

Conditional Augmentation for Aspect Term Extraction via Masked Sequence-to-Sequence Generation

Kun Li¹, Chengbo Chen¹, Xiaojun Quan^{1*}, Qing Ling¹, Yan Song²

¹School of Data and Computer Science, Sun Yat-sen University, China

²Sinovation Ventures

{likun36, chenchb7}@mail2.sysu.edu.cn

{quanxj3, lingqing556}@mail.sysu.edu.cn, clksong@gmail.com

Abstract

Aspect term extraction aims to extract aspect terms from review texts as opinion targets for sentiment analysis. One of the big challenges with this task is the lack of sufficient annotated data. While data augmentation is potentially an effective technique to address the above issue, it is uncontrollable as it may change aspect words and aspect labels unexpectedly. In this paper, we formulate the data augmentation as a conditional generation task: generating a new sentence while preserving the original opinion targets and labels. We propose a masked sequence-to-sequence method for conditional augmentation of aspect term extraction. Unlike existing augmentation approaches, ours is controllable and allows us to generate more diversified sentences. Experimental results confirm that our method alleviates the data scarcity problem significantly. It also effectively boosts the performances of several current models for aspect term extraction.

1 Introduction

Aspect term extraction (ATE), which aims to identify and extract the aspects on which users express their sentiments (Hu and Liu, 2004; Liu, 2012), is a fundamental task in aspect-level sentiment analysis. For example, in the sentence of “*The screen is very large and crystal clear with amazing colors and resolution*”, “*screen*”, “*colors*” and “*resolution*” are the aspect terms to extract in this task.

ATE is typically formulated as a sequence labeling problem (Xu et al., 2018, 2019; Li et al., 2018), where each word is appended with a label indicating if it identifies an aspect. Sentence and label sequence are both used to train a ATE model. One of the remaining challenges with this task is the shortage of annotated data. While data augmentation appears to be a solution to this problem, it

*Corresponding author.

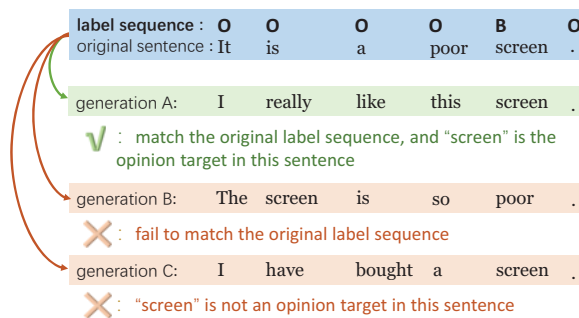


Figure 1: Examples of ATE augmentation, where *B*, *I* and *O* denote that a word is the beginning, inside and outside of opinion target, respectively.

faces two main obstacles here. First, the new sentences must adhere to their original label sequences strictly. As shown in Figure 1, the *generation A* is an effective augmentation as the original label sequence is preserved, whereas *B* is not even though it can be a valid review. Second, a noun phrase is regarded as aspect term only if it is an opinion target. In the *generation D* of Figure 1, although the term “*screen*” remains where it is in the original sentence, the new context makes it just an ordinary mention rather than an opinion target. To sum up, the real difficulty of data augmentation in ATE is generating a new sentence while aligning with the original label sequence and making the original aspect term remain an opinion target. Existing augmentation models such as GAN (Goodfellow et al., 2014) and VAE (Kingma and Welling, 2013) tend to change the opinion target unpredictably and thus are not applicable for this task.

Another genre of augmentation strategy is based on word replacement. It generates a new sentence by replacing one or multiple words with their synonyms (Zhang et al., 2015) or with words predicted by a language model (Kobayashi, 2018). This approach seems to be able to address the above issue in ATE augmentation, yet it only brings very lim-

ited changes to the original sentences and cannot produce diversified sentences. Intuitively, augmentation strategies are effective when they increase the diversity of training data seen by a model.

We argue in this paper that the augmentation for aspect term extraction calls for a conditional approach, which is to be formulated as a masked sequence-to-sequence generation task. Specifically, we first mask several consecutive tokens for an input sentence. Then, our encoder takes the partially masked sentence and its label sequence as input, and our decoder tries to reconstruct the masked fragment based on the encoded context and label information. The process of reconstruction keeps the opinion target unchanged and is therefore controllable. Moreover, compared with replacement-based approaches (Zhang et al., 2015; Kobayashi, 2018) which replace words separately, ours replaces a segment each time and has the potential to generate more diversified new sentences in content.

