
ABAE+HRSMap	54.9	62.2	54.7	58.9	59.9	54.1	57.5
MATE	46.2	52.2	45.6	43.5	48.8	42.3	46.4
ISWD	61.4	66.5	52.0	57.5	63.0	60.4	60.2
SSCL	**65.5**	**69.5**	60.4	62.3	67.0	61.0	64.3
Skip-gram	38.6	36.8	30.8	32.4	31.4	32.4	34.0
Skip-gram + Max	49.2	55.4	45.5	54.4	52.4	48.5	50.9
UCE	63.7	68.1	**62.9**	**67.3**	**68.0**	**62.6**	**65.4**

Table 2: Quantitative evaluation of aspect detection on Amazon product reviews. The results of ABAE and MATE were taken from (Angelidis and Lapata, 2018), ISWD from (Karamanolakis et al., 2019), ABAE+HRSMap and SSCL from (Shi et al., 2020).

and ISWD on both typical aspects and General.

The results on Restaurant/Laptop[6] are reported in Table 3. The weakly-supervised methods outperform their unsupervised counterparts by a large margin. UCE is superior on Laptop, but on Restaurant, it lags behind JASen. Their manual seed words provide a good testbed for evaluation. We further trained our model, replacing automatic seed words by these manual seed words while keeping the other settings identical (UCE*). This replacement yielded a new state-of-the-art on Laptop and brought a remarkable improvement on the other.

Despite its simplicity, Skip-gram performs comparably on both datasets. As shown in Table 2, there is a large gap between Skip-gram and the methods having a specific solution for General. As General was omitted on Restaurant/Laptop, it performs better. One can see that UCE significantly outperforms Skip-gram, showing the effectiveness of uncertainty-aware learning.

When looking at performance on each aspect on the Restaurant/Laptop datasets, UCE outperforms Skip-gram on 11 out of 13 aspects, which shows a consistent improvement. On the OPOSUM dataset, UCE yields a remarkable improvement on 35 out of 54 aspects, including General, while it shows significantly less reduction on the others (mostly on recall). Our guess is that as a major aspect, General possibly makes a negative effect on the other aspects.

Based on error analysis, we found that UCE correctly predicted some ambiguous examples. For example, the true label for the sentence *“she replied, well it would be more convenient for us if you ordered now, since you are a larger party, and it might get crowded”.* is *Service*, but Skip-gram predicted *Ambience*. Perhaps *“larger party”* and *“crowded”* caused Skip-gram to make incorrect prediction.

5 Analysis

In this section, we conduct an in-depth analysis of our model on the number of seed words and embedding space learning.

5.1 Number of seed words

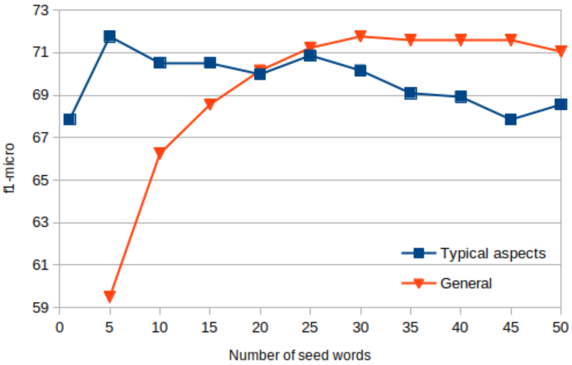

Figure 4: The performance of detecting typical aspects (square) and General (triangle) when the number of seed words varies.

Firstly, the effect of the number of seed words L on typical aspects was investigated. The results are demonstrated on the dev set of the OPOSUM *Laptop Bags* domain. L was chosen in the range of $[1, 50]$ with an interval of 5. The number of seed words for General was fixed to 30. As shown in Figure 4, the performance reaches a peak at 5 seed words. It gradually decreases when more seed words are added.

The effect of choosing L for General is similarly studied. Here, the number of seed words for the

[6] The standard deviations of UCE on the F1 metric over the 5 runs with different random seeds are less than 0.2% on both datasets.

Method	Restaurant				Laptop			
	Acc	**Precision**	**Recall**	**macro-F1**	**Acc**	**Precision**	**Recall**	**macro-F1**
ABAE	67.3	46.6	50.8	45.3	59.8	60.0	59.6	56.2
CAt	66.3	49.2	50.6	46.2	58.0	65.2	59.9	58.6
JASen	**83.8**	64.7	**73.0**	**66.3**	71.0	69.6	71.3	69.7
Skip-gram	67.5	53.7	62.3	53.5	67.8	69.5	70.2	67.4
UCE*	83.1	**66.1**	67.4	66.1	**72.0**	**72.9**	**73.9**	**72.2**
UCE	77.5	56.7	64.7	58.8	71.3	72.2	72.7	71.3

Table 3: Quantitative evaluation of aspect detection on Restaurant and Laptop reviews. The results of ABAE, CAt and JASen were taken from (Huang et al., 2020).

Model	Bags	B/T	Boots	Kbs	TVs	VCs
Skip-gram	48.1	59.5	50.5	67.1	60.2	59.2
ISWD	70.9	**78.2**	67.9	75.2	75.2	74.5
UCE	**72.5**	77.5	**72.6**	**79.1**	**78.1**	**75.5**

Table 4: The performance of UCE on the General aspect on OPOSUM. The performance of ISWD is reported by running their code available at https://github.com/gkaramanolakis/ISWD.

