CP3106 INDEPENDENT PROJECT

A Robust Framework for Targeted Sentiment Analysis

By

Xiang Pan

Department of Computer Science

School of Computing

National University of Singapore

2020/04

CP3106 INDEPENDENT PROJECT

A Robust Framework for Targeted Sentiment Analysis

By

Xiang Pan

Department of Computer Science

School of Computing

National University of Singapore

2020/04

Project No: CP3106
Advisor: Prof. Lee Wee Sun
Deliverables:
Report: 1 Volume

Abstract

In the report, we distinguish the differences between the general sentiment analysis and Targeted Sentiment Analysis. Further more, we analyzed the existing problems of Targeted Sentiment Analysis. To address those problems, we proposed a new BERT-based framework to solve the targeted sentiment analysis problem. Based on the framework, we introduce some auxiliary training methods to improve the accuracy of the results. To illustrate the existing methods' robustness problem toward new (unseen) targets, we introduce a new data set setting, which explicitly makes the targets in the training set and test set to be different. Then, we use the adversarial training methods to enhance the robustness of our framework training. Overall, our framework reach the performance of the state of the art in the traditional targeted sentiment analysis setting and showed robustness in the new re-split data set setting. Finally, we describe future work in targeted sentiment analysis.

Keywords:
Targeted Sentiment Analysis, Robustness, BERT, Adversarial Training, Auxiliary Training

Implementation Software and Hardware:
Python, Pytorch, RTX 2080TI

Acknowledgement

I would like to thank my friends, families, members of the laboratory and advisors. Without them, I would not have been able to complete this project. In addition, thanks to the NGNE program for giving me the opportunity to do research.

List of Figures

List of Tables

Table of Contents

Chapter 1

Introduction

1.1 Background

In this section, we briefly discuss the history and background of the targeted sentiment analysis problem. A detail literature survey is presented in Chapter 2.

Sentiment analysis (also known as opinion mining) refers to methods such as identifying and extracting subjective information in the original material. Generally speaking, the purpose of sentiment analysis is to find out the attitude of the speaker or the author on certain topics or against the bipolar views of a text. This attitude may be his or her personal judgment or evaluation, or his emotional state at the time (the author 's emotional state when making this speech), or the author 's intentional emotional communication (that is, the author wants the reader to experience Emotions)

1.2 The Problem

In different papers, the problem has different names. The target sentence analysis is a task with several subtasks, such as target term extraction, target term sentiment classification. In this work, we focus on the target term sentiment classification.

(Pei, Sun, & Li, 2019) proposed a detailed classification for the target sentiment analysis: We denote the f_{cls} as the classification model.

Table 1.1: Categorization of the data

Task	Dataset	Coherence	Source	Collection	Target Structure	Example application domain
TG-ABSA	SemEval 2014	Strong	Online Review	Crawling	Aspect (Entity)	Product, service, movie, Apps
TN-ABSA	Twitter	Weak	Twitter	Filtering	Entity	Event, people, organization
T-ABSA	Sentihood, Baby Care	Moderate	Forum	Crawling	(Entity, Aspect)	product, service

Target-grounded Aspect-based Sentiment Analysis (TG-ABSA):

$$f_{cls}(sentence, aspect\ terms\ related\ to\ a\ aspectcategory) = sentiment \tag{1.1}$$

Targeted Non-aspect-based Sentiment Analysis(TN-ABSA)

$$f_{cls}(sentence, target) = sentiment \tag{1.2}$$

For a general TSA task, or more precisely, target entity sentiment classification(we use TSA to illustrate in the subsequent paper), we adopt the 1.2 definition as our task setting and treat the aspect entity as the target term, which is also a general setting.

1.3 Our Contributions

Our Contributions is mainly on three aspects:

1. A general framework for target sentiment analysis
2. Auxiliary training methods for TSA

3. Proposing a new robustness test setting

4. Adversarial training methods for model's robustness towards new(never appeared in training) targets.

Chapter 2

Related Work

2.1 Text classification

The text classification is a traditional application of NLP. As a typical sequence data format, various sequence models have been applied to the text classification problem.

