

Variational Semi-supervised Aspect-term Sentiment Analysis via Transformer

Xingyi Cheng*, Weidi Xu*, Taifeng Wang, Wei Chu,
Weipeng Huang, Kunlong Chen and Junfeng Hu
Ant Financial Services Group
fanyin.cxy@antfin.com

Abstract

Aspect-term sentiment analysis (ATSA) is a long-standing challenge in natural language processing. It requires fine-grained semantical reasoning about a target entity appeared in the text. As manual annotation over the aspects is laborious and time-consuming, the amount of labeled data is limited for supervised learning. This paper proposes a semi-supervised method for the ATSA problem by using the Variational Autoencoder based on Transformer. The model learns the latent distribution via variational inference. By disentangling the latent representation into the aspect-specific sentiment and the lexical context, our method induces the underlying sentiment prediction for the unlabeled data, which then benefits the ATSA classifier. Our method is classifier-agnostic, i.e., the classifier is an independent module and various supervised models can be integrated. Experimental results are obtained on the SemEval 2014 task 4 and show that our method is effective with different five specific classifiers and outperforms these models by a significant margin.

1 Introduction

Aspect based sentiment analysis (ABSA) has two sub-tasks, namely aspect-term sentiment analysis (ATSA) and aspect-category sentiment analysis (ACSA). ACSA is to infer the sentiment polarity with regard to the predefined categories, e.g., the aspect *food, price, ambience*. On the other hand, ATSA aims at classifying the sentiment polarity of a given aspect word or phrase in the text. For example, given a review about a restaurant “the [pizza]_{aspect} is the best if you like thin crusted pizza, however, the [service]_{aspect} is awful.”, the sentiment implications with regard to “pizza” and “service” are contrary. For the aspect

“pizza”, the sentiment polarity is “positive” while “negative” for the aspect “service”. In contrast to document-level sentiment analysis, ATSA requires more fine-grained reasoning about the textual context. The task is worthy of investigation as it can obtain the attitude with regard to a specific entity which we are interested in. The task is widely applied in analyzing the comments, such as opinion generation. Recently, many attempts (Tang et al., 2016b; Pan and Wang, 2018; Liu et al., 2018; Xue and Li, 2018; Li et al., 2018a) focus on supervised learning and pay much attention to the interaction between the aspect and the context. However, the amount of labeled data is quite limited as the annotation about the aspects is laborious. Currently available data sets, e.g. SemEval, only has around 2K unique sentences and 3K sentence-aspect pairs, which is insufficient to fully exploit the power of the deep models. Fortunately, a large amount of unlabeled data is available for free and can be accessed easily from the websites. It will be of great significance if numerous unlabeled samples can be utilized to further facilitate the supervised ATSA classifier. Therefore, the semi-supervised ATSA is a promising research topic.

In ATSA, achieving the sentiment of the aspect-term is semantically complicated and it is non-trivial for a model to capture sentimental similarity of the aspects, which causes the difficulties for semi-supervised learning. In this paper, we proposed a classifier-agnostic framework which named Aspect-term Semi-supervised Variational Autoencoder (Kingma and Welling, 2014) based on Transformer (ASVAET). The variational autoencoder offers the flexibility to customize the model structure. In other words, the proposed framework is compatible with other supervised neural networks to boost their performance. Our proposed model learns the latent representation

*: Equal Contribution

of the input data and disentangles the representations into two independent parts, i.e., the aspect-term sentiment and the representation of the lexical context. By regarding the aspect sentiment polarity of the unlabeled data as the discrete latent variable, the model implicitly induces the sentiment polarity via the variational inference. Specifically, the representation of the lexical context is extracted by the encoder and the aspect-term sentiment polarity is inferred from the specific ATSA classifier. The decoder takes these two representations as inputs and reconstructs the original sentence by two unidirectional language models. In contrast to the conventional auto-regressive models, the latent representations have their specific meanings and are obtained from the encoder and the classifier to the input examples. Therefore, it is also possible to condition the sentence generation on the sentiment and lexical information w.r.t. a certain target entity. In addition, by separating the representation of the input sentence, the classifier becomes an independent module in our framework, which endows the method with the ability to integrate different classifiers. The method is presented in detail in Sec. 3.