To implement the above conditional augmentation strategy, we adopt Transformer (Vaswani et al., 2017) as our basic architecture and train it like MASS (Song et al., 2019), a pre-trained model for masked sequence-to-sequence generation.

The contributions of this work are as follows.

- To our knowledge, this work is the first effort towards data augmentation of aspect term extraction through conditional text generation.
- We propose a controllable data augmentation method by masked sequence-to-sequence generation, which is able to generate more diversified sentences than previous approaches.
- We provide qualitative analysis and discussions as to why our augmentation method works, and test its implementation with other language models to illustrate why this masked sequence-to-sequence framework is favored.

2 Related Work

2.1 Aspect Term Extraction

Aspect term extraction (ATE) and sentiment classification are two fundamental subtasks of aspect-based sentiment analysis. While the former aims to extract aspect terms in review sentences, the latter tries to determine their sentiment polarities. To deal with ATE, many traditional techniques like syntactic rules (Qiu et al., 2011), hidden Markov models

(Jin et al., 2009), and conditional random fields (Li et al., 2010; Toh and Su, 2016) have been explored.

Recently, neural network techniques such as LSTM (Liu et al., 2015), CNN (Xu et al., 2018), and attention (Li et al., 2018; Devlin et al., 2019) have been applied for ATE. Luo et al. (2019) and He et al. (2019) further proposed to predict aspect term and polarity jointly in a multi-task learning approach so as to take advantage of their relatedness. Generally, the above approaches treat ATE as a sequence labeling problem. In their pioneering work, Ma et al. (2019) formulated ATE as a sequence-to-sequence task. So far, one of the remaining challenges for ATE lies in the lack of annotated data, especially when today’s neural models are becoming increasingly large and complex.

2.2 Text Data Augmentation

Generative adversarial network (GAN) (Goodfellow et al., 2014) and variational autoencoder (VAE) (Kingma and Welling, 2013) are two neural network based generative models that are capable of generating text conditioned on input text and can be applied for data augmentation of sentence-level sentiment analysis (Gupta, 2019; Hu et al., 2017). These methods encode an input text into latent variables and generate new texts by decoding the latent variables in continuous space. However, they can hardly ensure high-quality sentences in terms of readability and label compatibility. Back translation (Edunov et al., 2018; Sennrich et al., 2016) is another augmentation approach for text data, but is less controllable, although it is good at maintaining the global semantics of an original sentence. As a class of replacement approach, Zhang et al. (2015) and Wang and Yang (2015) proposed to substitute all replaceable words with corresponding synonyms from WordNet (Miller, 1995). Differently, Kobayashi (2018) and Wu et al. (2019) proposed to randomly replace words with those predicted by a pre-trained language model.

Nevertheless, none of the above augmentation approaches is applicable for aspect term extraction task, as they are all targeted at sentence-level classification and may change opinion targets and aspect labels unexpectedly during augmentation.

2.3 MASS

Pre-training a large language model and fine-tuning it on downstream tasks has become a new paradigm. MASS (Song et al., 2019) is such a model for language generation. Unlike GPT (Radford et al.,

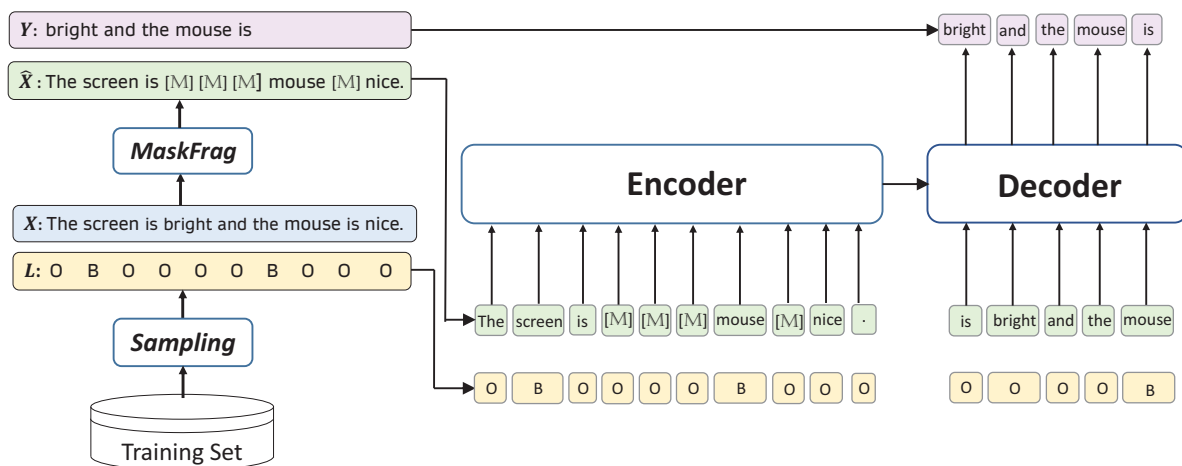


Figure 2: Framework of our augmentation method.

