Increasing Capacity of SVM based Approach for Aspect Term Sentiment Analysis

Zeyi Wen[†], Zhishang Zhou[‡], Hanfeng Liu[‡], Bingsheng He[‡], Jian Chen, Xia Li
[†]UWA Australia, [‡]NUS Singapore, South China University of Technology
zeyi.wen@uwa.edu.au

Abstract

Aspect term sentiment analysis (ATSA) is an important task in understanding natural language. In recent years, almost all of the solutions to the ATSA task are based on neural networks. However, in the ATSA task, SVMs have demonstrated reasonable predictive accuracy and are more efficient in training and inference. In this paper, we develop a novel SVM based approach for ATSA by increasing the model capacity. Thus, our solution is able to better fit the ATSA task. Experimental results on two commonly used data sets namely *Laptop* and *Restaurant* show that our solution consistently outperforms the state-of-the-art methods based on neural networks without using BERT. When compared with the neural network methods based on BERT, our solution achieves better or competitive predictive accuracy, but with much smaller model sizes.

1 Introduction

Aspect term sentiment analysis (ATSA) aims to identify the polarity (e.g., positive, negative, neutral) of each aspect (e.g., food and service) rather than the polarity of the whole review. The finer granularity of analysis in ATSA brings new challenges in extracting richer semantic information. Since the ATSA task was introduced [Kiritchenko *et al.*, 2014], two types of methods have been proposed. The first type of methods are based on shallow learning methods (e.g., SVMs), where the ATSA task is modeled as a three-class sentiment classifier with hand-crafted features. These methods have advantage on low computation cost, but are limited by their model capacity to well fit the ATSA task due to the small number of learnable parameters. The second type of methods are based on deep neural networks. This type of methods can learn both features and the sentiment classifier, and the models can be much more sophisticated by adding more layers and neurons. Thus, the deep neural network based methods have achieved promising results. However, the computation cost of the deep neural network based methods is much higher in both training and inference.

To address the problem of small capacity of SVMs and high computation cost of deep neural networks, we accomplish the ATSA task with SVMs of higher model capacity in this paper. We revisit this ATSA task with SVMs, because SVMs had demonstrated good results on the ATSA task [Kiritchenko *et al.*, 2014] and the computation cost is much lower. Hence, it is feasible to introduce more learnable parameters to the SVM based methods. We refer the SVMs with more learnable parameters as *increased capacity SVMs*. The increased capacity SVMs are able to better fit the ATSA task than the existing SVM based approach.

However, there are three key challenges from the increased capacity SVMs. *First*, unlike neural networks where the number of learnable parameters can be easily adjusted (e.g., adding one more layer or a few more neurons), the number of learnable parameters for SVMs cannot be easily increased to better fit the ATSA task, since the parameters in SVMs are fixed. *Second*, the existing hand-crafted features in the SVM based approach can capture global properties of a review, but cannot capture the local properties introduced by our newly introduced learnable parameters. *Third*, a review may have multiple adjectives and multiple aspect terms, and distinguishing which adjective describes which aspect term is crucial for SVM classifiers.

To address the first challenge, we propose two mechanisms to increase the number of learnable parameters. First, we divide the ATSA task into multiple sub-problems, and dedicate a dependent SVM classifier to each sub-problem. Thus, each SVM classifier can be optimized to better fit the sub-problem. Second, we formulate the learning problem to allow the hyper-parameters to be treated as learnable parameters in training. To address the second challenge, we propose techniques to group similar aspect terms together, such that we can extract features specifically for the similar aspect terms. Thus, the extracted features are able to well represent the local properties of the reviews. To address the third challenge, we propose a two-level mechanism to locate the relevant adjectives to an aspect term. For each aspect level, we exploit the dependency parsing result of a review to select adjectives related to a specific aspect term; for aspect cluster level, we obtain generic adjectives related to the aspect term from the aspect clusters.

Our proposed solution can train the sentiment classifier efficiently on multi-core CPUs, and can be much faster using GPUs. In comparison, the deep neural network based methods heavily rely on special hardware such as GPUs and TPUs. To summarize, we make the following contributions in this paper.

- We formulate the aspect term sentiment analysis problem

with SVMs as an ATSA-SVM problem which introduces more learnable parameters into the model training process. The ATSA-SVM problem is flexible in including more learnable parameters to better fit the ATSA task.

- To support more learnable parameters, we allow our solution to extract features specifically for each sub-problem, such that each SVM classifier can be specifically optimized to achieve high predictive accuracy.
- We conduct comprehensive experiments to verify the quality of our proposed solution. The results show that our solution outperforms all the neural network based models without using BERT. When compared with the neural network based models using BERT, our solution achieves better or competitive results.

