# Exploring Sequence-to-Sequence Learning in Aspect Term Extraction

**Dehong Ma**[♣], **Sujian Li**[♣], **Fangzhao Wu**[♠], **Xing Xie**[♠], **Houfeng Wang**[♣]
[♣]MOE Key Lab of Computational Linguistics, Peking University, Beijing, 100871, China
[♠] Microsoft Research Asia, Beijing, China
{madehong, lisujian, wanghf}@pku.edu.cn
wufangzhao@gmail.com, Xing.Xie@microsoft.com

## Abstract

Aspect term extraction (ATE) aims at identifying all aspect terms in a sentence and is usually modeled as a sequence labeling problem. However, sequence labeling based methods cannot make full use of the overall meaning of the whole sentence and have the limitation in processing dependencies between labels. To tackle these problems, we first explore to formalize ATE as a sequence-to-sequence (Seq2Seq) learning task where the source sequence and target sequence are composed of words and labels respectively. At the same time, to make Seq2Seq learning suit to ATE where labels correspond to words one by one, we design the gated unit networks to incorporate corresponding word representation into the decoder, and position-aware attention to pay more attention to the adjacent words of a target word. The experimental results on two datasets show that Seq2Seq learning is effective in ATE accompanied with our proposed gated unit networks and position-aware attention mechanism.

## 1 Introduction

Aspect term extraction (ATE) is a fundamental task in aspect-level sentiment analysis, and aims at extracting all aspect terms present in the sentences (Hu and Liu, 2004; Pontiki et al., 2014, 2015, 2016). For example, given a restaurant review *"The staff is friendly, and their cheese pizza is delicious"*, the ATE system should extract aspect terms *"staff"* and *"cheese pizza"*.

Early works focus on detecting the pre-defined aspects in a sentence (Hu and Liu, 2004; Zhuang et al., 2006; Popescu and Etzioni, 2007). Then, some works regard ATE as a sequence labeling task and utilize Hidden Markov Model (Jin et al., 2009) or Conditional Random Fields (Jin et al., 2009; Ma and Wan, 2010; Jakob and Gurevych, 2010; Liu et al., 2013) to extract all possible aspect terms. With the development of deep learning techniques, neural networks based methods (Wang et al., 2016; Liu et al., 2015; Li and Lam, 2017; Xu et al., 2018) have achieved good performances in ATE task, and they still treat ATE as a sequence labeling problem and extract more useful features surrounding a word. Obviously, the overall meaning of the sentence is important to predict the label sequence. For example, the word *memory* should be an aspect term in the laptop review *"The memory is enough for use."*, but it is not an aspect term in the sentence *"The memory is sad for me."*. However, sequence labeling methods are not good at grasping the overall meaning of the whole sentence because they cannot read the whole sentence in advance. In addition, neural networks based sequence labeling methods have the limitation in processing label dependencies because they only use transition matrix to encourage valid label paths and discourage other paths (Collobert et al., 2011). As we know, the label of each word is conditioned on its previous label. For example, "O" is followed by "B/O" but not "I" in the B-I-O tagging schema. To the best of our knowledge, no neural networks based method utilizes the previous label to improve their performances directly.

Recently, sequence to sequence (Seq2Seq) learning has been successfully applied to many generation tasks (Cho et al., 2014b; Sutskever et al., 2014; Bahdanau et al., 2014; Nallapati et al., 2016). Seq2Seq learning encodes a source sequence into a fixed-length vector based on which a decoder generates a target sequence. It just has the benefits of first collecting comprehensive information from the source text and then paying more attention to the generation of the target sequence. Thus, we propose to formalize the ATE task as a sequence-to-sequence learning problem, where



*Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3538–3547
Florence, Italy, July 28 - August 2, 2019. 

the source and target sequences are word and label sequence respectively. Our proposed method can make full use of the overall meaning of the sentence when decoding the target sequence because the fix-length vector stores all useful information of a sentence and will be used in the decoding process. At the same time, Seq2Seq learning can remedy the label dependencies problem because each label is conditioned on the previous label when generating the label sequence.