2016, 2019) and BERT (Devlin et al., 2019) which only have either an encoder or a decoder, MASS includes both of them and trains them jointly: the encoder takes as input a sentence with a fragment masked and outputs a set of hidden states; the decoder estimates the probability of a token in the masked fragment conditioned on its preceding tokens and the hidden states from the encoder. This pre-training approach enables MASS to perform representation learning and language generation simultaneously. MASS has achieved significant improvements in several sequence-to-sequence tasks, such as neural machine translation and text summarization (Song et al., 2019).

Our augmentation method has a similar training objective as MASS, and includes a label-aware module to constrain the generation process.

3 Conditional Augmentation for ATE

As mentioned before, we formulate the data augmentation of aspect term extraction (ATE) as a conditional generation task. In this section, we first introduce the problem formulation, and then describe our augmentation method in detail.

3.1 Problem Formulation

Given a training set \mathcal{D} of review texts, in which each sample includes a sequence of n words $X = [x_1, x_2, \dots, x_n]$ and a label sequence $L = [l_1, l_2, \dots, l_n]$, where $l_i \in \{B, I, O\}$. Here, B , I and O denote if a word is at the *beginning*, *inside* or *outside* of an aspect term, respectively. The objective of our augmentation task is to generate a new sentence consistent with L and the aspect term.

3.2 Our Approach

The above augmentation is modeled as a fine-grained conditional language generation task implemented by a masked sequence-to-sequence generation model. As depicted in Figure 2, the model adopts Transformer (Vaswani et al., 2017) as its basic architecture, and consists of a 6-layer encoder and a 6-layer decoder with 12 attention heads in each layer. The embedding size and hidden size are both 768, and the feed-forward filter size is 3072. The generation model is initialized with the pre-trained weights of MASS. To further incorporate the domain knowledge, we perform in-domain pre-training as in (Howard and Ruder, 2018).¹

3.2.1 Training

The training process is illustrated in Algorithm 1. For each batch, we first sample a few examples from the training set with replacement (Line 4) according to a probability p specified in Equation (1). The chosen examples are then masked using the Fragment Masking Strategy function (Line 6) to generate training examples for our model. We elaborate on Algorithm 1 in the following paragraphs.

Fragment Masking Strategy

The function *MaskFrag* (Line 6) is performed on the chosen examples to mask positions from u to $v = u + r * \text{length}(X)$, where $\text{length}(X)$ is the length of sentence X . Each masking position is replaced by $[M]$ only if its label is O . As a result, we obtain a partially masked sentence \hat{X} and a fragment $Y = [y_1, y_2, \dots, y_m] = [x_u, x_{u+1}, \dots, x_v]$,

¹The Amazon and Yelp review datasets are used as the Laptop and Restaurant domain corpora, respectively.

Algorithm 1 Training

Input:

\mathcal{D} : training set;
 θ : model parameters;
 p : sampling probability from \mathcal{D} ;
 K : total training iterations;
 B : batch size;
 r : proportion of sentence to mask; 0.5 by default;

```
1: for  $i \leftarrow 1$  to  $K$  do
2:    $Batch^{(i)} \leftarrow \emptyset$ ;
3:   for  $j \leftarrow 1$  to  $B$  do
4:     Randomly sample an example  $(X, L)$ 
       from  $\mathcal{D}$  by probability  $p$ ;
5:     Randomly sample a start position  $u$ ,
       where  $u \in (1, (1-r) * length(X))$ ;
6:      $\hat{X}, Y \leftarrow MaskFrag(X, L, u, r)$ ;
7:      $Batch^{(i)} \leftarrow Batch^{(i)} \cup \{(\hat{X}, L, Y)\}$ ;
8:   end for
9:    $\theta \leftarrow Train(Batch^{(i)}, \theta)$ ;
10: end for
11: Return  $\theta$ ;
```

where $m = v - u + 1$ is the length of the fragment.