The remainder of this paper is structured as follows. We first elaborate the details of our proposed solution in Section 2. Then, we show the experimental results in Section 3. After that, we present the related work in Section 4. Finally, we draw a conclusion in Section 5.

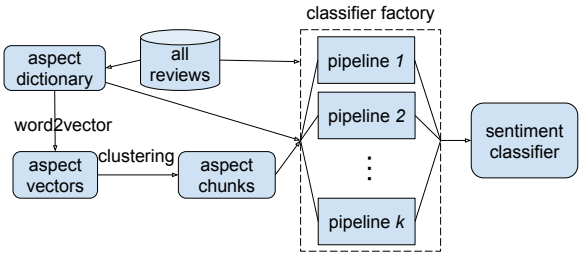
2 Our Proposed Solution

In this section, we present the details of redesigning the SVM based training and prediction process for aspect term sentiment analysis (ATSA). We observe that the previous SVM based approach for aspect term sentiment analysis is limited by the model capacity (i.e., much fewer number of learnable parameters than deep learning models) [Kiritchenko *et al.*, 2014]. The key idea of our solution is to introduce more learnable parameters in the training process. First, we allow the training process to learn the best hyper-parameters (i.e., kernel type and the kernel hyper-parameters), rather than manually setting them. Second, we train more SVM classifiers in the process to include more learnable parameters, instead of using only one SVM classifier as the previous SVM based approach does. Next, we elaborate the whole training process in greater details.

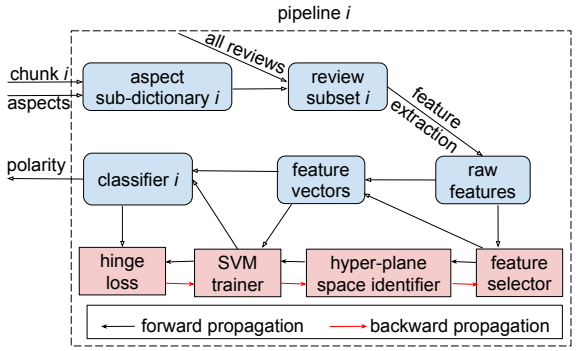
2.1 Overview of Our Solution

The overview of our proposed solution is shown in Figure 1a. First, we construct a dictionary for all the aspect terms appearing in the reviews. Then, the aspect term dictionary is transformed to a dictionary of aspect vectors using word embedding. After that, we perform clustering to group the aspect vectors into non-overlapping chucks. For an aspect chuck, we train an SVM classifier on the corresponding subset of reviews. Finally, all of those SVM classifiers together form the final sentiment classifier for the aspect term sentiment analysis problem.

The key steps of training each SVM classifier are shown in Figure 1b. Initially, we perform feature extraction on the reviews to obtain raw features (e.g., the sentiment score of each keyword) for each SVM. Then, a feature selector is dedicated to selecting aspect chunk specific features from the raw features to form a feature vector for each review. Meanwhile, we have a hyper-plane space identifier for setting proper hyper-parameters for SVMs (i.e., identifying a space for the separating hyper-plane). After that, the feature vectors and



(a) The overall architecture of our solution



(b) A pipeline of classifier training

Figure 1: Overview of our proposed solution

the hyper-parameters together are fed to the SVM trainer to learn an SVM classifier. Finally, the loss of the classifier is computed and backpropogated to improve the feature selector, hyper-plane space identifier and SVM trainer. Therefore, the SVM classifier trained in this process is well-tuned, as the features, hyper-parameters and the parameters of the SVM classifier are thoroughly learned. Next, we formulate the aspect term sentiment analysis problem with SVMs and elaborate more details of our solution.

2.2 The ATSA-SVM Problem

We formulate the aspect term sentiment analysis problem with SVMs as the "ATSA-SVM" problem. Let $\mathcal{T}$ and $\mathcal{V}$ denote the training data set and the validation data set, respectively. The training set and validation set are further clustered into k chunks denoted by $\mathcal{T}^1, \ldots, \mathcal{T}^k$ and $\mathcal{V}^1, \ldots, \mathcal{V}^k$, respectively. The learnable hyper-parameters include: $\mathcal{F}$ which is a set of features; $\mathcal{H}$ which is the candidate SVM kernel types; Λ which contains the kernel hyper-parameters and the regularization constant of SVMs; and k which is the number of chunks. Thus, the learnable hyper-parameters space can be defined as $\Theta = \mathcal{F} \times \mathcal{H} \times \Lambda$. Then, the ABSA-SVM problem is to minimize the following objective function:

$$\underset{k \in \mathbb{N}^+, \boldsymbol{\theta}^i \in \Theta}{\arg\min} \sum_{i=1}^{k} \frac{|\mathcal{V}^i|}{|\mathcal{V}|} \mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)$$

where $\mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)$ denotes the loss of the i-th SVM on $\mathcal{V}^i$, $\boldsymbol{\theta}^i = \{f^i, h^i, \lambda^i\}$, f^i denotes the features used in the i-th SVM, h^i is the used SVM kernel type, λ^i denotes the corresponding kernel hyper-parameters, $\frac{|\mathcal{V}^i|}{|\mathcal{V}|}$ is the weight of the i-th SVM and $\mathbb{N}^+$ is the set of positive natural numbers.

Optimization on the ATSA-SVM problem: The idea of the gradient based approach can be used to solve the ATSA problem. More specifically, the derivative of the ATSA-SVM problem over $\boldsymbol{\theta}^i$ is shown below.

$$\sum_{i=1}^{k} \frac{|\mathcal{V}^i|}{|\mathcal{V}|} \cdot \frac{\partial \mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)}{\partial \boldsymbol{\theta}^i}$$

Since $\boldsymbol{\theta}^i = \{f^i, h^i, \lambda^i\}$, the derivative can be written as

$$\sum_{i=1}^{k} \frac{|\mathcal{V}^i|}{|\mathcal{V}|} \cdot \left(\frac{\partial \mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)}{\partial f^i \cdot h^i \cdot \lambda^i} + \frac{\partial \mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)}{\partial h^i \cdot f^i \cdot \lambda^i} + \frac{\partial \mathcal{L}(\boldsymbol{\theta}^i, \mathcal{T}^i, \mathcal{V}^i)}{\partial \lambda^i \cdot f^i \cdot h^i}\right).$$

As $\Theta = \mathcal{F} \times \mathcal{H} \times \Lambda$ has discrete and conditional variables (e.g., the degree d in Λ is discrete and is used only for the polynomial kernel), there is no closed form for computing the gradients. In our solution, we use the sequential model-based optimization (SMBO) with Tree Parzen Estimator and Expected Improvement [Bergstra *et al.*, 2011] to solve the ATSA-SVM problem. This method can solve optimization problems where the search space is noncontinuous and conditional variables exist. In the following, we explain the process of clustering data into k chunks, extracting features for each chunk and training an SVM classifier for each chunk.

2.3 Clustering Aspect Terms into Chunks

Here, we describe the first component of our solution: clustering aspect terms into chunks. As we have discussed earlier in this section, using a single SVM classifier to deal with the ATSA task leads to poor predictive accuracy, due to the limited model capacity of one SVM classifier. We divide the reviews of the ATSA task into chunks, where each chunk contains the aspect terms sharing similar semantics. Hence, our solution is able to use more SVMs to handle the ATSA task, and each SVM classifier is specifically trained for a chunk. The key intuition of constructing an aspect chunk is that an aspect may be described using different aspect terms. For instance, aspect terms including "manager", "staff" and "chef" may be used to rate the "personnel" aspect of a restaurant. Moreover, similar aspect terms tend to be described by similar adjectives. For example, adjectives including "delicious" and "yummy" may be used to describe the food aspect, while "expensive" and "pricy" may be used to describe the price aspect. When performing sentiment analysis for the price aspect, "delicious" and "yummy" are noise. By clustering aspect terms into chunks, our solution is able to learn knowledge of similar aspect terms and exclude the noise from the irrelevant adjectives.

Formally, let $\{a_1, a_2, a_3, \ldots, a_n\}$ denote the aspect terms in the training data set $\mathcal{T}$, where a_j corresponds to the aspect term of the j-th training instance. We use k-means to cluster the training instances based on the word embedding of the aspect terms. After the clustering, the training data set $\mathcal{T}$ is divided into k chunks denoted by $\mathcal{T}^1, \mathcal{T}^2, \mathcal{T}^3, \ldots, \mathcal{T}^k$. As the chunks are independent, our solution can train a customized SVM classifier for each chunk to achieve high predictive accuracy. When a test instance is fed in, we easily assign it to a chunk with the centers of the chunks.

Data Balancing

A key challenge raises from clustering the training data into chunks is that the data of a chunk is unbalanced (e.g., more positive reviews than neutral reviews) which may downgrade the generalization ability of the SVM classifiers. To make each chunk balanced, we employ an upsampling technique. First, we use the largest class as a reference (e.g., positive class). Then, we aim to increase the number of training instances for the other two classes (e.g., neutral and negative classes) until the two classes have the same number of training instances as the referenced class. For increasing the number of training instances of a class (e.g., negative class), we randomly select a chuck (except the current chunk) and then randomly select a training instance of the class (e.g., negative class). The sampling process is repeated until the three classes have the same number of training instances. In our solution, whether to use sampling is a learnable parameter for each chunk.

