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

Tian Shi[1], **Liuqing Li**[2], **Ping Wang**[1], **Chandan K. Reddy**[1]
[1]Department of Computer Science, Virginia Tech
[2]Verizon 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 aspects of interest. We also propose using a knowledge distilling 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 distilling 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 from Amazon platform 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 gain 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). 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 Gravano 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.

Sentence	Aspect
Replaced my 27” jvc clunker with this one.	General
It fits perfectly inside our armoire.	General
Good picture.	Image
Easy to set up and program.	Ease of Use
Descent sound, not great...	Sound
We have the 42” version of this set downstairs.	General
Also a solid set.	General

Table 1: An example from Amazon product reviews about a television and aspect annotations for every sentence.

We focus our attention towards the problem of unsupervised aspect detection (UAD) since massive amount of reviews are generated everyday 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 nearest words. To learn better representations for both aspects and review segments, we

Preprint.

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 (Chen et al. 2020; He et al. 2020). In addition to the learning framework, 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 distilling* (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 distilling 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). 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 Al-location (LDA) based topic models (Brody and Elhadad 2010; Zhao et al. 2010; Chen, Mukherjee, and Liu 2014; García-Pablos, Cuadros, and Rigau 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 *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), we target at predicting 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, constrastive learning, aspect interpretation and mapping, and knowledge distilling, 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 aspects can be interpreted by 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 represents 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.

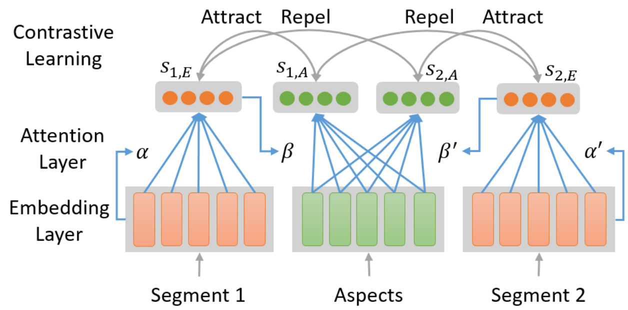


Figure 1: The proposed self-supervised contrastive learning framework. Attract and Repel represent positive and negative pairs, respectively.

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.,

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

where α_t is an attention weight and is calculated as follows:

$$\alpha_t = \frac{\exp(u_t)}{\sum_{\tau=1}^{T} \exp(u_\tau)} \qquad u_t = \lambda \cdot \tanh\left(q^\top (W_E E_{x_t} + b_E)\right) \tag{2}$$

Here, u_t is an alignment score and $q = \frac{1}{T}\sum_{t=1}^{T} E_{x_t}$ 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 λ 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 the RSA without smoothness gets worse performance than 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.,

$$s_{x,A} = \sum_{n=1}^{N} \beta_n A_n \tag{3}$$

The attention weight β_n is obtained by

$$\beta_n = \frac{\exp(v_{n,A}^\top s_{x,E} + b_{n,A})}{\sum_{\eta=1}^{N} \exp(v_{\eta,A}^\top s_{x,E} + b_{\eta,A})}, \tag{4}$$

where $v_{n,A} \in \mathbb{R}^M$ and $b_{n,A} \in \mathbb{R}$ are learnable parameters. $\beta = \{\beta_1, \beta_2, ..., \beta_N\}$ can be also interpreted as **soft-labels (probability distribution) over model-inferred aspects** for a review segment.