Sampling Strategy

Line 5 of Algorithm 1 shows that during the training process each sentence is masked every time it is sampled. Since long sentences have more different segments to mask than short ones, they should be sampled more frequently. We define the sampling probability p_i of each example i as follows:

$$p_i \propto \begin{cases} \sqrt{d_i} & , d_i > 5, \\ 0 & , otherwise \end{cases} \quad (1)$$

where d_i denotes the sequence length of example i .

Training Objective

The training objective (Line 9) takes the masked sentence \hat{X} and label sequence L as input, and reconstructs the masked fragment Y . The inputs of the encoder are obtained by summing up the embeddings of a token \hat{x} , its aspect label l , and position q . The output is the hidden state $H = [h_1, h_2, \dots, h_n]$:

$$H = Enc(\hat{X}, L), \quad (2)$$

where Enc represents the encoder, and $h_t \in \mathbb{R}^{s_h}$ denotes the hidden state of size s_h for word \hat{x}_t .

Each self-attention head of the encoder learns a representation for the sentence based on tokens

\hat{X} , label sequence L and position Q . The objective of the decoder is to generate a sequence Y based on \hat{X} and L . In particular, it predicts next token y_t based on context representations H , current aspect label l_t and previous tokens $[y_1, \dots, y_{t-1}]$.

$$P(Y|X, L) = \prod_{t=1}^m P(y_t|y_{1:t-1}, l_t, H), \quad (3)$$

where the conditional probability of token y_t is defined by:

$$P(y_t|y_{1:t-1}, l_t, H) = softmax(Ws_t + b). \quad (4)$$

Here, $W \in \mathbb{R}^{|V| \times s_h}$, $|V|$ is the vocabulary size, and s_t is the hidden state of the decoder at time step t , being calculated as:

$$s_t = z_t + Emb_l(l_t), \quad (5)$$

$$z_t = Dec(x_{t-1}, l_{t-1}), \quad (6)$$

where Emb_l is the label embedding function and Dec is the decoder.

In Equation (5), each decoding step is conditioned on the context information and the whole label sequence, making the generation controllable.

The encoder and the decoder are jointly trained by maximizing the log-likelihood loss:

$$\mathcal{J} = \sum_{t=1}^m \log(P_\theta(y_t|y_{1:t-1}, l_t, H)), \quad (7)$$

where θ includes the trainable parameters.

3.2.2 Augmentation

After training for a few epochs, our model is used to predict the words in a masked fragment. Specifically, given an example (X, L) from the training set \mathcal{D} , we choose a start position u and apply $MaskFrag(X, L, u, r)$ to obtain \hat{X} . To avoid that same positions are chosen repeatedly, we manually choose the start position u for the augmentation. At generation time, we use beam search with a size of 5 for the auto-regressive decoding. After the decoder produces all the tokens compatible with the original label sequence and aspect terms, we obtain a new example (\tilde{X}, L) . Empirically, we find the model tends not to generate a same segment as the old one when the masked segment is longer than 4. The above process can be run multiple times with different start positions, and generates multiple new examples from a source example. In this approach, each source example is augmented in turn.

4 Experiment

In this section, we first introduce the experimental datasets and several popular ATE models. Then, we report the experimental results, which are obtained by averaging five runs with different initializations.

4.1 Datasets

Two widely-used datasets, the Laptop from SemEval 2014 Task 4 (Pontiki et al., 2014) and the Restaurants from SemEval 2016 Task 5 (Pontiki et al., 2016), are used for our evaluations. The statistics of the two datasets are shown in Table 1, which tells clearly that there are only a limited number of samples in both datasets.

Dataset	Training		Testing	
	#Sent	#Aspect	#Sent	#Aspect
Laptop	3045	2358	800	654
Restaurant	2000	1743	676	622

Table 1: Statistics of our datasets. #Sent and #Aspect denote the count of sentence and aspect, respectively.

4.2 Experimental Setup

4.2.1 Dataset Augmentation

For each of the two datasets, we hold out 150 examples from the original training set for validation. For each remaining training example, we generate four augmented sentences according to Section 3.2.2 with the proportion r set to 0.5. The four new sentences are allocated to four different sets. This leads to four generated datasets.

4.2.2 ATE Models

To examine our data augmentation method, we use the original training sets and the augmented training sets to train several ATE models. The details of these models are as follows.