Though Seq2Seq learning has its obvious advantages of generating a sequence, it faces the difficulties of how to precisely map each word with its corresponding label. As we know, the label of each word is highly related to its own meaning. For example, an aspect term tends to be some words used to identify any of a class of people, places, or things (e.g. staff, restaurant, pizza), while some words to describe an action, state, or occurrence (e.g. hear, become, happen) are rarely a part of an aspect term. Furthermore, our proposed method can know for which word it generates a label, and this kind of one-to-one match does not exist in other Seq2Seq task (e.g. machine translation). To incorporate the exact meaning of each word into Seq2Seq learning, we propose the gated unit networks (GUN) which contain a gated unit produced based on the hidden states of encoder and decoder. The gated unit can automatically integrate information from the encoder and decoder hidden states of the current word when decoding its label.

Furthermore, the label of each word is dependent on its adjacent words because the adjacent words of an aspect term tend to be article, verb, adjective and etc. As the example in the first paragraph, the adjacent words of *staff*: *The*, *is* and *friendly* have positive effect on predicting its label, while the rest words are not key factors. This shows the importance of adjacent words of each word in predicting its label. In classic Seq2Seq learning, attention mechanism is used to make the decoder select important parts of source sequence to form a context vector for decoding current word (Bahdanau et al., 2014). However, this kind of attention mechanism cannot pay more attention to the adjacent words of a word because it does not take distance into account. To overcome this shortage, we introduce the position-aware attention which first computes the weight of each word with regard to previous hidden state $s_{i-1}$. Then, the weight of word $i$ will be decreased based on the distance between word $i$ and current word $t$. The more distant, the lower important. Therefore, our position-aware attention model can force the decoder to pay more attention to the adjacent words of the current word when decoding its label.

We conduct experiments on two datasets, and the experimental results demonstrate that our proposed method achieves comparable results compared with existing methods.

## 2 Model

Our proposed method is based on sequence-to-sequence learning framework, plus two supplementary components namely position-aware attention and gated unit networks, which are used to capture features from the current word and its adjacent words. In this section, we will introduce our model in detail, whose overall architecture is displayed in Figure 1.

### 2.1 Sequence-to-Sequence Learning

For convenience, we first define the notations which will be used next. Let $X = [x_1, x_2, ..., x_n]$ denote a sentence which contains $n$ words, and $x_i \in \mathbb{R}^d$ is word embedding which can be learned by a neural language model (Bengio et al., 2003; Mikolov et al., 2013). Let $Y = [y_1, y_2, ..., y_n]$ denote the aspect term labels of sentence $X$ where $y_i \in \{B, I, O\}$. we call $X$ and $Y$ as source and target sequence respectively.

The sequence-to-sequence learning method is composed of two basic components: encoder and decoder. The encoder reads the embeddings of the source sequence and learns the hidden states $H = [h_1, h_2, ..., h_n]$ for all words, and the commonly used method is the Recurrent Neural Networks (RNN). In our model, we use a bidirectional gated recurrent unit (Bi-GRU) (Cho et al., 2014b) to obtain the hidden states:

$$h_t = \text{Bi-GRU}(x_t, h_{t-1}), \quad (1)$$

where Bi-GRU represents the operations of bidirectional GRU. $h_t \in \mathbb{R}^{s_e}$ represents the hidden state of word $t$, and $s_e$ is the hidden state size of the encoder.

The decoder is also a RNN which generates the target sequence $Y$ based on $X$, and predicts the next label $y_t$ based on the context vector $c_t$ and all previous labels $[y_1, y_2, ..., y_{t-1}]$ predicted by the

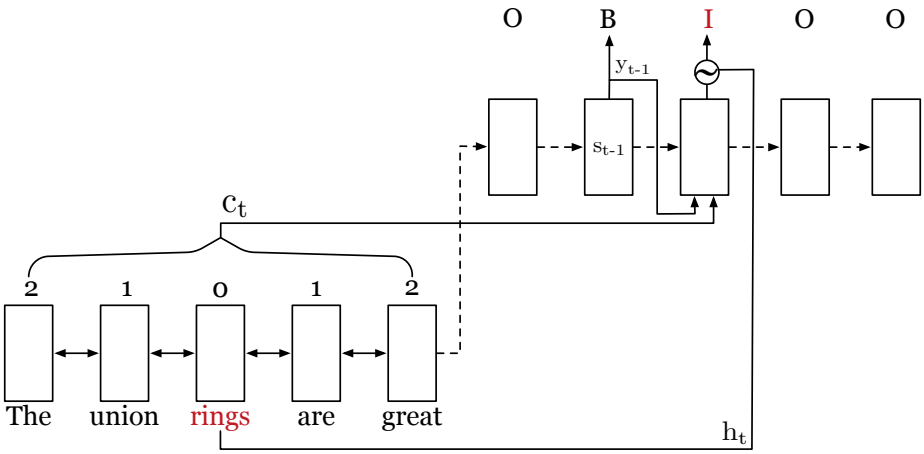


Figure 1: The overall architecture of our model.