Algorithm 1: The SSCL Algorithm

Input: Batch size X; constant λ and τ; 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);
6      end
7      for i=1,X; j=1,X do
8          Calculate sim(s_{j,E}, s_{i,A}) with Eq. (6);
9      end
10     for i=1,X do
11         Calculate l_i with Eq. (5);
12     end
13     Calculate regularization term Ω with Eq. (7);
14     Define Loss function L = (1/X) Σ_{i=1}^{X} l_i + Ω;
15     Update learnable parameters to minimize L.
16 end
```

Contrastive Learning Inspired by recent contrastive learning algorithms (Chen et al. 2020), SSCL learns aspect embeddings by introducing a contrastive loss to maximize the agreement between two representations of the same review segment. During training, we randomly sample a mini-batch of X examples and define the contrastive prediction task on pairs of segment representations from the mini-batch, which is denoted as $\{(s_{1,E}, s_{1,A}), (s_{2,E}, s_{2,A}), ...(s_{X,E}, s_{X,A})\}$. Similar to (Chen et al. 2017), we treat $(s_{i,E}, s_{i,A})$ as a positive pair and $\{(s_{j,E}, s_{i,A})\}_{j\neq i}$ as negative pairs within the mini-batch. The contrastive loss function for a positive pair of examples is defined as

$$l_i = -\log \frac{\exp(\mathrm{sim}(s_{i,E}, s_{i,A})/\mu)}{\sum_{j=1}^{X} \mathbb{I}_{[j\neq i]} \exp(\mathrm{sim}(s_{j,E}, s_{i,A})/\mu)}, \tag{5}$$

where $\mathbb{I}_{[j\neq i]} \in \{0, 1\}$ is an indicator function that equals 1 iff $j \neq i$ and μ represents a temperature hyperparameter. We utilize cosine similarity to measure the similarity between $s_{j,E}$ and $s_{i,A}$, which is calculated as follows:

$$\mathrm{sim}(s_{j,E}, s_{i,A}) = \frac{(s_{j,E})^\top s_{i,A}}{\|s_{j,E}\|\|s_{i,A}\|}, \tag{6}$$

where $\|\cdot\|$ denotes L_2-norm.

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

$$\Omega = \|\mathcal{A}\mathcal{A}^\top - I\|, \tag{7}$$

where each row of matrix $\mathcal{A}$, denoted as $\mathcal{A}_j$, is obtained by normalizing corresponding row in A, i.e., $\mathcal{A}_j = A_j/\|A_j\|$.

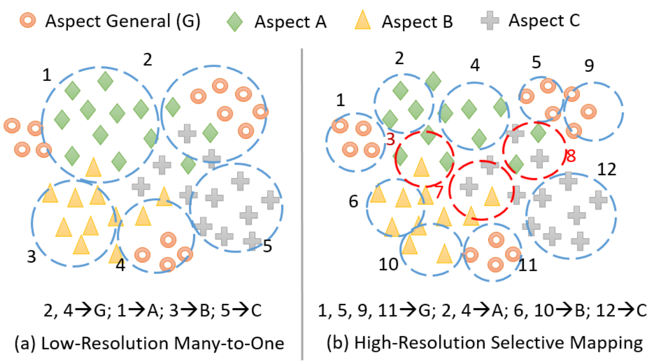


Figure 2: Comparison of Aspect Mappings. For HRSMap, aspects 3, 7, and 8 are not mapped to gold-standard aspects.

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 can first calculate a similarity matrix $G = AE^{\top}$, 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 than with cosine similarity; see Table 5.

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 to a gold-standard aspect (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 gold-standard aspects. Weakly supervised approaches leverage human-annotated datasets to extract aspects 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. More details are described in the Appendix.

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

$$\gamma_k = \sum_{n=1}^{N} \mathbb{I}_{[f(\beta_n)=\gamma_k]} \beta_n, \tag{8}$$

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

$$\hat{y} = \text{argmax}\{\gamma_1, \gamma_2, ...\gamma_K\}, \tag{9}$$

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

Knowledge Distilling

Given both soft- and hard-labels of gold-standard aspects for review segments, we utilize a simple knowledge distilling 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 as SSCLS, where the last S represents **student**. SSCLS learns knowledge from the **teacher** model, i.e., SSCL. The loss function is defined as

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

where y_k is the probability of aspect k predicted by SSCLS. $\hat{y}_k$ is 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 $\xi_k = \chi_{\text{G}}$ if aspect k is *General*, otherwise, $\xi_k = \chi_{\text{NG}}$. Both χ_{G} and χ_{NG} are hyperparameters. Hereafter, we will refer $\mathbb{I}_{[H(\gamma)<\xi_k]}$ as an **Entropy Filter**.

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 less confident predictions by SSCL, so that the student model can focus on high confident training samples. Moreover, the student model also benefits from pretrained encoders and overcomes the disadvantages of data preprocessing for SSCL, since we have removed out-of-vocabulary words and punctuation, and lemmatize tokens in SSCL. Therefore, SSCLS gets better performance in segment aspect predictions compared with 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*.

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