BiLSTM-CRF is a popular model for sequence labeling tasks. Its structure includes a BiLSTM followed by a CRF layer (Lafferty et al., 2001). The word embeddings for this model are initialized by GloVe-840B-300d (Pennington et al., 2014) and fixed during training. The hidden size is set to 300, and we use Adam (Kingma and Ba, 2014) with a learning rate of $1e-4$ and L2 weight decay of $1e-5$ to optimize this model.

Seq2Seq for ATE (Ma et al., 2019) is the first effort to apply a sequence-to-sequence model for aspect term extraction. It adopts GRU (Cho et al.,

2014) for both the encoder and the decoder. The encoder takes a source sentence as input, and the decoder generates a label sequence as the result. This approach is also equipped with a gated unit network and a position-aware attention network.

BERT for token classification (Devlin et al., 2019) uses pre-trained BERT with a linear layer. We implement this model using open source² and initialize its parameters with the pre-trained BERT-BASE-UNCASED model. We refer to this model as BERT-FTC in the following paragraphs.

DE-CNN (Xu et al., 2018) uses two types of word embeddings: general-purpose and domain-specific embeddings.³ While the former adopt GloVe-840B-300d, the latter are trained on a review corpus. They are concatenated and fed to a CNN model of 4 stacked convolutional layers.

BERT-PT (Xu et al., 2019)⁴ utilizes the weights of pre-trained BERT for initialization. To adapt to both domain knowledge and task-specific knowledge, it is then post-trained on a large-scale unsupervised domain dataset and a machine reading comprehension dataset (Rajpurkar et al., 2016, 2018). So far, it is the state of the art for ATE.

The above models are all open-sourced and their default settings are employed in our experiments.

4.3 Results and Analysis

4.3.1 Effect of Double Augmentation

We combine the original training set with each of the four generated datasets (refer to 4.2.1) and obtain four augmented training sets, each doubling the original training set in size. For each model, we train it on the four augmented training sets, respectively, and take their average F1-scores on the test set. By comparing this score with the model trained on the original training set, we can examine if the augmented datasets improve the model.⁵

As shown in Table 2, all the models are improved more or less based on the augmented datasets. Even for the state-of-the-art DE-CNN and BERT-PT models, our augmentation also brings considerable improvements, which confirms that our augmentation approach can generate useful sentences for training a more powerful model for aspect term extraction.

²<https://github.com/huggingface/transformers>

³<https://howardhsu.github.io/>

⁴<https://github.com/howardhsu/BERT-for-RRC-ABSA>

⁵The scores here may be slightly different from that in the original papers, but each pair of experiments in comparison were conducted strictly under the same setting.

Model	Laptop		Restaurant	
	source	augmentation	source	augmentation
BiLSTM-CRF	73.42	74.28	69.16	71.44
Seq2Seq for ATE	76.68	78.68	73.71	74.01
BERT-FTC	79.39	81.14	74.75	75.89
DE-CNN	81.08	81.58	74.52	75.19
BERT-PT	84.59	85.33	77.49	80.29

Table 2: F1-score(%) obtained on the tests for various models, where *source* denotes the original datasets.

4.3.2 Effect of Multiple Augmentation

The above results show the effect of double augmentation. In this subsection, we further combine any two of the four generated datasets with the original training set to form triple-augmented datasets, leading to six new datasets. In a similar approach, we can create quadruple-augmented and quintuple-augmented datasets. Then, we train the DE-CNN and BERT-FTC models on the new datasets and take the average F1-score for each model as before. The results are shown in Figure 3.

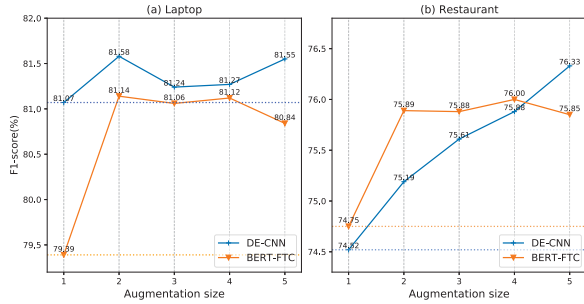


Figure 3: Performances of DE-CNN and BERT-FTC on different-sized augmentation datasets, where 1 means the original datasets without augmentation. All the results are based on the average scores of five runs.

