11747 Assignment 4 Report
Aspect Extraction with Adversarial Training

Fuyu Tang fuyut

Yifan Song yifanson

Jiameng Du jiamengd

Abstract

Aspect-Based Sentiment Analysis and its sub-task Aspect Extraction play an important role in real commerce and industry. One of the obstacle for Aspect Extraction is data collection due to the expensiveness of this process. We propose to leverage Adversarial Training on existing Aspect Extraction models to serve as a regularization method and improve robustness. In this report, we first go over the history and important related works of Aspect Extraction, as well as Adversarial Training. After that, we selected two State-of-the Art models: DE-CNN and BERT-PT, and apply adversarial training process on them. From the experiments, we show that our proposed models have outperformed the original models (without adversarial training) for both models on both benchmark datasets. At last, we discuss our progress on the remaining "boundary" error and analyze the negative results for current attempts.

1 Introduction

Knowing how people feel about the products is extremely important for industries. Thus, sentiment analysis on product reviews and customer opinions is increasingly viewed as crucial commercial success. Most approaches of sentiment analysis attempt to detect the overall polarity of a text, paragraph, or sentence. However, it is also valuable to learn the sentiment polarity on specific aspects (screen, battery, food quality, service) for target entities (laptop, restaurant). This task is called Aspect Based Sentiment Analysis (ABSA), whose goal is to identify aspects and the sentiment for each aspect (Pontiki et al., 2014, 2016). For instance, in the sentence "*The laptop has an incredible speed.*", the aspect is "*speed*" and a positive sentiment is mentioned towards it. The ABSA task can then be divided into two sub-tasks - Aspect Extraction (AE) and Aspect Sentiment Classification (ASC).

In AE task, the objective is to extract all aspects of the target entity, such as "*I liked the **food**, but the **service** was bad.*" The aspect can also be a multi-word term (but should be treated as a single aspect) such as "*The **hard disk** is too noisy.*" The AE task can be formalized as a sequence labeling task, where each word will be assigned one of the three labels - beginning word of aspect terms (*B*), in the aspect terms (*I*), or out of the aspect terms (*O*). For example, "*The **retina display** is great.*" can be labeled as "*O B I O O O*" where "*retina display*" is the aspect we would like to extract.

In ASC task, the objective is to determine the sentiment polarity of each aspect term which can be either *positive*, *negative*, or *neutral*. For example:

- "I loved their **fajitas**. (**fajitas**: *positive*)"
- "I hated their **fajitas**. (**fajitas**: *negative*)"
- "The **fajitas** are their first plate. (**fajitas**: *neutral*)".

There are many great works on this vital Natural Language Processing (NLP) task or one of its sub-tasks, especially AE which is a key challenge in ABSA. To narrow down the topic, we'll only focus on the AE task in this report.

One main obstacle for Aspect Extraction is data collection. Unlike some other NLP tasks, such as Machine Translation, where a large amount of natural data exists, collecting data, i.e. labeling aspects in product reviews/sentences, is very laborious and time-consuming. Due to this reason, most Aspect Extraction works are still experimented on a few benchmark datasets, each containing only

around 3000 data points. To resolve this issue, instead of collecting more data, one approach is to use adversarial training, i.e. generate adversarial examples that are similar to real data and train the model with both original samples and adversarial samples. This process can not only help the model to generalize better as a regularization method but also increase the robustness of the model. We then apply the adversarial training process on two selected State of the Art (SOTA) Aspect Extraction models and show that the new models with adversarial training outperform the original ones on two benchmark datasets.

2 Related Work

2.1 Aspect Extraction

Aspect Based Sentiment Analysis was first proposed by Hu and Liu (2004). As one of key subtask of ABSA, Aspect Extraction has been studied for more than a decade. The AE task has been performed by both unsupervised and supervised approaches. One main unsupervised approach is frequent item mining and syntactic rule-based extraction (Zhuang et al., 2006; Qiu et al., 2011). These models depend on pre-defined rules so they only work well when aspects are restricted to a small group of nouns. The other unsupervised approach is to use topic modeling and Latent Dirichlet Allocation (LDA) based models (Titov and McDonald, 2008; Zhao et al., 2010). In these models, corpus is a mixture of topics (aspects), and topics are distributions over words. However, the corpus might be well described by the mixture of aspects but it's easy to have poor-quality individual aspects.

