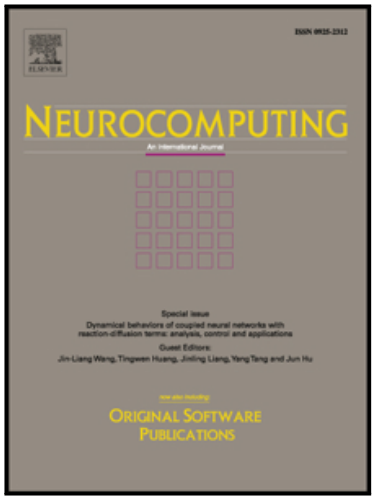
Communicated by Prof. Zidong Wang

Accepted Manuscript



Semi-supervised Target-oriented Sentiment Classification

Weidi Xu, Ying Tan

PII: S0925-2312(19)30086-4
DOI: https://doi.org/10.1016/j.neucom.2019.01.059
Reference: NEUCOM 20364

To appear in: *Neurocomputing*

Received date: 18 November 2018
Accepted date: 21 January 2019

Please cite this article as: Weidi Xu, Ying Tan, Semi-supervised Target-oriented Sentiment Classification, *Neurocomputing* (2019), doi: https://doi.org/10.1016/j.neucom.2019.01.059

Highlights

- Target-oriented Aspect-based Sentiment Analysis (TABSA) is an important research topic in NLP field, which requires fine-grained reasoning. Most of the existing works focus on model design for supervised learning.
- This paper aims to improve the performance from the semi-supervised learning perspective. To best of our knowledge, it is the first time that a deep generative model is used for semi-supervised TABSA task.
- To be suitable for TABSA, special structures are presented and investigated in our method.
- Both quantitative and qualitative experiments are conducted. Experiment results indicate that our method is effective. When combing with a powerful classifier, state-of-the-art results on the standard SemEval 2014 task 4 benchmark can be obtained.
- This work is a fundamental research which can shed light on other fine-grained semisupervised classification tasks.

Semi-supervised Target-oriented Sentiment Classification

Weidi Xu, Ying Tan*

Key Laboratory of Machine Perception (Ministry of Education), School of Electronics Engineering and Computer Science, Peking University, Beijing, 100871, China

Abstract

Target-oriented aspect-based sentiment analysis (TABSA) is a sentiment classification task that requires performing fine-grained semantical reasoning about a given aspect. The amount of labeled data is usually insufficient for supervised learning because the manual annotation w.r.t. the aspects is both time-consuming and laborious. In this paper, we propose a novel semi-supervised method to derive and utilize the underlying sentiment of unlabeled samples via a deep generative model. This method assumes that when given the aspect, the sentence is generated by two stochastic variables, i.e., the context variable and the sentiment variable. By explicitly disentangling the representation into the context and sentiment, the meaning of sentiment variable can be kept clean during the training phase. An additional advantage is that the proposed method uses a standalone classifier, and as such, our system is able to integrate with various supervised models. In terms of the implementation, since capturing the conditional input is non-trivial for a sequential model, special structures are put forward and investigated. We conducted experiments on SemEval 2014 task 4 and the results indicate that our method effectively handles five kinds of advanced classifiers. The proposed method outperforms two general semi-supervised methods and achieves state-of-the-art performance on this benchmark.

*Corresponding author
Email address: wead_hsu@pku.edu.cn, ytan@pku.edu.cn (Weidi Xu, Ying Tan)

Keywords: Semi-supervised Learning, Generative Model, Sentiment Analysis, Variational Inference

1. Introduction

Target-oriented aspect-based sentiment analysis (TABSA) aims to predict the sentiment category of a certain aspect within a sentence. For instance, a review about a restaurant is "*the* $[food]_{aspect}$ *is the best in San Francisco, however, the* $[service]_{aspect}$ *is unsatisfactory.*". In this sentence, the sentiment categories for the "*food*" and "*service*" are different. With respect to the "*food*", the review implies a positive connotation, while for the "*service*", the prediction should be negative. Contrary to traditional sentiment analysis task, which focuses on extracting global semantical features, the TABSA task requires the model to extract the local context that relates to the aspect, and then derive the appropriate sentiment prediction.

The TABSA task is of great significance as it can obtain more specific information about what we are interested in. Also, since fine-grained analysis is the subject of intense interest in the NLP field, the investigation into TABSA may help to enlighten us to the benefit of other related tasks. Recently, many effective supervised TABSA models have been proposed [1, 2, 3, 4, 5, 6]. They focus on inducing patterns between the aspect and the context.

Despite the success of supervised TABSA models, their performance ultimately depends on the available amount of labeled data. However, annotating TABSA samples is both laborious and time-consuming. To annotate an unlabeled sample, one has to first find all of the relevant aspects mentioned in the text, perform sophisticated reasoning and then give the predictions for these aspects. The TABSA datasets [7, 8] that have been recently made available are usually too small to fully exploit the ability of deep models. On the other hand, unlike the scarcity of labeled data, unlabeled data is in abundance and can be easily accessed from web-sites. Utilizing unlabeled data to improve the classification performance may bring forth a positive and significant contribution

to the research on the TABSA task. Previous related semi-supervised works primarily focus on the task of aspect term extraction, as the terms with the same category have a word-level clustering property [9, 10, 11, 12]. But for the TABSA task, the aspect-based sentimental similarity between different samples is beyond the word-level semantics, which results in more difficulties.

To achieve semi-supervised TABSA, the proposed method captures the target-oriented sentiment similarity by means of a generative model. Specifically, the data is represented by two stochastic variables. The first one is the context variable, which captures the lexical information of the given sentence, and the second one is the category variable, which represents the sentiment category related to the aspect. By explicitly disentangling the latent representation, the variables are forced to have their own specific meanings. This prevents the sentiment information from being vanished when learning the representation for the unlabeled data. And it is also possible to condition the sentence generation on the sentiment and context variables w.r.t. the aspect. To maximize the generative probability, we resort to variational inference and the model is implemented via neural networks [13, 14, 15]. It consists of three main components: a classifier, an encoder, and a decoder. 1) The classifier is responsible for extracting the sentiment category when given the sentence and the concerned aspect. 2) The role of the encoder is to compress the sentence into a continuous vector, which captures the lexical information but excludes the sentiment. The meanings conveyed by the outputs of the classifier and encoder are enforced by the labeled data. 3) The decoder takes the outputs of both classifier and encoder as the input to reconstruct the original sentence. An additional advantage of separating the representation is that the classifier becomes independent, which endows the method with the ability to exploit various kinds of advanced classifiers. For simplicity, the combination of the encoder and the decoder is referred to as auto-encoder in the following.

In TABSA, in addition to the text sequence, the aspect is also provided as the input to the auto-decoder. It is desirable to integrate the connections between the aspect and the contextual words when encoding and decoding the

sentence. However, it is non-trivial for a sequential model to adequately capture the conditional input [16, 17, 18]. This work introduces and evaluates novel auto-encoders that emphasize the content and the position of the given aspect. The encoder uses two recurrent neural networks (RNN) to encode the parts that are before and after the aspect terms. The position tag technique is also employed in the input to enhance the location information. The decoder is the inverse of the encoder. And it is equipped with an extension of Long Short-Term Memory (LSTM) [19]. This module is able to model the relationship between the aspect and its context, and carry the label information during the decoding.