Model	Bags	B/T	Boots	Kbs	TVs	Vcs
Skip-gram	46.5	47.4	37.2	41.3	42.7	39.1
ISWD	41.2	42.0	31.6	26.9	40.4	40.5
UCE	**49.4**	**48.4**	**45.7**	**48.0**	**47.6**	**41.2**

Table 5: The performance of UCE on typical aspects on OPOSUM.

typical aspects is fixed to 5. As can be seen in Figure 4, adding more seed words results in a steady improvement until L reaches 30. After that, the performance slightly decreases.

5.2 Embedding space learning

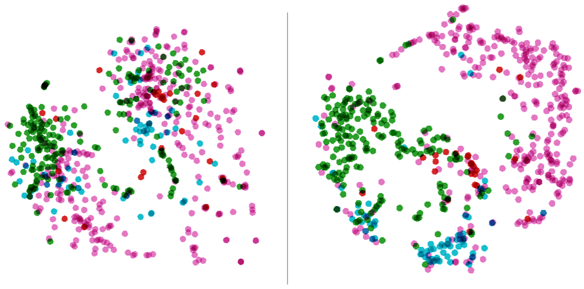

Figure 5: The embeddings space before (left) and after (right) learning. Each data point is a review segment in the test set of Restaurant. The segments of the same aspect have the same color (Location: dark blue, Drink: red, Food: pink, Ambience: light blue, Service: green).

Here, we focus on analyzing the embedding space before and after parameter learning. T-SNE (van der Maaten and Hinton, 2008) was used to project high-dimensional segment vectors into a two-dimensional space. We used the test set of Restaurant for this visualization.

Before learning, the two dominant aspects, i.e. Service and Food, overlap each other as shown in the right-top region of the left part of Figure 5. The two less frequent aspects, i.e., Ambience and Drink, scatter around the space. After learning, there is a clear distinction between Service and Food. Both Ambience and Drink are now more coherent. Since the number of the examples in Location is too small, we could not draw a conclusion on this aspect. Although Service and Food have been largely improved, one can see that these two dominant aspects still interfere with the other aspects, which could mislead the model in prediction.

6 Conclusions and Future Work

In this paper, we have presented a novel neural encoder for aspect detection. Uncertain-aware learning has been proposed to exploit seed words for the task. The model has a selective mechanism to effectively detect the General aspect. Our method consistently achieves the state-of-the-art on several benchmarks.

However, there is still a large room for improvement. Firstly, one should further investigate the distribution of aspects, possibly taking the many-to-one mapping, HRSMap and our selective mechanism as a starting point. In addition to heterogeneity, aspects in related domains typically form a hierarchical structure. Secondly, more general settings should be based on. For example, a segment might belong to multiple aspect categories. Thirdly, as seed words play a central role in weakly-supervised methods, more attention should be paid to the methods to extract this resource. Last but not least, the multi-task perspective is a potential direction, by simultaneously resolving aspect detection, aspect term extraction and sentiment analysis.

Acknowledgements

We would like to thank anonymous reviewers for their constructive feedback. This work was supported by the collaboration program Naver x HUST, funded by Naver corporation.

References

Stefanos Angelidis and Mirella Lapata. 2018. Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3675–3686, Brussels, Belgium. Association for Computational Linguistics.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 347–358, Baltimore, Maryland. Association for Computational Linguistics.

Steve Cronen-Townsend, Yun Zhou, and W. Bruce Croft. 2002. Predicting query performance. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '02, page 299–306, New York, NY, USA. Association for Computing Machinery.

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.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 388–397, Vancouver, Canada. Association for Computational Linguistics.

Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272.

Jiaxin Huang, Yu Meng, Fang Guo, Heng Ji, and Jiawei Han. 2020. Weakly-supervised aspect-based sentiment analysis via joint aspect-sentiment topic embedding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6989–6999, Online. Association for Computational Linguistics.

Giannis Karamanolakis, Daniel Hsu, and Luis Gravano. 2019. Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training. 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 4611–4621, Hong Kong, China. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Ling Luo, Xiang Ao, Yan Song, Jinyao Li, Xiaopeng Yang, Qing He, and Dong Yu. 2019. Unsupervised neural aspect extraction with sememes. In *IJCAI*, pages 5123–5129.

Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Arjun Mukherjee and Bing Liu. 2012. Aspect extraction through semi-supervised modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 339–348.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Know.-Based Syst.*, 108(C):42–49.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Tian Shi, Liuqing Li, Ping Wang, and Chandan K. Reddy. 2020. A simple and effective self-supervised contrastive learning framework for aspect detection.

Stéphan Tulkens and Andreas van Cranenburgh. 2020. Embarrassingly simple unsupervised aspect extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3182–3187, Online. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605.

Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard De Melo. 2015. Sentiment-aspect extraction based on restricted boltzmann machines. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 616–625.

Lei Zhang and Bing Liu. 2014. *Aspect and Entity Extraction for Opinion Mining*, pages 1–40. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis : A survey. *CoRR*, abs/1801.07883.