2.2 Pre-trained Language models

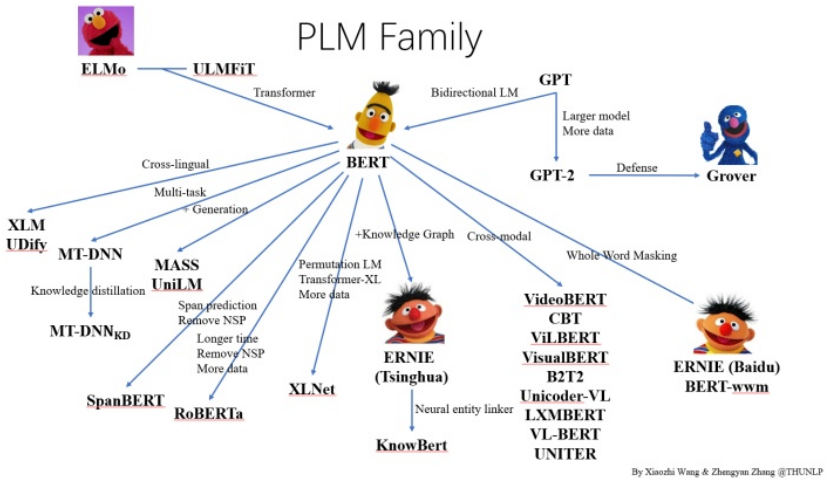

Figure 2.1: Pre-trained Language Models Family

In order to perform the downstream tasks of natural language processing, it is usually necessary to represent the original text to some features. From the early statistical language

model to the model based on neural networks in recent years, it is to achieve this goal. Starting from word2vec, how to learn a reasonable word vector representation has become the key to different downstream tasks. The pre-training model is trained through reasonable structure design and large-scale corpus training, so as to obtain a more general word vector representation. Using these word vectors, many NLP downstream tasks can be performed. However, due to the different corpus domains and language domain differences, the task corpus needs to be used to fine-tune the model. At the same time, due to different downstream tasks, the use of word vectors is strictly related to downstream models.

BERT (Devlin, Chang, Lee, & Toutanova, 2019) as a typical and successful pre-trained model for various NLP tasks can be seen as a baseline. How to use the pre-trained language model to adapt to various downstream tasks and the efficient fine-tuning is focused by researchers.

2.3 Aspect Based Sentiment Analysis

We briefly review the related work in ABSA.

Using the Memory Network architecture (Tang, Qin, & Liu, 2016b), which uses a memory network to remember contextual words and explicitly model attention to target words and context. It was found that compared with the previous model (Tang, Qin, Feng, & Liu, 2016a) that used the left or right context alone, making full use of context words can improve its model.

The Attention Encoder Network (AEN) (Song, Wang, Jiang, Liu, & Rao, 2019a) was tested using GloVe word vectors as input word vector representations and BERT as word vectors representation word vector representations (AEN-BERT) (a modification of the transformer architecture). The author divides the Multi-Headed Attention (MHA) layer into the MHA inner layer and MHA outer layer in order to model the target words and context differently. Such a structure is lighter than the transformer architecture.

Graph Convolutional Neural Network (GCN) (Zhaoa, Houb, & Wua, 2019) achieves another recent performance improvement, using graph convolutional neural network to explicitly establish the dependence between sentiment words in sentences with multiple aspects. They show that if there are multiple aspects in a sentence, the performance of their model's architecture

will be particularly good.

However, the designs of architecture in these works utilize various characteristics of targeted sentiment analysis. It is hard to unify to a generic format to combine those designs for further improvement. For a more generic framework and better utilize the powerful pre-trained language models, we proposed our framework.

Chapter 3

Problem and Algorithm

3.1 Formal Description of Problem

Given a text sequence with n words text=$\{w_1, w_2, w_3, w_4, ..., w_n\}$ and a target with m words, target text= $\{t_1, t_2, ..., t_m\}$ with its begin position b, the problem is to classify the sentiment polarity $polarity = \{positive, neutral, negative\}$ towards the given target in the context. We followed the SemEval 2014 Task 4(Pontiki, Galanis, Pavlopoulos, Papageorgiou, Androutsopoulos, & Manandhar, 2014) subtask 2.

3.2 Design of Algorithm

3.3 BERT for Text Classification

BERT for sentence text classification As a powerful pre-trained universal language model, BERT can be used in various downstream tasks. To utilize bert, we have several direct ways:

1. Using BERT embeddings as the input of a sequence

2. Fine-tuned BERT by [CLS] classification token.

To enhance the performance, (Sun, Qiu, Xu, & Huang, 2019) proposed several methods for text classification:

Trick for text classification

1. Various fine tuning methods

   (a) Within-task pre-training

   (b) In-domain pre-training

   (c) Cross-domain pre-training

2. Different learning rates are used for different layers of Bert

Those methods only consider the sentence-level classification. For TSA, the classification problem is more fine-grained, hence we will introduce our BERT-based framework for TSA.

