Solving Aspect Category Sentiment Analysis as a Text Generation Task

Jian Liu[1], **Zhiyang Teng**[2,3], **Leyang Cui**[2,3], **Hanmeng Liu**[2,3], **Yue Zhang**[2,3]
[1]School of Computer Science, Fudan University
[2]School of Engineering, Westlake University
[3]Institute of Advanced Technology, Westlake Institute for Advanced Study
jianliu17@fudan.edu.cn,
{tengzhiyang, cuileyang, liuhanmeng, zhangyue}@westlake.edu.cn

Abstract

Aspect category sentiment analysis has attracted increasing research attention. The dominant methods make use of pre-trained language models by learning effective aspect category-specific representations, and adding specific output layers to its pre-trained representation. We consider a more direct way of making use of pre-trained language models, by casting the ACSA tasks into natural language generation tasks, using natural language sentences to represent the output. Our method allows more direct use of pre-trained knowledge in seq2seq language models by directly following the task setting during pre-training. Experiments on several benchmarks show that our method gives the best reported results, having large advantages in few-shot and zero-shot settings.

1 Introduction

Aspect-based sentiment analysis (ABSA) is a fine-grained sentiment analysis task that includes a number of subtasks, two of which are aspect category sentiment analysis (ACSA) and aspect category detection (ACD). Figure 1 shows an example, where the input is "*The restaurant was expensive, but the menu was great*". ACD detects the aspect categories, such as *price* and *food*, and ACSA predicts the sentiment polarities toward each aspect category. In this work, we focus on these two tasks as well as the joint task that combines both.

Previous studies have investigated various methods that treat ACSA and ACD as classification tasks, learning aspect-specific sentence representations (Wang et al., 2016; Ruder et al., 2016). Recently, pre-trained language models (PLM) have shown their effectiveness to this end (Jiang et al., 2019). The main idea is to make use of pre-trained models such as BERT (Devlin et al., 2019a) for representing an aspect-specific form of the input (e.g., by concatenating the aspect category to the end of the input sentence (Figure 3(a))), which provides useful semantic features for ACSA and ACD classifiers. Such methods have given highly competitive results (Sun et al., 2019; Li et al., 2020b).

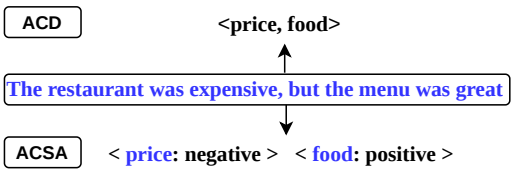


Figure 1: Example of aspect category detection (ACD) and aspect category sentiment analysis (ACSA).

The above classification models benefit from contextualized representations, which contain knowledge learned by pre-training over large data (Lin et al., 2019). However, their use of pre-trained knowledge can be viewed as indirect due to at least two reasons. First, the classification task is performed by using a neural network on top of pre-trained representation, with separate network parameters. Second, the integration of aspect category makes the aspect-specific input representation not exactly a natural language sentence, which differs from the pre-training setting. Intuitively, more pre-trained knowledge could be leveraged by connecting pre-training and ACSA at the *task* level, rather than only at the *representation* level.

We investigate the above potentials by casting the sentiment classification tasks into language modelling tasks. In particular, as shown in Figure 2, both ACSA and ACD are transformed into sequence-to-sequence (seq2seq) tasks, where the encoder takes the input sentence and the decoder generates a natural language sentence. For ACD, the output follows a template stating whether the specific aspect is discussed (e.g., "*The* ⟨`category_type`⟩ *category is discussed*"); for ACSA, the sentiment polarity of a specific aspect is stated (e.g., "*The sentiment polarity of* ⟨`given_category`⟩ *is* ⟨`polarity_type`⟩"). The setting corresponds closely to the denoising auto-

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 4406–4416
November 7–11, 2021.

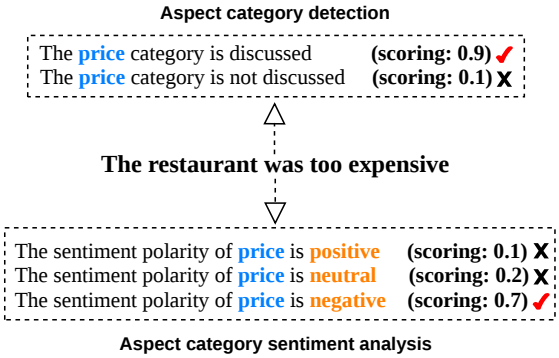


Figure 2: ACSA as a generation task.

encoder training scheme of BART (Lewis et al., 2020), which we use as the pre-trained model. Compared with classification-based methods, our method does not include more network parameters, and thus can potentially generalize better to new domains (Brown et al., 2020; Gao et al., 2020). Given a new domain with completely unseen aspect categories and sentiment labels, our method can be applied without changing output layer structure.

In addition to classification-based methods, we take masked language models (MLM) as a baseline also, for which a natural counterpart of our method is a mask-refilling task. As shown in Figure 3(b), different from our method, the output template is concatenated to the input, with the keyword being masked for prediction. This MLM task corresponds closely to BERT (Devlin et al., 2019a) pre-training. In comparison to this MLM method, a generation method can better learn the correlation between the input and output template as two related sequences, which has been demonstrated by the strong performance of BART for abstractive text summarization (Lewis et al., 2020).