same decoder. Therefore, the joint probability of the target sequence is defined as:

$$P(Y|X) = \prod_{t=1}^{n} P(y_t|y_{[1:t-1]}, c_t), \quad (2)$$

where $y_{[1:t-1]} = [y_1, ..., y_{t-1}]$ and the conditional probability of label $y_t$ can be modeled by the decoder, and defined as:

$$P(y_t|y_{[1:t-1]}, c_t) = \text{softmax}(W_o s_t + b_o), \quad (3)$$

where $W_o \in \mathbb{R}^{|V| \times s_d}$, $b_o \in \mathbb{R}^{|V|}$, $|V|$ is the target vocabulary size, and $s_d$ is the hidden state size of decoder. $s_t \in \mathbb{R}^{s_d}$ is the hidden state in the decoder at time step $t$, and computed as:

$$s_t = \text{GRU}(s_{t-1}, y^e_{t-1} \oplus c_t), \quad (4)$$

where GRU is a unidirectional GRU. $\oplus$ is the concatenation operation, and $y^e_{t-1}$ is label embedding for label $y_{t-1}$. The context vector $c_t$ will be explained in the next section. It is noticed that the initial hidden state of the decoder is the last hidden state of the encoder. This means that the decoder can be aware of the meaning of the whole source sequence during the decoding process.

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

$$\text{Loss} = -\frac{1}{n}\sum_{t=1}^{n} l_t \log(P_\theta(y_t|y_{[1:t-1]}, c_t)), \quad (5)$$

where $l_t$ is the ground truth label of word $t$, and $\theta$ denotes the parameters of the encoder and the decoder.

From Eq. (3) and (4), we can see that the previous label is regarded as input when decoding the label for the current word. However, existing neural network based sequence labeling methods first compute the label scores of each word simultaneously, and obtain the globally optimized label sequence (Collobert et al., 2011). Therefore, they do not know the label of previous word when computing the label scores for the current word. By contrast, our proposed model generates the label for current word based on the label of previous word. This is the main difference between our proposed model and existing methods in solving label dependencies for ATE task.

## 2.2 Position-Aware Attention

In ATE task, the adjacent words of each word have important effects on predicting its label, while the distant words make less contribution to its label. The reason is that aspect terms are often surrounded by their modifiers. To the best of our knowledge, the current widely-used attention mechanism usually ignores the influence of positions when measuring the weights of each word. Therefore, we propose a **P**osition-**A**ware **A**ttention (PAA) model which regularly decreases the weight of word $i$ with respect to the distance between word $i$ and word $t$. Supposing that we compute the context vector $c_t$ at position $t$, PAA first computes the weight for each word by:

$$\alpha^i_t = \frac{\exp(f(s_{t-1}, h_i))}{\sum_{j=1}^{n} \exp(f(s_{t-1}, h_j))}, \quad (6)$$

where $f(s_{t-1}, h_i)$ is the score function which computes the weight of $h_i$ given previous decoder hidden state $s_{t-1}$ and the corresponding distance.

The score function is defined as:

$$f(s_{t-1}, h_i) = \frac{1}{\mathrm{d}(w_i, w_t)}(W_s[s_{t-1}, h_i] + b_s)v_s^T, \tag{7}$$

where $\frac{1}{\mathrm{d}(w_i,w_t)}$ calculates the weight decay rate for word $i$, $W_s \in \mathbb{R}^{(s_d+s_e)\times(s_d+s_e)}$, $v_s \in \mathbb{R}^{(s_d+s_e)}$ and $b_s \in \mathbb{R}^{(s_d+s_e)}$ are weight matrix, weight vector and bias separately. $v_s^T$ means the transpose of $v_s$. In our model, we set $\mathrm{d}(w_i, w_t)$ as the function $\log_2(2 + l)$, where $l$ is the distance between word $w_i$ and current word $w_t$. As the example in Figure 1, when computing the context vector for *rings*, the $\mathrm{d}(\textit{union}, \textit{rings})$ is $\log_2(2 + 1)$.