3.4 BERT for TSA

As described in the related work, BERT-SPC (Song, Wang, Jiang, Liu, & Rao, 2019b) repeats the target words at the end of the context sentence.

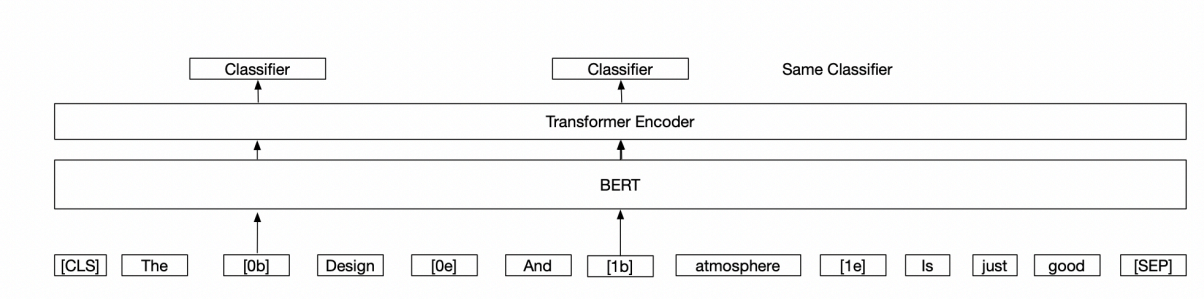

Figure 3.1: Our framework for TSA

As shown in the image 4.1, we add the beginning token and end token around the target. Such an idea was inspired by the (Soares, FitzGerald, Ling, & Kwiatkowski, 2019), which add special tokens for relationship extraction.

To test the token add methods, in our initial experiments, we add the same token for different targets in the context sentence. However, the result is worse than only taken for the target you need to do the classification. The reason may be that BERT model can not distinguish which token should aggregate the classification information.

3.5 Auxiliary training methods for TSA

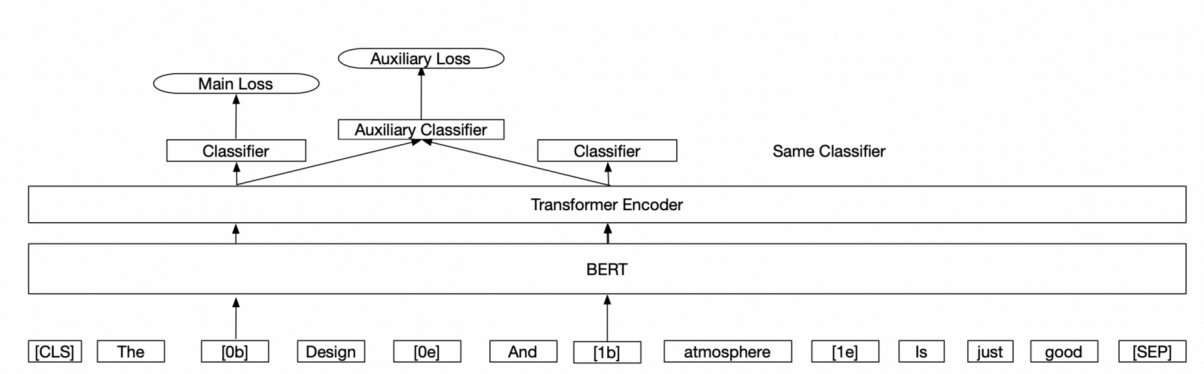

Figure 3.2: auxiliary training

To enhance bert by utilizing the targets' position information and the relationship between different targets in the same sentence, we use auxiliary training methods to better fine-tune BERT.

we denote the auxiliary classification function by f_{aux}, the classification can be presented as:

$$f_{aux}(main\ target, other\ target) = sentiment\ pair \tag{3.1}$$

For a sentence with more than 2 targets, we iterate all the other targets in the sentence. Thus, the auxiliary loss can be denoted as:

$$loss_{aux}(main\ target) = \sum_{other\ target_i \in other\ targets} f_{aux}(main\ target, other\ target_i) \tag{3.2}$$

3.6 Adversarial training methods for Robustness of TSA

(Karimi, Rossi, Prati, & Full, 2020) use adversarial training methods from (Miyato, Dai, & Goodfellow, 2016) to enhance BERT-PT(Xu, Liu, Shu, & Yu, 2019) model's performance. We utilize a similar method in our framework.