The traditional supervised approach typically uses Hidden Markov Models (HMM) and Conditional Random Fields (CRF) (Jakob and Gurevych, 2010). On top of this basis, part-of-speech and named entity features are incoporated in Chernyshevich (2014) while syntactic features and word embeddings are used in Toh and Su (2016). They won the aspect extraction task in 2014 and 2016 SemEval Challenge respectively.

In recent years, deep learning has become one of the most emerging techniques. Deep neural networks have also been applied to Aspect Extraction. Liu et al. (2015) may be the first work to use vanilla Long Short-Term Memory (LSTM), while Convolutional Neural Networks (CNN) may be the first to propose use on AE in Poria et al. (2016). Later, He et al. (2017) has incorporated the attention mechanism to improve the coherence of aspects by exploiting the distribution of word co-occurrences through neural word embeddings. A joint model has been proposed in Wang et al. (2017) to use multy-layer attention mechanism to jointly extract aspect and opnion terms. Li et al. (2018) has further strengthened the joint model using truncated history-attention and selective transformation network. In Li et al. (2019), the authors have designed a network to transfer aspect knowledge learned from a coarse-grained network. More recently, Graph Convolutional Networks (GCN) has been brought into this field (Zhao et al., 2020).

BERT is one of the key innovations in recent NLP research (Devlin et al., 2019). With the power of BERT, BERT-based models have demonstrated very competitive performance in Aspect Extraction. For instance: Sun et al. (2019) constructs a sequence of auxiliary sentences using the sequence of aspects then fine-tunes BERT with both sequences; Rietzler et al. (2020) first uses self-supervised fine-tuning on domain specific data, then follows the second task-specific fine-tuning stage. More BERT-based works are coming out, and it will not be surprising that they dominate the SOTA list of Aspect Extraction.

2.2 Adversarial Training

The term "Adversarial Examples" was firt proposed in Szegedy et al. (2014). Adversarial Examples are "fake" data points created by making small perturbations on original inputs that can fool a machine learning (neural net) model and significantly drop its performance. Inspired from it, Goodfellow et al. (2015) proposed the method called "Adversarial Training" that generates Adversarial Examples during training which can potentially enhance the performance of original model by acting as a regularization method while improving the robustness to small and approximately worst case perturbations.

Adversarial Training was first proposed and shown success in image classification. It would be much more difficult to apply it on text-related tasks because of the discrete nature of text input. However, Miyato et al. (2017) later succeed in applying Adversarial Training on text classification by bringing this technique into embedding

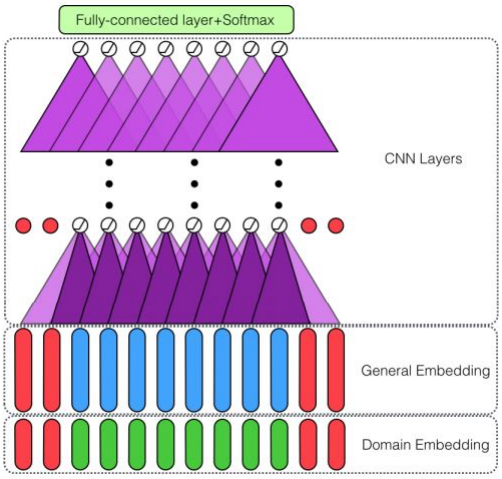

Figure 1: Overview of DE-CNN, figure from original paper

space that generates perturbations and adversarial examples on word embeddings. We will follow this approach in our work for Aspect Extraction models.

3 Method

After reviewing the history and relevant works of AE, we then pick two typical recent AE models that achieve competitive performance as our baseline models, one non-BERT based model and one BERT based model. More impressively, these two highly-cited works are from the same authors.