Experimental results on three standard benchmarks datasets show that both generation and MLM methods outperform classification methods using the same pre-trained language models. Finally, generation methods give stronger performances than MLM methods, outperforming the previous state-of-the-art methods by a large margin. In addition, using the generation method, we show that jointly performing ACSA and ACD leads to better results than the traditional pipeline. To our knowledge, we are the first to employ a generative pre-trained language model to address an ACSA/ACD problem. We release our code at `https://github.com/lgw863/ACSA-generation`.

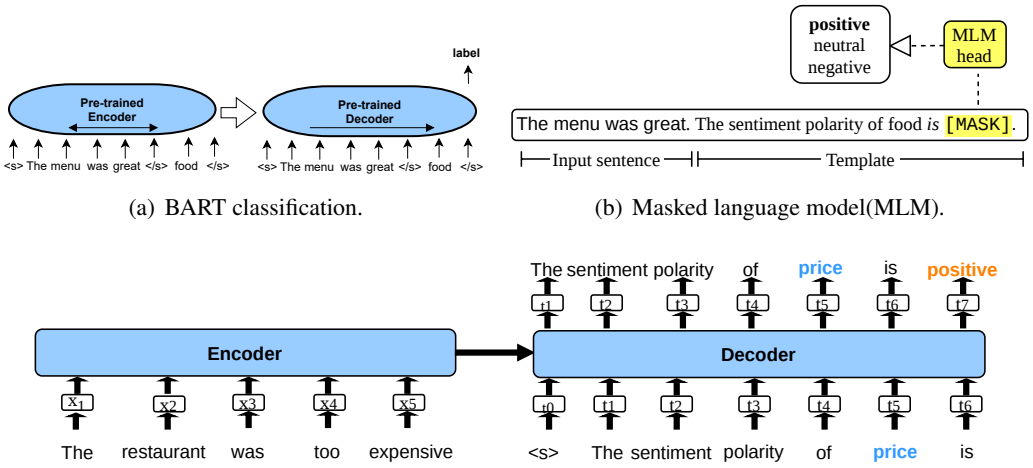
2 Related Work

Aspect Category Sentiment Analysis Wang et al. (2016) propose an attention-based LSTM network, which can concentrate on different parts of a sentence when different aspect categories are taken as input. Ruder et al. (2016) model the interdependencies of sentences in a text with a hierarchical bidirectional LSTM. Yin et al. (2017) model the task as a machine comprehension problem by constructing pseudo question-answer pairs. Xue and Li (2018) extract sentiment features with CNN and selectively output aspect category related features with gating mechanisms. Xing et al. (2019), Liang et al. (2019) and Zhu et al. (2019) incorporate aspect category information into sentence encoders in the context modeling stage. Sun et al. (2019) construct auxiliary sentences from the aspect categories and convert ACSA to a sentence-pair classification task. Li et al. (2020b) predict the sentiment of an aspect category mentioned in a sentence by aggregating the sentiments of the words indicating the aspect category in the sentence.

Several joint models were proposed to avoid error propagation, which perform ACD and ACSA jointly. Schmitt et al. (2018) propose two joint models: end-to-end LSTM and end-to-end CNN, which produce all the aspect categories and their corresponding sentiment polarities at once. Hu et al. (2019) propose constrained attention networks (CAN) to constrain the attention weight allocation. Wang et al. (2019) propose the aspect-level sentiment capsules model (AS-Capsules), which utilizes the correlation between aspect category and sentiment through shared components. Li et al. (2020a) propose a novel joint model which contains a shared sentiment prediction layer.

All the models above are classification methods, which use a separate output network to give the output label. In contrast, we investigate natural language generation methods by directly following the pre-training process of language models.

Masked Language Model Methods There is a line of work using the masked language model (MLM) for natural language understanding tasks. The basic idea is to leverage information from pre-trained models by defining specific sentence prompt in a language modelling task. Brown et al. (2020) use prompt for few-shot learning in text classification tasks. Schick and Schütze (2020) rephrase inputs as cloze questions for text classification. Schick et al. (2020) and Gao et al. (2020)



(a) BART classification.

(b) Masked language model(MLM).

(c) BART generation.

Figure 3: A comparison of aspect category sentiment analysis methods.

extend Schick and Schütze (2020) by automatically generating label words and templates, respectively. Petroni et al. (2019) extract relation between entities from BERT by constructing cloze-style templates. We are the first to apply such methods to ACSA, taking it as a baseline. Different from these template-based models, our final model uses BART for text generation, which better models the correlations between the input sentence and the output sentence compared with BERT.

Generation Methods There has been work casting NLP problems as sequence generation tasks (Vinyals et al., 2015; Ma et al., 2017; Stanovsky and Dagan, 2018; Raffel et al., 2020), where the output is a sequence of tokens rather than a natural language sentence. Daza and Frank (2018) treat semantic role labelling as a sequence-to-sequence process. Li et al. (2019) solve the entity-relation extraction task as a multi-turn question answering generation method. Our work is similar in casting an NLP task as a generation task. Different from the above methods, our goal is to make the most of pre-trained knowledge in BART for ACSA.

3 Methods

Formally for ACD, the input is a sentence $\mathbf{X} = \{x_1, \ldots, x_n\} = x_{1:n}$, where x_i denotes the i-th word. For ACSA, a set of pre-identified aspect categories are also given. We introduce relevant pre-trained language models in 3.1, classification methods in Section 3.2, MLM methods in Section 3.3, and our generation method in Section 3.4.

