A Simple and Effective Self-Supervised Contrastive Learning Framework for Aspect Detection

Tian Shi¹, Liuqing Li², Ping Wang¹, Chandan K. Reddy¹
¹Department of Computer Science, Virginia Tech
²Verizon Media
tshi@vt.edu, liuqing.li@verizonmedia.com, ping@vt.edu, reddy@cs.vt.edu

Abstract

Unsupervised aspect detection (UAD) aims at automatically extracting interpretable aspects and identifying aspect-specific segments (such as sentences) from online reviews. However, recent deep learning based topic models, specifically aspect-based autoencoder, suffer from several problems such as extracting noisy aspects and poorly mapping aspects discovered by models to the aspects of interest. To tackle these challenges, in this paper, we first propose a self-supervised contrastive learning framework and an attention-based model equipped with a novel smooth self-attention (SSA) module for the UAD task in order to learn better representations for aspects and review segments. Secondly, we introduce a high-resolution selective mapping (HRSMap) method to efficiently assign aspects discovered by the model to the aspects of interest. We also propose using a knowledge distillation technique to further improve the aspect detection performance. Our methods outperform several recent unsupervised and weakly supervised approaches on publicly available benchmark user review datasets. Aspect interpretation results show that extracted aspects are meaningful, have a good coverage, and can be easily mapped to aspects of interest. Ablation studies and attention weight visualization also demonstrate effectiveness of SSA and the knowledge distillation method.

Introduction

Aspect detection, which is a vital component of aspect-based sentiment analysis (Pontiki et al. 2014, 2015), aims at identifying predefined aspect categories (e.g., Price, Quality) discussed in segments (e.g., sentences) of online reviews. Table 1 shows an example review about a television from several different aspects, such as Image, Sound, and Ease of Use. With a large number of reviews, automatic aspect detection allows people to efficiently retrieve review segments of aspects they are interested in. It also benefits many downstream tasks, such as review summarization (Angelidis and Lapata 2018) and recommendation justification (Ni, Li, and McAuley 2019).

There are several research directions for aspect detection. Supervised approaches (Zhang, Wang, and Liu 2018) can leverage annotated labels of aspect categories but suffer from domain adaptation problems (Rietzler et al. 2020). Another research direction consists of unsupervised approaches and has gained a lot of attention in recent years. Early unsupervised systems are dominated by Latent Dirichlet Allocation (LDA) based topic models (Brody and Elhadad 2010; Mukherjee and Liu 2012; García-Pablos, Cuadros, and Rigau 2018; Rakesh et al. 2018; Zhang et al. 2019). However, several recent studies have revealed that LDA-based approaches do not perform well in aspect detection and extracted aspects are of poor quality (incoherent and noisy) (He et al. 2017). Compared with LDA-based approaches, deep learning models, such as aspect-based autoencoder (ABAE) (He et al. 2017; Luo et al. 2019), have shown excellent performance in extracting coherent aspects and identifying aspect categories for review segments. However, these models require some human effort to manually map model discovered aspects to aspects of interest, which may lead to inaccuracies in mapping especially when model discovered aspects are noisy. Another research direction is based on weakly supervised approaches that leverage a small number of aspect representative words (namely, seed words) for the fine-grained aspect detection (Angelidis and Lapata 2018; Karamanolakis, Hsu, and Gra-vano 2019). Although these models outperform unsupervised approaches, they do make use of human annotated data to extract high-quality aspect seed words, which may limit their application. In addition, they are not able to automatically discover new aspects from review corpus.

We focus our attention towards the problem of unsupervised aspect detection (UAD) since massive amount of reviews are generated every day and many of them are for newer products. It is difficult for humans to efficiently capture new aspects and manually annotate segments for them at scale. Motivated by ABAE, we learn interpretable aspects
by mapping aspect embeddings into word embedding space, so that aspects can be interpreted by the nearest words. To learn better representations for both aspects and review segments, we formulate UAD as a self-supervised representation learning problem and solve it using a contrastive learning algorithm, which is inspired by success of self-supervised contrastive learning in visual representations (Cheng et al. 2020; He et al. 2020). In addition to the learning algorithm, we also resolve two problems that deteriorate the performance of ABAE, including its self-attention mechanism for segment representations and aspect mapping strategy (i.e., many-to-one mapping from aspects discovered by the model to aspects of interest). Finally, we discover that the quality of aspect detection can be further improved by knowledge distillation (Hinton, Vinyals, and Dean 2015). The contributions of this paper are summarized as follows:

- Propose a self-supervised contrastive learning framework for the unsupervised aspect detection task.
- Introduce a high-resolution selective mapping strategy to map model discovered aspects to aspects of interest.
- Utilize knowledge distillation to further improve the performance of aspect detection.
- Conduct systematic experiments on seven benchmark datasets, and demonstrate the effectiveness of our models, both quantitatively and qualitatively.