We apply our method on SemEval 2014 task 4 [7]. The experimental results indicate that our method is effective with five typical classifiers. The method consistently outperforms the pure-supervised classifier, as well as two general semi-supervised learning methods, i.e., in-domain word embedding pretraining and self-training. The qualitative results are also provided to analyze the learned latent space. Finally, we show that the results achieved by the best classifier in our experiments can obtain state-of-the-art results on this benchmark. The code has been made publicly available from https://github.com/wead-hsu/tssgm.

2. Related Work

Sentiment analysis is a long-standing research problem in the NLP community [20, 21]. Recently, with the release of several online datasets, abundant supervised models were put forward to tackle the TABSA problem. Tang *et al.* [1] made use of a model to encode the sentence from start and end to the aspect words. This model verifies that the neural network is able to achieve competitive performance. Tang *et al.* [2] then presented a model based on the memory network, which reasons over the sentence in multiple hops. Zhang *et al.* [22] proposed a model that treats the input sentence as three parts, i.e., words before the aspect, the aspect itself and the words that follow the aspect. These three parts are combined using a gating module. Besides these works, there is also a great deal of extant research into the TABSA task [3, 4, 5, 6]. The

dominant principle of these supervised models is to better capture and utilize the connection between the context and the aspect.

Semi-supervised learning in text classification is another related topic. A popular one, which is almost a standard practice, is to initialize the parameters with pre-trained models, e.g., word embedding [23]. Several recently proposed methods extend the scope of the pre-trained parameters from the embedding layer to more layers [24, 25, 26]. They can be used as a foundational component to provide the contextual embedding for other supervised models. These methods require extensive additional computational resources and are complementary with our method because our classifier is independent. Combining with our method may yield better performance than either technique alone, but this is a subject work that is beyond the scope of this paper.

Our method is founded on the basis of the generative model. The generative model has been successfully applied in many semi-supervised NLP tasks, e.g., text classification [18], relation extraction [27], sequence tagging [28], and semantic parsing [29]. By regarding the sentiment polarity of the unlabeled data as a latent variable, the approach implicitly induces the sentiment orientation when maximizing the data log-likelihood. For TABSA, the main problem lies in that the information of conditional aspect should be carefully modeled. Recall that different from the vanilla text classification problem where sentiment-related information corresponds to the entire sentence, TABSA solely focuses on the aspect-related content. To circumvent this problem, our method uses two explicitly disentangled variables and novel neural network structures.

3. Method Description

This section first clarifies the definition of the problem and then presents our method.

3.1. Problem Definition

In TABSA, the goal is to predict the sentiment polarity y, when given the input sentence $\mathbf{x} = \{x_1, ..., x_T\}$ and the aspect $\mathbf{a} = \{a_1, ..., a_m\}$, where $\mathbf{a}$ is a

subsequence of $\mathbf{x}$. Typically, the sentiment polarity is $\{P, O, N\}$, where P, O, N denote "positive", "neutral" and "negative" respectively. This paper considers the following semi-supervised scenario. The dataset consists of both labeled data and unlabeled data, where the labeled data is $\mathbf{S}_l = \{(\mathbf{x}_l^{(i)}, \mathbf{a}_l^{(i)}, y_l^{(i)})\}_{i=1}^{N_l}$ and the unlabeled data is $\mathbf{S}_u = \{(\mathbf{x}_u^{(i)}, \mathbf{a}_u^{(i)}\}_{i=1}^{N_l}$. In the unlabeled data, the sentence and the aspect are provided, but the label is absent. The semi-supervised TABSA aims to improve the predictive accuracy by using the unlabeled data.

3.2. The Model

A generative model is proposed for the semi-supervised TABSA. It includes two stochastic variables. The sentiment variable y is used to capture the sentiment polarity of the sentence about the aspect $\mathbf{a}$. And the context variable $\mathbf{z}$ is used to capture the lexical information. Then the sentence $\mathbf{x}$ can be generated conditioned on these two representations as well as the aspect.

Specifically, to generate a sentence $\mathbf{x}$ with the given aspect $\mathbf{a}$:

1. Draw a sentiment variable (discrete scalar) $y \sim p(y)$, where $p(y)$ is the category distribution in the dataset.

2. Draw a context variable (continuous vector) [1] $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$.

3. Draw the sentence $\mathbf{x} \sim p(\mathbf{x}|\mathbf{a}, y, \mathbf{z})$.

This generation procedure assumes that the text is represented by three parts ($\mathbf{a}$, y and $\mathbf{z}$). Our method implements this generative model using variational inference. In the following, we refer to it as Target-oriented Semi-supervised Sequential Generative Model (TSSGM). The dependency of the variables is given in Fig. 1, and the framework is depicted in Fig. 2.

[1] We choose the Gaussian distribution for more flexibility in the sequence prediction problem, as well as the advantage realized during the inference.

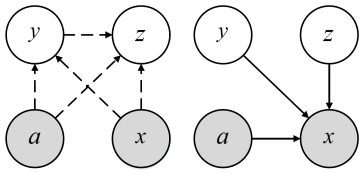


Figure 1: Illustration of TSSGM as a directed graph. **Left**: Dashed lines are used to denote variational approximation $q_\phi(y|\mathbf{x},\mathbf{a})q_\phi(\mathbf{z}|\mathbf{x},\mathbf{a},y)$. **Right**: Solid lines are used to denote generative model $p_\theta(\mathbf{x}|y,\mathbf{a},\mathbf{z})$.

3.3. Model Inference

In the semi-supervised learning setting, the objective functions differ for the labeled and unlabeled data. For the labeled data, our method aims to maximize $p(\mathbf{x},y|\mathbf{a})$, where $\mathbf{z}$ is marginalized. For the unlabeled data, the objective is to maximize $p(\mathbf{x}|\mathbf{a})$, where y is also marginalized. Direct optimization of these two objectives is intractable so we use variational inference for approximating the marginal probability.

Similar to [15], we construct the variational objective of $p(\mathbf{x},y|\mathbf{a})$ for the labeled data, also referred to as the variational evidence lower bound (ELBO), as follows:

$$\begin{aligned}\log p_\theta(\mathbf{x},y|\mathbf{a}) \geq\ & \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{a},y)}[\log p_\theta(\mathbf{x}|y,\mathbf{a},\mathbf{z})] + \log p_\theta(y) \\ & - \mathrm{KL}(q_\phi(\mathbf{z}|\mathbf{x},\mathbf{a},y)||p_\theta(\mathbf{z})) = \mathcal{L}(\mathbf{x},\mathbf{a},y)\,,\end{aligned} \tag{1}$$

where KL is the Kullback-Leibler divergence.

In dealing with the unlabeled data, the ELBO of $\log p(\mathbf{x}|\mathbf{a})$ can be extended from Eq. 1 following the variable dependency:

$$\begin{aligned}\log p_\theta(\mathbf{x}|\mathbf{a}) &\geq \mathbb{E}_{q_\phi(y|\mathbf{x},\mathbf{a})}\mathcal{L}(\mathbf{x},\mathbf{a},y) + \mathcal{H}(q_\phi(y|\mathbf{x},\mathbf{a})) \\ &= \mathcal{U}(\mathbf{x},\mathbf{a})\,,\end{aligned} \tag{2}$$

where $\mathcal{H}$ is the entropy function.