3.1 Pre-trained language Models

We take BERT (Devlin et al., 2019a) and BART (Lewis et al., 2020) as the pre-trained language models. Both are built on the Transformer (Vaswani et al., 2017) architecture. BERT (Devlin et al., 2019a) is an encoder stack of Transformer for masked text filling, where a model uses the context words to predict masked words. BART (Lewis et al., 2020) is a denoising auto-encoder seq2seq model pre-training for natural language generation. Its training applies document corruption such as randomly deleting tokens from the input and corrupting text with an arbitrary noising function. BART is trained to reconstruct the original text.

3.2 The Classification Method

We use a multi-layer perceptrons network as the classifier model, which takes a representation vector as input. Both BERT and BART are considered as the encoders.

BERT Classification BERT adopts "*[CLS] input sentence [SEP] given_category [SEP]*" as input. The final hidden state corresponding to "[CLS]" is used as the representation for classification.

BART Classification BART adopts "$\langle$S$\rangle$ *input sentence* $\langle$/S$\rangle$ *given_category* $\langle$/S$\rangle$" as input and predicts the sentiment polarity of the sentence towards the given category. The same input is fed into the encoder and decoder (see Figure 3(a)). Formally, suppose that the query category is a, $x_0 = \langle$S$\rangle$, $x_{n+1} = \langle$/S$\rangle$, $x_{n+2} = a$, $x_{n+3} = \langle$/S$\rangle$, then the input to BART is $x_{0:n+3} = \langle$S$\rangle$ $x_1, \ldots, x_n$ $\langle$/S$\rangle$ a $\langle$/S$\rangle$. The output hidden vec-

tors obtained by the BART encoder (ENCODER) and BART decoder (DECODER) are:

$$\mathbf{h}^{enc} = \text{ENCODER}(x_{0:n+3})$$

$$\mathbf{h}_0 \dots \mathbf{h}_{n+3} = \text{DECODER}(\mathbf{h}^{enc}; x_{0:n+3})$$

The output vector $\mathbf{h}_{n+3}$ is then taken as the representation vector for classification.

3.3 The MLM Method

Masked language models (MLM) (Devlin et al., 2019a) complete a given prompt by filling missing tokens. We refer to the template including a given category and MASK token together as a prompt. For sentiment analysis tasks, *BERT MLM* adopts the input sentence and the prompt as the model input and predicts the sentiment polarity label word towards the given category. For *BART MLM*, the same input is fed into the encoder and decoder, and the highest decoder prediction from label words of the MASK token is the predicted polarity label(see Figure 3(b)). We use the same template in the MLM method and generation method, following the template creation method in section 3.4.1.

3.4 The Generation Method

We take both ACSA and ACD as language model ranking problems under a seq2seq framework (see Figure 3(c)). The target sequence $\mathbf{T}_{a_i,p_k}(\mathbf{T}_{a_i}) = \{t_1, \dots, t_m\}$ is a template filled by the given category a_i and the polarity type p_k. We first introduce how to create templates in Section 3.4.1, and then show the inference and training details in Section 3.4.2 and Section 3.4.3, respectively.

3.4.1 Template Creation

For ACSA, we manually create templates containing one slot for the `given_category` and another slot for the `polarity_type` label. We set a category word set $\mathbf{A} = \{a_1, \dots, a_{|C|}\}$, $|C|$ is the category type size (e.g., a_i="*price*") and polarity type word set $\mathbf{P} = \{p_1, \dots, p_{|L|}\}$, $|L|$ is the polarity type size (e.g., p_k="*positive*"), and use words to define templates $\mathbf{T}_{a_i,p_k}$ (e.g. "*The sentiment polarity of price is positive*"). The template $\mathbf{T}$ is "*The sentiment polarity of* $\langle a_i \rangle$ *is* $\langle p_k \rangle$". For a given category a_i, we can obtain a list of templates $\mathbf{T}_{\mathbf{a_i}} = [\mathbf{T}_{a_i,p_1}, \dots, \mathbf{T}_{a_i,p_{|L|}}]$.

For ACD, we use a_i to create a sentiment template $\mathbf{T}^+_{a_i}$ for an existing aspect category, and a none-category template $\mathbf{T}^-_{a_i}$. $\mathbf{T}^+$ is "*The* $\langle a_i \rangle$ *category is discussed*" and $\mathbf{T}^-$ is "*The* $\langle a_i \rangle$ *category is not discussed*".

3.4.2 Inference

For ACSA, we first enumerate all possible polarities for the given category of the sentence $\mathbf{X}$ and fill them in the prepared templates, and then use the fine-tuned pre-trained generative language model to assign a score for each template $\mathbf{T}_{a_i,p_k} = \{t_1, \dots, t_m\}$, formulated as:

$$f(\mathbf{T}_{a_i,p_k}) = \sum_{c=1}^{m} \log P(t_c | t_{1:c-1}, \mathbf{X}) \quad (1)$$

We calculate a score $f(\mathbf{T}_{a_i,p_k})$ for each possible polarity by employing the pre-trained generative language model (i.e., BART) to score the templates, and then choose the polarity of category a_i with the largest score.