Related Work

Aspect detection is an important problem of aspect-based sentiment analysis (Zhang, Wang, and Liu 2018; Shi et al. 2019). Existing studies attempt to solve this problem in several different ways, including rule-based, supervised, unsupervised, and weakly supervised approaches. Rule-based approaches focus on lexicons and dependency relations, and utilize manually defined rules to identify patterns and extract aspects (Qiu et al. 2011; Liu et al. 2016), which require domain-specific knowledge or human expertise. Supervised approaches usually formulate aspect extraction as a sequence labeling problem that can be solved by hidden Markov models (HMM) (Jin, Ho, and Srihari 2009), conditional random fields (CRF) (Li et al. 2010; Mitchell et al. 2013; Yang and Cardie 2012), and recurrent neural networks (RNN) (Wang et al. 2016; Liu, Joty, and Meng 2015). These approaches have shown better performance compared to the rule-based ones, but require large amounts of labeled data for training. Unsupervised approaches do not need labeled data. Early unsupervised systems are dominated by Latent Dirichlet Allocation (LDA)-based topic models (Brody and Elhadad 2010; Zhao et al. 2010; Chen, Mukherjee, and Liu 2014; Garcia-Pablos, Cuadros, and Rigau 2018; Shi et al. 2018). Wang et al. (2015) proposed a restricted Boltzmann machine (RBM) model to jointly extract aspects and sentiments. Recently, deep learning based topic models (Srivastava and Sutton 2017; Luo et al. 2019; He et al. 2017) have shown strong performance in extracting coherent aspects. Specifically, aspect-based autoencoder (ABAE) (He et al. 2017) and its variants (Luo et al. 2019) have also achieved competitive results in detecting aspect-specific segments from reviews. The problem is that they need some human effort for aspect mapping. Tulkens and van Cranenburgh (2020) propose a simple heuristic model that can use nouns in the segment to identify and map aspects, however, it strongly depends on the quality of word embeddings, and its applications have so far been limited to restaurant reviews. Weakly-supervised approaches usually leverage aspect seed words as guidance for aspect detection (Angelidis and Lapata 2018; Karamanolakis, Hsu, and Gravano 2019; Zhuang et al. 2020) and achieve better performance than unsupervised approaches. However, most of them rely on human annotated data to extract high-quality seed words and are not flexible to discover new aspects from a new corpus. In this paper, we are interested in unsupervised approaches for aspect detection and dedicated to tackle challenges in aspect learning and mapping.

The Proposed Framework

In this section, we describe our self-supervised contrastive learning framework for aspect detection shown in Fig. 1. The goal is to first learn a set of interpretable aspects (named as model-inferred aspects), and then extract aspect-specific segments from reviews so that they can be used in downstream tasks.

Problem Statement

The Aspect detection problem is defined as follows: given a review segment \( x = \{x_1, x_2, ..., x_T\} \) such as a sentence or an elementary discourse unit (EDU) (Mann and Thompson 1988), the goal is to predict an aspect category \( y_k \in \{y_1, y_2, ..., y_K\} \), where \( x_t \) is the index of a word in the vocabulary, \( T \) is the total length of the segment, \( y_k \) is an aspect among all aspects that are of interest (named as gold-standard aspects), and \( K \) is the total number of gold-standard aspects. For instance, when reviewing restaurants, we may be interested in the following gold-standard aspects: Food, Service, Ambience, etc. Given a review segment, it most likely relates to one of the above aspects.

The first challenge in this problem is to learn model-inferred aspects from unlabeled review segments and map them to a set of gold-standard aspects. Another challenge is to accurately assign each segment in a review to an appropriate gold-standard aspect \( y_k \). For example, in restaurants reviews, “The food is very good, but not outstanding.” → Food. Therefore, we propose a series of modules in our framework, including segment representations, contrastive learning, aspect interpretation and mapping, and knowledge distillation, to overcome both challenges and achieve our goal.

Self-Supervised Contrastive Learning (SSCL)

To automatically extract interpretable aspects from a review corpus, a widely used strategy is to learn aspect embeddings in the word embedding space so that the aspects can be interpreted using their nearest words (He et al. 2017; Angelidis and Lapata 2018). Here, we formulate this learning process as a self-supervised representation learning problem.

Segment Representations

For every review segment in a corpus, we construct two representations directly based on (i) word embeddings and (ii) aspect embeddings. Then, we develop a contrastive learning mechanism to map aspect
embeddings to the word embedding space. Let us denote a word embedding matrix as \( E \in \mathbb{R}^{V \times M} \), where \( V \) is the vocabulary size and \( M \) is the dimension of word vectors. The aspect embedding matrix is represented by \( A \in \mathbb{R}^{N \times M} \), where \( N \) is the number of model-inferred aspects.