We also include the additional classification loss for $q_\phi(y|\mathbf{x},\mathbf{a})$ with the labeled data. Combining the above objectives, the overall objective function to

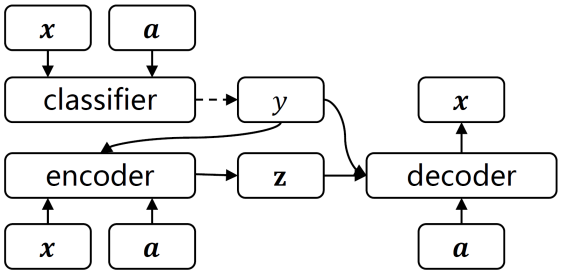


Figure 2: This is the sketch of our model. **Classifier**: When using unlabeled data, the distribution of $y \sim q_\phi(y|\mathbf{x}, \mathbf{a})$ is provided by the classifier. **Encoder**: The sequence is encoded by the encoder. The encoding and the label y are used to parameterize the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$. **Decoder**: The context variable $\mathbf{z}$ (sampled from $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$) and the label y are passed to the generative network to estimate the probability $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$.

minimize for the entire data set is:

$$G = \sum_{(\mathbf{x},\mathbf{a},y)\in S_l} -\mathcal{L}(\mathbf{x}, \mathbf{a}, y) + \sum_{(\mathbf{x},\mathbf{a})\in S_u} -\mathcal{U}(\mathbf{x}, \mathbf{a}) + \gamma \sum_{(\mathbf{x},\mathbf{a},y)\in S_l} -\log q_\phi(y|\mathbf{x}, \mathbf{a}) \,, \tag{3}$$

where γ is a hyper-parameter which controls the weight of the additional classification loss.

This objective function includes three learnable terms, i.e., $q_\phi(y|\mathbf{x}, \mathbf{a})$, $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$ and $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ and they are modeled by three independent neural networks respectively. In the following, we refer to them as the classifier, the encoder, and the decoder. The role of the classifier is to predict the sentiment polarity when given an aspect and the corresponding sentence. Because it is a stand-alone module in our method, it can be implemented using various supervised TABSA models. The encoder extracts the lexical feature from the data. The meanings of y and $\mathbf{z}$ are guaranteed through learning from the labeled data. With y and $\mathbf{z}$ obtained, the decoder can reconstruct the input sentence for a given aspect.

Since the choice the classifier is optional, the description of this component will be brief. Rather, this paper focuses on the implementation of the auto-encoder. Specifically, two auto-encoder structures and two kinds of decoder

RNNs are investigated.

3.4. Classifier

Many choices of the classifier are currently available and integrable with our system. For labeled data, the classifier is trained using the cross-entropy loss between the prediction and the target label. When the labels are unknown, the label predictive distribution $q_\phi(y|\mathbf{x}, \mathbf{a})$ is tuned through maximizing the ELBO of the $\log p(\mathbf{x}|\mathbf{a})$. In this work, we experiment with five canonical classifiers from the recent literature.

3.5. Encoder

The encoder models the term $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$, which obtains the context variable from the sentence when supplied with the aspect and sentiment. The extracted context variable is independent of the label y. Here two kinds of structures are investigated.

Uni-directional Encoder. The first is an uni-directional encoder. It uses a single LSTM to encode the input sequence $\mathbf{x}$ and integrate y using a transformation layer. Denote that the LSTM network maps x_t to the hidden state $\mathbf{h}_t^e$. The hidden state of the last time-step $\mathbf{h}_T^e$ is used to derive the distribution of variable $\mathbf{z}$. Then $\mathbf{h}_T^e$ and y is concatenated into a vector, which is then used to compute the mean and variance of $\mathbf{z}$ [2]:

$$\mathbf{z} \sim \mathcal{N}(\mu(\mathbf{x}, y), diag(\sigma^2(\mathbf{x}, y))) \,, \tag{4}$$

$$\mu(\mathbf{x}, y) = \tanh(\mathbf{W}_\mu[\mathbf{h}_T^e : \mathbf{y}] + \mathbf{b}_\mu) \,, \tag{5}$$

$$\log \sigma(\mathbf{x}, y) = \tanh(\mathbf{W}_\sigma[\mathbf{h}_T^e : \mathbf{y}] + \mathbf{b}_\sigma) \,. \tag{6}$$

In this way, the aspect $\mathbf{a}$ is excluded from the computation and the encoder is unaware of its existence. To capture the aspect $\mathbf{a}$ in the uni-directional encoder, we employ the approach of adding position tags to the input sequence. The

[2] To propagate through $\mathbf{z}$, the reparameterization technique [13, 14, 15] is used.

position tag is widely used in other NLP tasks, e.g., relation extraction [30, 31, 32] and machine translation [33], to indicate the position of tokens. Here the position tag $\mathbf{d} = \{d_0, d_1, ..., d_T\}$ is used to denote the relative distance from the current word to the aspect. Each d_t is clipped by $[-10, 10]$. They are mapped to vectors $\in R^{d_p}$ using a transformation matrix, which is initialized as in [33]. Then the vectors are concatenated with input embedding of $\mathbf{x}$. In the presented models, the position tag is used by default for both encoder and decoder. And for the purpose of clarity, the mention of the position tag will be omitted in the following.

Bi-directional Encoder. We also study another way to implement $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$. Instead of encoding the sentence from left to right, it is desirable to capture the location of the aspect and emphasize its content. To accomplish this, we use two LSTMs to encode the sentence (cf. Fig. 3). They process the sentence from left and right sides to the aspect respectively. To clarify, the words to the left and the words to the right of the aspect are treated as two distinct parts. Denote that $\mathbf{x}$ is composed of three parts $(\mathbf{x}_l, \mathbf{a}, \mathbf{x}_r)$. A forward LSTM $\overrightarrow{LSTM}$ is used to obtain the representation vector of $[\mathbf{x}_l : \mathbf{a}]$, where : means vector concatenation. The sequence $[\mathbf{a} : \mathbf{x}_r]$ is processed by another backward LSTM $\overleftarrow{LSTM}$. Then the stochastic variable $\mathbf{z}$ can be derived as:

$$\begin{aligned}
\mathbf{z} &\sim \mathcal{N}(\mu(\mathbf{x}, \mathbf{a}, y), diag(\sigma^2(\mathbf{x}, \mathbf{a}, y)))\,, \\
\mathbf{g} &= \tanh(\mathbf{W}_g[\mathbf{g}_l : \mathbf{g}_r] + \mathbf{b}_g) \\
\mu(\mathbf{x}, \mathbf{a}, y) &= \tanh(\mathbf{W}_\mu[\mathbf{g} : \mathbf{y}] + \mathbf{b}_\mu)\,, \\
\log\sigma(\mathbf{x}, \mathbf{a}, y) &= \tanh(\mathbf{W}_\sigma[\mathbf{g} : \mathbf{y}] + \mathbf{b}_\sigma)\,,
\end{aligned}$$

where $\mathbf{g}_l$ ($\mathbf{g}_r$) corresponds to the output of last step of $\overrightarrow{LSTM}([\mathbf{x}_l : \mathbf{a}])$ ($\overleftarrow{LSTM}([\mathbf{a} : \mathbf{x}_r])$). By splitting the sequence into two parts, the position and the content of the aspect term $\mathbf{a}$ can be better recognized and aggregated by the encoder.