From the 3.3 we can see that our model still has some dependence on the bottom layer of BERT. Such dependence may contribute to a higher score in reappearing target sentiment anal-

Table 3.1: Statistics of re-split dataset.

stat-type	train	test	
twitter	size	6248	619
	target-number	104	358
restaurant	size	3608	393
	target-number	606	304
laptop	size	2328	282
	target-number	461	232

ysis. However, when a target is never seen before, the dependence may decrease the robustness. We would like to make the framework to be less reliant on the target tokens.

3.6.1 Robustness Test Settings

For simplicity, we remove those samples, which own the same or similar targets in the train set, from the test set.

3.6.2 Adversarial training

The adversarial training method searches the worst perturbations which can make the largest classification error. Towards the main optimization function, the adversarial training target is to maximize the main loss function. For maximizing the error rate, the following perturbations are added to the input embeddings to create new adversarial sentences in the embedding space.

$$r_{adv} = -\epsilon \frac{g}{||g||_2} \tag{3.3}$$

$$g = \nabla_x \log p(y|x; \hat{\theta}) \tag{3.4}$$

and $p_{adv} = \epsilon$ is the size of the perturbations.

$$loss_{aux} = -\log p(y|x + r_{adv}; \theta)$$

For robustness test, the total training loss is:

$$loss(main\ target) = loss_{main}(begin\_token) + p_{aux} * loss_{aux}(main\ target) + loss_{adv} \quad (3.5)$$

For the hyper parameters p_{aux} and p_{adv} is adjusted by the experiment results.