Given a review segment \( x = \{x_1, x_2, ..., x_T\} \), we construct a vector representation \( s_{x,E} \) based on its word embeddings \( \{E_{x_1}, E_{x_2}, ..., E_{x_T}\} \), along with a novel self-attention mechanism, i.e.,

\[
s_{x,E} = \sum_{t=1}^{T} \alpha_t E_{x_t},
\]

where \( \alpha_t \) is an attention weight and is calculated as follows:

\[
\alpha_t = \frac{\exp(u_t)}{\sum_{t=1}^{T} \exp(u_t)}
\]

\[
u_t = \lambda \cdot \tanh(q^\top (W_E E_{x_t} + b_E))
\]

Here, \( u_t \) is an alignment score and \( q = \frac{1}{T} \sum_{i=1}^{T} E_{x_i} \) is a query vector. \( W_E \in \mathbb{R}^{M \times M} \), \( b_E \in \mathbb{R}^M \) are trainable parameters, and the smooth factor \( \lambda \) is a hyperparameter. More specifically, we call this attention mechanism as Smooth Self-Attention (SSA). It applies an activation function \( \tanh \) to prevent the model from using a single word to represent the segment, thus increasing the robustness of our model. For example, for the segment “plenty of ports and settings”, SSA will attend both “ports” and “settings”, while regular self-attention may only concentrate on “settings”. Hereafter, we will use RSA to represent regular self-attention adopted in (Angelidis and Lapata 2018). In our experiments, we discover that RSA without smoothness gets worse performance compared to a simple average pooling mechanism.

Further, we also construct a vector representation \( s_{x,A} \) for the segment \( x \) with global aspect embeddings \( \{A_1, A_2, ..., A_N\} \) through another attention mechanism, i.e.,

\[
s_{x,A} = \sum_{n=1}^{N} \beta_n A_n
\]

The attention weight \( \beta_n \) is obtained by

\[
\beta_n = \frac{\exp(v_{n, A}^\top s_{x,E} + b_{n,A})}{\sum_{n=1}^{N} \exp(v_{n, A}^\top s_{x,E} + b_{n,A})},
\]

\[
v_n, A \in \mathbb{R}^M \quad \text{and} \quad b_n, A \in \mathbb{R}
\]

where \( v_n, A \in \mathbb{R}^M \) and \( b_n, A \in \mathbb{R} \) are learnable parameters.

**Algorithm 1:** The SSCL Algorithm

**Input:** Batch size \( X \); constants \( \lambda \) and \( \tau \); network structures;

**Output:** Aspect embedding matrix \( A \); model parameters \( W_E, b_E, v_A, b_A \);

1. Initialize Matrix \( E \) with pre-trained word vectors; matrix \( A \) with k-means centroids;
2. for sampled mini-batch of size \( X \) do
   3. for \( i=1,X \) do
      4. Calculate \( s_{i,E} \) with Eq. (1);
      5. Calculate \( s_{i,A} \) with Eq. (3);
   end
   7. for \( i=1,X; j=1,X \) do
      8. Calculate \( \text{sim}(s_{j,E}, s_{i,A}) \) with Eq. (6);
   end
   10. for \( i=1,X \) do
        11. Calculate \( l_i \) with Eq. (5);
   end
12. Calculate regularization term \( \Omega \) using Eq. (7);
14. Define Loss function \( L = \frac{1}{X} \sum_{i=1}^{X} l_i + \Omega \);
15. Update learnable parameters to minimize \( L \);
16. end

where \( \|\cdot\| \) denotes \( L_2 \)-norm.

We summarize our SSCL framework in Algorithm 1. Specifically, in line 1, the aspect embedding matrix \( A \) is initialized with the centroids of clusters by running k-means on the word embeddings. We follow (He et al. 2017) to penalize...
the aspect embedding matrix and ensure diversity of different aspects. In line 13, the regularization term $\Omega$ is defined as

$$\Omega = \| AA^T - I \|,$$  \hspace{1cm} (7)

where each row of matrix $A$, denoted by $A_j$, is obtained by normalizing the corresponding row in $A$, i.e., $A_j = A_j / \| A_j \|$.  

Aspect Interpretation and Mapping

Aspect Interpretation In the training stage, we map aspect embeddings to the word embedding space in order to extract interpretable aspects. With embedding matrices $A$ and $E$, we first calculate a similarity matrix

$$G = AE^T,$$

where $G \in \mathbb{R}^{N \times V}$. Then, we use the top-ranked words based on $G_n$ to represent and interpret each model-inferred aspect $n$. In our experiments, the matrix with inner product similarity produces more meaningful representative words compared to using the cosine similarity (see Table 6).