Finally, the context vector $c_t$ is computed as a weighted sum of these encoder hidden states:

$$c_t = \sum_{i=1}^{n} \alpha_t^i h_i. \tag{8}$$

We can see that PAA can tune the weights of each word according to the distance. Therefore, compared with vanilla attention, our model can pay more attention to its adjacent words given a word.

### 2.3 Gated Unit Networks

When solving ATE by our proposed method, there exists a consistent one-to-one mapping between source sequence and target sequence. This means that the word representation can be used to help the decoder to generate its label. For example, some kinds of words (e.g. food, place, and people) tend to be aspect term, while other words (e.g. verb, adjective and adverb) have less opportunity to be a part of aspect term. Therefore, we design the **G**ated **U**nit **N**etworks (GUN) to incorporate word information into our model.

The main component of GUN is a *merge gate* which integrates information from encoder hidden state $h_t$ and decoder hidden state $s_t$. To make $s_t$ and $h_t$ have the same dimension $s_g$, we apply full-connection layers on $s_t$ and $h_t$ to obtain new representations $s'_t \in \mathbb{R}^{s_g}$ and $h'_t \in \mathbb{R}^{s_g}$. The *merge gate* is defined as:

$$g_t = \sigma(W_g h'_t + U_g s'_t + b_g), \tag{9}$$

where $\sigma$ is sigmoid function. $W_g, U_g \in \mathbb{R}^{s_g \times s_g}$ are weight matrices and $b_g \in \mathbb{R}^{s_g}$ is bias.

The *merge gate* automatically controls how much information should be taken from $h_t$ and $s_t$ for decoding the label for word $t$ by:

$$r_t = g_t h'_t + (1 - g_t)s'_t. \tag{10}$$

Finally, we feed $r_t$ to softmax rather than $s_t$ used in Eq. (3) to obtain the label distribution for word $t$. $h'_t$ plays a more important role than $s'_t$ if $g_t$ is greater than 0.5, and vice versa. In such way, GUN can make full use of the corresponding word representation to help the decoder to generate its label.

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

Table 1: The statistics of two datasets. #Sent and #Aspect mean the number of sentence and aspect term separately.

## 3 Experiments

In this section, we first introduce the datasets and hyper-parameters used in our experiments. Then, we show the baselines for comparison. Finally, we compare the performance of our model with the baselines and analyze the reason why our model work.

### 3.1 Dataset & Hyperparameter Setting

We conduct experiments on two widely used datasets of the ATE task (Li and Lam, 2017; Li et al., 2018; Xu et al., 2018), which are the laptop dataset from SemEval 2014 Task 4 (Pontiki et al., 2014)[1] and the restaurant dataset from SemEval 2016 Task 5 (Pontiki et al., 2016)[2] respectively. The details of the two datasets are shown in Table 1. All sentences are tokenized by NLTK[3]. In our experiments, we randomly split 10% of the training data as validation data. We adopt F1-Measure to evaluate the performance of the baselines and our model.

In our experiments, all word embeddings are initialized by pre-trained GloVe embeddings (Pennington et al., 2014)[4]. We also use fastText (Joulin

[1] http://alt.qcri.org/semeval2014/task4/

[2] http://alt.qcri.org/semeval2016/task5/

[3] https://www.nltk.org/

[4] Pre-trained GloVe embeddings can be downloaded from https://nlp.stanford.edu/projects/glove/

et al., 2016)[5] to compute word vector for out-of-vocabulary (OOV) words. The label embeddings are initialized randomly. The word and label embedding size are set as 300 and 50 respectively. The parameters of our model are initialized by uniform distribution $u \sim (-0.1, 0.1)$. Both the encoder and decoder have two layers of GRU, and their hidden size is set to 300. We use Adam (Kingma and Ba, 2014) to optimize our model with the learning rate of 0.001, and two momentum coefficients are set to 0.9 and 0.999 respectively. The batch size is set to 8. To avoid overfitting, we use dropout on word embedding and label embedding, and the dropout rate is set to 0.5.

### 3.2 Baselines

To evaluate the effectiveness of our approach, we compare our model with three groups of baselines. The first group of baselines utilizes conditional randomly fields (CRF):

- **CRF** trains a CRF model with basic feature templates[6] and word embeddings (Pennington et al., 2014) for ATE.
- **IHS_R&D** is the best system of laptop domain, and uses CRF with features extracted using named entity recognition, POS tagging, parsing, and semantic analysis (Chernyshevich, 2014).
- **NLANGP** utilizes CRF with the word, name list and word cluster feature to tackle the task and obtains the best results in the restaurant domain. It also uses the output of a Recurrent Neural Network (RNN) as additional features to enhance their performances (Toh and Su, 2016).
- **WDEmb** first learns embeddings of words and dependency paths based on the optimization objective formalized as $w_1 + r \approx w_2$, where $w_1$, $w_2$ are words, $r$ is the corresponding dependency path. Then, the learned embeddings of words and dependency paths are utilized as features in CRF for ATE (Yin et al., 2016).