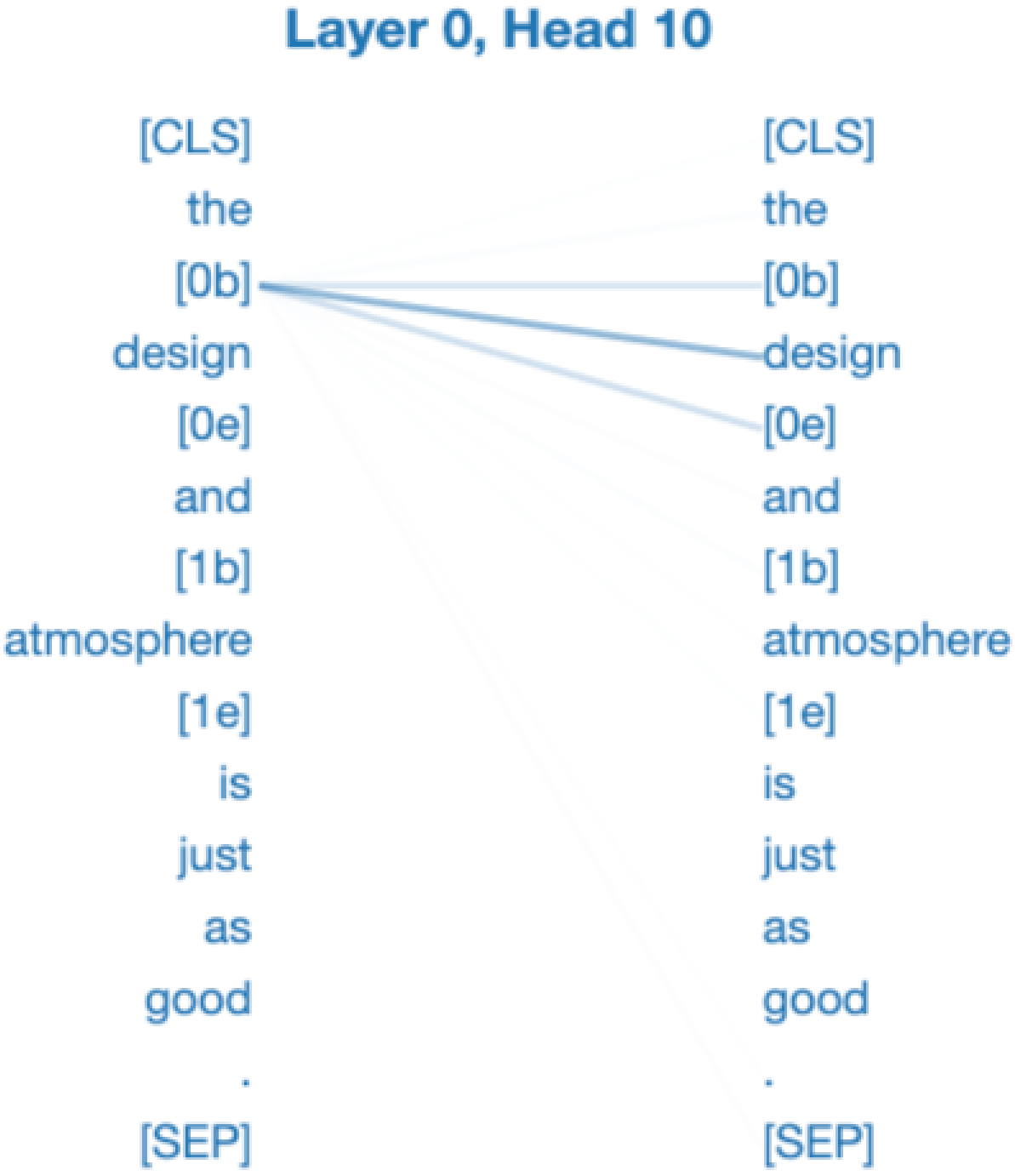

Figure 3.3: Bottom layer of BERT visualization

Chapter 4

Evaluation

In this section, we describe the experiment environment, evaluation details and results of our experiments.

4.1 Implementation Details

We list our experiment environment:

OS: Ubuntu 18.04 LTS (Bionic Beaver)

Kernel: x86_64 Linux 4.15.0-70-generic2

Shell: zsh 5.4.2

CPU: Intel Xeon W-2123 @ 8x 3.9GHz

GPU: GeForce RTX 2080 Ti

RAM: 31859MiB

Our experiment is based on the code of ABSA-Pytorch. We release our code in TSA-Pytorch.

Our bert model is based on the huggingface's transformers library. For some bottom-modified, we direct modified the source code of the library, details refer to our source code.

4.2 Experimental Setup

batch-size: 32

optimizer: adam

valset-ratio: 0.05

max-seq-len: 128

num-epoch: 10(The best performance model usually trained with 2 or 3 epochs for general bert, bert-multi-target(our framework) usually takes up to 10 and adversarial training usually takes up to 10)

4.3 Visualization

4.3.1 Pre-trained BERT Visualization

From the visualization, we can find that the bert_spc model's attention is densely connected to the [SEP] token and [CLS] token. A similar structure can be found in our framework, the beginning token gives additional positional information. Thus we can enhance the model performance by using aggregate classification token embeddings.

4.3.2 BERT-SPC Visualization

We attached the full visualization of whole the bert-spc model's attention in the appendix. The bert's different attention is less modified on the fine-tuning stage. When we add an additional transformer encoder layer above the bert model, the model training is getting much easier and a little performance increases.