Aspect Mapping Most unsupervised aspect detection methods focus on the coherence and meaningfulness of model-inferred aspects, and prefer to map every model-inferred aspect (MIA) to a gold-standard aspect (GSA) (He et al. 2017). Here, we call this mapping as many-to-one mapping, since the number of model-inferred aspects are usually larger than the number of gold-standard aspects. Weakly supervised approaches leverage human-annotated datasets to extract the aspect representative words, so that model-inferred aspects and gold-standard aspects have one-to-one mapping (Angelidis and Lapata 2018). Different from the two mapping strategies described above, we propose a high-resolution selective mapping (HRSMap) strategy as shown in Fig. 2. Here, high-resolution means that the number of model-inferred aspects should be at least 3 times more than the number of gold-standard aspects, so that model-inferred aspects have a better coverage. Selective mapping means noisy or meaningless aspects will not be mapped to gold-standard aspects.

In our experiments, we set the number of MIAs to 30, considering the balance between aspect coverage and human-effort to manually map them to GSAs. First, we automatically generate keywords of MIAs based aspect interpretation results, where the number of the most relevant keywords for each aspect is set to 10. Second, we create several rules for aspect mapping: (i) If keywords of a MIA are clearly related to one specific GSA (not General), we map this MIA to the GSA. For example, we map “apps, app, netflix, browser, hulu, youtube, stream” to Apps/Interface (see Table 6). (ii) If keywords are coherent but not related to any specific GSA, we map this MIA to General. For instance, we map “pc, xbox, dvd, ps3, file, game” to General. (iii) If keywords are related to more than one GSA, we treat this MIA as a noisy aspect and it will not be mapped. For example, “excellent, amazing, good, great, outstanding, fantastic, impressed, superior” may be related to several different GSAs. (iv) If keywords are not quite meaningful, their corresponding MIA will not be mapped. For instance, “ago, within, last 30, later, took, couple, per, every” is a meaningless MIA. Third, we further verify the quality of aspect mapping using development sets.

Given the soft-labels of model-inferred aspects $\beta$, we calculate soft-labels $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_K\}$ over gold-standard aspects for each review segment as follows:

$$\gamma_k = \sum_{n=1}^{N} \mathbb{I}(f(\beta_n) = \gamma_k) \beta_n,$$  \hspace{1cm} (8)

where $f(\beta_n)$ is the aspect mapping for model-inferred aspect $n$. The hard-label $\hat{y}$ of gold-standard aspects for the segment is obtained by

$$\hat{y} = \arg\max \{\gamma_1, \gamma_2, ..., \gamma_K\},$$  \hspace{1cm} (9)

which can be converted to a one-hot vector with length $K$.

Knowledge Distillation

Given both soft- and hard-labels of gold-standard aspects for review segments, we utilize a simple knowledge distillation method, which can be viewed as classification on noisy labeled data. We construct a simple classification model, which consists of a segment encoder such as BERT encoder (Devlin et al. 2019), a smooth self-attention layer (see Eq. (2)), and a classifier (i.e., a single-layer feed-forward network followed by a softmax activation). This model is denoted by SSCLS, where the last S represents student. SSCLS learns knowledge from the teacher model, i.e., SSCL. The loss function is defined as

$$\mathcal{L} = - \frac{1}{K} \sum_{k=1}^{K} \mathbb{I}[H(\gamma) < \xi_k] \cdot \hat{y}_k \log(\hat{y}_k),$$  \hspace{1cm} (10)

where $\hat{y}_k$ is the probability of aspect $k$ predicted by SSCLS. $\hat{y}_k$ is a hard-label given by SSCL. $H(\gamma)$ represents the Shannon entropy for the soft-labels and is calculated by

$$H = - \sum_{k=1}^{K} \gamma_k \log(\gamma_k).$$

Here, the scalar $\xi_k = \chi_G$ if aspect $k$ is General and $\xi_k = \chi_{NG}$, otherwise. Both $\chi_G$ and $\chi_{NG}$ are hyperparameters. Hereafter, we will refer to $\mathbb{I}[H(\gamma) < \xi_k]$ as an Entropy Filter.

\footnote{Usually, it takes less than 15 minutes to assign 30 MIAs to GSAs.}
Aspects Battery, Comfort, Connectivity, Durability, Color, Comfort, Durability, Look, Materials, which the model prediction are more confident. Moreover, with low confidence predictions from the SSCL model, thus to avoid optimizing any models on the testing set, we use Table 2: The annotated aspects for Amazon reviews across different domains.

<table>
<thead>
<tr>
<th>Domains</th>
<th>Aspects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bluetooth</td>
<td>Battery, Comfort, Connectivity, Durability, Ease of Use, Look, Price, Sound, General</td>
</tr>
<tr>
<td>Keyboards</td>
<td>Build Quality, Connectivity, Extra Function, Feel Comfort, Layout, Looks, Noise, Price, General</td>
</tr>
<tr>
<td>TVs</td>
<td>Apps/Interface, Connectivity, Customer Service, Ease of Use, Image, Price, Size/look, Sound, General</td>
</tr>
<tr>
<td>Vacuums</td>
<td>Accessories, Build Quality, Customer Service, Ease of Use, Noise, Price, Suction Power, Weight, General</td>
</tr>
</tbody>
</table>

Table 2: The annotated aspects for Amazon reviews across different domains.