Table 2: The vocabulary size and 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, 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 2 shows statistics of different datasets.

Comparison Methods

We compare our methods against five baselines on 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 auto-encoder 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 framework, where **TS-Teacher** is a bag-of-words classifier (teacher) based on seed words. **TS-Stu-W2V** and **TS-Stu-BERT** are student networks that use word2vec embeddings and BERT model to encode text segments, respectively.

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 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[1] to train word embeddings from scratches and set both window and negative sample size to 5. Aspect embedding matrix is initialized with the centroids of clusters by running k-means on word embeddings. We set the number of aspects to 30 for all datasets because the model can achieve competitive performance while it will still be relatively easier to map model-inferred aspects to gold-standard aspects. The smooth factor λ is tuned in $\{0.5, 1.0, 2.0, 3.0, 4.0, 5.0\}$ and set to 0.5 for all datasets. The temperature μ is set to 1. For *SSCLS*, we have experimented with two pretrained encoders, i.e., BERT (Devlin et al. 2019) and DistilBERT (Sanh et al. 2019). We tune smooth factor λ in $\{0.5, 1.0\}$, χ_{G} in $\{0.7, 0.8, 1.0, 1.2\}$, and χ_{NG} in $\{1.4, 1.6, 1.8\}$. We set $\chi_{\mathrm{G}} < \chi_{\mathrm{NG}}$ to alleviate label imbalance problem, since majority of the sentences in the corpus are labeled as *General*.

For both SSCL and SSCLS, model parameters are optimized using Adam optimizer (Kingma and Ba 2014) with $\beta_1 = 0.9, \beta_2 = 0.999$, and $\epsilon = 10^{-8}$. 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.

Performance on Amazon Product Reviews

Following previous work (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 3, 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 that weakly supervised ABAE$_{init}$, MATE and MATE-MT perform significantly better than unsupervised ABAE since they leverage aspect representative words extracted from human-annotated datasets and thus leading 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 low performance of traditional ABAE. Therefore, we have re-implemented ABAE and combined it with HRSMap. The new model (i.e., ABAE + HRSMap) gets significantly better results compared to the traditional ABAE on all datasets (performance improvement 51.7%), which shows HRSMap is

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

Methods	Bags	B/T	Boots	KBs	TVs	VCs	AVG
Unsupervised Methods							
ABAE (2017)	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
Weakly Supervised Methods							
ABAE$_{init}$ (2018)	41.6	48.5	41.2	41.3	45.7	40.6	43.2
MATE (2018)	46.2	52.2	45.6	43.5	48.8	42.3	46.4
MATE-MT (2018)	48.6	54.5	46.4	45.3	51.8	47.7	49.1
TS-Teacher (2019)	55.1	50.1	44.5	52.0	56.8	54.5	52.2
TS-Stu-W2V (2019)	59.3	66.8	48.3	57.0	64.0	57.0	58.7
TS-Stu-BERT (2019)	61.4	66.5	52.0	57.5	63.0	60.4	60.2
SSCL	61.0	65.2	57.3	60.6	64.6	57.2	61.0
SSCLS-BERT	**65.5**	**69.5**	60.4	**62.3**	**67.0**	**61.0**	**64.3**
SSCLS-DistilBERT	64.7	68.4	**61.0**	62.0	66.3	59.9	63.7

Table 3: Micro-averaged F1 scores for 9-class EDU-level aspect detection in Amazon reviews. **AVG** denotes the average of F1 scores across all domains.

	Food			Staff			Ambience			Overall		
Methods	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**	**P**	**R**	**F**
SERBM (2015)	89.1	85.4	87.2	81.9	58.2	68.0	80.5	59.2	68.2	86.0	74.6	79.5
ABAE (2017)	95.3	74.1	82.8	80.2	72.8	75.7	81.5	69.8	74.0	89.4	73.0	79.6
W2VLDA (2018)	96.0	69.0	81.0	61.0	86.0	71.0	55.0	75.0	64.0	80.8	70.0	75.8
AE-CSA (2019)	90.3	92.6	91.4	92.6	75.6	77.3	91.4	77.9	77.0	85.6	86.0	85.8
CAt (2020)	91.8	92.4	92.1	82.4	75.6	78.8	76.6	80.1	76.6	86.5	86.4	86.4
ABAE + HRSMap	93.0	88.8	90.9	85.8	75.3	80.2	67.4	89.6	76.9	87.0	85.8	86.0
SSCL	91.7	94.6	93.1	88.4	75.9	81.7	79.1	86.1	82.4	88.8	88.7	88.6
SSCLS-BERT	89.6	97.3	93.3	95.5	71.9	82.0	84.0	87.6	85.8	90.0	89.7	89.4
SSCLS-DistilBERT	91.3	96.6	**93.9**	92.4	75.9	**83.3**	84.4	88.0	**86.2**	90.4	90.3	**90.1**

Table 4: 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.

effective in mapping model-inferred aspects to gold-standard aspects. Compared with TS-* baselines, our SSCL achieves better results on Boots, KBs, and TVs, and competitive results on Bags, B/T, and VCs. On an 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%, demonstrating knowledge distilling is effective in improving the quality of aspect prediction.