For ACD, we first create templates $\mathbf{T}^+_{a_i}$ and $\mathbf{T}^-_{a_i}$ for all possible categories of the sentence $\mathbf{X}$, and then use the fine-tuned pre-trained generative language model to assign a score for each template $\mathbf{T}_{a_i} = \{t_1, \dots, t_m\}$, in a similar way as Equation 1. Also, we decide whether the a_i category is discussed or not in the input sentence according to the higher score between $\mathbf{T}^+_{a_i}$ and $\mathbf{T}^-_{a_i}$.

3.4.3 Training

For ACSA, suppose that the polarity type of a_i is p_k. We fill the given category a_i and the polarity type p_k into template $\mathbf{T}$ to create a gold target output $\mathbf{T}_{a_i,p_k}$. Similarly for ACD, if the category of a_i is discussed, the gold target $\mathbf{T}^+_{a_i}$ is obtained by filling a_i into $\mathbf{T}^+$, and otherwise is $\mathbf{T}^-_{a_i}$.

For ACSA, we use all gold polarities in the training set to construct $(\mathbf{X}, \mathbf{T})$ pairs. For ACD, we use all gold categories in the training set to construct $(\mathbf{X}, \mathbf{T}^+)$ pairs, and additionally create negative samples $(\mathbf{X}, \mathbf{T}^-)$ by sampling all none existing categories in the input. Finally, we obtain $\{(\mathbf{X}, \mathbf{T})\} = \{(\mathbf{X}, \mathbf{T}^+) \cup (\mathbf{X}, \mathbf{T}^-)\}$

Given a sequence pair $(\mathbf{X}, \mathbf{T})$, we feed the input $\mathbf{X} = x_{1:n}$ to the BART encoder, obtaining hidden representations of the sentence:

$$\mathbf{h}^{enc} = \text{ENCODER}(x_{1:n}) \quad (2)$$

At the c th step of the decoder, $\mathbf{h}^{enc}$ and previous output tokens $t_{1:c-1}$ are then as inputs, yielding a representation using attention (Vaswani et al., 2017)

$$\mathbf{h}^{dec}_c = \text{DECODER}(\mathbf{h}^{enc}, t_{1:c-1}) \quad (3)$$

The conditional probability of the word t_c is defined as:

$$P(t_c | t_{1:c-1}, \mathbf{X}) = \text{SOFTMAX}(\mathbf{h}^{dec}_c \mathbf{W}_{lm} + \mathbf{b}_{lm}), \quad (4)$$

where $\mathbf{W}_{lm} \in \mathbb{R}^{d_h \times |\mathcal{V}|}$ and $\mathbf{b}_{lm} \in \mathbb{R}^{|\mathcal{V}|}$, $|\mathcal{V}|$ represents the vocab size of pre-trained BART. The

cross-entropy between the decoder's output and the original template is used as the loss function:

$$\mathcal{L} = -\sum_{c=1}^{m} \log P(t_c | t_{1,c-1}, \mathbf{X}) \quad (5)$$

4 Experiments

We choose the SemEval-2014 restaurant review (Rest14) (Pontiki et al., 2014a), a variant of Rest14 (Rest14-hard) (Xue and Li, 2018) and the multi-aspect multi-sentiment (MAMS) (Jiang et al., 2019) datasets for sentence-level sentiment , the TripAdvisor (Wang et al., 2010) and BeerAdvocate (McAuley et al., 2012; Lei et al., 2016) datasets for document-level sentiment. Standard splits of training/development/testing sets are adopted following previous work Tay et al. (2018), the details of which are shown in Appendix A.

We use the pre-trained BERT-base[1] and BART-base[2] models for task fine-tuning. We select the fine-tuning learning rate from {4e-5, 2e-5, and 1e-5} and batch size from {8, 16, 24} for different models. The dropout probability is 0.1. The best model configuration is selected according to the highest performance on the development set. The details of settings are shown in Appendix A.

4.1 Baseline Methods

We compare our generation method with classification and MLM baselines (Figure 3) using the same encoder. In particular, *BART generation* (i.e., Figure 3(c)) is compared with *BART classification* (Figure 3(a)) and *BART MLM* (Figure 3(b)), as well as *BERT classification* and *BERT MLM*. In addition, our method is also compared with other models in the literature as follows.

For sentence-level ACSA, we also compare our method with the following state-of-the-art methods in the literature. (1) non-BERT models: GCAE (Xue and Li, 2018), As-capsule (Wang et al., 2019) and CapsNet (Jiang et al., 2019); (2) BERT (Devlin et al., 2019b) based models: BERT-pair-QA-B (Sun et al., 2019), CapsNet-BERT (Jiang et al., 2019) and AC-MIMLLN-BERT (Li et al., 2020b).

For document-level ACSA, we compare our method with the following methods. (1) non-BERT models: LSTM (Tang et al., 2015), HAN (Yang et al., 2016) and MR (machine comprehension pattern) (Yin et al., 2017); (2) BERT (Devlin et al., 2019b) based model: *BERT classification*.

For ACD, we compare our method with the following methods. (1) non-BERT models: XRCE (Brun et al., 2014), NRC-Canada (Kiritchenko et al., 2014); (2) BERT (Devlin et al., 2019b) based models: *BERT classification*, BERT-pair-NLI-B (Sun et al., 2019) , CNE-net (Dai et al., 2020).

ACSA Template $\mathbf{T}$	Dev accuracy
The sentiment polarity of $\mathbf{a}_i$ is $\mathbf{p}_k$	**83.78**
The sentiment is $\mathbf{p}_k$ for $\mathbf{a}_i$	83.44
The $\mathbf{a}_i$ category has a $\mathbf{p}_k$ label	82.31

Table 1: ACSA results using different templates. $\mathbf{a}_i$ indicates given category, $\mathbf{p}_k$ indicates polarity type.