Experimental results are obtained on the two classical datasets from SemEval 2014 task 4 (Pontiki et al., 2014). Five recent available models are implemented as the classifier in ASVAET. Our method is able to utilize the unlabeled data and consistently improve the performance against the supervised models. Compared with other semi-supervised methods, i.e., in-domain word embedding pre-training and self-training, the proposed method also demonstrates better performance. We also evaluate the effectiveness of labeled data and sharing embeddings, and show that the structure can provide the separation between lexical context and sentiment polarity in the latent space.

2 Related Work

Sentiment analysis is a traditional research hotspot in the NLP field (Wang and Manning, 2012). Rather than obtaining the sentimental inclination of the entire text, ATSA instead aims to extract the sentimental expression w.r.t. a target entity. With the release of online completions, abundant methods were proposed to explore the limits of current models. Tang et al. (Tang et al., 2016a) proposed to make use of bidirectional Long Short-Term Memory (LSTM) (Hochreiter and Schmid-

huber, 1997) to encode the sentence from the left and right to the aspect-term. This model primarily verifies the effectiveness of deep models for ABSA Tang et al. (Tang et al., 2016b) then put forward a neural reasoning model in analogy to the memory network to perform the reasoning in many steps. There are also many other works dedicating to solve this task (Pan and Wang, 2018; Liu et al., 2018; Zhang and Liu, 2017).

Another related topic is semi-supervised learning for the text classification. Recently, Data augmentation methods (Xie et al., 2019; Berthelot et al., 2019) achieve a great success on low-resource datasets. Moreover, A simple but efficient method is to use pre-trained modules, e.g., initializing the word embedding or bottom layers with pre-training. Word embedding technique has been widely used in NLP models, e.g., Glove (Pennington et al., 2014) and ELMo (Peters et al., 2018). Recently, Bidirectional Encoder Representations from Transformer (BERT) (Devlin et al., 2018) replaces the embedding layer to context-dependent layer with the pre-trained bidirectional language model to capture the contextual representation. BERT is complementary to the encoder of the proposed method. To keep our framework neat, these pre-training investigations are not conducted in this paper.

VAE-based semi-supervised methods, on the other hand, are able to cooperate with various kinds of classifiers. VAE has been applied in many semi-supervised NLP tasks, ranging from text classification (Xu et al., 2017), relation extraction (Marcheggiani and Titov, 2016) to sequence tagging (Chen et al., 2018). Different from text classification where sentiment polarity is related to an entire sentence, ATSA just interested in related information of a given aspect-term. To circumvent this problem, a novel structure is proposed.

3 Method Description

In this section, the problem definition is provided and then the model framework is presented in detail.

The ATSA task aims to classify a data sample with input sentence $\mathbf{x} = \{x_1, \dots, x_n\}$ and corresponding aspect ¹ $\mathbf{a} = \{a_1, \dots, a_m\}$, where \mathbf{a} is a subsequence of \mathbf{x} , into a sentiment polarity y , where $y \in \{P, O, N\}$. P, O, N denotes

¹If an input sentence has n aspect-terms, then n data samples are generated.

“positive”, “neutral”, “negative”. For the semi-supervised ATSA, we consider the following scenario. Given a dataset consisting of labeled samples \mathbf{S}_l and unlabeled samples \mathbf{S}_u , where the $\mathbf{S}_l = \{(\mathbf{x}_l^{(i)}, \mathbf{a}_l^{(i)}, y_l^{(i)})\}_{i=1}^{N_l}$ and $\mathbf{S}_u = \{(\mathbf{x}_u^{(i)}, \mathbf{a}_u^{(i)})\}_{i=1}^{N_u}$, the goal is to utilize \mathbf{S}_u to improve the classification performance over the supervised model using \mathbf{S}_l only.