The second group of baselines employs neural networks methods to address the ATE problem:

- **Bi-LSTM** applies different kinds of Bi-RNN (Elman/Jordan-type RNN) with different kinds of embeddings in the ATE task (Liu et al., 2015).
- **GloVe-CNN**[7] uses multi-layer Convolution Neural networks (CNN) model with GloVe embeddings to extract aspect-term (Xu et al., 2018).
- **BiLSTM-CNN-CRF** is the state-of-the-art system for named entity recognition task, which adopts CNN and Bi-LSTM to learn character-level and word-level features respectively, and CRF is used to avoid the illegal transition between labels (Reimers and Gurevych, 2017).

The third group of baselines are joint methods for aspect term and opinion term extraction, and they take advantages of opinion label information to improve their performances.

- **MIN** is an LSTM-based deep multi-task learning framework for ATE, opinion word extraction and sentimental sentence classification. It has two LSTMs equipped with extended memories, and neural memory operations are designed for jointly handling the extraction tasks of aspects and opinions via memory interactions (Li and Lam, 2017).
- **CMLA** is made up of multi-layer attention network, where each layer consists of a couple of attention with tensor operators. One attention is for extracting aspect terms, while the other is for extracting opinion terms (Wang et al., 2017).
- **RNCRF** [8] learns structure features for each word from parse tree by Recursive Neural Networks, and the learned features are fed to CRF to decode the label for each word (Wang et al., 2016).
- **HAST** tackles ATE by exploiting two useful clues, namely opinion summary and aspect detection history (Li et al., 2018).

[5] https://github.com/facebookresearch/fastText

[6] https://sklearn-crfsuite.readthedocs.io/en/latest/

[7] To make it fair, we compare our method with GloVe-CNN which only uses GloVe embeddings because our model just uses Glove embeddings but DE-CNN uses additional domain embeddings trained with large domain corpus.

[8] They also use handcraft features to improve their performances.

| Method | Laptop | Restaurant |
|---|---|---|
| CRF | 74.01 | 69.56 |
| IHS_RD | 74.55 | - |
| NLANGP | - | 72.34 |
| WDEmb | 75.16 | - |
| Bi-LSTM | 75.25 | 71.26 |
| GloVe-CNN | 77.67 | 72.08 |
| BiLSTM-CNN-CRF | 77.80 | 72.50 |
| MIN$^\sharp$ | 77.58 | 73.44 |
| CMLA$^\sharp$ | 77.80 | 72.77$^*$ |
| RNCRF$^\sharp$ | 78.42 | 69.72$^*$ |
| HAST$^\sharp$ | 79.52 | 73.61 |
| Seq2Seq4ATE | **80.31** | **75.14** |

Table 2: The performances (F1:%) of all baselines and our model. All results of baselines are taken from their papers, and "-" means that the result is not available. The model with ♯ means that it uses opinion information. The result with $*$ is from HAST.

### 3.3 Results Discussion

In this section, we report the performances of all models and analyze the advantages and disadvantages of them. The results of baselines and our model are displayed in Table 2.

From the first part, we can see that *CRF* model obtains the worst performances on both datasets. Compared with the CRF model, *IHS_RD* and *NLANGP* achieves better performances because they add more handcraft features to CRF. This shows that useful features are key factors for CRF based methods. Different from three previous approaches, *WDEmb* only uses word embeddings as inputs and performs better than IHS_RD model. In fact, the CRF model also uses GloVe embeddings, but its results are much worse than WDEmb. The reason may be that embeddings used in WDEmb are trained with parsing information which plays important roles in ATE task. For example, the subject and object have a higher probability to be an aspect term than other components. We can find that the CRF based methods are heavily dependent on the quality of features. However, it is hard to extract effective features, and this prevents CRF based methods from improving their results.