2.4 Extracting Features

After dividing the training data into subsets, the next important step for our solution is to extract features from the reviews. In our solution, we extract features including (i) surface features, (ii) parse features, (iii) word similarity features, (iv) sentiment lexicon features and (v) aspect term independent features.

(i) Surface feature: A training instance $\boldsymbol{x}$ has an aspect term a and l sentences denoted by $\boldsymbol{s}_1, ..., \boldsymbol{s}_l$. The training instance $\boldsymbol{x}$ can be expressed as $\boldsymbol{x} = (a, \boldsymbol{s}_1, ..., \boldsymbol{s}_l)$. The surface features are the unigrams extracted from the sentences. We use the bag-of-word model to represent the surface features, and each dimension is a value computed by term frequency-inverse document frequency (TF-IDF).

(ii) Parse feature: An aspect term may be separated from its modifying words which contain sentiment polarity. The surface features are insufficient for capturing this property, because the surface features are purely based on locality. To address the limitation of the surface features, we use the parse tree rooted on the aspect term to obtain unigrams and use the bag-of-word model to represent the parse features.

(iii) Word similarity features: When dealing with the unseen instances where out-of-vocabulary often occurs, the surface and parse features are not enough. To further strengthen the quality of the features, we use word embedding to improve the generalization ability of our proposed solution. We cluster the words in the reviews corresponding to an aspect chunk into W word *packs*. As a result, the words in a word pack have the similar meaning. Then we generate a W-dimension vector for each review, where each dimension of the vector corresponds to the total word counts in a word pack. Intuitively, each dimension of the W-dimension vector stores the number of words belonging to the same word pack.

(iv) Sentiment lexicon feature: We also extract the sentiment lexicon features using sentiment lexicon dictionaries, where each word is associated with a sentiment score. Given a word w from the review, a corresponding sentiment score can be found in the sentiment lexicon dictionaries [Kiritchenko *et al.*, 2014]. The sentiment scores of positive words are positive, while the scores of negative words are negative. The closer the score is to 0, the less sentiment the word carries. We extract seven sentiment score related features including (1) the number of positive words, (2) the number of negative words, (3) the total score of positive words, (4) the total score of negative words, (5) the maximal score of positive words,

(6) the minimal score of negative words, and (7) the total sentiment score.

(v) Aspect term independent features: The above features are dependent on the aspect terms within an aspect chunk. To further enhance the quality of the features, we extract aspect term independent features (i.e., generic features) by considering information from the other aspect chunks. Our key idea is that we extract features which are the least related to the current aspect chunk. The relativity score of word w in the i-th aspect chunk is computed by the following formula based on chi-squared.

$$chi(w) = \frac{N(N_w^i N_{\bar{w}}^{\bar{i}} - N_{\bar{w}}^i N_w^{\bar{i}})^2}{(N_w^i + N_w^{\bar{i}})(N_w^i + N_{\bar{w}}^i)(N_{\bar{w}}^i + N_w^{\bar{i}})(N_{\bar{w}}^i + N_{\bar{w}}^{\bar{i}})}$$

where N denotes the total number of training instances in all the aspect chunks; N_w^i denotes the number of training instances that contain the word w in the i-th aspect chunk; $N_{\bar{w}}^i$ denotes the number of training instances that do not contain the word w in the i-th aspect chunk; $N_w^{\bar{i}}$ denotes the number of training instances that contain the word w but not in the i-th aspect chunk; $N_{\bar{w}}^{\bar{i}}$ is the number of training instances that neither contain the word w nor belong to the i-th aspect chunk. We select b words with the smallest relativity scores (e.g., b equals to 10% of the vocabulary size).

2.5 Training the Model

One important property in our solution is that the feature selector, hyper-plane space identifier and SVM trainer are all learnable components. They can be improved based on the loss obtained from the current SVM classifier. Thus, the SVM classifier trained in our solution is well-tuned for features, hyper-parameters and the parameters in the SVM classifier. Here we elaborate the details of training the model including learning the hyper-parameters and training the SVMs.

Learning to Select Features

We have used five groups of features presented in Section 2.4. Those features are extracted for each aspect chunk. The extracted features may work well for one SVM classifier but poorly for another SVM classifier. It is important that different aspect chunks use different groups of features. Hence, we need to find out the best feature combination among the five feature groups for each SVM classifier. In this paper, our solution enumerates the feature combination from the five feature groups, and chooses the best feature combination for each aspect chunk. For example, the SVM classifier for the first aspect chunk may use surface features and word similarity features only, while that for the second aspect chunk may use all the five groups of features.