Entropy scores have been used to evaluate the confidence of predictions (Mandelbaum and Weinshall 2017). In the training stage, we set thresholds to filter out training samples with low confidence predictions from the SSCL model, thus allowing the student model to focus on training samples for which the model prediction are more confident. Moreover, the student model also benefits from pre-trained encoders and overcomes the disadvantages of data pre-processing for SSCL, since we have removed out-of-vocabulary words and punctuation, and lemmatized tokens in SSCL. Therefore, SSCLS achieves better performance in segment aspect predictions compared to SSCL.

Experiments

Datasets

We train and evaluate our methods on seven datasets: Citysearch restaurant reviews (Ganu, Elhadad, and Marian 2009) and Amazon product reviews (Angelidis and Lapata 2018) across six different domains, including Laptop Cases (Bags), Bluetooth Headsets (B/T), Boots, Keyboards (KBs), Televisions (TVs), and Vacuums (VCs).

The Citysearch dataset only has training and testing sets. To avoid optimizing any models on the testing set, we use restaurant subsets of SemEval 2014 (Pontiki et al. 2014) and SemEval 2015 (Pontiki et al. 2015) datasets as a development set, since they adopt the same aspect labels as Citysearch. Similar to previous work (He et al. 2017), we select sentences that only express one aspect, and disregard those with multiple and no aspect labels. We have also restricted ourselves to three labels (Food, Service, and Ambience), to form a fair comparison with prior work (Tulkens and van Cranenburgh 2020). Amazon product reviews are obtained from the OPOSUM dataset (Angelidis and Lapata 2018). Different from Citysearch, EDUs (Mann and Thompson 1988) are used as segments and each domain has eight representative aspect labels as well as aspect General (see Table 2).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vocab</th>
<th>W2V</th>
<th>Train</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Citysearch</td>
<td>9,088</td>
<td>279,862</td>
<td>279,862</td>
<td>2,686</td>
<td>1,490</td>
</tr>
<tr>
<td>Bags</td>
<td>6,438</td>
<td>244,546</td>
<td>584,332</td>
<td>598</td>
<td>641</td>
</tr>
<tr>
<td>B/T</td>
<td>9,619</td>
<td>573,206</td>
<td>1,419,812</td>
<td>661</td>
<td>656</td>
</tr>
<tr>
<td>Boots</td>
<td>6,710</td>
<td>408,169</td>
<td>957,309</td>
<td>548</td>
<td>611</td>
</tr>
<tr>
<td>KBs</td>
<td>6,904</td>
<td>241,857</td>
<td>603,379</td>
<td>675</td>
<td>681</td>
</tr>
<tr>
<td>TVs</td>
<td>10,739</td>
<td>579,526</td>
<td>1,422,192</td>
<td>699</td>
<td>748</td>
</tr>
<tr>
<td>VCs</td>
<td>9,780</td>
<td>588,369</td>
<td>1,453,651</td>
<td>729</td>
<td>725</td>
</tr>
</tbody>
</table>

Table 3: The vocabulary size and the number of segments in each dataset. Vocab and W2V represent vocabulary size and word2vec, respectively. Refer to Appendix for more details.

In order to train SSCL, all reviews are preprocessed by removing punctuation, stop-words, and less frequent words (<10). For Amazon reviews, reviews are segmented into elementary discourse units (EDUs) through a Rhetorical Structure Theory parser (Feng and Hirst 2014). We have converted EDUs back to sentences to avoid training word2vec (Mikolov et al. 2013) on very short segments. However, we still use EDU-segments for training and evaluating different models following previous work (Angelidis and Lapata 2018). Table 3 shows statistics of different datasets.

Comparison Methods

We compare our methods against five baselines on the Citysearch dataset. SERBM (Wang et al. 2015) is a sentiment-aspect extraction restricted Boltzmann machine, which jointly extracts review aspects and sentiment polarities in an unsupervised manner. W2VLDA (García-Pablos, Cuadros, and Rigau 2018) is a topic modeling based approach, which combines word embeddings (Mikolov et al. 2013) with Latent Dirichlet Allocation (Blei, Ng, and Jordan 2003). It automatically pairs discovered topics with pre-defined aspect names based on user provided seed-words for different aspects. ABAE (He et al. 2017) is an autoencoder that aims at learning highly coherent aspects by exploiting the distribution of word co-occurrences using neural word embeddings, and an attention mechanism that can put emphasis on aspect-related keywords in segments during training. AE-CSA (Luo et al. 2019) improves ABAE by leveraging sememes to enhance lexical semantics, where sememes are obtained via WordNet (Miller 1995). CAT (Tulkens and van Cranenburgh 2020) is a simple heuristic model that consists of a contrastive attention mechanism based on Radial Basis Function kernels and an automated aspect assignment method.