From the second part, we can observe that the *Bi-LSTM* model obtains the worst performances on both datasets compared with the other neural networks based methods. Although Bi-LSTM model only takes embeddings as features, it achieves comparable results compared with the best CRF based methods. The main reason is that Bi-LSTM can learn dependencies between words, and this phenomenon demonstrates that neural networks based methods have bigger advantages than CRF-based methods in solving the ATE task. Compared with Bi-LSTM, the *GloVe-CNN* model improves 2.42% and 0.82% on laptop and restaurant datasets respectively. It is noticed that the GloVe-CNN just extracts features in a fixed-size window of each word for predicting its label. That is to say, the adjacent words are key factors for ATE, and this important information is also incorporated into our model by PAA. The *BiLSTM-CNN-CRF* model takes advantages of Bi-LSTM and CNN and achieves better performances than both systems. This shows that Bi-LSTM and CNN can complement each other.

From the third part, we can see that *MIN*, *CMLA*, *RNCRF* and *HAST* achieve good performances on both datasets. This implies that joint learning is a new direction for ATE task. However, they take advantage of opinion information to improve their performances, and the opinion information is not accessible in many situations. It is noticed that *HAST* also use the information of previous words to predict the current label, and they find that previous word information (not the predicted label of the previous word) is important to model the label dependencies.

Finally, we can see that *Seq2Seq4ATE* raises its performances about 0.79% and 1.53% on two datasets compared with HAST. In addition, Seq2Seq4ATE does not take advantage of any extra features such as handcraft/syntactic features and opinion information. This demonstrates the effectiveness of our model.

In a word, our proposed method can make use of the overall meaning of the sentence to better deal with polysemous words (e.g. *memory*) and remedy the label dependencies through decoding current word conditioned on previous label. In addition, we propose the PAA and GUN to make Seq2seq learning method better suit the ATE task.

### 3.4 Ablation Study

In this section, we study the effectiveness of the key components (e.g. PAA and GUN) in our proposed model and conduct an extensive ablation study. There are two main ablation baselines: (1)Seq2Seq4ATE-w/o-PAA removes the PAA from the Seq2Seq4ATE, (2)Seq2Seq4ATE-w/o-

| Method | Laptop | Restaurant |
|---|---|---|
| Seq2Seq4ATE-w/o-GUN | 75.43 | 71.93 |
| Seq2Seq4ATE-w/o-PAA | 74.45 | 72.66 |
| Seq2Seq+VAM | 77.39 | 72.47 |
| Seq2Seq4ATE | **80.31** | **75.14** |

Table 3: The performances (F1:%) of our model's variants on two datasets.

| Method | Laptop | | Restaurant | |
|---|---|---|---|---|
| | F1 | IT-Rate | F1 | IT-Rate |
| BiLSTM | 75.08 | 6.72 | 68.41 | 8.98 |
| BiLSTM+CRF | 77.72 | 3.97 | 71.94 | 3.69 |
| Seq2Seq4ATE | **80.31** | **0.02** | **75.14** | **0.03** |

Table 4: The performances (F1:%) and illegal transition rate (IT-Rate:%) of three models.

GUN removes the GUN from the Seq2Seq4ATE. In addition, we also use vanilla attention mechanism (VAM) to compute the context vector (named Seq2Seq+VAM) for verifying the advantage of PAA. Table 3 reports the results of Seq2Seq4ATE and its variants.

From Table 3, we can first observe that both PAA and GUN are important components in our model because removing any of them from our model would result in heavily drop in performances on both datasets.

Secondly, we can see that Seq2Seq4ATE-w/o-GUN performs better on the laptop dataset but Seq2Seq4ATE-w/o-PAA performs better on the restaurant dataset. The reason may be that the aspect terms in the laptop domain are fixed words such as *CPU*, *memory* and etc. But the aspect terms in the restaurant domain are more arbitrary such as *The Mom Kitchen*, *Hot Pizzeria* and etc. Therefore, GUN is more important in the laptop domain because it can incorporate the word representation into Seq2Seq by merge gate, but PAA is more important for the restaurant domain because it can leverage the adjacent words of each word to help predict its label.

In addition, we also find that the Seq2Seq4ATE removing both PAA and GUN performs very bad in both datasets. We think the main reason is that the number of aspect term is much smaller compared with all words. Therefore, our model can hardly learn useful information from data. We analyze the datasets and find that the words of aspect term make up 8.8% and 6.9% of the training data of restaurant and laptop domain.