3.6. Decoder

The decoder models the term $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$. Conditional sequence generation is well-known to be semantically complex [16, 17, 18]. It is non-trivial for an

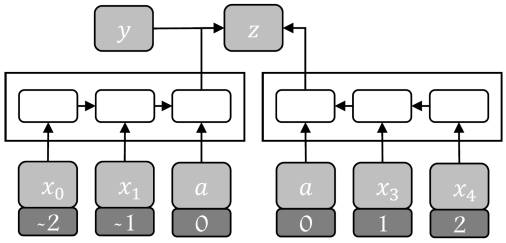


Figure 3: Bi-directional Encoder: This figure considers a sentence $\mathbf{x}$ with 5 tokens $(x_0, x_1, x_2, x_3, x_4)$, where x_2 is the aspect. It is split into two parts and each part is processed by an LSTM network. The last states of these two LSTMs, as well as the label, are used for computing the distribution of $\mathbf{z}$. The position tags is included in the input.

RNN to capture the conditional input. Therefore, the main question here is how to implement the model to estimate the generative probability $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ without neglecting the aspect $\mathbf{a}$ and sentiment variable y. To capture $\mathbf{a}$, we investigate two decoder structures similar to the encoder. And to capture y, two plausible RNNs are presented and verified.

Uni-directional Decoder. An RNN f_d is adopted here, which takes $\mathbf{z}$ as the initial state and y at each time-step. When given a sequence $\mathbf{x}$, conditional input y and $\mathbf{z}$, the decoding process can be presented as:

$$\mathbf{h}_0^d = \tanh(\mathbf{W}_d([\mathbf{y} : \mathbf{z}]))\,, \tag{7}$$

$$\mathbf{h}_t^d = f_d(x_t, \mathbf{y}, \mathbf{h}_{t-1}^d), \quad t = 1, ..., T\,, \tag{8}$$

$$p(x_{t+1}) = \text{softmax}(\mathbf{W}_p(\mathbf{h}_t^d))\,, \tag{9}$$

$$\log p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z}) = \sum_{x_t} \log p(x_t), \quad x_t \in \mathbf{x} \wedge x_t \notin \mathbf{a}\,, \tag{10}$$

where the bold $\mathbf{y}$ is the one-hot encoded vector. The formulation of decoder RNN f_d will be illustrated in the following.

In this implementation, 1) the reconstruction loss of aspect terms is omitted, given that the aspect is a conditional input. 2) The conditional label y is fed into the sequential model at each time-step because doing so will prevent the decoder from neglecting y [18, 34]. Otherwise, the classifier is unable to obtain the valid gradient from the decoder in Eq. 2.

Bi-directional Decoder. Another way to implement $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ is to reverse the process of the bi-directional encoder (sketched in Fig. 4). That is, two RNNs are adopted and each perceives the concatenation of y and $\mathbf{z}$ as the initial state. And the sentence is generated as follows:

$$
\begin{aligned}
\overleftarrow{\mathbf{h}}_0^d = \overrightarrow{\mathbf{h}}_0^d &= \tanh(\mathbf{W}_d[\mathbf{y} : \mathbf{z}] + b_d)\,, \\
\overleftarrow{\mathbf{h}}_t^d &= \overleftarrow{f}_d(x_t, \mathbf{y}, \overleftarrow{\mathbf{h}}_{t-1}^d), \quad x_t \in \overleftarrow{[\mathbf{x}_l : \mathbf{a}]} \\
p(x_{t+1}|\cdot) &= \mathrm{softmax}(\mathbf{W}_p \overleftarrow{\mathbf{h}}_t^d + b_p)\,, \\
\log p_\theta(\mathbf{x}_l|\mathbf{a}, y, \mathbf{z}) &= \sum_{x_t} \log p(x_t|\cdot), \quad x_t \in \mathbf{x_l}\,, \\
\overrightarrow{\mathbf{h}}_t^d &= \overrightarrow{f}_d(x_t, \mathbf{y}, \overrightarrow{\mathbf{h}}_{t-1}^d), \quad x_t \in \overrightarrow{[\mathbf{a} : \mathbf{x}_r]} \\
p(x_{t+1}|\cdot) &= \mathrm{softmax}(\mathbf{W}_p \overrightarrow{\mathbf{h}}_t^d + b_p)\,, \\
\log p_\theta(\mathbf{x}_r|\mathbf{a}, y, \mathbf{z}) &= \sum_{x_t} \log p(x_t|\cdot), \quad x_t \in \mathbf{x_r}\,.
\end{aligned}
$$

It is equivalent to generating the left and right part using two different uni-directional decoders. As the aspect appears prior to other tokens, the decoder is able to carry its information when processing the context.

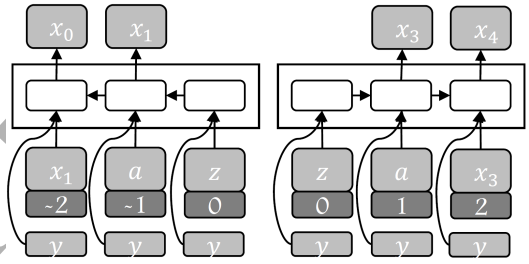


Figure 4: Bi-directional Decoder: The left and right parts of the sentence are reconstructed by two RNNs. Each RNN takes $\mathbf{a}$, $\mathbf{z}$ and y as input, and estimates the probability of generating a part of $\mathbf{x}$, excluding $\mathbf{a}$.

3.7. Implementations of f_d

Vanilla LSTM is unable to handle the conditional input. Here two extensions of the LSTM network are presented to implement $f_d(x_t, y, \mathbf{h}_{t-1})$.

CLSTM. To perceive y along the decoding, the conditional LSTM (CLSTM) is used, which receives y as an additional input at each time-step. Specifically, when the previous state $\mathbf{h}_{t-1}$ is given, CLSTM implements $f_d(x_t, y, \mathbf{h}_{t-1})$ as:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i x_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{V}_i \mathbf{y} + b_i)\,, \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f x_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{V}_f \mathbf{y} + b_f)\,, \\
\hat{\mathbf{c}}_t &= \tanh(\mathbf{W}_c x_t + \mathbf{U}_c \mathbf{h}_{t-1} + \mathbf{V}_c \mathbf{y} + b_c)\,, \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o x_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{V}_o \mathbf{y} + b_o)\,, \\
\mathbf{c}_t &= \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \hat{\mathbf{c}}_t\,, \\
\mathbf{h}_t &= \mathbf{o}_t \cdot \tanh(\mathbf{c}_t)\,,
\end{aligned}
$$

where $\mathbf{V}_i, \mathbf{V}_f, \mathbf{V}_c, \mathbf{V}_o$ are extra parameters used to incorporate the information from y.

FcLSTM. The study in [35, 36] demonstrates that the hidden units of the LSTM are able to represent the complex semantical feature, e.g., sentiment category. Therefore, it is feasible to append y as an additional part of the LSTM cell. This kind of implementation is referred to as Fixed-cell LSTM (FcLSTM), which works as follows:

$$
\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i x_t + \mathbf{U}_i \mathbf{h}_{t-1} + b_i)\,, \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f x_t + \mathbf{U}_f \mathbf{h}_{t-1} + b_f)\,, \\
\hat{\mathbf{c}}_t &= \tanh(\mathbf{W}_c x_t + \mathbf{U}_c \mathbf{h}_{t-1} + b_c)\,, \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o x_t + \mathbf{U}_o \mathbf{h}_{t-1} + b_o)\,, \\
\mathbf{c}_t &= \mathbf{f}_t \mathbf{c}_{t-1} + \mathbf{i}_t \hat{\mathbf{c}}_t\,, \\
\mathbf{h}_t &= \mathbf{o}_t \cdot \tanh([\mathbf{c}_t : \mathbf{y}])\,,
\end{aligned}
$$

where the size of $\mathbf{o}_t$ is $d_h + d_y$ and the size of $\mathbf{i}_t, \mathbf{f}_t, \hat{\mathbf{c}}_t$ is d_h. During the decoding, the value of y can be extracted by opening o_t, which is computed by x_t and $\mathbf{h}_{t-1}$.