3.1 DE-CNN

Before BERT came out, Dual Embeddings CNN (DE-CNN), which integrates GloVe and domain-specific embeddings, was the SOTA model of Aspect Extraction (Xu et al., 2018). This work has shown that employing both (pre-trained) general-purpose embeddings and embeddings trained from the domain-specific data would produce a competitive performance even with a relatively simple model to avoid over sophisticated models like many other previous works that cause problems in real deployment. The performance of this model is still very close to the current SOTA and higher than the performance of vanilla BERT and multiple BERT-based models, while requiring much less time and computation resource for training.

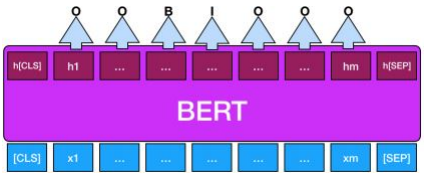

Figure 2: Overview of BERT for AE, figure from original paper

Algorithm 1: Post-training Algorithm

Input: $\mathcal{D}_{\text{DK}}$: one batch of DK data;
$\mathcal{D}_{\text{MRC}}$ one batch of MRC data;
u: number of sub-batches.

$\nabla_\Theta \mathcal{L} \leftarrow 0$
$\{\mathcal{D}_{\text{DK},1}, \ldots, \mathcal{D}_{\text{DK},u}\} \leftarrow \text{Split}(\mathcal{D}_{\text{DK}}, u)$
$\{\mathcal{D}_{\text{MRC},1}, \ldots, \mathcal{D}_{\text{MRC},u}\} \leftarrow \text{Split}(\mathcal{D}_{\text{MRC}}, u)$
for $i \in \{1, \ldots, u\}$ **do**
  $\mathcal{L}_{\text{partial}} \leftarrow \frac{\mathcal{L}_{\text{DK}}(\mathcal{D}_{\text{DK},i}) + \mathcal{L}_{\text{MRC}}(\mathcal{D}_{\text{MRC},i})}{u}$
  $\nabla_\Theta \mathcal{L} \leftarrow \nabla_\Theta \mathcal{L} + \text{BackProp}(\mathcal{L}_{\text{partial}})$
end
$\Theta \leftarrow \text{ParameterUpdates}(\nabla_\Theta \mathcal{L})$

Figure 3: Post-training algorithm, *DK* represents domain knowledge data and *MRC* represents task knowledge data, figure from original paper

3.2 BERT-PT

The second model is called BERT Post-Training (BERT-PT), which is currently one of the SOTA models for Aspect Extraction, and yet another simple but brilliant approach (Xu et al., 2019). In BERT-PT, the authors propose a joint post-training technique that takes BERT's pre-trained weights as the initialization for basic language understanding and post-train BERT with both domain knowledge and task knowledge data before fine-tuning on the end-tasks.

3.3 Adversarial Process

The architecture of our adversarial training process is shown in Figure 4.

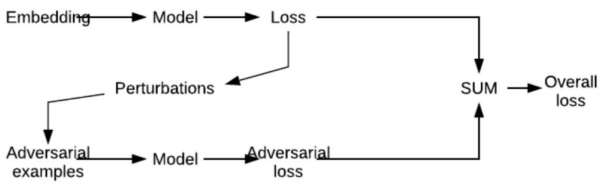

Figure 4: Adversarial Training Architecture

"**Embedding**" is the input sentence embedding, where DE-CNN integrates both GloVe as general-purpose embedding and pre-trained domain-specific word embedding while BERT-PT

leverages BERT embedding summing from token, segment and position embedding; "**Model**" represents the AE model which is either DE-CNN or BERT-PT while "**Loss**" is calculated by cross-entropy loss function. After getting the original loss, "**Perturbations**" are then calculated based on the gradient of the original loss with the following equation:

$$r_{adv} = \underset{r,\|r\|\leq\epsilon}{\operatorname{argmin}} \log p(y|x+r;\hat{\theta}) \qquad (1)$$

where r is the perturbation and $\hat{\theta}$ is a constant copy of model parameters θ so that the actual parameters are not updated during back propagation. Then the worst perturbations (r_{adv}) are chosen to construct "**Adversarial examples**" that can best improve the original model by adding the perturbation on original embedding, $x + r_{adv}$. This can be done by solving the equation 1. However, the exact solution of this minimization problem might not be solved directly since it is intractable in many scenarios. Thus, Goodfellow et al. (2015) proposed an approximation method that solve this equation by linearizing $\log p(y|x;\theta)$ with a $L2$ norm constraint, the final perturbations then follow the equation below:

$$r_{adv} = \frac{-\epsilon * g}{\|g\|_2} \qquad (2)$$

where g is the gradient of original loss: $\nabla_x \log p(y|x;\hat{\theta})$. Here, ϵ is a scaling hyper-parameter of perturbations that we'll tune during experiments. We then pass the generated "**Adversarial examples**" to the same "**Model**" (either DE-CNN or BERT-PT) and get the "**Adversarial Loss**" with same loss function (cross entropy): $-\log p(y|x+r_{adv};\theta)$. The last step is to sum up the original loss and adversarial loss to get the overall model loss and back propagate with this overall loss.

4 Experiments

4.1 Datasets

Most AE works are performed on two benchmark datasets which consist of review sentences with aspect terms labeled: the *laptop* dataset from subtask 1 of SemEval-2014 Task 4 (Pontiki et al., 2014) and the *restaurant* dataset from subtask 1 of SemEval-2016 Task 5 (Pontiki et al., 2016). Our models have also been experimented on these two datasets and the performance is measured by *F1 score*. The statistics of these two datasets are shown in Table 1.

	Laptop	***Restaurant***
Training	3045 S./2358 A.	2000 S./1743 A.
Testing	800 S./654 A.	676 S./622 A.

Table 1: Summary statistics of datasets, S: number of sentences; A: number of aspects

4.2 Results

The experiment results can be seen in Table 2. We have tuned the hyper-parameters and incorporated adversarial training in the two models, which improved F1 scores in general. In a3, we have run the DE-CNN model and BERT-PT model on the two datasets by fine-tuning with the AE task. In our experiments, both of their adversarial training versions have outperformed the original model significantly.

We have tried several sets of parameters on the laptop and restaurant dataset. In our experiment, the pattern of rising validation loss resulting from an increment of the number of epochs occurs after three epochs. However, as the number of epochs increases, the f1 score of restaurant test data improves, which is not the case in laptop test data. This trend occurs because, for laptop dataset, the distribution of its validation data and test data are more alike, whereas, for restaurant dataset, this resemblance lies in its train data and test data. The best epsilon is 0.2 for the laptop task and 0.5 for the restaurant task in BERT-PT. With a larger epsilon, the BERT-PT model attacked with the worst-case adversarial examples should get better performance. But with a large epsilon value, the adversarial examples may have a huge distinction from the input embeddings, which may be harmful to the model and lead to worse performance.

Comparing the increase in performance of DE-

Model	***Laptop** (F1)*	***Restaurant** (F1)*
BERT(vanilla)	77.02	72.96
DE-CNN	81.35	74.45
DE-CNN (adv)	82.08	76.23
BERT-PT	82.98	77.50
BERT-PT (adv)	83.98	79.89

Table 2: F1 results for our reproduced models and models with adversarial training

Task	Dropout	Epsilon
DE-CNN(laptop)	0.5	0.01
DE-CNN(rest)	0.5	0.01
BERT-PT(laptop)	0.0	0.2
BERT-PT(rest)	0.0	0.5

Table 3: Parameters for best tuned models with adversarial training

CNN and BERT-PT after implementing adversarial training, we noticed that BERT-PT has considerably improved its performance (+2.39 for restaurant and +1.00 for laptop) compared to DE-CNN (+1.78 for restaurant and +0.73 for laptop). This could be attributed to the fact that the two models use different embeddings. BERT-PT model contains both domain knowledge and test knowledge through pre-training and post-training; BERT embedding summing from token, segment, and position embedding is more robust to the attack adversarial examples. DE-CNN model leverages both general and domain embeddings, which are word embeddings that contain less information and are more susceptive to attacks.