Learning to Set the Hyper-Plane Space

The hyper-parameters of SVMs have significant influence on the SVM model quality. The hyper-parameters define the hyper-plane space of the SVMs. In our solution, the hyper-parameters of SVMs are learned rather than manually set. We use the the sequential model-based optimization (SMBO) to help select the kernel type and the kernel hyper-parameters for each SVM classifier. The key idea is that we use the history of the hyper-parameters to train a machine learning model which guides the search for the best kernel and its corresponding hyper-parameters. Moreover, our solution also learns to decide whether to use sampling to balanced the training instances for the SVMs. After each training instance is represented as a feature vector and the hyper-parameters of the SVMs are set, we train an SVM classifier for each aspect chunk using ThunderSVM [Wen *et al.*, 2018].

2.6 Sentiment Prediction

Suppose we have a test instance $\boldsymbol{x}_t = \{a_t, \boldsymbol{s}_1, ..., \boldsymbol{s}_l\}$ where a_t is the aspect term and $\boldsymbol{s}_l$ is a sentence in the review. Our solution first assigns $\boldsymbol{x}_t$ to the corresponding SVM classifier whose center of the aspect chunk is the most similarity to a_t. Then the features are extracted and selected using the techniques presented in Section 2.4. Finally, a sentiment label (e.g., positive) is predicted by the SVM classifier.

2.7 Time Complexity Analysis for Training

Deep learning models generally consume huge computation time compared with SVMs. Here, we provide the time complexity analysis for our proposed solution, in comparison with HAPN [Li *et al.*, 2018a] which is based on deep neural networks and achieves high predictive accuracy on the ATSA task. We denote α as the average sentence length, n as the number of training instances, d as the number of dimensions of the training instances, and t as the training rounds (e.g., the number of epochs). For the deep learning based model (i.e., HAPN), the most time consuming operations are matrix multiplications on Bi-GRU and hierarchical attention. Hence, the time complexity for HAPN is $\mathcal{O}(t \cdot n \cdot \alpha \cdot d^3)$, where the matrix multiplication takes $\mathcal{O}(d^3)$ for each training instance. In comparison, the time complexity of SVMs is $\mathcal{O}(t \cdot n \cdot d)$ for the SVM training using the Sequential Minimal Optimization algorithm [Keerthi *et al.*, 2001]. As we can see, the SVM based solution has a much lower time complexity than the deep neural network based solution. We acknowledge that the number of rounds t and the dimension of training instances in SVMs and neural networks may be different. However, this time complexity analysis provides insights of the training cost of SVMs and neural networks.

3 Experimental Study

In this section, we study the predictive accuracy and efficiency of our proposed solution.

3.1 Experimental Setup

Our solution was implemented in Python and the source code to reproduce our experiments is available in this GitHub URL (omitted due to double blind peer review policy).

Data sets: We conducted our experiments using two popular aspect term sentiment analysis data sets from SemEval 2014 Task 4: *Restaurant* and *Laptop*. The detailed information of the two data sets are listed in Table 1, where "#positive", "#negative", "#neutral" and "#total" denote the number of positive reviews, negative reviews, neutral reviews and the total number of reviews, respectively.

Dictionaries for sentiment lexicon: In our experiments, we used eight sentiment lexicon dictionaries [Kiritchenko *et*

data set		#positive	#negative	#neutral	#total
Laptop	Train	987	866	460	2313
	Test	341	128	169	638
Restaurant	Train	2164	807	637	3608
	Test	728	196	196	1120

Table 1: Details of the *Laptop* and *Restaurant* data sets

al., 2014]: (1) *tweet sentiment lexicons*, (2) *hashtag sentiment lexicons* and (3) *sentiment140*, (4) *NRC emotion lexicons*, (5) *Bing Liu's lexicons*, (6) *MPQA subjectivity lexicons*, (7) *Yelp restaurant word–aspect association lexicons* and (8) *Amazon laptop word-aspect association lexicons*. Each dictionary contributes to seven features as described in Section 2.4, and hence we have 56 features from the sentiment lexicon dictionaries in total.

Hyper-parameter settings: The number of chunks was searched from 1 to 35. The regularization constant, C, of the SVMs was selected from 1 to 2^{20}. The SVM kernel functions considered include Radial Basis Function (RBF), polynomial, Sigmoid and linear kernels. The degree for the polynomial kernel was searched from 1 to 5. The γ term of the RBF kernel was selected from 10^{-3} to 10 divided by the size of the training data set. Which features used in the SVM classifier were automatically learned from the data, and the features were selected from the five groups of features discussed in Section 2.4.