4. Experiments

4.1. Datasets and Preparation

The model was evaluated on two datasets from the SemEval 2014 task 4 ABSA Challenge [7]: the Restaurant (`REST`) dataset and the Laptop (`LAPTOP`) dataset. The `REST` dataset contains comments relevant to the restaurant domain, while the `LAPTOP` dataset contains descriptive comments about laptop products. The statistics for these datasets are listed in Table 1. For consistency in comparing methods, we follow the same data-preprocessing procedure as the other work [6]. The data set has some samples labeled as "Conflict". These samples are removed. All of the text in the sample is lowercase, and there is no additional preprocessing, such as deleting stop words, symbols, or numbers.

The unlabeled data is obtained from other datasets in the same domain. For the `REST` dataset, we used the samples from a sentiment classification competition hosted by Kaggle [3]. This dataset is composed of 82K training samples and each is annotated with a coarse-grained sentiment label. For the `LAPTOP` dataset, the "Six Categories of Amazon Product Reviews" [4] dataset is adopted. Among the six categories of product, we draw 412K laptop-related samples.

The NLTK software [37] is utilized to split the long paragraphs into sentences. These sentences are used to construct the data samples. Two aspect extractors [5] are trained with the `REST` and `LAPTOP` dataset respectively and their F1 score is 88.42 and 80.12. And then, they are used to label the aspect for the unlabeled sentences. After the sequence labeling, the samples without aspect are filtered. In addition, those with a sequence length that is greater than 80 are also removed. The statistics for the final dataset are shown in Table. 2.

[3] https://inclass.kaggle.com/c/restaurant-reviews
[4] http://times.cs.uiuc.edu/∼wang296/Data/
[5] https://github.com/guillaumegenthial/sequence_tagging

		# Positive	# Negative	# Neutral
REST	Train	2159	800	632
	Test	730	195	196
LAPTOP	Train	980	858	454
	Test	340	128	171

Table 1: The statistics of the datasets.

		Avg. Length	Std. Length
REST	Labeled	20.06	10.38
	Unlabeled	22.70	12.38
LAPTOP	Labeled	21.95	11.80
	Unlabeled	29.89	17.33

Table 2: The statistics of the reviews.

4.2. Model Configuration & Classifiers

The system is implemented using Tensorflow [38]. For consistency and accuracy in our evaluation, all experiments share a common set of the hyper-parameters. The size of the LSTM cell is 100 and the dimension of latent variable $\mathbf{z}$ is 50. As mentioned in [39], the KL weight (denoted as klw) in Eq. 1 should be tuned so that the latent variable will not get stuck in the local optimum where $\mathbf{z}$ carries no information. During the experiments, we set the klw to be 1e-4. The pre-trained embedding, i.e., GloVe [23], is used to initialize the word embedding matrix where the out-of-vocabulary words are excluded [6]. The hyper-parameter γ is set to be 10 and the dimension of position embedding is 50. The number of randomly selected unlabeled samples is 10K in our experiments. We have tested the method using larger amounts of unlabeled data and found that the performance does not improve significantly. The test accuracy w.r.t. γ, klw and number of unlabeled samples is shown in Fig. 5. An open-source imple-

[6] http://nlp.stanford.edu/data/glove.840B.300d.zip

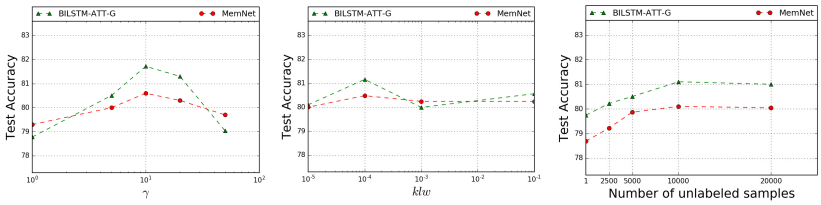


Figure 5: Hyper-parameter tuning of TSSGM.

mentation of our method is available from https://github.com/wead-hsu/tssgm.

Five classical TABSA classifier are tested, i.e., MemNet [2], TC-LSTM [1], BILSTM-ATT-G [5], IAN [40] and TNet [6].

- **TC-LSTM**: TC-LSTM makes use of two LSTMs to separately encode the sentence from each end of the text, left and right, up to the aspect. Then the concatenation of the LSTM outputs is used to predict the sentiment polarity.

- **MemNet**: MemNet utilizes attention mechanism to reason over the input words in multiple hops, so the model can perform reasoning with respect to the given aspect. After the multi-hops computation, the vector of the final round is used as input into a fully-connected layer to predict the sentiment polarity.

- **IAN**: IAN also uses two LSTMs to interactively derive the representation vectors of the context and the aspect terms. They are then concatenated to form the input of the final prediction layer.

- **BILSTM-ATT-G**: BILSTM-ATT-G treats the sentence as a combination of three parts. It adopts two attention-based LSTMs to obtain the left and right representation vectors. And a contextualized attention module is used to combine these two vectors. The resulting vector is then supplied as input to the final transformation layer.

- **TNet-AS**: Instead of using an attention module, TNet employs a CNN layer to extract salient features from the transformed word representations originated from a bi-directional RNN layer. Among current supervised models, TNet achieves currently state-of-the-art results on the SemEval 2014 task 4 datasets. Among TNet variants, TNet-AS is adopted in our experiments.

We re-implemented these classifiers so that they can be integrated into our Tensorflow code. The hyper-parameters, as well as the training setting, are guaranteed to be the same as in their original paper. Experiments are conducted with different classifiers to verify the robustness of the proposed method. The experimental results show that it can consistently improve the classification performance for various classifiers.

4.3. Main Results

The main experimental results are shown in Table 3 for the `REST` and `LAPTOP` dataset. We adopted two evaluation metrics here, i.e., the classification accuracy and the Macro-averaged F1 score. The first one is commonly used in standard classification problem and the latter is better suitable for the multi-label classification tasks, especially in cases of imbalanced datasets. In this table, the full TSSGM model that employs the bi-directional auto-encoder and position tag is used. We report the mean and the standard deviation after 5 runs for each experiment. Four comparison experiments are conducted for each classifier. Denote *clf* as the classifier being used:

- *clf*: We perform supervised training for the classifier using only labeled data. This is performed to demonstrate whether the proposed method achieves the goal of semi-supervised learning.

- *clf* (EMB): We also perform the semi-supervised experiments using in-domain pre-trained word embedding. The CBOW method [41] is used to pre-train the embedding vectors using both labeled and unlabeled data.