We can observe from the figure that both models are generally improved as the size of augmentation increases on the Restaurant dataset. There is even a 1.8 boost for DE-CNN. On the Laptop dataset, however, the highest scores are seen at double-augmentation for both models. One of the reasons could be the relatively large volume of the original dataset. Another possible reason is that the aspect terms in this dataset are often regular nouns such as *screen* and *keyboard*, which can be successfully extracted just based on their own meanings. Differently, aspect terms in the Restaurant dataset are more arbitrary and diverse such as *Cafe Spice* and *Red Eye Grill*, the names of dish or restaurant. This requires a model to pay more attention to the

contexts while determining whether the candidate is an aspect terms. As our augmentation approach can generate different contexts for an aspect term, it works better on the Restaurant dataset.

5 Discussion

In this section, we present more qualitative analysis and discussions about our augmentation approach.

5.1 Does Larger Masked Length Help?

In the augmentation stage, the masked proportion r is a hyperparameter and set to the half of the length of a sentence in the above experiments. In this subsection, we explore its influence by changing it from 30% to 70% of sentence length stepped by 10%. We use DE-CNN model for this evaluation on the double-augmented datasets.

As shown in Figure 4, the overall trend for F1-scores is moving up as r increases. The reason is that sentences with short masked fragments are likely to be restored to their original forms by our generation model. As the proportion r increases, the content of a sentence has increasingly more chances to be changed significantly, resulting in diversified new sentences. This can be confirmed by the declining BLEU scores in Figure 4.

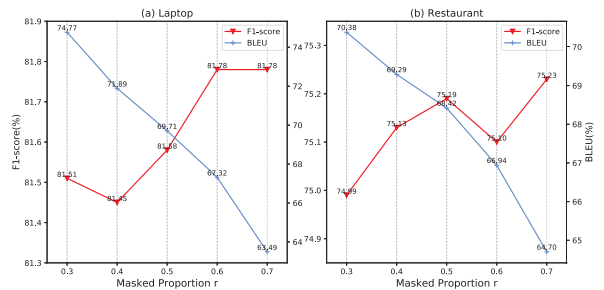


Figure 4: Performance of DE-CNN with different masked proportion r for augmentation. BLEU is used to measure the correspondence between a new sentence after augmentation and the original sentence.

Dataset	Laptop			Restaurant		
	Precision	Recall	F1	Precision	Recall	F1
Source	81.24	80.91	81.08	70.62	78.88	74.52
Ours w/o LEM	80.75	79.66	80.20	70.63	78.23	74.24
Ours	81.88	81.29	81.58	70.86	80.08	75.19

Table 3: Results of ablation study on whether label embeddings are used, where *Source* denotes the original dataset, and *Ours w/o LEM* denotes our augmentation model without label embeddings.

5.2 Does Label Sequence Matter?

Our augmentation model introduces label embeddings into Transformer to force the new sentences to be task-competent. We conduct an ablation study to verify the effectiveness by removing these embeddings during augmentation. The DE-CNN model is used again for this study.

As shown in Table 3, the removal of label embeddings causes considerable performance drops, and the results are even worse than that on the original dataset. This is probably due to the poor *Recall* performance that can be explained as follows. When label sequence information is not present, the augmentation is prone to produce decayed examples in which some new aspect terms are generated in the positions of label *O*, or vice vice. The model trained with such decayed examples is misled not to extract these aspect terms in the test stage. As a result, the model makes many false-negative errors, leading to poor *Recall* scores. This indicates that label embeddings are helpful for generating qualified sentences for aspect term extraction.

5.3 Why Sequence-to-Sequence Generation?

As mentioned before, we formulate the data augmentation for aspect term extraction as a conditional generation problem that is solved by masked sequence-to-sequence learning. One may argue that other pre-trained language models like BERT and GPT-2 are also competent for this task as in (Wu et al., 2019; Sudhakar et al., 2019; Keskar et al., 2019). Here we compare them and demonstrate the superiority of our approach in this task.

Following some previous work (Wu et al., 2019; Sudhakar et al., 2019; Keskar et al., 2019), we modify the settings of BERT and GPT-2 to make them fit this task. Readers are recommended to refer to Appendix for more details. Moreover, a widely-used replacement-based method is implemented for comparison, in which half of the tokens are randomly replaced by their synonyms from Word-

Net (Miller, 1995). We use *fluency*⁶ and BLEU⁷ to evaluate the generated sentences. Note that these datasets do not contain the original training examples because we want to focus more on the generated ones. We employ BERT-FTC as the implementation model and train it on these datasets. The results on the test sets are presented in Table 4.