Finally, we can see that Seq2Seq4ATE improves about 2.92% and 2.67% on laptop and restaurant compared with Seq2Seq+VAM. The great improvements again prove that the adjacent words play important roles in ATE. The reason is that the weights of distant words in VAM may be large in VAM. However, the weights of distant words in PAA will be heavily decayed by the position information and the weights of adjacent words will be decayed little because $\mathrm{d}(w_i, w_t)$ is proportional to the distance.

### 3.5 Analysis of Label Dependencies

In this section, we conduct experiments to validate the effectiveness of our proposed model in handling label dependencies.

Collobert et al. (2011) have demonstrated that it is important to model label dependencies in sequence labeling task. To validate the effectiveness of our model in addressing this problem, we compare our model Seq2Seq4ATE with two models: *BiLSTM*[9] and *BiLSTM+CRF*. BiLSTM does not take the label dependencies into account, and BiLSTM+CRF uses transition matrix (Collobert et al., 2011) to address label dependencies problem.

To evaluate the effectiveness of model in modeling label dependencies, we propose an evaluation criterion: *Illegal Transition Rate* (IT-Rate) which is computed by: IT-Rate $= \frac{\text{\#illegal transition}}{\text{\#aspect term}} \times 100$ where "#illegal transition" is the number of illegal transition (e.g. O→I) occurrences in predicted label sequence, and "#aspect term" is the number of aspect term. Generally speaking, lower IT-Rate means better performance in modeling label dependencies.

Table 4 shows the results of three models on testing data. First, we can observe that the higher F1 is accompanied by lower IT-Rate. This once again demonstrates the importance of modeling label dependencies. Secondly, we can observe that BiLSTM+CRF decreases IT-Rate about 2.75% and 5.29% on two datasets compared with the BiLSTM model. This indicates that the transition matrix is a good way to model label dependencies. However, they also do not utilize the previous label to improve their performances directly. The most impressive results are that the IT-Rate of Seq2Seq4ATE is 0.02% and 0.03% which almost can be ignored compared with BiLSTM and BiL-

[9]We only use GloVe embeddings for words and utilize the same hyper-parameters used in Seq2Seq4ATE. Thus, its ATE results are not the same with LSTM in Table 2.

STM+CRF. The main reason is that Seq2Seq4ATE leverages previous label information $y_{t-1}$ to decode label $y_t$ for word $t$. Consequently, $y_t$ is compatible with $y_{t-1}$. This indicates the advantages of our model in handling label dependencies compared with previous methods.

## 4 Related Work

Aspect-based sentiment analysis (ABSA) is a subfield of sentiment analysis (Hu and Liu, 2004; Pontiki et al., 2014, 2015, 2016). In this paper, we only focus on the ATE task, and we solve this task by Seq2Seq learning which is often used in the generative task. We will introduce the recent study progresses in ATE and Seq2Seq learning.

### 4.1 Aspect Term Extraction

Hu and Liu (2004) first propose to evaluate the sentiment of different aspects in a document, and all aspects are predefined artificially. The key step is to extract all possible aspects of a document (Zhuang et al., 2006; Popescu and Etzioni, 2007; Mei et al., 2007; Titov and McDonald, 2008; He et al., 2017). However, predefined aspects may not cover all the aspects appearing in a document. Therefore, many works turn to extract all possible aspect terms in a document. The mainstream methods for aspect term extraction include the unsupervised method and supervised method. The typical unsupervised methods include bootstrapping (Wang and Wang, 2008), double propagation (Qiu et al., 2011) and others. The supervised methods contain Hidden Markov Model (Jin et al., 2009), Conditional Random Fields (Jakob and Gurevych, 2010; Li et al., 2010; Yang and Cardie, 2013; Chernyshevich, 2014; Toh and Su, 2016; Yin et al., 2016; Shu et al., 2017) and other approaches (Wu et al., 2009; Ma and Wan, 2010; Liu et al., 2013). With the developments of deep learning, neural networks based method such as recurrent NN (Liu et al., 2015; Li and Lam, 2017), recursive NN (Wang et al., 2016), convolution NN (Poria et al., 2016; Xu et al., 2018) and attention model (Wang et al., 2017) have achieved good performances in ATE. In addition, many works utilize multi-task learning (Yang and Cardie, 2013; Wang et al., 2016, 2017; Li et al., 2018) and other resources (Xu et al., 2018) to improve their performances.