Performance on Restaurant Review

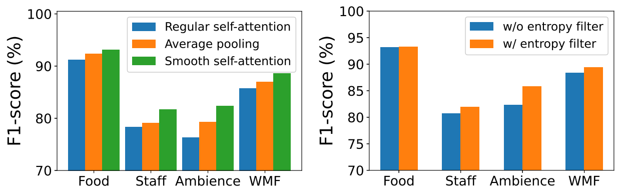
We have conducted more detailed comparisons on the Citysearch dataset, which has been widely used to benchmark aspect detection models. Following previous work (Tulkens and van Cranenburgh 2020), we use weighted macro averaged precision, recall and F1 score as metrics to evaluate the overall performance. 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 4. Results of compared models are obtained from corresponding published papers.

From Table 4, we also observe that ABAE + 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

Aspects	**Representative Keywords**
Apps/Interface	apps app netflix browser hulu youtube
Connectivity	channel antenna broadcast signal station
	optical composite hdmi input component
Customer Serv.	service process company contact support
	call email contacted rep phone repair
Ease of Use	button remote keyboard control use qwerty
Image	setting brightness mode contrast color
	motion scene blur action movement effect
Price	dollar cost buck 00 pay tax
Size/Look	32 42 37 46 55 40
Sound	speaker bass surround volume sound stereo
General	forum read reading review cnet posted
	recommend research buy purchase decision
	plastic glass screw piece metal base
	foot wall mount stand angle cabinet
	football watch movie kid night game
	pc xbox dvd ps3 file game
	series model projection plasma led sony

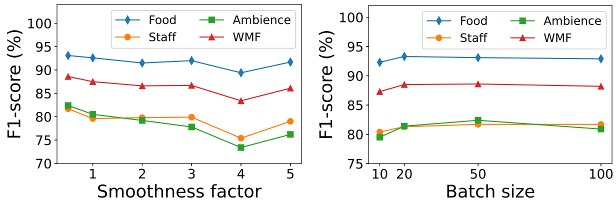
Table 5: Left: Gold-standard aspects for TVs reviews. Right: Model-inferred aspects presented by representative words.

observe that, for each individual aspect, our SSCL, SSCLS-BERT and SSCLS-DistilBERT performs consistently better than compared 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 distilling,



(a) Smooth Self-Attention (b) Entropy Filter

Figure 3: Ablation study on the Citysearch testing set. **WMF** represents weighted macro averaged F1-score.



(a) F1 vs. smoothness factor λ (b) F1 vs. Batch Size

Figure 4: Parameter sensitivity analysis on Citysearch.

which are able to capture high-quality aspects, effectively map model-inferred aspects to gold-standard aspects, and accurately predict aspect labels for given segments.

Aspect Interpretation

As SSCL achieves promising performance on aspect detection compared with baselines in the quantitative analysis, we further show some qualitative results to interpret extracted concepts. From Table 5, we notice that there is at least one model-inferred aspect corresponding to each of gold-standard aspects, which indicates model-inferred aspects based on HRSMap have 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 aspects mapped to *General*, we may label "*pc, xbox, dvd, ps3, file, game*" as *Connected Devices*, and "*plastic glass screw piece metal base*" as *Build Quality*.

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. Therefore, 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 smooth self-attention (SSA) layer with a regular self-attention (RSA) layer used in (Angelidis and Lapata 2018) and an average pooling (AP) layer, respectively. 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 model performance. Then, we study the effect of smoothness factor λ in SSA and observe that our model achieves promising and stable results when $\lambda \leq 1$. 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.

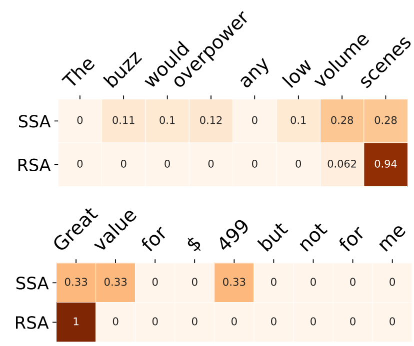


Figure 5: Visualization of attention weights. SSA and RSA represent smooth and regular self-attention, respectively.

Case Study

Fig. 5 compares heat-maps of attention weights obtained from SSA and RSA on two segments from 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 5) 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 attention module over word embeddings, we introduce a SSA mechanism. Thus, our model can learn robust representations, since SSA encourages 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 distilling. BERT-based student models can benefit from pretrained encoders and overcome disadvantages of data preprocessing for the teacher model. During training, we introduce entropy filters in the loss function to ensure student models concentrate on high confidence training samples. Our models have shown better performance compared with several recent unsupervised and weakly-supervised models on a restaurant review dataset and six Amazon reviews 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.