The architecture is depicted in Fig. 1. The method consists of three main components, i.e., the classifier, the encoder, and the decoder. The classifier can be any differentiable supervised ATSA model, which takes \mathbf{x} and \mathbf{a} as input, and outputs the prediction about y . The encoder transform the data into a latent space that is independent of the label y . And the decoder combines the outputs from the classifier and the encoder to reconstruct the input sentence. For the labeled data, the classifier and the autoencoder are trained with the given label y . For the unlabeled data, the y is regarded as the latent discrete variable and it is induced by maximizing the generative probability. As the classifier can be implemented by various models, the description of the classifier will be omitted. We present a autoencoder structure based on Transformer (Vaswani et al., 2017). In the following, the objective functions are clarified, followed by the model description.

3.1 Variational Inference

Using generative models is a common approach for semi-supervised learning, which tries to extract the information from the unlabeled data by modeling the data distribution. In VAE, the data distribution is modeled by optimizing the evidence lower bound (ELBO) of data log-likelihood, which leads to two objectives for labeled data and unlabeled data respectively. For the labeled data, VAE maximizes the ELBO of $p(\mathbf{x}, y|\mathbf{a})$. For the unlabeled data, it optimizes the ELBO of $p(\mathbf{x}|\mathbf{a})$, where the y is latent and integrated. Specifically, the dependency between variables is illustrated in Fig. 2. The ELBO of $\log p(\mathbf{x}, y|\mathbf{a})$ can be given as follows:

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

where \mathbf{z} is the latent variable which represents lexical information over the sentence and D_{KL} is the

KullbackLeibler divergence.

In terms of the unlabeled data, the ELBO of $\log p(\mathbf{x}|\mathbf{a})$ can be extended from Eq. 1.

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

where \mathcal{H} is the entropy function and $q_\phi(y|\mathbf{x}, \mathbf{a})$ is the classification function.

And $q_\phi(y|\mathbf{x}, \mathbf{a})$ can also be trained in the supervised manner using the labeled data. Combining the above objectives, the overall objective for the entire data set is:

$$\begin{aligned} J &= \sum_{(\mathbf{x}, \mathbf{a}, y) \in \mathbf{S}_l} -\mathcal{L}(\mathbf{x}, \mathbf{a}, y) + \sum_{\mathbf{x} \in \mathbf{S}_u} -\mathcal{U}(\mathbf{x}, \mathbf{a}) \\ &\quad + \gamma \sum_{(\mathbf{x}, \mathbf{a}, y) \in \mathbf{S}_l} -\log q_\phi(y|\mathbf{x}, \mathbf{a}), \end{aligned} \quad (3)$$

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

To implement this objective, three components are required to model the $q_\phi(y|\mathbf{x}, \mathbf{a})$, $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$ and $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ respectively.

3.2 Classifier

Various currently available models can be used as the classifier. For the unlabeled data, the classifier is used to predict the distribution of label y for the decoder, i.e., $y \sim q_\phi(y|\mathbf{x}, \mathbf{a})$. The distribution $q_\phi(y|\mathbf{x}, \mathbf{a})$ will be tuned during maximizing the objective in Eq. 2. In this work, five classifiers are implemented in ASVAET and they are also used as the supervised baselines for the comparison.

3.3 Transformer Encoder

The encoder plays the role of $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$. This module attempts to extract the lexical feature that is independent of the label y when given data sample (\mathbf{x}, \mathbf{a}) . In this way, the \mathbf{z} and \mathbf{y} jointly form the representative vector for the input data.