ACD Template $\mathbf{T}^+/\mathbf{T}^-$	Dev F1
The $\mathbf{a}_i$ category is discussed The $\mathbf{a}_i$ category is not discussed	**93.13**
The sentence discusses the $\mathbf{a}_i$ category The sentence discusses no $\mathbf{a}_i$ category	92.67
It is about the $\mathbf{a}_i$ category It is not about the $\mathbf{a}_i$ category	92.44

Table 2: ACD results using different templates. $\mathbf{a}_i$ indicates category type.

4.2 Development Experiments

Different templates can be used for expressing the same meaning. For instance, "*The sentiment polarity of* ⟨given_category⟩ *is positive*" can also be expressed by "*The sentiment is positive for* ⟨given_category⟩". For ACSA, we investigate the impact of manual templates using the MAMS development set. Table 1 shows the impact of different choice of templates. For instance, "*The* ⟨given_category⟩ *category has a* ⟨polarity_type⟩ *label*" and "*The sentiment polarity of* ⟨given_category⟩ *is* ⟨polarity_type⟩" give 82.31% and 83.78% accuracy, respectively, indicating that the template has influence on the final performance. This is consistent with finds of Gao et al. (2020) for the few-shot task. Based on the development results, we use the top performing template "*The sentiment polarity of* ⟨given_category⟩ *is* ⟨polarity_type⟩" in our ACSA experiments.

For ACD, we investigate the impact of templates using the Rest14 development set. Table 2 shows the performance impact of different templates. We use the top performing template "*The* ⟨category_type⟩ *category is discussed*" as template $\mathbf{T}^+$ and "*The* ⟨category_type⟩ *category is not discussed*" as template $\mathbf{T}^-$ in our ACD experiments.

[1] https://github.com/google-research/bert

[2] https://huggingface.co/facebook/bart-base/tree/main

Category	Model	Rest14	Rest14-hard	MAMS
Classification w/o PLM	GCAE (Xue and Li, 2018)	81.336(±0.883)	54.717(±4.920)	72.098†
	As-capsule (Wang et al., 2019)	82.179(±0.414)	60.755(±2.773)	75.116(±0.473)
	CapsNet (Jiang et al., 2019)	81.172(±0.631)	53.962(±0.924)	73.986†
	AC-MIMLLN (Li et al., 2020b)	81.603(±0.715)	65.283(±2.264)	76.427(±0.704)
Classification w PLM	BERT classification	87.482(±0.906)	67.547(±5.894)	78.292†
	BART classification	88.289(±0.943)	68.698(±3.407)	78.761(±0.752)
	BERT-pair-QA-B (Sun et al., 2019)	87.523(±1.175)	69.433(±4.368)	79.134(±0.973)
	CapsNet-BERT (Jiang et al., 2019)	86.557(±0.943)	51.321(±1.412)	79.461†
	AC-MIMLLN-BERT (Li et al., 2020b)	89.250(±0.720)	74.717(±3.290)	81.198(±0.606)
Masked language model	BERT MLM	88.446(±0.825)	69.021(±2.753)	79.019(±0.935)
	BART MLM	88.667(±0.768)	69.585(±2.529)	79.243(±0.854)
Generation	BART generation	**90.545(±0.315)***	**77.358(±2.160)***	**83.130(±0.478)***

Table 3: Results of the sentence-level ACSA in terms of accuracy (%, mean±(std)). † refers to Jiang et al. (2019). * means the result is significant at $p < 0.01$ using paired t-test comparing to *BART MLM* and *BART classification*.

Model	TripAdvisor	BeerAdvocate
LSTM	44.02	34.78
HAN	44.68	36.03
MR	46.56	38.06
BERT classification	47.03	39.85
BART classification	47.45	40.44
BERT MLM	48.03	40.58
BART MLM	48.36	40.72
BART generation	**49.51***	**41.42***

Table 4: Results of the document-level ACSA in terms of accuracy (%). * means the result is significant at $p < 0.01$ using paired t-test comparing to *BART MLM* and *BART classification*.

Model	P	R	F1
XRCE	83.23	81.37	82.29
NRC-Canada	91.04	86.24	88.58
BERT classification	92.78	89.07	90.89
BERT-pair-NLI-B	93.57	**90.83**	92.18
CNE-net	93.76	**90.83**	92.27
BART classification	93.01	89.92	91.44
BART MLM	93.44	89.83	91.60
BART generation	**95.18**	90.54	**92.80**

Table 5: Rest14 results: Aspect Category Detection. We use the results reported in XRCE (Brun et al., 2014), NRC-Canada (Kiritchenko et al., 2014), BERT-pair-NLI-B (Sun et al., 2019) and CNE-net (Dai et al., 2020).