References

Angelidis, S.; and Lapata, M. 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*, 3675–3686. ACL.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research* 3(Jan): 993–1022.

Brody, S.; and Elhadad, N. 2010. An unsupervised aspect-sentiment model for online reviews. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 804–812. ACL.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709* .

Chen, T.; Sun, Y.; Shi, Y.; and Hong, L. 2017. On Sampling Strategies for Neural Network-based Collaborative Filtering. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 767–776. ACM.

Chen, Z.; Mukherjee, A.; and Liu, B. 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)*, 347–358. ACL.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 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)*, 4171–4186.

Feng, V. W.; and Hirst, G. 2014. A linear-time bottom-up discourse parser with constraints and post-editing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 511–521.

Ganu, G.; Elhadad, N.; and Marian, A. 2009. Beyond the stars: improving rating predictions using review text content. In *Proceedings of the 12th International Workshop on the Web and Databases (WebDB 2009)*, 1–6. Citeseer.

García-Pablos, A.; Cuadros, M.; and Rigau, G. 2018. W2VLDA: almost unsupervised system for aspect based sentiment analysis. *Expert Systems with Applications* 91: 127–137.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

He, R.; Lee, W. S.; Ng, H. T.; and Dahlmeier, D. 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)*, 388–397. ACL.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* .

Jin, W.; Ho, H. H.; and Srihari, R. K. 2009. OpinionMiner: a novel machine learning system for web opinion mining and extraction. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1195–1204. ACM.

Karamanolakis, G.; Hsu, D.; and Gravano, L. 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)*, 4603–4613. ACL.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Li, F.; Han, C.; Huang, M.; Zhu, X.; Xia, Y.; Zhang, S.; and Yu, H. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, 653–661. ACL.

Liu, P.; Joty, S.; and Meng, H. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 1433–1443. ACL.

Liu, Q.; Liu, B.; Zhang, Y.; Kim, D. S.; and Gao, Z. 2016. Improving opinion aspect extraction using semantic similarity and aspect associations. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, 2986–2992. ACM.

Luo, L.; Ao, X.; Song, Y.; Li, J.; Yang, X.; He, Q.; and Yu, D. 2019. Unsupervised neural aspect extraction with sememes. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 5123–5129. AAAI Press.

Mandelbaum, A.; and Weinshall, D. 2017. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844* .

Mann, W. C.; and Thompson, S. A. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse* 8(3): 243–281.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, 3111–3119. ACM.

Miller, G. A. 1995. WordNet: a lexical database for English. *Communications of the ACM* 38(11): 39–41.

Mitchell, M.; Aguilar, J.; Wilson, T.; and Van Durme, B. 2013. Open domain targeted sentiment. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1643–1654. ACL.

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

Ni, J.; Li, J.; and McAuley, J. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. 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)*, 188–197. ACL.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in PyTorch .

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Androutsopoulos, I.; Manandhar, S.; Mohammad, A.-S.; Al-Ayyoub, M.; Zhao, Y.; Qin, B.; De Clercq, O.; et al. 2016. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, 19–30.

Pontiki, M.; Galanis, D.; Papageorgiou, H.; Manandhar, S.; and Androutsopoulos, I. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 486–495. ACL.

Pontiki, M.; Galanis, D.; Pavlopoulos, J.; Papageorgiou, H.; Androutsopoulos, I.; and Manandhar, S. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, 27–35. ACL.

Qiu, G.; Liu, B.; Bu, J.; and Chen, C. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics* 37(1): 9–27.

Rietzler, A.; Stabinger, S.; Opitz, P.; and Engl, S. 2020. Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, 4933–4941.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* .