In our implementation, we use a bidirectional encoder to construct sentences embeddings. It is referred as the Transformer encoder that is actually a sub-graph of the Transformer architecture (Vaswani et al., 2017), the architecture is shown in the left part of the Fig. 2. The encoder employs residual connections around each of the multi-head attention sub-layers, followed by

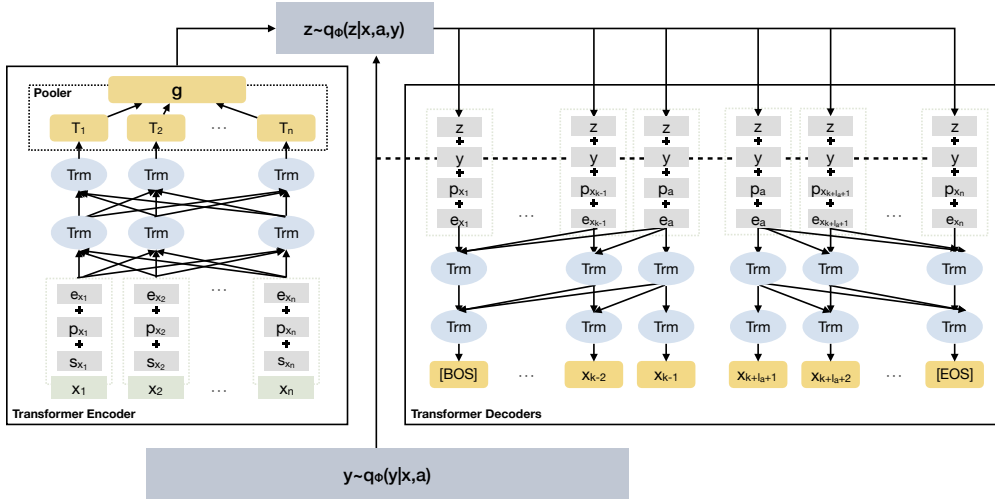


Figure 1: This is the sketch of our model with bidirectional encoder and decoder. Assuming the aspect-term starts at the k -th position in \mathbf{x} . **Bottom:** When using unlabeled data, the distribution of $y \sim q_\phi(y|\mathbf{x}, \mathbf{a})$ is provided by the classifier. **Left:** The sequence is encoded by a Transformer block, which receives the summation of three embeddings, i.e., segment (used to distinguish aspect words) \mathbf{s}_{x_n} , position \mathbf{p}_{x_n} and word \mathbf{e}_{x_n} . The encoding and the label y are used to parameterize the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$. **Right:** A sample \mathbf{z} from the posterior $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$ and label y are passed to the generative network which estimates the probability $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ by two unidirectional Transformer decoders. The number of aspect tokens is l_a .

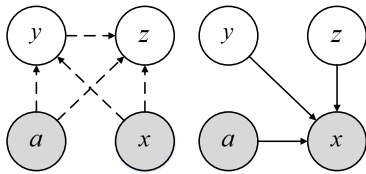


Figure 2: Illustration of ASVAET as a directed graph. **Left:** Dashed lines are used to denote variational approximation $q_\phi(y|\mathbf{x}, \mathbf{a})q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{a}, y)$. **Right:** Solid lines are used to denote generative model $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$.

layer normalization. To capture the aspect-term, we treat the aspect-term and its context differently by segment embeddings. To further emphasize the position of the conditional aspect, the position tag is also included for each token. The position tag indicates the distance from the token to the aspect. And then the position tag is transformed into a vector as defined in (Vaswani et al., 2017), which is added with the word embedding and segment embedding as the input of the Transformer encoder. Let \mathbf{g} denote the output of the Transformer encoder after pooling which simply averaging the hidden states of the aspect-terms (the number of tokens is equal or greater than one) of the last layer, y is the indicator vector of the polar-

ity. Then the distribution of \mathbf{z} can be given as:

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

The sequences are divided into two parts by using segment embedding, the encoder can be aware of the position and the content of the aspect-term \mathbf{a} by multi-head attention operation in the Transformer encoder. The information from two sides are aggregated into the aspect-term \mathbf{a} , and therefore the resulting \mathbf{z} can gather the information related to the aspect.