Machine information: The experiments were conducted on a workstation running Linux with a Xeon E5-2640v4 12 core CPU and 64GB main memory.

Baselines: We identify the latest solutions to the aspect term sentiment analysis problems. The solutions include the latest neural network based ones without pre-trained models and the latest ones using BERT. The neural network based solutions without pre-trained models include: (1) **HAPN** [Li *et al.*, 2018a] which introduces position-aware encoding and fuses the information of aspects and relevant contexts for sentence representation; (2) **IMN** [He *et al.*, 2019] which is an interactive multi-task learning network to jointly learn multiple related tasks through a shared set of latent variables; (3) **BILSTM-ATT-G** [Cheng *et al.*, 2018] which uses the variational autoencoder based on transformer and exploits the underlying sentiment prediction for the unlabeled data; (4) **RAM** [Chen *et al.*, 2017] which adopts a Bi-LSTM to weight the words according to their relative positions to the target, and non-linearly combines a series of the attention results; (5) **LSTM+SynATT+TarRep** [He *et al.*, 2018] which constructs an aspect embedding matrix to captures the semantic meaning and uses an attention model to incorporate dependency parsing information; (6) **PF-CNN** [Huang and Carley, 2018] which introduces a parameterized filters to incorporate aspect information into convolutional neural networks.

The four latest pre-trained neural model based solutions include: (1) **BERT-SPC** [Song *et al.*, 2019] which models the sentiment analysis problem into the sentence pair classification task of BERT; (2) **SDGCN-BERT** [Zhao *et al.*, 2019] which extracts the aspect specific representations from BERT and then, exploits sentiment dependencies with a graph convolutional network; (3) **AEN-BERT** [Song *et al.*, 2019] which employs multi-head attention based encoders for the modeling between context and target with word embedding from BERT. (4) **BERT-PT** [Xu *et al.*, 2019] which fine-tune BERT in the proposed Review Reading Comprehension (RRC) task with domain-specific corpora. We notice that it is possible to further improve our solution by using extra training data, as demonstrated in the existing work [Rietzler *et al.*, 2019; Yang *et al.*, 2019]. In our experiments, we focus on training the models without using extra training data.

Models	Restaurant		Laptop	
	Acc	Macro-F1	Acc	Macro-F1
BERT-SPC [Song *et al.*, 2019]	84.46	**76.98**	78.99	75.03
SDGCN-BERT [Zhao *et al.*, 2019]	83.57	76.47	**81.35**	**78.34**
AEN-BERT [Song *et al.*, 2019]	83.12	73.76	79.93	76.31
BERT-PT [Xu *et al.*, 2019]	**84.95**	76.96	78.07	75.08
HAPN [Li *et al.*, 2018a]	82.23	-	77.27	-
IMN [He *et al.*, 2019]	83.89	75.66	75.36	72.02
BILSTM-ATT-G [Cheng *et al.*, 2018]	81.11	72.19	75.44	70.52
RAM [Chen *et al.*, 2017]	80.23	70.80	74.49	71.35
LSTM+SynATT+TarRep [He *et al.*, 2018]	80.63	71.32	71.94	69.23
PF-CNN [Huang and Carley, 2018]	79.20	-	70.06	-
existing svm based approach [Kiritchenko *et al.*, 2014]	82.23	73.75	72.27	65.60
ours (single svm)	78.57	63.78	72.26	67.61
ours (multiple svms)	**86.79**	**78.81**	**80.25**	**77.07**

Table 2: Accuracy and Macro-F1 comparison

3.2 Accuracy and Macro-F_1 Comparison

Here, we report the experimental results on predictive accuracy which is shown in Table 2. As we can see, our proposed solution achieves an accuracy of nearly 87% on *Restaurant* and 80% on *Laptop*. Our solution outperforms all the neural network based solutions without using the pre-trained model (i.e., BERT). When compared with the BERT based models, our solution outperforms all of them on the *Restaurant* data set. The predictive accuracy of our solution outperforms most of the BERT based models on *Laptop* and is competitive to the best one. We also compared the Macro-F_1 scores among the different methods [Song *et al.*, 2019]. Similar to accuracy results, our solution outperforms all the neural network based methods without using the pre-trained model.