5 Discussion on Boundary Error

One remaining error that both models are suffering from is what we called "boundary" error. The "boundary" error can be classified into two types:

- the extracted aspect is not complete or only a part of it, for instance, in the sentence "*I tried several monitors with HDMI and this was the case each time.*", the model predicts only "*HDMI*" (or "*monitors*") as the aspect but the true aspect should be "*monitors with HDMI*""
- the extracted aspect includes leading or tailing terms, typically adjectives, for example, in the sentence "*Buy the separate RAM memory and you will have a rock*", the model predict "*separate RAM memory*" as the aspect but the true aspect should be "*RAM memory*"

We first proposed to train a separate "reposition" network that can correct the boundary by modifying the starting and ending position of the aspects with both positive and negative examples directly generated from original models. But this would cause cascading errors and involve overfitting issues if we train only with samples from training sets (otherwise, to use both train and test samples, it will violate the assumption of unseen data).

We then tried to use a margin-based method by updating the original cross entropy loss function to a multi-label margin loss.

$$loss(x, y) = \sum_i \max(0, 1 - (x[y] - x[i])) \quad (3)$$

where y is true label and $x[i]$ is the score of label i and $i \neq y$.

The intuition here is to penalize heavier on misclassification to find a larger and more solid margin between labels, comparing to the objective of better learning on true labels with higher accuracy for original cross entropy loss, so that the boundary error would be minimized. We have also experimented with different weights including both up-weighting the correct classifications and down-weighting the wrong ones. But after tuning the model with different parameter settings, the average test f1 score decreased by around 0.2 for laptop and 0.1 for restaurant, while the best ones also failed to outperform the original models.

As discussed above, we failed to tackle this problem in this work and we decided to leave it for future work due to time limitation. One approach worth to try is to reformat the task as span prediction. The difficulty here is how to incorporate the span length and start/end position information into the model as the this information is ideally unknown for testing set. Although the boundary error seems to only cause a minor difference between extracted aspects and "gold" labels while doesn't appear very frequently, it still harms performance significantly and causes errors in practical use. Thus, we believe that solving this error would further improve the model performance in AE task.

References

Maryna Chernyshevich. 2014. Crossdomain extraction of product features using crf. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014*.

Jacob Devlin, Ming-Wei Chang, and Kenton Lee. 2019. Bert: Pre-training of deep bidirectional

transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Ian J Goodfellow, Jonathon Shlens, , and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Proceedings of International Conference on Learning Representations 2015*.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Mingqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Xin Li, Lidong Bing, Piji Li, Wai Lam, , and Zhimou Yang. 2018. Aspect term extraction with history attention and selective transformation. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*.

Zheng Li, Ying Wei, Yu Zhang, Xiang Zhang, Xin Li, and Qiang Yang. 2019. Exploiting coarse-to-fine task transfer for aspect-level sentiment classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*.

Takeru Miyato, Andrew M Dai, and Ian Goodfellow. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of International Conference on Learning Representations 2017*.

Maria Pontiki, Dimitris Galanis, and Haris Papageorgiou. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th international workshop on semantic evaluation (SemEval-2014)*.

Maria Pontiki, Dimitris Galanis, and Haris Papageorgiou. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*.

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

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. In *Computational Linguistics*.

Alexander Rietzler, Sebastian Stabinger, Paul Opitz, and Stefan Engl. 2020. Adapt or get left behind: Domain adaptation through bert language model finetuning for aspecttarget sentiment classification. In *Proceedings of The 12th Language Resources and Evaluation Conference*.

Chi Sun, Luyao Huang, and Xipeng Qiu. 2019. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Proceedings of International Conference on Learning Representations 2014*.

Ivan Titov and Ryan McDonald. 2008. Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International World Wide Web Conference*.

Zhiqiang Toh and Jian Su. 2016. Nlangp at semeval2016 task 5: Improving aspect based sentiment analysis using neural network features. In *Proceedings of the 10th International Workshop*

on Semantic Evaluation, SemEval@NAACL-HLT 2016.

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

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. Bert posttraining for review reading comprehension and aspect-based sentiment analysis. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*.

Pinlong Zhao, Linlin Hou, and Ou Wu. 2020. Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. In *Knowledge-Based Systems*.

Wayne Xin Zhao, Jing Jiang, Hongfei Yan, and Xiaoming Li. 2010. Jointly modeling aspects and opinions with a maxent-lda hybrid. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*.

Li Zhuang, Feng Jing, and Xiao-Yan Zhu. 2006. Movie review mining and summarization. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*.