3.4 Transformer Decoders

The decoder is also a sub-graph of Transformer architecture (Vaswani et al., 2017) which focus on reconstructing original text. The main difference from the Transformer encoder is that the Transformer decoder is unidirectional by modifying the self-attention sub-layer to prevent positions from paying attention to subsequent positions. The textual sequence is well-known to be semantically complex and it is non-trivial for a Transformer decoder to capture the high-level semantics. Here we investigate two questions. How to implement $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ without losing the information of \mathbf{a} and how to capture the semantic polarity by a sequential model. For the first question, denoting

that \mathbf{x} is composed of three parts ($\mathbf{x}_l, \mathbf{a}, \mathbf{x}_r$), we use two Transformer decoders to model the left and right content. For the second question, we let each token is generated conditioned on the summation of the variables \mathbf{z} and embedding \mathbf{y} .

One way to achieve $p_\theta(\mathbf{x}|y, \mathbf{a}, \mathbf{z})$ is to separate the sequence into two parts, reversing the process in the two unidirectional decoder. For each decoder, the input state is represented by the summation of the four input i.e., the polarity indicator vector \mathbf{y} from the classifier or the labeled dataset, the context vector \mathbf{z} from the encoder, the input token embedding \mathbf{e}_{x_t} and the position embedding \mathbf{p}_{x_t} :

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

It is equivalent to generate two sequences using two decoders. When decoding left part (or right part), the aspect will first get processed by the decoder and hence the decoder is aware of the aspect-terms. The position tag is also used in the decoder.

4 Experiments

4.1 Datasets and Preparation

The models are evaluated on two benchmarks: Restaurant (REST) and Laptop (LAPTOP) datasets from the SemEval ATSA challenge (Pontiki et al., 2014). The REST dataset contains the reviews in the restaurant domain, while the LAPTOP dataset contains the reviews of Laptop products. The statistics of these two datasets are listed in Table 1. When processing these two datasets, we follow the same procedures as in another work (Lam et al., 2018). The dataset has a few samples that are labeled as “conflict” and these samples are removed. All tokens in the samples are lowercased without other preprocess, e.g., removing the stop words, symbols or digits.

In terms of the unlabeled data, we obtained samples in the same domain for the REST and LAPTOP datasets. For the REST, the unlabeled

		# 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.

samples are obtained from a sentiment analysis competition in Kaggle². The competition consists of 82K training samples and 34K test samples. For the LAPTOP, the unlabeled samples are obtained from the “Six Categories of Amazon Product Reviews”³, which has 412K samples. The reviews about the laptops are used among six product categories.

The NLTK sentence tokenizer is utilized to extract the sentences from the raw comments. And each sentence is regarded as a sample in our model for both REST and LAPTOP. To obtain the aspects in the unlabeled samples, an open-sourced aspect extractor⁴ is pre-trained using labeled data. The resulting test F1 score is 88.42 for the REST and 80.12 for the LAPTOP. Then the unlabeled data is processed by the pre-trained aspect extractor to obtain the aspects. The sentences that have no aspect are removed. And the sentences are filtered with maximal sentence length 80. The statistic of the resulting sentences is given in Table 2.

4.2 Model Configuration & Classifiers

In the experiments, the model is fixed with a set of universal hyper-parameters. The number of units in the encoder and the decoder is 100 and the latent variable is of size 50 and the number of layers of both Transformer blocks is 2, the number of self-attention heads is 8. The KL weight klw should be carefully tuned to prevent the model from trapping in a local optimum, where \mathbf{z} carries no useful information. In this work, the KL weight is set to be $1e-4$. In term of word embedding, the pre-trained GloVe (Pennington et al., 2014) is used as the in-