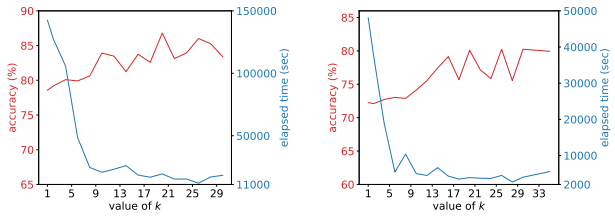
We also looked into the hyper-parameters of each SVM classifier. We have found that about 80% of the SVM classifiers use the sampling technique to form a balanced training data set (cf. Section 2.3). The parameter b, which is used to select the number of words with the smallest relativity scores (cf. Section 2.4), was set to 10% for most of the SVM classifiers.

3.3 Effect on varying the number of aspect chunks

Figure 2 shows the effect of varying the number of chunks (i.e., k) on accuracy and the elapsed time. As we can see from the figure, the elapsed time for training decreases as the number of chunks increases. This is because the training cost of each SVM classifier is lower when the chunk size is smaller. The accuracy tends to increase as the number of chunks increases. However, the accuracy hits the highest score when the number of chunks is 20 for *Restaurant* and 30 for *Laptop*, respectively.

3.4 Inference efficiency comparison

Figure 3 shows the batch inference efficiency of BERT and our solution on the two data sets. We used BERT-SPC [Song *et al.*, 2019] as an example to study the inference efficiency, as



(a) *Restaurant* (b) *Laptop*

Figure 2: Effect of # of chunks on accuracy and elapsed time

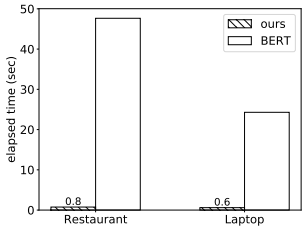


Figure 3: Inference efficiency

BERT-SPC achieves high score in *Restaurant* and *Laptop*. The batch size used in the experiments was 16 which leads to the best efficiency for BERT on both of the data sets according to our experiments. As we can see from the figure, the efficiency of our solution is over 40 times better than BERT. This is an important property of our solution which produces better or competitive predictive accuracy to BERT based solutions, but our solution is an order of magnitude faster in inference.

3.5 The Trained Model Size

We also calculated the number of parameters in the final trained models. We have found that our solution leads to a much smaller number of parameters than the other models. Specifically, our solution only requires about 1 million parameters, while HAPN [Li *et al.*, 2018a] requires over 2 million parameters and BERT-SPC [Song *et al.*, 2019] even requires about 110 million parameters.

4 Related Work

4.1 Aspect Term Sentiment Analysis (ATSA)

Aspect term sentiment analysis (ATSA) is a fine-grained sentiment classification task [Pontiki *et al.*, 2014]. The key to solving this task highly depends on the extraction of semantic relatedness between aspect terms and their corresponding context. Traditional machine learning methods, including rule based methods and machine learning based methods [Kiritchenko *et al.*, 2014], are based on a set of linguistically inspired lexical, semantic and sentiment lexicon features. Recently, deep neural network based methods achieve the state-of-the-art results in the ATSA task. The neural networks tend to capture richer semantic relatedness between an aspect term and its context.

4.2 Deep Learning for the ATSA Task

Deep neural networks (DNNs) have achieved good results in aspect term sentiment analysis. DNNs (e.g., MemNet [Tang *et al.*, 2016]) contain millions of parameters which are effective for modeling complex language dependencies, and thus DNNs outperform the previous statistic-based methods such as NRC-Canada [Kiritchenko *et al.*, 2014] and DCU [Wagner *et al.*, 2014]. A series of studies based on deep neural networks have been dedicated to solving the ATSA task. TD-LSTM [Tang *et al.*, 2015] adopts a forward and a backward LSTM to model the left and the right contexts of the aspect, and associate target with contextual features separately. TNet [Li *et al.*, 2018b] stacks a proposed CPT module to fuse the embedding of aspect term into the embedding of whole context words and output a dense vector representing the information of an aspect term and its context. ATAE-LSTM [Wang *et al.*, 2016] concatenates aspect embeddings with context word representations and adopts an attention mechanism which lets aspect participate in. TNet-ATT [Tang *et al.*, 2019] proposes an incremental approach to automatically extract attention supervision information for neural aspect term sentiment classification models. SDGCN-BERT [Zhao *et al.*, 2019] employs Graph Convolutional Networks [Kipf and Welling, 2016] over the attention mechanism to capture the sentiment dependencies and achieves particularly well predictive accuracy.