For Amazon reviews, we compare our methods with several weakly supervised baselines, which explicitly leverage seed words extracted from human annotated development sets (Karamanolakis, Hsu, and Gravano 2019) as supervision for aspect detection. ABAE_{init} (Angelidis and Lapata 2018) replaces each aspect embedding vector in ABAE with the corresponding centroid of seed word embeddings, and fixes aspect embedding vectors during training. MATE (Angelidis and Lapata 2018) uses the weighted average of seed word embeddings to initialize aspect embeddings. MATE-MT extends MATE by introducing an additional multi-task training objective. TS-* (Karamanolakis, Hsu, and Gravano 2019) is a weakly supervised student-teacher co-training
We implemented all deep learning models using PyTorch (Paszke et al. 2017). For each dataset, the best parameters and hyperparameters are selected based on the development set.

For our SSCL model, word embeddings are pre-loaded with 128-dimensional word vectors trained by skip-gram model (Mikolov et al. 2013) with negative sampling and fixed during training. For each dataset, we use gensim\(^3\) to train word embeddings from scratch and set both window and negative sample size to 5. The aspect embedding matrix is initialized with the centroids of clusters by running k-means and hyperparameters are selected based on the development set.

### Implementation Details

We implemented all deep learning models using PyTorch (Paszke et al. 2017). For each dataset, the best parameters and hyperparameters are selected based on the development set.

For our SSCL model, word embeddings are pre-loaded with 128-dimensional word vectors trained by skip-gram model (Mikolov et al. 2013) with negative sampling and fixed during training. For each dataset, we use gensim\(^3\) to train word embeddings from scratch and set both window and negative sample size to 5. The aspect embedding matrix is initialized with the centroids of clusters by running k-means and hyperparameters are selected based on the development set.

We have experimented with two pretrained encoders, i.e., BERT (Devlin et al. 2019) and DistilBERT (Sanh et al. 2019). We tune smooth factor \(\lambda\) in \([0.5, 1.0)\), \(\chi_G\) in \([0.7, 0.8, 1.0, 1.2]\), and \(\chi_{NG}\) in \([1.4, 1.6, 1.8]\). We set \(\chi_G < \chi_{NG}\) to alleviate the label imbalance problem, since the majority of sentences in the corpus are labeled as General.

For both SSCL and SSCLS, model parameters are optimized using the Adam optimizer (Kingma and Ba 2014) with \(\beta_1 = 0.9, \beta_2 = 0.999\), and \(\epsilon = 1 \times 10^{-8}\). Batch size is set to 50. For learning rates, we adopt a warmup schedule strategy proposed in (Vaswani et al. 2017), and set warmup step to 2000 and model size to \(10^5\). Gradient clipping with a threshold of 2 has also been applied to prevent gradient explosion. Our codes are available at https://github.com/tshi04/AspDecSSCL.

### Performance on Amazon Product Reviews

Following previous works (Angelidis and Lapata 2018; Karamanolakis, Hsu, and Gravano 2019), we use micro-averaged F1 score as our evaluation metric to measure the aspect detection performance among different models on Amazon product reviews. All results are shown in Table 4, where we use bold font to highlight the best performance values. The results of the compared models are obtained from the corresponding published papers. From this table, we can observe

### Table 5: Aspect-level precision (P), recall (R), and F-scores (F) on the Citysearch testing set. For overall, we calculate weighted macro averages across all aspects.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Food P</th>
<th>Food R</th>
<th>Food F</th>
<th>Staff P</th>
<th>Staff R</th>
<th>Staff F</th>
<th>Ambience P</th>
<th>Ambience R</th>
<th>Ambience F</th>
<th>Overall P</th>
<th>Overall R</th>
<th>Overall F</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABAE (2017)</td>
<td>89.1</td>
<td>85.4</td>
<td>87.2</td>
<td>81.9</td>
<td>58.2</td>
<td>68.0</td>
<td>80.5</td>
<td>59.2</td>
<td>68.2</td>
<td>86.0</td>
<td>74.6</td>
<td>79.5</td>
</tr>
<tr>
<td>ABAE + HRSMap</td>
<td>93.0</td>
<td>88.8</td>
<td>90.9</td>
<td>85.8</td>
<td>75.3</td>
<td>80.2</td>
<td>67.4</td>
<td>89.6</td>
<td>76.9</td>
<td>87.0</td>
<td>85.8</td>
<td>86.0</td>
</tr>
<tr>
<td>SSCL</td>
<td>91.7</td>
<td>94.6</td>
<td>93.1</td>
<td>88.4</td>
<td>75.9</td>
<td>81.7</td>
<td>79.1</td>
<td>86.1</td>
<td>82.4</td>
<td>88.8</td>
<td>88.7</td>
<td>88.6</td>
</tr>
<tr>
<td>SSCLS-BERT</td>
<td>90.3</td>
<td>97.3</td>
<td>93.3</td>
<td>95.5</td>
<td>71.9</td>
<td>82.0</td>
<td>84.0</td>
<td>87.6</td>
<td>85.8</td>
<td>90.0</td>
<td>89.7</td>
<td>89.4</td>
</tr>
<tr>
<td>SSCL-SSDIbert</td>
<td>91.3</td>
<td>96.6</td>
<td>93.9</td>
<td>92.4</td>
<td>75.9</td>
<td>83.3</td>
<td>84.4</td>
<td>88.0</td>
<td>86.2</td>
<td>90.4</td>
<td>90.3</td>
<td><strong>90.1</strong></td>
</tr>
</tbody>
</table>