4.3 ACSA Experiments

The results of sentence-level ACSA are shown in Table 3. We can see that, first, the performance of *BERT MLM* and *BART MLM* is better than *BERT classification* and *BART classification*, respectively. In particular, *BERT MLM* gives a strong baseline, outperforming all non-BERT and *BERT classification* baselines. This shows that making use of pre-training at the *task* level can achieve better results than that at the *representation* level. Also, the BART *MLM* and *classification* models perform better than the corresponding BERT models. Second, *BART generation* outperforms all baselines on all three datasets, which indicates that our model can better detect multiple sentiment polarities in one sentence toward different aspect categories. Third, *BART generation* performs significantly better than *BART MLM*, giving absolutely 3.89% stronger accuracy on MAMS, demonstrating the effectiveness of the generation method. This shows the strength of BART pre-training for generating semantically related content, which was also reflected by the strong performance of BART on abstractive summarization (Lewis et al., 2020). In contrast, the MLM method concatenates the input and output into one sequence, and thus fails to model their correlation in encoder-decoder pre-trainng.

The performances of our model on document-level ACSA are shown in Table 4. Compared with LSTM, HAN and MR, *BERT classification* and *BART classification* outperform all baselines, which shows the effectiveness of pre-training. *BERT MLM* and *BART MLM* surpass *BERT classification* and *BART classification*, respectively. Our *BART generation* model achieves improvements of 1.15% and 0.70% over *BART MLM* on TripAdvisor and BeerAdvocate, respectively, demonstrating that the generation method can more effectively make use of BART for ACSA.

4.4 ACD Experiments

Results on the Rest14 ACD subtask are presented in Table 5. Following Pontiki et al. (2014b), we use Micro-F1 for evaluating. Again *BART generation* achieves better results than *BART classification* and *BART MLM*. Our model outperforms all baselines

Model	Rest14			MAMS		
	P	R	F1	P	R	F1
Pipeline BART generation	82.03	76.46	79.15	77.04	71.92	74.39
Joint BERT classification	77.75	76.07	76.90	74.14	71.92	73.01
Joint BART classification	81.92	73.59	77.53	74.59	74.13	74.36
Joint BART MLM	81.88	76.73	79.22	75.32	75.07	75.19
Joint BART generation	**82.76**	**81.91**	**82.33**	**77.18**	**76.58**	**76.88**

Table 6: Performance on combination setting.

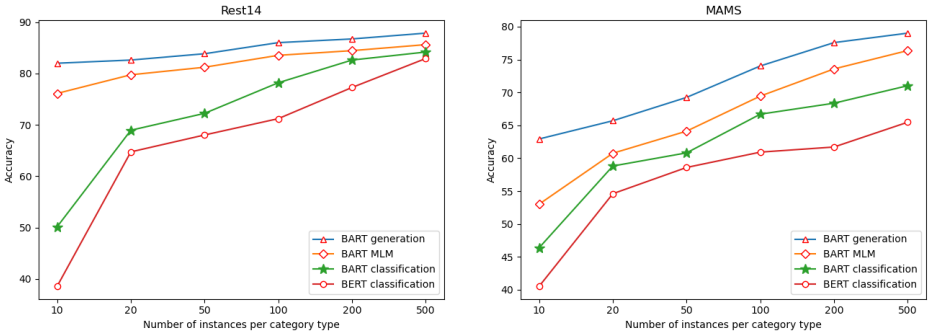


Figure 4: Few-shot ACSA performance on different test sets.

Model	P	R	F1
BERT classification	90.50	86.68	88.50
BART classification	90.67	88.34	89.49
BART MLM	90.57	88.86	89.71
BART generation	**90.71**	**90.16**	**90.43**

Table 7: MAMS results: Aspect Category Detection.

Model	R → M	M → R
BERT classification	43.38	62.28
BART classification	46.61	68.55
BART MLM	47.86	70.64
BART generation	**49.84**	**72.46**

Table 8: Zero-Shot results: ACSA. R → M indicates training on Rest14 and testing on MAMS. M → R indicates training on MAMS and testing on Rest14.

on precision and F-1 score. In particular, a more than 95% precision score is achieved, which shows that our model can effectively exclude the aspect categories not mentioned in the input.

We also investigate the performance on the MAMS dataset, which consists of at least two unique aspect categories with different sentiment polarities in each input sentence. Table 7 shows that *BART generation* outperforms all baselines, indicating better ability of our model to detect multiple aspect categories in one sentence.

4.5 A Joint Model

The generation method allows us to build a straightforward joint model by extending the first template in Table 1, using "*The sentiment polarity of <given_category> is none*" as a template for non-existing aspect categories. The results on Rest-14 and MAMS are presented in Table 6. We find that joint *BART generation* achieves better results on this task with improvements over pipeline *BART generation*. Joint *BART generation* outperforms all baselines on precision, recall and F-1 score, which shows the advantage of joint learning.

4.6 Few-Shot and Zero-Shot Learning

We evaluate the model performance on ACSA where only a small amount of labelled data is available for training, simulating the low-resource data scenarios by randomly sampling training instances from a large training set. In particular, we use different numbers of instances for training, randomly sampling a fixed number of instances per category type (10, 20, 50, 100, 200, 500 instances per category type for Rest14 and MAMS). The results are shown in Figure 4, where the methods of *BERT classification*, *BART classification* and *BART MLM* are also compared.