Srivastava, A.; and Sutton, C. A. 2017. Autoencoding variational inference for topic models. In *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*. ICLR, OpenReview.net.

Tulkens, S.; and van Cranenburgh, A. 2020. Embarrassingly simple unsupervised aspect extraction. In *Proceedings of the 58nd Annual Meeting of the Association for Computational Linguistics*, 3182–3187. ACL.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008. ACM.

Wang, L.; Liu, K.; Cao, Z.; Zhao, J.; and De Melo, G. 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)*, 616–625. ACL.

Wang, W.; Pan, S. J.; Dahlmeier, D.; and Xiao, X. 2016. Recursive Neural Conditional Random Fields for Aspect-based Sentiment Analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 616–626. ACL.

Yang, B.; and Cardie, C. 2012. Extracting opinion expressions with semi-markov conditional random fields. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 1335–1345. ACL.

Zhang, L.; Wang, S.; and Liu, B. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8(4): e1253.

Zhao, W. X.; Jiang, J.; Yan, H.; and Li, X. 2010. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, 56–65. ACL.

Zhuang, H.; Guo, F.; Zhang, C.; Liu, L.; and Han, J. 2020. Joint Aspect-Sentiment Analysis with Minimal User Guidance. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1241–1250.

Supplementary Materials

Datasets

In this section, we provide more details about the datasets used in our experiments.

Amazon Reviews We obtain Amazon product reviews from the OPOSUM dataset (Angelidis and Lapata 2018), which has six subsets across different domains, including Laptop Cases, Bluetooth Headsets, Boots, Keyboards, Televisions, and Vacuums. For each subset, reviews are segmented into elementary discourse units (EDUs) through a Rhetorical Structure Theory parser (Feng and Hirst 2014). Then, each segment in development and test sets is manually annotated with eight representative aspect labels as well as aspect *General*. We show the annotated aspect labels in Table A1. In our experiments, we use exactly the same segments and aspect labels as (Angelidis and Lapata 2018).

Domains	**Aspects**
Bags	Compartments, Customer Service, Handles, Looks, Price, Quality, Protection, Size/Fit, General.
Bluetooth	Battery, Comfort, Connectivity, Durability, Ease of Use, Look, Price, Sound, General
Boots	Color, Comfort, Durability, Look, Materials, Price, Size, Weather Resistance, General
Keyboards	Build Quality, Connectivity, Extra Function, Feel Comfort, Layout, Looks, Noise, Price, General
TVs	Apps/Interface, Connectivity, Customer Service, Ease of Use, Image, Price, Size/Look, Sound, General
Vacuums	Accessories, Build Quality, Customer Service, Ease of Use, Noise, Price, Suction Power, Weight, General

Table A1: The annotated aspects for Amazon reviews across different domains.

Restaurant Reviews For restaurant reviews, training and testing sets are from the Citysearch dataset (He et al. 2017), while the development set is a combination of restaurant subsets of SemEval 2014 and SemEval 2015 Aspect-Based Sentiment Analysis datasets (Pontiki et al. 2014, 2015). Similar to previous work (He et al. 2017), sentences are treated as segments. In the development and testing sets, 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 (i.e., *Food*, *Service*, and *Ambience*), to form a fair comparison with prior work (He et al. 2017; Tulkens and van Cranenburgh 2020).

In our experiments, we have also exploited the English restaurant review dataset from SemEval-2016 Aspect-based Sentiment Analysis task (Pontiki et al. 2016) containing reviews for multiple domains and languages, which has been used in prior work (Karamanolakis, Hsu, and Gravano 2019) for aspect detection. However, we find that the dataset suffers from severe label-imbalance problem. For example, there are only 3 and 13 out of 676 sentences labeled as *drinks#prices* and *location#general*, respectively.

Aspect Mapping

In this section, we provide more details of high-resolution selective mapping (HRSMap). High-resolution refers to the fact that the number of model-inferred aspects **(MIAs)** should be at least 3 times more than the number of gold-standard aspects **(GSAs)**, so that model-inferred aspects have a better coverage. Selective mapping implies that 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 gold-standard aspects. Usually, it takes less than 15 minutes to assign 30 MIAs to GSAs. First, we automatically generate keywords of MIAs and dump them into a text file, where the number of the most relevant keywords for each aspect is 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*. (ii) If keywords are coherent but not related to any specific GSA, we map this MIA to *General*. For instance, we map "*football, watch, movie, kid, night, family*" 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.