\(^3\)https://radimrehurek.com/gensim/
that weakly supervised $\text{ABAE}_{\text{init}}$, MATE and MATE-MT perform significantly better than unsupervised ABAE since they leverage aspect representative words extracted from human-annotated datasets and thus leads to more accurate aspect predictions. TS-Teacher outperforms MATE and MATE-MT on most of the datasets, which further demonstrates that these words are highly correlated with gold-standard aspects. The better performance of both TS-Stu-W2V and TS-Stu-BERT over TS-Teacher demonstrates the effectiveness of their teacher-student co-training framework.

In our experiments, we conjecture that low-resolution many-to-one aspect mapping may be one of the reasons for the low performance of traditional ABAE. Therefore, we have reimplemented ABAE and combined it with HRSMap. The new model (i.e., $\text{ABAE} + \text{HRSMap}$) obtains significantly better results compared to the traditional ABAE on all datasets (performance improvement of 51.7%), showing HRSMap is effective in mapping model-inferred aspects to gold-standard aspects. Compared to the TS-* baseline methods, our SSCL achieves better results on Boots, KBs, and TVs, and competitive results on Bags, B/T, and VCs. On average, it outperforms TS-Teacher, TS-Stu-W2V, and TS-Stu-BERT by 16.9%, 3.9%, and 1.3%, respectively. SSCLS-BERT and SSCLS-DistilBERT further boost the performance of SSCL by 5.4% and 4.4%, respectively, thus demonstrating that knowledge distillation is effective in improving the quality of aspect prediction.

**Performance on Restaurant Reviews**

We have conducted more detailed comparisons on the City-search dataset, which has been widely used to benchmark aspect detection models. Following previous work (Tulken and van Cranenburgh 2020), we use weighted macro averaged precision, recall and F1 score as metrics to evaluate the overall performance. We also evaluate performance of different models for three major individual aspects by measuring aspect-level precision, recall, and F1 scores. Experimental results are presented in Table 5. Results of compared models are obtained from the corresponding published papers.

From Table 5, we also observe that $\text{ABAE} + \text{HRSMap}$ performs significantly better than traditional ABAE. Our SSCL outperforms all baselines in terms of weighted macro averaged F1 score. SSCLS-BERT and SSCLS-DistilBERT further improve the performance of SSCL, and SSCLS-DistilBERT achieves the best results. From aspect-level results, we can observe that, for each individual aspect, our SSCL, SSCLS-BERT and SSCLS-DistilBERT performs consistently better than compared baseline methods in terms of F1 score. SSCLS-DistilBERT gets the best F1 scores across all three aspects. This experiment demonstrates the strength of the contrastive learning framework, HRSMap, and knowledge distillation, which are able to capture high-quality aspects, effectively map model-inferred aspects to gold-standard aspects, and accurately predict aspect labels for the given segments.

**Aspect Interpretation**

As SSCL achieves promising performance quantitatively on aspect detection compared to the baselines, we further show some qualitative results to interpret extracted concepts. From Table 6, we notice that there is at least one model-inferred aspect corresponding to each of the gold-standard aspects, which indicates model-inferred aspects based on HRSMap have a good coverage. We also find that model-inferred concepts, which are mapped to non-general gold-standard aspects, are fine-grained, and their representative words are meaningful and coherent. For example, it is easy to map “app, netflix, browser, hulu, youtube” to Apps/Interface. Compared to weakly supervised methods (such as MATE), SSCL is also able to discover new concepts. For example, for as-