Models	REST		LAPTOP	
	Accuracy	Macro-F1	Accuracy	Macro-F1
CNN-ASP	77.82 ♮	-	72.46 ♮	-
AE-LSTM	76.60 ♮	-	68.90 ♮	-
ATAE-LSTM	77.20 ♮	-	68.70 ♮	-
GCAE	77.28 (0.32) ♮	-	69.14 (0.32) ♮	-
TC-LSTM	77.97 (0.16)	67.55 (0.32)	68.42 (0.56)	62.42 (1.10)
TC-LSTM (EMB)	77.18 (0.38)	65.97 (0.44)	67.51 (0.72)	60.31 (1.28)
TC-LSTM (ST)	78.19 (0.36)	67.65 (0.43)	68.47 (0.47)	62.54 (0.74)
TC-LSTM (TSSGM)	**78.64** (0.28)	**68.71** (0.82)	**69.08** (0.56)	**63.06** (0.51)
MemNet	78.68 (0.23)	68.18 (0.58)	70.28 (0.32)	64.38 (0.86)
MemNet (EMB)	79.47 (0.38)	69.06 (0.21)	72.17 (0.44)	65.06 (0.73)
MemNet (ST)	78.83 (0.20)	68.92 (0.20)	69.52 (0.36)	64.39 (0.67)
MemNet (TSSGM)	**80.18** (0.26)	**69.46** (0.43)	**72.22** (0.58)	**65.88** (0.45)
IAN	79.20 (0.19)	68.71 (0.59)	69.48 (0.52)	62.90 (0.99)
IAN (EMB)	79.46 (0.38)	69.45 (0.38)	70.89 (0.48)	65.27 (0.34)
IAN (ST)	79.45 (0.11)	69.36 (0.71)	**73.25** (0.81)	**68.25** (0.76)
IAN (TSSGM)	**80.23** (0.17)	**70.32** (1.00)	72.04 (0.39)	65.39 (0.85)
BILSTM-ATT-G	79.74 (0.22)	69.16 (0.53)	74.26 (0.35)	69.54 (0.53)
BILSTM-ATT-G (EMB)	80.27 (0.44)	70.33 (0.51)	73.61 (0.30)	68.25 (0.63)
BILSTM-ATT-G (ST)	80.54 (0.23)	71.28 (0.19)	74.70 (0.41)	70.31 (0.60)
BILSTM-ATT-G (TSSGM)	**81.10** (0.37)	**72.17** (0.26)	**75.34** (0.22)	**70.80** (0.49)
TNet-AS	80.57 (0.22)	71.17 (0.54)	76.44 (0.47)	71.38 (0.79)
TNet-AS (EMB)	80.92 (0.54)	71.01 (0.98)	76.56 (0.67)	71.51 (0.88)
TNet-AS (ST)	80.76 (0.22)	71.32 (0.67)	76.88 (0.38)	71.74 (0.64)
TNet-AS (TSSGM)	**81.76** (0.17)	**72.57** (0.32)	**77.57** (0.31)	**72.31** (0.69)

Table 3: Experimental results (%). For each classifier, we performed three experiments as follows: the standard supervised classifier, the supervised classifier with pre-trained embedding using unlabeled data and our model with the classifier. The results are obtained after 5 runs, and we report both the mean and standard deviation of the test accuracy and the Macro-averaged F1 score. For clarity, better results are in bold. ♮ denotes that the results are extracted from the original paper.

The resulting vectors are used to initialize the embedding matrix, rather than the pre-trained GloVe used in the original work.

- *clf* (ST): The self-training (ST) method is another semi-supervised baseline. It works by iteratively increasing the labeled data by selecting 1K samples with the highest confidence from the unlabeled data. Pseudo labels are assigned by the trained classifier. This process continues until all the unlabeled data is labeled.

- *clf* (TSSGM): The final set of experiments involves the proposed method TSSGM that uses *clf* as the classifier. As aforementioned, the classifier is an independent module in our method and thus it is easy to integrate various classifiers. All of the hyper-parameters are the same as those that were used in the supervised setting.

The table also includes the results of several supervised results of other models in the first block, i.e., CNN-ASP [6], AE-LSTM [42], ATAE-LSTM [42], GCAE [43].

Experimental results indicate that our method is able to consistently improve the performance as compared to the supervised models. For instance, the test accuracy can be improved from 78.68% to 80.18% (1.5% absolute improvement) when using the MemNet classifier. Among five classifiers, TNet-AS achieves the best performance. When TSSGM uses TNet-AS as the classifier, the state-of-the-art results are obtained on this dataset.

The effectiveness of TSSGM is further seen in the comparison against other two semi-supervised methods. Specifically, for all of the experiments except using IAN on the `LAPTOP` dataset, TSSAVE demonstrates superior performance than the ST and EMB methods. Regarding IAN, it is observed to be prone to over-fitting in the early-training phase. However, the auto-encoder in TSSAVE converges at a slower rate than the classifier and therefore the improvement afforded by TSSGM is not very significant in this case.

From the table, the usage of in-domain pre-trained word embedding is generally beneficial compared to GloVe. It is noteworthy that when the EMB and

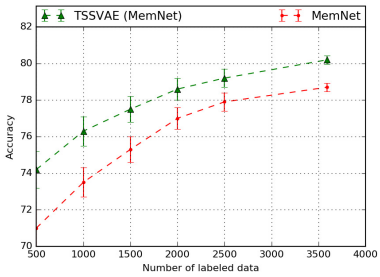


Figure 6: The test accuracy with respect to the number of labeled samples from the REST dataset with the MemNet classifier.

TSSAVE methods are combined together, BILSTM-ATT-G (TSSGM) is able to achieve the test accuracy of 81.22% with the REST dataset.

4.4. Effect of Labeled Data

Here we study how the performance of TSSGM varies with the number of labeled samples. Without loss of generality, we use MemNet as the basic classifier. Different amount of labeled data is sampled from the labeled set and the results are reported in Fig. 6. As the test curve depicts, the testing accuracy decreases with fewer labeled samples, but the improvement against supervised results is more evident. With 500 labeled samples, TSSGM can achieve 3% accuracy improvement, which illustrates the proposed method performs effectively even with a very small amount of labeled data.

4.5. Share Embedding or Not?

In previous works, the encoder and decoder typically share the word embedding matrix, as well as the classifier, to reduce the demand on the computational resources. In other words, the embedding vectors are also trained by learning to reconstruct the input sequence. This leads to a question of whether the improvement is a result of multi-task learning, i.e., the classification and the text reconstruction. In our experiments, the classifier and the auto-encoder possess their own embedding matrices, which ensures that the improvement comes from the usage of TSSGM.

Accuracy	Not sharing	Sharing
TC-LSTM (TSSGM)	**78.64**	77.98
MemNet (TSSGM)	**80.18**	78.66
IAN (TSSGM)	**79.88**	79.42
BILSTM-ATT-G (TSSGM)	**81.10**	78.28
TNet-AS (TSSGM)	**81.76**	79.11

Table 4: Comparison between sharing embedding and not on the REST dataset.

Here, we investigate whether sharing the embedding will benefit the classifier. The results are given in Table 4. From the results, we can see that the joint training of the word embedding is negative for the final performance in TSSGM. This suggests that in this problem the gradients from these two objectives may collide with each other, which leads to a performance degradation.

4.6. Ablation Study

To investigate the impact of each individual components, such as bi-directional LSTM and position embedding, we perform a comparison between the full TSSGM model and its ablations (cf. Table 5). Without loss of generality, we consider the performance with the BILSTM-ATT-G and TNet-AS classifiers on the REST dataset. After replacing the bi-directional encoder and decoder with the uni-directional version, the results become inferior. In particular, it shows that using the bi-directional auto-encoder is beneficial for TSSGM, which indicates the effective integration of context information into the aspect term is crucial for good performance.

In comparing the results between TSSGM and TSSGM w/o position embedding, we observe that performance of TSSGM degrades without position embedding. The position embedding benefits TSSGM by providing relative distance between tokens and the aspect terms, which allows TSSGM to better recognize the expression related to the aspect.