We provide more qualitative results to demonstrate: (i) MIAs are meaningful and interpretable. (ii) MIAs based on HRSMap have good coverage. (iii) Our model is able to discover new aspects. All results are summarized in Tables A2, A3, A4, A5, A6, and A7.

Aspects	**Representitive Keywords**
Compartments	zippered velcro flap main zipper front
Customer Serv.	service customer warranty shipping contacted email
	shipping arrived return shipped sent amazon
Handles	shoulder strap chest comfortable weight waist
Looks	color blue pink purple green bright
Price	50 cost spend paid dollar price
Protection	protect protection protects protecting protected safe
Quality	scratch dust drop damage scratched bump
	material plastic fabric soft foam leather
Size/Fit	inch perfectly snug tight dell nicely
	plenty lot amount enough ton extra
	17 15 13 14 11 16
General	purchased bought ordered buying buy owned
	review read people mentioned reviewer reading
	airport security tsa friendly checkpoint luggage
	trip travel carry seat traveling school

Table A2: Left: GSAs for Laptop Cases reviews. Right: MIAs presented by representative words.

Aspects	Representative Keywords
Battery	charge recharge life standby battery drain
Comfort	uncomfortable hurt sore comfortable tight pressure
Connectivity	usb cable charger adapter port ac
	paired htc galaxy android macbook connected
Durability	minute hour foot day min second
Ease of Use	button pause track control press forward
Look	red light blinking flashing color blink
Price	00 buck spend paid dollar cost
Sound	bass high level low treble frequency
	noisy wind environment noise truck background
General	rating flaw consider star design improvement
	christmas gift son birthday 2013 new husband
	warranty refund shipping contacted sent email
	motorola model plantronics voyager backbeat jabra
	gym walk house treadmill yard kitchen
	player video listen streaming movie pandora
	read reading website manual web review
	purchased bought buying ordered buy purchase

Table A3: Left: GSAs for Bluetooth Headsets reviews. Right: MIAs presented by representative words.

Aspects	Representative Keywords
Color	color darker brown dark grey gray
Comfort	calf leg ankle shaft top knee
	hurt blister pain sore break rub
Durability	ago wore apart worn started last
Look	casual stylish cute compliment dressy sexy
Materials	slippery traction sole grip tread rubber
	insole lining insert wool liner padding
Price	price paid pay spend cost money
Size	16 13 14 knee circumference 15
	room large big wide tight bigger
Weather Resist.	snow dry water cold wet weather
General	box rubbed weird near cut make
	brand owned miz marten mooz clark
	walking walk floor office town walked
	christmas store local gift daughter birthday
	suggest recommend buy probably consider thinking
	amazon best description future satisfied needle
	reviewer review people others everyone someone
	shipping service seller return delivery amazon

Table A4: Left: GSAs for Boots reviews. Right: MIAs presented by representative words.

Ablation Study and Parameter Sensitivity

In this section, we provide more results for ablation study and parameter sensitivity. Tables A8 and A9 show models with SSA achieve better performance than those with RSA and AVGP. Tables A10 and A11 show effects of the smoothness factor on the performance of our SSCL model. We find that our model achieves promising and stable results when $\lambda \leq 1.0$ and λ is fixed to 0.5 for all datasets. From Table A12 and A13, we can see that F1 scores increase with batch size and become stable when batch size is greater than 20. According to Algorithm 1 line 7-8, we calculate similarities for X^2 times at each training step, where X is the batch size. Since large batch size requires extra computations, we set batch size to 50 for all our experiments as a trade-off between performance and computational complexity.

Aspects	Representative Keywords
Build Qual.	plastic case stand cover bag angle
Connectivity	cable port receiver cord usb dongle
Extra Func.	volume pause mute medium music player