²<https://inclass.kaggle.com/c/restaurant-reviews>

³<http://times.cs.uiuc.edu/wang296/Data/>

⁴https://github.com/guillaumegenthial/sequence_tagging

Classifier	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	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 (ASVAET)	78.34 (0.18)	68.41 (0.92)	70.04 (0.53)	64.23 (0.71)
MemNet	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 (ASVAET)	80.58 (0.23)	70.06 (0.53)	73.21 (0.55)	65.88 (0.45)
IAN	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 (ASVAET)	80.23 (0.17)	70.32 (1.00)	74.02 (0.42)	69.39 (0.75)
BILSTM-ATT-G	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)	71.88 (0.19)	74.70 (0.41)	70.31 (0.60)
	BILSTM-ATT-G (ASVAET)	81.11 (0.34)	72.19 (0.27)	75.44 (0.32)	70.52 (0.33)
TNet-AS	TNet-AS	80.56 (0.23)	71.17 (0.43)	76.75 (0.35)	71.88 (0.35)
	TNet-AS (EMB)	80.96 (0.49)	69.99 (0.87)	76.45 (0.40)	71.52 (0.73)
	TNet-AS (ST)	80.76 (0.23)	71.32 (0.56)	76.88 (0.41)	71.74 (0.63)
	TNet-AS (ASVAET)	81.77 (0.20)	72.57 (0.32)	77.57 (0.31)	72.31 (0.69)

Table 3: Experimental results (%). For each classifier, we performed five experiments, i.e., the 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 the mean and the standard deviation of the test accuracy, and the Macro-averaged F1 score. Better results are in bold. ‡ denotes that the results are extracted from the original paper.

put of the encoder and the decoder⁵ and the out-of-vocabulary words are excluded. And it is fixed during the training. The γ is set to be 10 across the experiments.

We implemented and verified four kinds of mainstream ATSA classifiers integrated into our model, i.e., TC-LSTM (Tang et al., 2016a), MemNet (Tang et al., 2016b), BILSTM-ATT-G (Zhang and Liu, 2017), IAN (Ma et al., 2017) and TNet (Li et al., 2018b).

- **TC-LSTM:** Two LSTMs are used to model the left and right context of the target separately, then the concatenation of two representations is used to predict the label.
- **MemNet:** It uses the attention mechanism over the word embedding over multiple rounds to aggregate the information in the sentence, the vector of the final round is used for the prediction.
- **IAN:** IAN adopts two LSTMs to derive the representations of the context and the target phrase interactively and the concatenation is fed to the softmax layer.
- **BILSTM-ATT-G:** It models left and right contexts using two attention-based LSTMs

and makes use of a special gate layer to combine these two representations. The resulting vector is used for the prediction.

- **TNet-AS:** Without using an attention module, TNet adopts a convolutional layer to get salient features from the transformed word representations originated from a bi-directional LSTM layer. Among current supervised models, TNet is currently one of the in-domain state-of-the-art methods and the TNet-AS is one of the two variants of TNet.

The configuration of hyper-parameters and the training settings are the same as in the original papers. Various classifiers are tested here to demonstrate the robustness of our method and show that the performance can be consistently improved for different classifiers.

4.3 Main Results

Table 3 shows the experimental results on the REST and LAPTOP datasets. Two evaluation metrics are used here, i.e., classification accuracy and Macro-averaged F1 score. The latter is more sensitive when the dataset is class-imbalance. In this table, the semi-supervised results are obtained with 10K unlabeled data. We didn't observe further improvement with more unlabeled data. The mean and the standard deviation are reported over

⁵<http://nlp.stanford.edu/data/glove.8B.300d.zip>

5 runs. For each classifier *clf*, we conducted the following experiments:

- *clf*: The classifier is trained using labeled data only.
- *clf* (EMB): We use CBOW (Mikolov et al., 2013) to train the word embedding vectors using both labeled and unlabeled data. And the resulting vectors, instead of pre-trained GloVe vectors, are used to initialize the embedding matrix of the classifier. This is the embedding-level semi-supervised learning as the embedding layer is trained using in-domain data.
- *clf* (ST): The self-training (ST) method is a typical semi-supervised learning method. We performed the self-training method over each classifier. At each epoch, we select the 1K samples with the best confidence and give them pseudo labels using the prediction. Then the classifier is re-trained with the new labeled data. The procedure loops until all the unlabeled samples are labeled.
- *clf* (ASVAET): The proposed method that uses *clf* as the classifier. Note again that the classifier is an independent module in the proposed model, and the same configuration is used as in the supervised learning.