	Accuracy	Macro-F1
BILSTM-ATT-G (TSSGM) w/ uni	80.44	71.34
BILSTM-ATT-G (TSSGM) w/o pe	80.35	71.11
BILSTM-ATT-G (TSSGM)	**81.10**	**72.17**
TNet-AS (TSSGM) w/ uni	80.77	71.54
TNet-AS (TSSGM) w/o pe	81.05	71.88
TNet-AS (TSSGM)	**81.76**	**72.57**

Table 5: Comparison between bi-directional TSSGM and its ablated variants on the `REST` dataset. *uni* denotes that the uni-directional auto-encoder is adopted and *pe* denotes the use of position embedding.

Accuracy	CLSTM	FcLSTM
TC-LSTM (TSSGM)	78.46	**78.64**
MemNet (TSSGM)	**80.18**	79.43
IAN (TSSGM)	**79.88**	79.44
BILSTM-ATT-G (TSSGM)	80.23	**81.10**
TNet-AS (TSSGM)	81.11	**81.76**

Table 6: Comparison between CLSTM decoder and FcLSTM decoder on the `REST` dataset.

4.7. Analysis of Decoder Structures

This work studies two sequential models in the decoder, i.e., CLSTM and FcLSTM. Table 6 describes the results obtained when using these two decoders. The performance of the decoder varies with the classifier used in TSSGM. This is due to the training dynamics inherent to these decoders and classifiers. CLSTM perceives the information of y more directly, hence it learns to capture the conditional input more quickly. This coordinates with the MemNet and IAN classifier, as they have fewer parameters and can converge faster as well. In contrast, in cases of the other classifiers, FcLSTM demonstrates better performance.

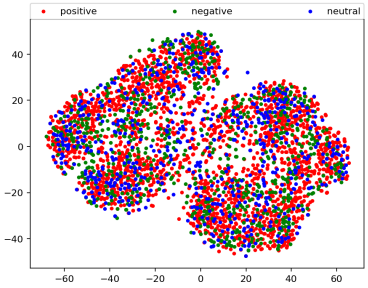


Figure 7: The distribution of the REST dataset in latent space $\mathbf{z}$ using t-SNE.

4.8. Analysis of the Latent Space

In TSSGM, the data is represented by two variables, i.e., y and $\mathbf{z}$. They are disentangled and have different meanings. Recall that y denotes the sentiment polarity and $\mathbf{z}$ captures the lexical information. We plot the scatter diagrams of $\mathbf{z}$ for three different sentiment polarities in Fig. 7. The figure illustrates that the distributions with different y are the same, indicating that $\mathbf{z}$ and y are successfully disentangled. The latent space consists of two clusters, as the description of the sentimental context locates in the different position, i.e., the left or the right to the aspect. When digging into local areas, its interesting to discover that sentences sharing similar syntactic and lexical structures around the aspect are learned to cluster together.

TSSGM is also capable of generating data samples that are conditioned on y and $\mathbf{z}$. We selects several generated sentences in Table 7. When $\mathbf{z}$ is given, the sentences are generated with different sentiments. Table 7 demonstrates that the sentences generated with the same z share the similar lexical structure but have completely different sentiment orientations. This verifies that the proposed method is able to successfully perceive and model the relationship between y and $\mathbf{z}$.

5. Conclusion and Future Work

A novel method has been proposed for the semi-supervised TABSA task based on the generative model. The analytical and experimental work has been

Positive	**Negative**	**Neutral**
... the best *food* i 've ever had !!! ...	.. the worst *food* i 've ever had !!! ...	... had the *food* in the restaurant ...
... the *lox* is very tasty ...	... the *lox* is a bit of boring ...	... lox with a glass of chilli sauce ...
... the *rice* is a great value ...	... the *rice* is awful ...	... the *rice* with a couple of olives salad ...

Table 7: Nice sentences that are generated by controlling the sentiment polarity y using the decoder trained on the `REST` dataset.

carried out to demonstrate its effectiveness. We conducted experiments with five classical classifiers and all of them are valid with TSSGM, which verifies its universality.

In future work, one question that we seek to answer is whether it is possible to reconstruct the aspect terms, rather than the entire sentence. Our motivation for this rests in the fact that it is difficult to generate the sentence conditionally. Secondly, it is assumed that information about this aspect is known in the present work, and there is a problem of error propagation when using a pre-trained aspect extractor. With this considered, we feel there is value in exploring whether it is possible to jointly learn the aspect, as well as the sentiment polarity of unlabeled data. If ultimately we can extract detailed knowledge from unlabeled data then it will be of great benefit in similar tasks.

Acknowledgment

Thanks to Xin Li for sharing his TNet code. This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61673025 and 61375119 and Supported by Beijing Natural Science Foundation (4162029), and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China under grant no. 2015CB352302.

References

References

[1] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: COLING 2016, 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, December 11-16, 2016, Osaka, Japan, 2016, pp. 3298–3307.
URL http://aclweb.org/anthology/C/C16/C16-1311.pdf

[2] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, 2016, pp. 214–224.
URL http://aclweb.org/anthology/D/D16/D16-1021.pdf

[3] S. J. Pan, W. Wang, Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, 2018, pp. 2171–2181.
URL https://aclanthology.info/papers/P18-1202/p18-1202

[4] F. Liu, T. Cohn, T. Baldwin, Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers), 2018, pp. 278–283.
URL https://aclanthology.info/papers/N18-2045/n18-2045

[5] Y. Zhang, J. Liu, Attention modeling for targeted sentiment, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2017, Valencia, Spain, April 3-7, 2017, Volume 2: Short Papers, 2017, pp. 572–577.
URL https://aclanthology.info/papers/E17-2091/e17-2091

[6] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: ACL, 2018, pp. 946–956.

[7] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, 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., 2014, pp. 27–35.
URL http://aclweb.org/anthology/S/S14/S14-2004.pdf

[8] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. D. Clercq, V. Hoste, M. Apidianaki, X. Tannier, N. V. Loukachevitch, E. Kotelnikov, N. Bel, S. M. J. Zafra, G. Eryigit, Semeval-2016 task 5: Aspect based sentiment analysis, in: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016, 2016, pp. 19–30.
URL http://aclweb.org/anthology/S/S16/S16-1002.pdf

[9] S. Brody, N. Elhadad, An unsupervised aspect-sentiment model for online reviews, in: Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 2-4, 2010, Los Angeles, California, USA, 2010, pp. 804–812.
URL http://www.aclweb.org/anthology/N10-1122

[10] A. Mukherjee, B. Liu, Aspect extraction through semi-supervised modeling, in: The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers, 2012, pp. 339–348.
URL http://www.aclweb.org/anthology/P12-1036

[11] Q. Liu, B. Liu, Y. Zhang, D. S. Kim, Z. Gao, Improving opinion aspect extraction using semantic similarity and aspect associations, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2016, pp. 2986–2992.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11973

[12] R. He, W. S. Lee, H. T. Ng, D. Dahlmeier, An unsupervised neural attention model for aspect extraction, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers, 2017, pp. 388–397. doi:10.18653/v1/

P17-1036.
URL https://doi.org/10.18653/v1/P17-1036

[13] D. P. Kingma, M. Welling, Auto-encoding variational bayes, in: The International Conference on Learning Representations (ICLR), Banff, Canada, 2014.