It can be seen that on all the datasets, our model outperforms *BERT classification*, *BART classification* and *BART MLM*, especially when the number of training instances is small. For example, when there are only 10 training instances, our model gives accuracy scores of 82.01% on Rest14, as compared to 38.57% by *BERT classification* and 50.16% by *BART classification*. When the number of instances grows as large as 500, our model gives 2.24% and 2.65% better accuracies than *BART MLM* on Rest14 and MAMS, respectively. One

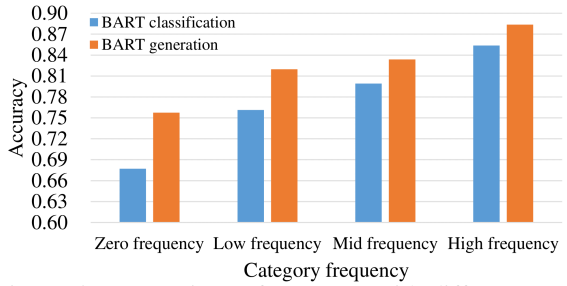


Figure 5: Comparison of accuracy with different category frequency on MAMS.

possible reason is that our method makes more use of direct sentiment knowledge in the pre-trained language model by directly adopting the original structure of BART mentioned earlier. In contrast, classification methods cannot achieve this due to transferring the sentiment bias indirectly.

The results of our zero-shot learning experiments are in Table 8. In all cases, our method outperforms all the baselines. In particular, the model trained on MAMS has a better performance on Rest14 than the reverse zero-shot setting, which proves that the MAMS dataset has a higher challenge.

5 Analysis

5.1 Influence of Category Frequency

Aspect categories can be implicit and do not necessarily occur as terms in the given sentence. To explore the correlation between ACSA accuracy and the occurrence frequency of a given category, we split the eight categories in the MAMS test set into four subsets based on the occurrence frequency. The category (i.e., *miscellaneous*) that never occurs in the given sentence is put into the *zero frequency* subset, the 15% least frequent (i.e., *ambience*, *staff*) are put into *low frequency* subset, the 30% most frequent (i.e., *menu*, *service*) are put into *high frequency* subset, and the remaining (i.e., *price*, *food*, *place*) are put into *mid frequency* subset.

Figure 5 shows the accuracy of *BART classification* and our model against the frequency. As the category occurrence frequency decreases, the relative gap of accuracy between the two models increases. In the *zero frequency*, our method gives absolutely 8.03% stronger accuracy than *BART classification*. This demonstrates that our method is more robust in summarizing the sentiment polarity of abstract or rare categories. Even if there are no explicit category terms in the sentence, the generation method can give the implicit category opinion of the whole sentence according to the context.

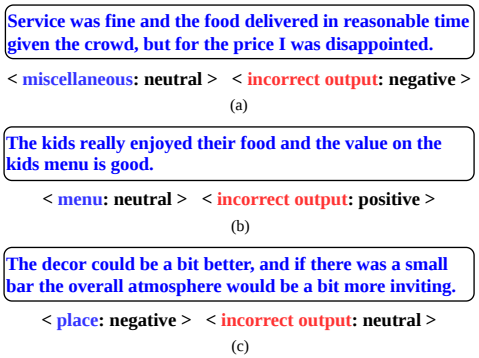


Figure 6: Examples of *BART classification*. (a) is an instance with category do not occur as term in sentence. (b) represents that our method is not affected by the surrounding interference information. (c) needs conditional reasoning for analysis. Our method can obtain correct sentiment polarity.

5.2 Case Study

Figure 6 shows typical examples from the test set which cannot be inferred by the *BART classification* model. In sentence (a), the given category *miscellaneous* does not occur as a term in the given sentence. Our method can synthesize different sentiment polarities with different aspects to obtain correct polarity. In sentence (b), "*the value on the kids menu is good*", *good* modifies *the value*, rather than the given category *menu*. Our method gives the correct polarity, not being affected by the surrounding other aspect sentiments. The last instance (c) has conditional reasoning which is difficult for *BART classification*. In contrast, *BART generation* gives the correct label by correctly recognizing the negativity in "*if there was ... would be a bit more inviting*". This is likely because our method makes use of pre-trained knowledge to infer the inter-sentential correlations between the input and the output sequences, which the *BART classification* model failed to achieve due to the indirect use of BART in the additional classification network.

6 Conclusion

We investigated a generation method for aspect category detection (ACD) and aspect category sentiment analysis (ACSA), which can make better use of BART's advantages in making semantic level summaries to the input by not introducing additional model parameters. Experiments show that our proposed method obtains superior performance over the baseline models for both sentence-level and document-level aspect sentiment analysis. In contrast to the traditional sentiment classification methods, our method is also more powerful on zero-shot and few-shot tasks.

Acknowledgements

Zhiyang Teng is the corresponding author. We would like to thank the anonymous reviewers for their insightful comments. We gratefully acknowledge funding from the National Natural Science Foundation of China (NSFC No.61976180).

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Caroline Brun, Diana Nicoleta Popa, and Claude Roux. 2014. XRCE: hybrid classification for aspect-based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 838–842. The Association for Computer Linguistics.

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 97–106.

Zehui Dai, Cheng Peng, Huajie Chen, and Yadong Ding. 2020. A multi-task incremental learning framework with category name embedding for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 6955–6965. Association for Computational Linguistics.

Angel Daza and A. Frank. 2018. A sequence-to-sequence model for semantic role labeling. In *Rep4NLP@ACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners.

Mengting Hu, Shiwan Zhao, Li Zhang, Keke Cai, Zhong Su, Renhong Cheng, and Xiaowei Shen. 2019. Can: Constrained attention networks for multi-aspect sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4593–4602.

Qingnan Jiang, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang. 2019. A challenge dataset and effective models for aspect-based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6281–6286.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 437–442. The Association for Computer Linguistics.

Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 107–117. The Association for Computational Linguistics.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 1340–1350. Association for Computational Linguistics.