Besides, we also include the results of several supervised models in the first block, i.e., CNN-ASP (Lam et al., 2018), AE-LSTM, ATAE-LSTM (Wang et al., 2016), GCAE (Li and Xue, 2018), from the original paper.

From the Table 3, the ASVAET is able to improve supervised performance consistently for all classifiers. For the MemNet, the test accuracy can be improved by about 2% by the TSSVAE, and so as the Macro-averaged F1. The TNet-AS outperforms the other three models.

Compared with the other two semi-supervised methods, the ASVAET also shows better results. The ASVAET outperforms the compared semi-supervised methods evidently. The adoption of in-domain pre-trained word vectors is beneficial for the performance compared with the Glove vectors.

4.4 Ablation Studies

4.4.1 Effect of Labeled Data

Here we investigated whether the ASVAET works with less labeled data. Without loss of general-

Accuracy	w/o sharing	w/ sharing
TC-LSTM (ASVAET)	78.34	77.65
MemNet (ASVAET)	80.58	78.82
IAN (ASVAET)	80.23	79.22
BILSTM-ATT-G (ASVAET)	81.11	78.36
TNet-AS (ASVAET)	81.77	79.53

Table 4: Comparison between with or without sharing embedding on the REST dataset.

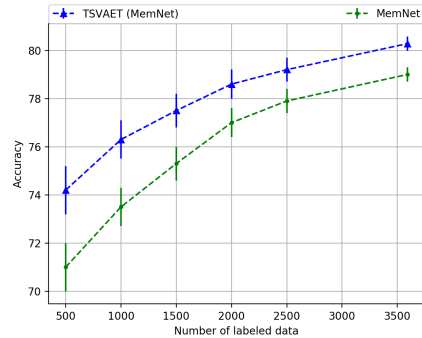


Figure 3: The test accuracy w.r.t. the number of labeled samples on the REST dataset with MemNet classifier.

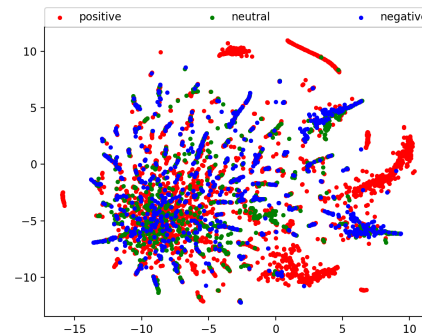


Figure 4: The distribution of the REST dataset in latent space z using t-SNE.

ity, the MemNet is used as the basic classifier. We sampled different amount of labeled data to verify the improvement by using ASVAET. The test accuracy curve w.r.t. the amount of labeled data used is shown in Fig. 3. With fewer labeled samples, the test accuracy decreases, however, the improvement becomes more evident. When using 500 labeled samples, the improvement is about 3.2%. With full 3591 labeled samples, 1.5% gain can be obtained. This illustrates that our method can improve the accuracy with limited data.

4.4.2 Effect of Sharing Embeddings

In previous works, the word embedding is shared among all the components. In other words, the word embedding is also tuned in learning to reconstruct the data. It is questionable whether the improvement is obtained by using VAE or multi-

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

Table 5: Nice sentences that are generated by controlling the sentiment polarity y using the decoder.

task learning (text generation and classification). In the aforementioned experiments, the embedding layer is not shared between the classifier and autoencoder. This implementation guarantees that the improvement does not come from learning to generate. To verify if sharing embedding will benefit, we also conducted experiments with sharing embedding, as illustrated in Table. 4. The results indicate that the joint training for the embedding layer is negative for improving the performance in this task. The gradient from the autoencoder may collide with the gradients from the classifier and therefore, interferes with the optimization direction.