[14] D. J. Rezende, S. Mohamed, D. Wierstra, Stochastic backpropagation and approximate inference in deep generative models, in: Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014, 2014, pp. 1278–1286.
URL http://jmlr.org/proceedings/papers/v32/rezende14.html

[15] D. P. Kingma, S. Mohamed, D. J. Rezende, M. Welling, Semi-supervised learning with deep generative models, in: Advances in Neural Information Processing Systems, 2014, pp. 3581–3589.

[16] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov, E. P. Xing, Controllable text generation, arXiv preprint arXiv:1703.00955.

[17] J. Li, W. Monroe, T. Shi, A. Ritter, D. Jurafsky, Adversarial learning for neural dialogue generation, arXiv preprint arXiv:1701.06547.

[18] W. Xu, H. Sun, C. Deng, Y. Tan, Variational autoencoder for semi-supervised text classification, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA., 2017, pp. 3358–3364.
URL http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14299

[19] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780. doi:10.1162/neco.1997.9.8.1735.
URL http://dx.doi.org/10.1162/neco.1997.9.8.1735

[20] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: NAACL HLT 2011, Association for Computational Linguistics, Portland, Oregon, USA, 2011, pp. 142–150.

[21] S. Wang, C. D. Manning, Baselines and bigrams: Simple, good sentiment and topic classification, in: Proceedings of the 50th Annual Meeting of the Asso-

ciation for Computational Linguistics: Short Papers-Volume 2, Association for Computational Linguistics, 2012, pp. 90–94.

[22] M. Zhang, Y. Zhang, D. Vo, Gated neural networks for targeted sentiment analysis, in: Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA., 2016, pp. 3087–3093.
URL http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12074

[23] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1532–1543.
URL http://aclweb.org/anthology/D/D14/D14-1162.pdf

[24] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer, Deep contextualized word representations, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers), 2018, pp. 2227–2237.
URL https://aclanthology.info/papers/N18-1202/n18-1202

[25] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training.

[26] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, CoRR abs/1810.04805. arXiv:1810.04805.
URL http://arxiv.org/abs/1810.04805

[27] D. Marcheggiani, I. Titov, Discrete-state variational autoencoders for joint discovery and factorization of relations, TACL 4 (2016) 231–244.
URL https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/761

[28] X. Zhang, Y. Jiang, H. Peng, K. Tu, D. Goldwasser, Semi-supervised structured prediction with neural CRF autoencoder, in: Proceedings of the 2017 Conference

on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, 2017, pp. 1701–1711.
URL https://aclanthology.info/papers/D17-1179/d17-1179

[29] T. Kociský, G. Melis, E. Grefenstette, C. Dyer, W. Ling, P. Blunsom, K. M. Hermann, Semantic parsing with semi-supervised sequential autoencoders, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, 2016, pp. 1078–1087.
URL http://aclweb.org/anthology/D/D16/D16-1116.pdf

[30] D. Zeng, K. Liu, S. Lai, G. Zhou, J. Zhao, Relation classification via convolutional deep neural network, in: COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland, 2014, pp. 2335–2344.
URL http://aclweb.org/anthology/C/C14/C14-1220.pdf

[31] Y. Wu, D. Bamman, S. J. Russell, Adversarial training for relation extraction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, 2017, pp. 1778–1783.
URL https://aclanthology.info/papers/D17-1187/d17-1187

[32] W. Zeng, Y. Lin, Z. Liu, M. Sun, Incorporating relation paths in neural relation extraction, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, 2017, pp. 1768–1777.
URL https://aclanthology.info/papers/D17-1186/d17-1186

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, 2017, pp. 6000–6010.
URL http://papers.nips.cc/paper/7181-attention-is-all-you-need

[34] Z. Yang, Z. Hu, R. Salakhutdinov, T. Berg-Kirkpatrick, Improved variational autoencoders for text modeling using dilated convolutions, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, 2017, pp. 3881–3890.
URL http://proceedings.mlr.press/v70/yang17d.html

[35] A. Radford, R. Józefowicz, I. Sutskever, Learning to generate reviews and discovering sentiment, CoRR abs/1704.01444. arXiv:1704.01444.
URL http://arxiv.org/abs/1704.01444

[36] A. Karpathy, J. Johnson, F. Li, Visualizing and understanding recurrent networks, CoRR abs/1506.02078. arXiv:1506.02078.
URL http://arxiv.org/abs/1506.02078

[37] S. Bird, NLTK: the natural language toolkit, in: ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, Sydney, Australia, 17-21 July 2006, 2006.
URL http://aclweb.org/anthology/P06-4018

[38] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, CoRR abs/1603.04467.
URL http://arxiv.org/abs/1603.04467

[39] S. R. Bowman, L. Vilnis, O. Vinyals, A. M. Dai, R. Józefowicz, S. Bengio, Generating sentences from a continuous space, in: Proceedings of the 20th SIGNLL Conference on Computational Natural Language Learning, CoNLL 2016, Berlin, Germany, August 11-12, 2016, 2016, pp. 10–21.
URL http://aclweb.org/anthology/K/K16/K16-1002.pdf

[40] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017, 2017, pp. 4068–4074. doi:10.24963/ijcai.2017/568.
URL https://doi.org/10.24963/ijcai.2017/568

[41] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, CoRR abs/1301.3781. arXiv:1301.3781.
URL http://arxiv.org/abs/1301.3781

[42] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based LSTM for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, 2016, pp. 606–615.
URL http://aclweb.org/anthology/D/D16/D16-1058.pdf

[43] T. Li, W. Xue, Aspect based sentiment analysis with gated convolutional networks, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, 2018, pp. 2514–2523.
URL https://aclanthology.info/papers/P18-1234/p18-1234

Weidi Xu was born in 1992. He received B.S. degree in the School of Software Engineering, the South China University of Technology. He is currently a fifth-year Ph.D student at School of Electronics Engineering and Computer Science, Peking University.

His current research mainly focuses on semi-supervised learning and deep generative models, and their applications in NLP problems, such as text classification, tagging, sequence generation and parsing.

Ying Tan is a full professor and PhD advisor at EECS, and director of Computational Intelligence Laboratory, Peking University and and a guest professor at Kyushu University, Japan. He received his BEng, MS, and PhD from Southeast Univ. in 1985, 1988, and 1997, respectively. His current research interests include swarm intelligence, machine learning, and data mining and their applications. He serves as the Editor-in-Chief of International Journal of Computational Intelligence and Pattern Recognition (IJCIPR), the Associate Editor of IEEE Transactions on Evolutionary Computation (TEC), IEEE Transactions on Cybernetics (CY-B), IEEE Transactions on Neural Networks and Learning Systems (NNLS), etc. He also served as an Editor of Springers Lecture Notes on Computer Science (LNCS) for 28+ volumes, and Guest Editors of several referred Journals, including IEEE/ACM Transactions on Computational Biology and Bioinformatics, Information Science, Neurocomputing, Natural Computing, Softcomputing, etc. He is a member of Emergent Technologies Technical Committee (ETTC) of IEEE Computational Intelligence Society since 2010. He is the founder and chair of the ICSI International Conference series. He was the general chair of joint general chair of 1st&2nd BRICS Congress of Computational Intelligence, program committee co-chair of IEEE WCCI 2014, etc. He won the 2nd-Class Natural Science Award of China in 2009. His research interests include computational intelligence, swarm intelligence, data mining, intelligent information processing for information security etc. He has published more than 280 papers in refereed journals and conferences in these areas, and authored/co-authored 11 books and 16 chapters in book, and received 4 invention patents.