Yuncong Li, Zhe Yang, Cunxiang Yin, Xu Pan, Lunan Cui, Qiang Huang, and Ting Wei. 2020a. A joint model for aspect-category sentiment analysis with

shared sentiment prediction layer. In *Proceedings of the 19th Chinese National Conference on Computational Linguistics*, pages 1112–1121, Haikou, China. Chinese Information Processing Society of China.

Yuncong Li, Cunxiang Yin, Sheng-hua Zhong, and Xu Pan. 2020b. Multi-instance multi-label learning networks for aspect-category sentiment analysis. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3550–3560. Association for Computational Linguistics.

Yunlong Liang, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019. A novel aspect-guided deep transition model for aspect based sentiment analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5572–5584.

Yongjie Lin, Yi Chern Tan, and Robert Frank. 2019. Open sesame: Getting inside BERT's linguistic knowledge. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 241–253, Florence, Italy. Association for Computational Linguistics.

Chunpeng Ma, L. Liu, Akihiro Tamura, T. Zhao, and E. Sumita. 2017. Deterministic attention for sequence-to-sequence constituent parsing. In *AAAI*.

Julian J. McAuley, Jure Leskovec, and Dan Jurafsky. 2012. Learning attitudes and attributes from multi-aspect reviews. *CoRR*, abs/1210.3926.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014a. SemEval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland. Association for Computational Linguistics.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014b. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 27–35. The Association for Computer Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Sebastian Ruder, Parsa Ghaffari, and John G Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 999–1005.

Timo Schick, Helmut Schmid, and Hinrich Schütze. 2020. Automatically identifying words that can serve as labels for few-shot text classification. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 5569–5578, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few shot text classification and natural language inference.

Martin Schmitt, Simon Steinheber, Konrad Schreiber, and Benjamin Roth. 2018. Joint aspect and polarity classification for aspect-based sentiment analysis with end-to-end neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1109–1114.

Gabriel Stanovsky and Ido Dagan. 2018. Semantics as a foreign language. In *EMNLP*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 380–385.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1422–1432. The Association for Computational Linguistics.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, pages 5998–6008. Curran Associates, Inc.

Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey E. Hinton. 2015. Grammar as a foreign language. In *NIPS*.

Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: a rating regression approach. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 25-28, 2010*, pages 783–792. ACM.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615.

Yequan Wang, Aixin Sun, Minlie Huang, and Xiaoyan Zhu. 2019. Aspect-level sentiment analysis using as-capsules. In *The World Wide Web Conference*, pages 2033–2044.

Bowen Xing, Lejian Liao, Dandan Song, Jingang Wang, Fuzheng Zhang, Zhongyuan Wang, and Heyan Huang. 2019. Earlier attention? aspect-aware lstm for aspect sentiment analysis. *arXiv preprint arXiv:1905.07719*.

Wei Xue and Tao Li. 2018. Aspect based sentiment analysis with gated convolutional networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2514–2523.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 1480–1489.

Yichun Yin, Yangqiu Song, and Ming Zhang. 2017. Document-level multi-aspect sentiment classification as machine comprehension. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pages 2044–2054.

Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect aware learning for aspect category sentiment analysis. *ACM Trans. Knowl. Discov. Data*, 13(6).

A Datasets

A.1 Sentence-Level Datasets

Rest14 (Pontiki et al., 2014a) Following previous work (Cheng et al., 2017; Tay et al., 2018; Hu et al., 2019), we remove samples with conflict polarities. Since there is no official development set for Rest14, we use the split offered by Tay et al. (2018).

Dataset		Pos.	Neg.	Neu.
Rest14	Train	1855	733	430
	Dev	324	106	70
	Test	657	222	94
Rest14-hard	Test	21	20	12
MAMS-ACSA	Train	1929	2084	3077
	Dev	241	259	388
	Test	245	263	393

Table 9: Statistics of the sentence-level datasets.

Dataset	#docs	#words/doc	words/sent
TripAdvisor	29,391	251.7	18.0
BeerAdvocate	51,020	144.5	12.1

Table 10: Statistics of the document-level datasets. The rating scale of TripAdvisor dataset is 1-5. The rating scale of BeerAdvocate dataset is 1-10.

Rest14-hard Following Xue and Li (2018), we construct Rest14-hard, where the training set and development set are the same as Rest14's, while test set is constructed from the test set of Rest14. The test set of Rest14-hard only includes sentences containing at least two aspect categories with different sentiment polarities.

MAMS Jiang et al. (2019) Since the test set of Rest14-hard is small, we also adopt the Multi-Aspect Multi-Sentiment dataset for Aspect Category Sentiment Analysis (denoted by MAMS). All sentences in MAMS contain multiple aspect categories with different sentiment polarities.

A.2 Document-Level Datasets

TripAdvisor (Wang et al., 2010) and BeerAdvocate (McAuley et al., 2012; Lei et al., 2016) contain seven aspects (value, room, location, cleanliness, check in/front desk, service, and business service) and four aspects (feel, look, smell, and taste) respectively. We randomly split them into training, development, and testing sets with 80/10/10%.

Statistics of these three sentence-level datasets are given in Table 9 and two document-level datasets are described in Table 10.

B Settings

Each method is trained for 30 epochs, during which the model with the best performance on the validation set is saved. We also apply early stopping in training, which means that the training will stop if the performance on validation set does not improve in 5 epochs.