4.5 Analysis of the Latent Space

Transformer encodes the data into two representations, i.e., y and z . These two latent variable represented sentiment polarity and lexical context individually from the input text. We expect the y and z are fully disentangled and represent different meanings. The scatters of latent variable z (cf. Fig. 4) helps us have a better understanding. As shown in the figure, the distributions of three different polarities are very similar, which indicates that the lexical context representation z is independent of the polarity y .

The generation ability of the decoder is also investigated. Several sentences are generated and selected in the Table 5. By controlling the sentiment polarity y with the same z , the decoder can generate sentences with different sentiment in a similar format. This indicates that the decoder is trained successfully to perceive the y and model the relationship between the y and x .

5 Conclusion

A VAE-based framework has been proposed for the ATSA task. In this work, the encoder and decoder are constructed from the Transformers. Both analytical and experimental work has been carried out to show the effectiveness of the ASVAET. The method is verified with various kinds of classifiers. For all tested classifiers, the improvement is obtained when equipped with ASVAET, which demonstrates its universality.

In this paper, the aspect-term is assumed to be known and there is an error accumulation problem when using the pre-trained aspect extractor. According to this, in future work, it is also interesting to show if it is possible to learn the aspect and sentiment polarity jointly for the unlabeled data. It will be of great importance if detailed knowledge can be extracted from the unlabeled data, which will shed light on other related tasks.

References

- David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. 2019. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.
- Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. 2018. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 215–226.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.
- Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *The International Conference on Learning Representations (ICLR)*, Banff, Canada.
- Wai Lam, Xin Li, Bei Shi, and Lidong Bing. 2018. Transformation networks for target-oriented sentiment classification. pages 946–956.
- Lishuang Li, Yang Liu, and AnQiao Zhou. 2018a. Hierarchical attention based position-aware network for aspect-level sentiment analysis. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 181–189.

- Tao Li and Wei Xue. 2018. [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*, pages 2514–2523.
- Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018b. Transformation networks for target-oriented sentiment classification. *arXiv preprint arXiv:1805.01086*.
- Fei Liu, Trevor Cohn, and Timothy Baldwin. 2018. [Recurrent entity networks with delayed memory update for targeted aspect-based sentiment analysis](#). pages 278–283.
- Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. [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*, pages 4068–4074.
- Diego Marcheggiani and Ivan Titov. 2016. [Discrete-state variational autoencoders for joint discovery and factorization of relations](#). *TACL*, 4:231–244.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. [Efficient estimation of word representations in vector space](#). *CoRR*, abs/1301.3781.
- Sinno Jialin Pan and Wenya Wang. 2018. [Recursive neural structural correspondence network for cross-domain aspect and opinion co-extraction](#). pages 2171–2181.
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. [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*, pages 1532–1543.
- Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [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)*, pages 2227–2237.
- Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. [Semeval-2014 task 4: Aspect based sentiment analysis](#). In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014.*, pages 27–35.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. [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*, pages 3298–3307.
- Duyu Tang, Bing Qin, and Ting Liu. 2016b. [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*, pages 214–224.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. [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*, pages 6000–6010.
- Sida Wang and Christopher D Manning. 2012. [Baselines and bigrams: Simple, good sentiment and topic classification](#). In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.
- Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. [Attention-based LSTM for aspect-level sentiment classification](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 606–615.
- Qizhe Xie, Zihang Dai, Eduard Hovy, Minh-Thang Luong, and Quoc V Le. 2019. [Unsupervised data augmentation](#). *arXiv preprint arXiv:1904.12848*.
- Weidi Xu, Haoze Sun, Chao Deng, and Ying Tan. 2017. [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.*, pages 3358–3364.
- Wei Xue and Tao Li. 2018. [Aspect based sentiment analysis with gated convolutional networks](#). *arXiv preprint arXiv:1805.07043*.
- Yue Zhang and Jiangming Liu. 2017. [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*, pages 572–577.