From the table, we note that the F1 scores of GPT-2 are the worst because of its low recall scores. This conforms with the architecture and the language modeling objective of GPT-2, which does not have an encoder to encode the label information. In this case, the decoding step is uncontrollable and cannot generate a sentence fitting the label sequence. In contrast, our framework contains an encoder to encode a sentence and the label sequence simultaneously, and a decoder to generate sentences conditional on the encoder output. That is, our decoder takes advantage of both context information and aspect label information, making the augmentation conditional and controllable.

BERT performs the worst in this task in fluency. This can be attributed to its independence assumption in the process of generation, which means that all masked tokens are independently reconstructed, likely leading to in-coherent word sequences. In contrast, our approach generates the sequence in an auto-regressive way, with each decoding step based on the result of its previous step, ensuring fluent new sentences.

The replacement-based method does not take into account the sentence context and leads to poor fluency scores. Also, there are limited words to choose for synonyms in such lexical databases as WordNet. Thus, such replacement-based methods can only produce sentences of limited diversity, which is confirmed by the BLEU scores.

⁶Fluency is measured by the *perplexity* of sentence, and is calculated by OpenAI GPT. In this metric, sentences with lower perplexity scores are more fluent. Note that the *GPT* here is different from GPT-2 that we use to generate text data.

⁷The original sentences are taken as reference.

Dataset	Laptop			Restaurant		
	Precision/Recall/F1	BLEU	Fluency	Precision/Recall/F1	BLEU	Fluency
Source	77.36/81.53/79.39	100.00	203	74.86/74.64/74.75	100.00	158
DATA_synonym	74.91/79.06/76.93	76.68	558	71.23/76.43/73.74	78.22	438
DATA_BERT	80.06/78.60/79.32	69.93	461	72.34/76.37/74.30	68.57	435
DATA_GPT-2	71.97/61.10/66.09	54.86	328	70.53/57.42/63.30	58.60	314
DATA_Ours	80.27/80.98/ 80.61	69.70	242	73.95/76.95/ 75.41	68.42	257

Table 4: Results on datasets generated by different augmentation approaches. *Source* denotes the original datasets; *DATA_synonym* denotes the datasets obtained by randomly replacing tokens with their synonyms. *DATA_BERT*, *DATA_GPT-2* and *DATA_Ours* denote the datasets generated by BERT, GPT-2 and our augmentation approach, respectively. BERT-FTC is used as the implementation model, and lower *Fluency* scores mean more fluent sentences.

Source:	Also, the space bar makes a noisy click every time you use it.
Augmented:	Also, the space bar will get stuck there every time you use it.
Source:	The hinge design forced you to place various connections all around the computer, left right ...
Augmented:	The hinge design also allows you to adjust the angle around the computer , left right ...
Source:	Their pad penang is delicious and everything else is fantastic.
Augmented:	Their pad penang is mediocre but everything else is fantastic.
Source:	I am learning the finger options for the mousepad that allow for quicker browsing of web pages.
Augmented:	I also enjoy the fact that it has a mousepad that allow for quicker browsing of web pages.
Source:	I charge it at night and skip taking the cord with me because of the good battery life .
Augmented:	I don't have to carry the cord with me because of the good battery life .

Table 5: Examples generated by our augmentation approach. Texts in bold, blue and purple represent aspect terms, masked fragments and generated fragments, respectively.

To sum up, our data augmentation model benefits considerably from its encoder-decoder architecture and the masked sequence-to-sequence generation mechanism, which is controllable to ensure qualified data augmentation for aspect term extraction. The results show that this sequence-to-sequence generation framework is non-replaceable by other language models such as BERT and GPT-2.

5.4 Case Study

We finally present several augmented examples in Table 5 to illustrate the effect of our augmentation method more intuitively. We observe that the contents of the masked fragments can be dramatically changed from their original forms after augmentation. In some cases, the sentiment polarities are even reversed. Nevertheless, the new contexts are still appropriate for the aspect terms, making them qualified and also diversified new training examples for aspect term extraction.

6 Conclusion

In this paper, we have presented a conditional data augmentation approach for aspect term extraction.

We formulated it as a conditional generation problem and proposed a masked sequence-to-sequence generation model to implement it. Unlike existing augmentation approaches, ours is controllable to generate qualified sentences, and allows more diversified new sentences. Experimental results on two review datasets confirm its effectiveness in this conditional augmentation scenario. We also conducted qualitative studies to analyze how this augmentation approach works, and tested other language models to explain why our masked sequence-to-sequence generation framework is favored. Moreover, the proposed augmentation method tends not to be unique to the current task and could be applied to other low-resource sequence labeling tasks such as chunking and named entity recognition.