### 4.2 Sequence-to-Sequence Learning

Sequence-to-sequence model is a generative model which is proposed by (Cho et al., 2014b; Sutskever et al., 2014), and first used in the field of machine translation. In addition, Cho et al. (2014a) improves the decoding by beam-search. However, vanilla Seq2Seq model performs worse in generating long sentences. The reason is that the encoder needs to compress the whole sentence into a fix length representation. To address this problem, Bahdanau et al. (2014) introduce an attention mechanism which selects important parts of the source sentence with respect to the previous hidden state in decoding the next state. Afterward, some studies focus on improving attention mechanism (Luong et al., 2015). So far, Seq2Seq models and attention mechanism have been applied to many fields such as dialog (Serban et al., 2016) generation, text summarization (Nallapati et al., 2016) and etc.

In this paper, we first attempt to formalize the ATE as a sequence-to-sequence learning task because it can make full use of both the meaning of the sentence and label dependencies compared with existing methods. Furthermore, we design a position-aware attention model and gated unit networks to make Seq2Seq model better suit to this task. Generally, Seq2Seq model is time-consuming in many fields because the target vocabulary size is very large, but the time costs in ATE is acceptable because the target vocabulary size is 3.

## 5 Conclusion and Future Work

In this paper, we propose a sequence-to-sequence learning based approach to address the ATE task. Our proposed method can take full advantage of the meaning of the whole sentence and the previous label during the decoding process. Furthermore, we find that each word's adjacent words and its own word representation are key factors for its label, and we propose a PAA and GUN model to incorporate two kinds of information into our model. The experimental results demonstrate that our approach can achieve comparable performances on ATE task. In our future work, we plan to apply our approach to other sequence labeling tasks, such as named entity recognition, word segmentation and so on.

## Acknowledgments

We thank reviewers for helpful comments. Our work is supported by the National Key Research and Development Program of China under Grant No.2017YFB1002101 and National Natural Science Foundation of China under Grant No.61433015. The corresponding author of this paper is Houfeng Wang.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research*, pages 1137–1155.

Maryna Chernyshevich. 2014. Ihs r&d belarus: Cross-domain extraction of product features using crf. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014)*, pages 309–313.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, pages 2493–2537.

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

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.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1035–1045.

Wei Jin, Hung Hay Ho, and Rohini K Srihari. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *Proceedings of 2009 International Conference on Machine Learning*, pages 465–472.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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 48th Annual Meeting of the Association for Computational Linguistic*, pages 653–661.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. *arXiv preprint arXiv:1805.00760*.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2886–2892.

Kang Liu, Heng Li Xu, Yang Liu, and Jun Zhao. 2013. Opinion target extraction using partially-supervised word alignment model. In *Proceedings of 2013 International Joint Conference on Artificial Intelligence*, pages 2134–2140.

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.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Tengfei Ma and Xiaojun Wan. 2010. Opinion target extraction in chinese news comments. In *Proceedings of The 23th International Conference on Computational Linguistics*, pages 782–790.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*, pages 171–180.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *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, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 486–495.

Maria Pontiki, John Galanis, Dimitris Pavlopoulos, Haris 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 19–30.

Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*, pages 9–28.

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, pages 42–49.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics*, pages 9–27.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. *arXiv preprint arXiv:1707.09861*.

Iulian Vlad Serban, Alessandro Sordoni, Yoshua Bengio, Aaron C Courville, and Joelle Pineau. 2016. Building end-to-end dialogue systems using generative hierarchical neural network models. In *Proceedings of The Thirtieth AAAI Conference on Artificial Intelligence*, pages 3776–3784.

Lei Shu, Hu Xu, and Bing Liu. 2017. Lifelong learning crf for supervised aspect extraction. *arXiv preprint arXiv:1705.00251*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ivan Titov and Ryan McDonald. 2008. A joint model of text and aspect ratings for sentiment summarization. *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistic*, pages 308–316.

Zhiqiang Toh and Jian Su. 2016. Nlangp 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.

Bo Wang and Houfeng Wang. 2008. Bootstrapping both product features and opinion words from chinese customer reviews with cross-inducing. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*, pages 289–295.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. *arXiv preprint arXiv:1603.06679*.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proceedings of The Thirty-First AAAI Conference on Artificial Intelligence*, pages 3316–3322.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1533–1541.

Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. *arXiv preprint arXiv:1805.04601*.

Bishan Yang and Claire Cardie. 2013. Joint inference for fine-grained opinion extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1640–1649.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. *arXiv preprint arXiv:1605.07843*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 43–50.