4.3 Using Pre-trained Models for the ATSA Task

More recently, the large pre-trained language models (e.g., BERT [Devlin *et al.*, 2018]) are used to tackle the ATSA task. There are two ways to leverage the benefits of large pre-trained language models. The first way is to fine-tune BERT for the ATSA task. The related work includes BERT-PT [Xu *et al.*, 2019] and BERT-ADA [Rietzler *et al.*, 2019]. The second way is to extract the word embedding from BERT. The examples include AEN-BERT [Song *et al.*, 2019] and SDGCN-BERT [Zhao *et al.*, 2019]. Despite their good accuracy, those BERT based models contain a large number of parameters (i.e., around 110 million parameters) and are slow in inference.

5 Conclusion and Future Work

In this paper, we have revisited tackling the ATSA task with increased capacity SVMs. Our proposed solution has more learnable parameters than the existing SVM based method, and hence it is able to better fit the ATSA task. Our experimental results on the *Laptop* and *Restaurant* data sets have showed that our solution consistently outperforms the state-of-the-art methods based on neural networks without using the pre-trained model (i.e., BERT). When compared with the neural network methods using BERT, our solution achieves better or competitive predictive accuracy, but with much smaller model sizes. This research finding may bring more attention from the research community to rethink about the conventional machine learning algorithms to tackle existing problems such as ATSA. One important property of our solution is that we only need CPUs to train the increased capacity SVMs for the ATSA task, which is different from the deep neural network based methods that require special hardware such as GPUs and TPUs.

Other machine learning algorithms such as decision trees and logistic regression can be used to replace the SVMs in our solution, which may achieve even higher predictive accuracy. We plan to explore in this direction in the future.

References

[Bergstra *et al.*, 2011] James S Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, pages 2546–2554, 2011.

[Chen *et al.*, 2017] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 452–461, 2017.

[Cheng *et al.*, 2018] Xingyi Cheng, Weidi Xu, Taifeng Wang, and Wei Chu. Variational semi-supervised aspect-term sentiment analysis via transformer. *arXiv preprint arXiv:1810.10437*, 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[He *et al.*, 2018] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. Effective attention modeling for aspect-level sentiment classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1121–1131, 2018.

[He *et al.*, 2019] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. *arXiv preprint arXiv:1906.06906*, 2019.

[Huang and Carley, 2018] Binxuan Huang and Kathleen M Carley. Parameterized convolutional neural networks for aspect level sentiment classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1091–1096, 2018.

[Keerthi *et al.*, 2001] S. Sathiya Keerthi, Shirish Krishnaj Shevade, Chiranjib Bhattacharyya, and Karuturi Radha Krishna Murthy. Improvements to platt's smo algorithm for svm classifier design. *Neural Computation*, 13(3):637–649, 2001.

[Kipf and Welling, 2016] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Kiritchenko *et al.*, 2014] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 437–442, 2014.

[Li *et al.*, 2018a] Lishuang Li, Yang Liu, and AnQiao Zhou. 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, 2018.

[Li *et al.*, 2018b] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. Transformation networks for target-oriented sentiment classification. *arXiv preprint arXiv:1805.01086*, 2018.

[Pontiki *et al.*, 2014] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27—35, 2014.

[Rietzler *et al.*, 2019] Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification. *arXiv preprint arXiv:1908.11860*, 2019.

[Song *et al.*, 2019] Youwei Song, Jiahai Wang, Tao Jiang, Zhiyue Liu, and Yanghui Rao. Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*, 2019.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. Effective LSTMs for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*, 2015.

[Tang *et al.*, 2016] Duyu Tang, Bing Qin, and Ting Liu. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*, 2016.

[Tang *et al.*, 2019] Jialong Tang, Ziyao Lu, Jinsong Su, Yubin Ge, Linfeng Song, Le Sun, and Jiebo Luo. Progressive self-supervised attention learning for aspect-level sentiment analysis. *arXiv preprint arXiv:1906.01213*, 2019.

[Wagner *et al.*, 2014] Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. DCU: Aspect-based polarity classification for semeval task 4. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 223–229, 2014.

[Wang *et al.*, 2016] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based LSTM for aspect-level sentiment classification. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 606–615, 2016.

[Wen *et al.*, 2018] Zeyi Wen, Jiashuai Shi, Qinbin Li, Bingsheng He, and Jian Chen. Thundersvm: A fast svm library on gpus and cpus. *The Journal of Machine Learning Research*, 19(1):797–801, 2018.

[Xu *et al.*, 2019] Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. BERT post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

[Yang *et al.*, 2019] Heng Yang, Biqing Zeng, JianHao Yang, Youwei Song, and Ruyang Xu. A multi-task learning model for chinese-oriented aspect polarity classification and aspect term extraction. *arXiv preprint arXiv:1912.07976*, 2019.

[Zhao *et al.*, 2019] Pinlong Zhao, Linlin Hou, and Ou Wu. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *arXiv preprint arXiv:1906.04501*, 2019.