Acknowledgments

The work was partially supported by the Fundamental Research Funds for the Central Universities (No.19lgpy220) and the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (No.2017ZT07X355).

References

- Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.
- Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.
- Rahul Gupta. 2019. Data augmentation for low resource sentiment analysis using generative adversarial networks. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7380–7384. IEEE.
- Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 504–515, Florence, Italy.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia.
- Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177.
- Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70, ICML’17*, pages 1587–1596.
- Wei Jin, Hung Hay Ho, and Rohini K Srihari. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of the 26th annual international conference on machine learning*, pages 465–472.
- Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.
- Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Sosuke Kobayashi. 2018. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana. Association for Computational Linguistics.
- John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Ying-Ju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *Proceedings of the 23rd international conference on computational linguistics*, pages 653–661.
- Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. In *IJCAI*, pages 4194–4200.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Pengfei Liu, Shafiq Joty, and Helen Meng. 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*, pages 1433–1443.
- Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: Dual cross-shared RNN for aspect term-polarity co-extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 591–601, Florence, Italy.
- Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang. 2019. Exploring sequence-to-sequence learning in aspect term extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547.
- George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. *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.
- Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2016. Improving language understanding by generative pre-training. 2018. URL https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8).
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.
- Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *ICML*.
- Akhilesh Sudhakar, Bhargav Upadhyay, and Arjun Maheswaran. 2019. “transforming” delete, retrieve, generate approach for controlled text style transfer. 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 3260–3270, Hong Kong, China. Association for Computational Linguistics.
- Zhiqiang Toh and Jian Su. 2016. Nlangu at semeval-2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 282–288.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.
- William Yang Wang and Diyi Yang. 2015. That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using# petpeeve tweets. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2557–2563.
- Xing Wu, Shangwen Lv, Liangjun Zang, Jizhong Han, and Songlin Hu. 2019. Conditional BERT contextual augmentation. In *Computational Science - ICCS 2019 - 19th International Conference, Faro, Portugal, June 12-14, 2019, Proceedings, Part IV*, pages 84–95.
- Hu Xu, Bing Liu, Lei Shu, and Philip Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. 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 2324–2335, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.
- Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

A Appendices

We modify the settings of BERT and GPT-2 to make them fit our augmentation task as follows.

BERT We follow Wu et al. (2019) who applied the masked language model task to augment their training data for text classification. The segment embeddings in the input layer are replaced by label embeddings. The objective is to predict the masked token t_i based on the conditional probability distribution $p(t_i|L, \mathcal{S} \setminus \{t_i\})$, where \mathcal{S} and L denote the sentence and its label sequence, respectively, and $\mathcal{S} \setminus \{t_i\}$ means the context of t_i . To predict t_i , both the context $\mathcal{S} \setminus \{t_i\}$ and the label sequence L are considered. The word with the highest probability among all vocabulary words is chosen.

In our experiments, 50% of the tokens in a sentence are masked individually. These tokens are then reconstructed in the masked language modeling task. Finally, the predicted tokens and the unmasked tokens constitute an augmented sentence.

GPT-2 We refer to the work of Sudhakar et al. (2019) and Keskar et al. (2019), which aim at style transfer and controllable generation, respectively. In the training stage, the format of input is: [ASP] $aspect_1$ [ASP] $aspect_2$ [BOS] $sentence$ [EOS], where [ASP] is a special symbol followed by an aspect term, and [BOS] is appended before the source sentence. For example, for a source sentence “*I was disappointed with this restaurant*”, where “*restaurant*” is the aspect term, the input takes the following format: [ASP] $restaurant$ [BOS] $I was disappointed with this restaurant.$ [EOS]. Let \mathcal{S} denote the source sentence. The autoregressive model learns to reconstruct \mathcal{S} given C_X , which is the fragment of the input from the first token [ASP] to [BOS]. The objective is to maximize:

$$L(\theta) = \log p(\mathcal{S}|C_X; \theta) \quad (8)$$

where θ denotes the parameters of GPT-2.

In the augmentation stage, this model takes the fragment C_X and half of the source sentence as input, and tries to reconstruct the other half. An augmented sentence is formed by joining the unmasked half of the sentence and the reconstructed half. The beam size is set to 5 in our experiments.