<table>
<thead>
<tr>
<th>Aspects</th>
<th>Representative Keywords</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery</td>
<td>charge recharge life standby battery drain</td>
</tr>
<tr>
<td>Comfort</td>
<td>uncomfortable hurt sore comfortable tight pressure</td>
</tr>
<tr>
<td>Connectivity</td>
<td>usb cable charger adapter port ac paired htc galaxy android macbook connected</td>
</tr>
<tr>
<td>Durability</td>
<td>minute hour foot day min second</td>
</tr>
<tr>
<td>Ease of Use</td>
<td>button pause track control press forward</td>
</tr>
<tr>
<td>Look</td>
<td>red light blinking flashing color blink</td>
</tr>
<tr>
<td>Price</td>
<td>00 buck spend paid dollar cost</td>
</tr>
<tr>
<td>Sound</td>
<td>bass high level low treble frequency noisy wind environment noise truck background</td>
</tr>
<tr>
<td>Price</td>
<td>dollar cost buck 00 pay tax</td>
</tr>
<tr>
<td>Size/Look</td>
<td>32 42 37 46 55 40</td>
</tr>
<tr>
<td>General</td>
<td>rating flaw consider star design improvement christmas gift son birthday 2013 new husband</td>
</tr>
<tr>
<td>General</td>
<td>gym walk house treadmill yard kitchen</td>
</tr>
<tr>
<td>General</td>
<td>player video listen streaming movie pandora</td>
</tr>
<tr>
<td>General</td>
<td>read reading website manual web review</td>
</tr>
<tr>
<td>General</td>
<td>purchased bought buying ordered buy purchase</td>
</tr>
</tbody>
</table>

Table 6: Left: Gold-standard aspects for TVs reviews. Right: Model-inferred aspects presented by representative words.

Table 7: Left: Gold-standard aspects for Bluetooth Headsets reviews. Right: Model inferred aspects presented by representative words.
Strengths mapped to General, we may label “pc, xbox, dvid, ps3, file, game” as Connected Devices, and “plastic glass screw piece metal base” as Build Quality. Similarly, we observe that model-inferred aspects based on Bluetooth Headsets reviews also have sufficient coverage for gold-standard aspects (see Table 7). We can easily map model inferred aspects to gold-standard ones since their keywords are meaningful and coherent. For instance, it is obvious that “red, light, blinking, flashing, color, blink” are related to Look and “charge, recharge, life, standby, battery, drain” are about Battery. For new aspect detection, “motorola, model, plantronics, voyager, backbeatjabra” can be interpreted as Brand. “player, video, listen, streaming, movie, pandora” are about Usage.

Ablation Study and Parameter Sensitivity

In addition to self-supervised contrastive learning framework and HRSMap, we also attribute the promising performance of our models to (i) Smooth self-attention mechanism, (ii) Entropy filters, and (iii) Appropriate batch size. Hence, we systematically conduct ablation studies and parameter sensitivity analysis to demonstrate the effectiveness of them, and provide the results in Fig. 3 and Fig. 4.

First, we replace the smooth self-attention (SSA) layer with a regular self-attention (RSA) layer used in (Angelidis and Lapata 2018) and an average pooling (AP) layer. The model with SSA performs better than the one with AP or RSA. Next, we examine the entropy filter for SSCLS-BERT, and observe that adding it has a positive impact on the model performance. Then, we study the effect of smoothness factor λ in SSA and observe that our model achieves promising and stable results when λ ≤ 1. Finally, we investigate the effect of batch size. F1 scores increase with batch size and become stable when batch size is greater than 20. However, very large batch size increases the computational complexity; see Algorithm 1. Therefore, we set batch size to 50 for all our experiments.

Case Study

Fig. 5 compares heat-maps of attention weights obtained from SSA and RSA on two segments from the Amazon TVs testing set. In each example, RSA attempts to use a single word to represent the entire segment. However, the word may be either a representative word for another aspect (e.g., “scene” for Image in Table 6) or a word with no aspect tendency (e.g., “great” is not assigned to any aspect). In contrast, SSA captures phrases and multiple words, e.g., “volume scenes” and “great value, 499”. Based on the results in Fig. 3 and Fig. 5, we argue SSA is more robust and intuitively meaningful than RSA for aspect detection.

Conclusion

In this paper, we propose a self-supervised contrastive learning framework for aspect detection. Our model is equipped with two attention modules, which allows us to represent every segment with word embeddings and aspect embeddings, so that we can map aspect embeddings to the word embedding space through a contrastive learning mechanism. In the attention module over word embeddings, we introduce a SSA mechanism. Thus, our model can learn robust representations, since SSA encourages the model to capture phrases and multiple keywords in the segments. In addition, we propose a HRSMap method for aspect mapping, which dramatically increases the accuracy of segment aspect predictions for both ABAE and our model. Finally, we further improve the performance of aspect detection through knowledge distillation. BERT-based student models can benefit from pretrained encoders and overcome the disadvantages of data preprocessing for the teacher model. During training, we introduce entropy filters in the loss function to ensure student models focus on high confidence training samples. Our models have shown better performance compared to several recent unsupervised and weakly-supervised models on several publicly available review datasets across different domains. Aspect interpretation results show that extracted aspects are meaningful, have a good coverage, and can be easily mapped to gold-standard aspects. Ablation studies and visualization of attention weights further demonstrate the effectiveness of SSA and entropy filters.
Acknowledgments
This work was supported in part by the US National Science Foundation grants IIS-1707498, IIS-1838730, and NVIDIA Corporation.

References