Feel Comfort	wrist hand pain easier typing finger
Layout	smaller size larger sized layout bigger
	backspace shift delete fn arrow alt
Looks	black white see finish color wear lettering print show glossy
	lighting light color bright lit dark
Noise	feedback tactile cherry sound loud noise
Price	price cost dollar buck pay money
General	galaxy tablet pair ipad samsung android
	web email text video movie document
	microsoft ibm natural purchased hp dell
	amazon sent customer seller contacted service
	driver software window install download
	recommend buy highly purchase gaming buying
	month week stopped ago year started
	room couch tv living pc desk
	star negative flaw complain complaint review

Table A5: Left: GSAs for Keyboards reviews. Right: MIAs presented by representative words.

Aspects	Representative Keywords
Accessories	extension powered turbo tool attachment accessory
	container cup bin bag canister tank
Build Quality	plastic screw clip tube tape hose
Customer Serv.	repair warranty send service called contacted
Ease of Use	height switch button setting adjust turn
Noise	difference quality noise design sound flaw
Price	00 cost dollar buck paid shipping
Suction Power	crumb food litter hair sand fur
Weight	easier difficult heavy awkward cumbersome lug
General	recommend thinking suggest money regret thought
	read mentioned reading negative agree complained
	purchased bought buying ordered buy purchasing
	died lasted broke stopped within last
	eureka kenmore electrolux hoover model upright
	corner table bed ceiling chair furniture

Table A6: Left: GSAs for Vacuums reviews. Right: MIAs presented by representative words.

Aspects	Representative Keywords
Ambience	room wall ceiling wood floor window
	music dj bar fun crowd band
	atmosphere romantic cozy feel decor intimate
	wall ceiling wood high black lit
Food	steak medium cooked fry dry tender
	pork chicken potato goat rib roast
	tuna shrimp pork lamb salmon duck
	chocolate coffee cake cream tea dessert
	large small big three four huge
	tomato sauce cheese onion oil crust
	american menu variety japanese italian cuisine
Staff	staff waiter server waitress waitstaff manager
	friendly attentive helpful prompt knowledgeable courteous
General	per tip bill 20 fixe dollar
	sunday night saturday friday weekend evening
	ago birthday anniversary recently last celebrate
	overpriced worth average quality bit pretty
	street west east park manhattan village
	minute year month min hour week
	review say heard believe read reading

Table A7: Left: GSAs for Restaurant reviews. Right: MIAs presented by representative words.

Smooth	Bags	B/T	Boots	KBs	TVs	VCs	AVG
SSA	61.0	65.2	57.3	60.6	64.6	57.2	61.0
RSA	55.9	62.3	52.9	59.5	59.5	53.9	57.3
AVGP	61.6	65.5	52.7	60.5	64.0	56.0	60.1

Table A8: Effects of SSA on micro-averaged F1 scores for Amazon review datasets. SSA, RSA, AVGP represent smooth self-attention, regular self-attention and average-pooling, respectively.

Smooth	Food	Staff	Ambience	WMF
SSA	93.1	81.7	82.4	88.6
RSA	91.2	78.4	76.3	85.7
AVGP	92.4	79.1	79.3	87.0

Table A9: Effects of SSA on aspect-level F1 scores and weighted macro-averaged F1 scores for the Citysearch dataset. WMF represents weighted macro averaged F1-score.

λ	Bags	B/T	Boots	KBs	TVs	VCs	AVG
0.5	61.0	65.2	57.3	60.6	64.6	57.2	61.0
1.0	61.6	65.1	58.3	61.8	66.4	55.6	61.5
2.0	60.7	63.9	57.3	59.8	67.0	55.0	60.6
3.0	61.8	64.6	57.6	59.9	63.0	55.3	60.4
4.0	58.2	64.2	54.0	59.9	64.3	56.1	59.4
5.0	57.4	63.0	54.2	59.3	66.4	54.9	59.2

Table A10: Effects of smoothness factor λ on micro-averaged F1 scores for Amazon review datasets.

λ	Food	Staff	Ambience	WMF
0.5	93.1	81.7	82.4	88.6
1.0	92.6	79.6	80.5	87.5
2.0	91.5	79.8	79.2	86.6
3.0	92.0	79.9	77.8	86.7
4.0	89.4	75.4	73.4	83.4
5.0	91.7	79.0	76.2	86.1

Table A11: Effects of smoothness factor λ on aspect-level F1 scores and weighted macro-averaged F1 scores for the Citysearch dataset.

Bsize	Bags	B/T	Boots	KBs	TVs	VCs	AVG
20	60.2	66.9	56.0	60.4	66.7	56.3	61.1
50	61.0	65.2	57.3	60.6	64.6	57.2	61.0
100	61.8	66.0	55.8	61.4	63.4	57.4	61.0
200	59.4	64.6	56.3	60.8	64.6	56.6	60.4

Table A12: Effects of batch size on micro-averaged F1 scores for Amazon review datasets.

Bsize	Food	Staff	Ambience	WMF
10	92.3	80.4	79.5	87.3
20	93.3	81.3	81.4	88.5
50	93.1	81.7	82.4	88.6
100	92.9	81.7	80.9	88.2
200	93.0	82.6	82.9	88.9

Table A13: Effects of batch size on aspect-level F1 scores and weighted macro-averaged F1 scores for the Citysearch dataset.