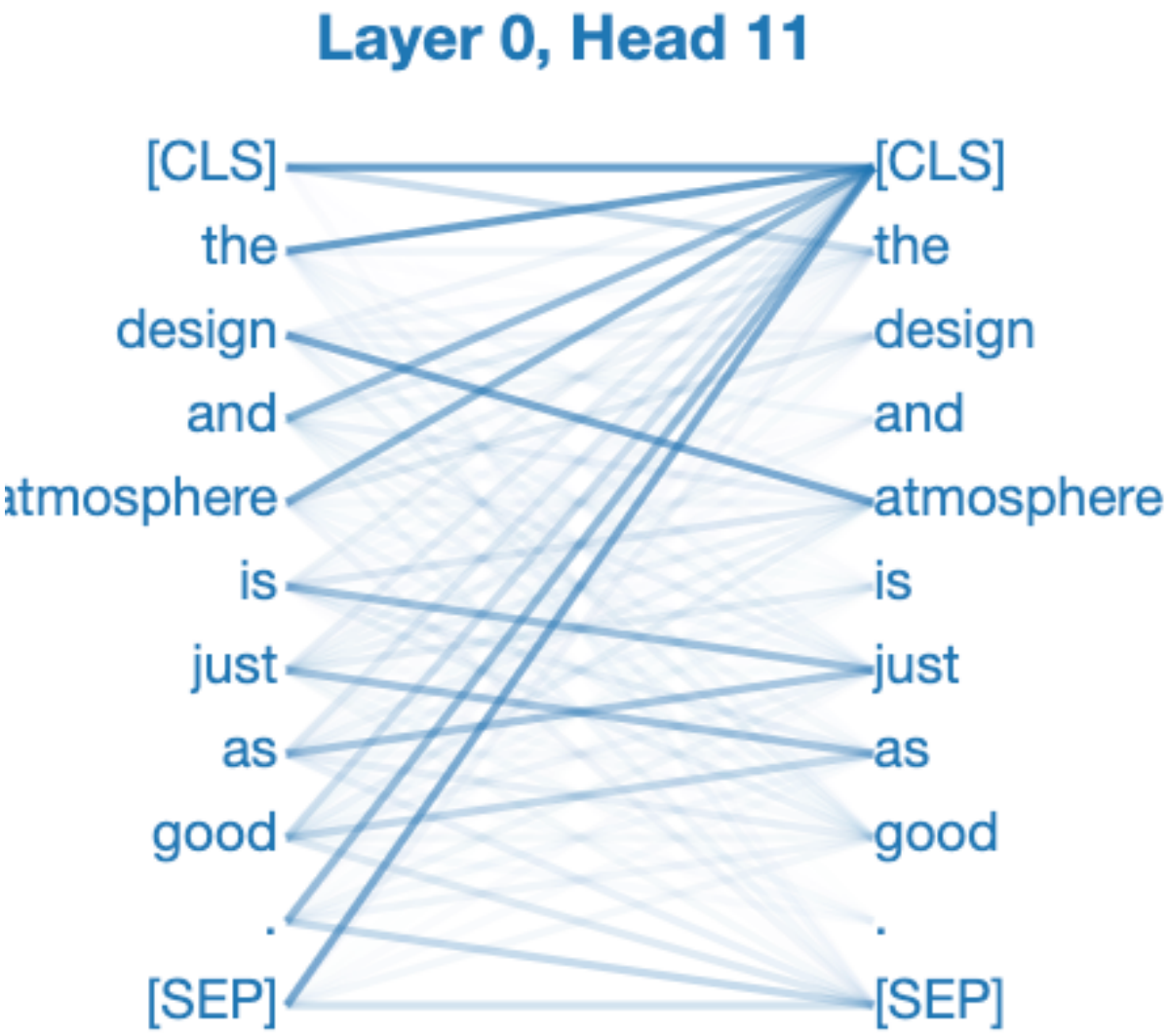

Figure 4.1: Layer 0 Head 11 of original BERT

4.4 Results

4.4.1 TSA results

For the test results, we experiment on the original semeval (Pontiki et al., 2014) data set and twitter data set (Mitchell, Aguilar, Wilson, & Van Durme, 2013). For comparison, we use the reported results from TD-BERT (Gao, Feng, Song, & Wu, 2019).

Table 4.1: Test results on three typical data sets.

	Models	Twitter		Restaurant		Laptop	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
RNN baselines	TD-LSTM	0.7080	0.6900	0.7563	-	0.6813	-
	ATAE-LSTM	-	-	0.7720	-	0.6870	-
	IAN	-	-	0.7860	-	0.7210	-
	RAM	0.6936	0.6730	0.8023	0.7080	0.7449	0.7135
Non-RNN baselines	Feature-based SVM	0.6340	0.6330	0.8016	-	0.7049	-
	Rec-NN	0.6630	0.6590	-	-	-	-
	MemNet	0.6850	0.6691	0.7816	0.6583	0.7033	0.6409
AEN-BERT	BERT	0.7464	0.7304	0.8128	0.6979	0.7654	0.7283
	BERT-SPC	0.7355	0.7214	0.8446	0.7698	0.7899	0.7503
	AEN-BERT	0.7471	0.7313	0.8312	0.7376	0.7993	0.7631
BERT-PT	BERT-PT	-	-	0.8495	0.7696	0.7807	0.7508
TD-BERT	TD-BERT	0.7669	0.7428	0.8510	0.7835	0.7807	0.7508
	TD-BERT-QA-CON	0.7731	0.7440	0.8456	0.7961	0.7842	0.7437
Our	Framework	0.7673	0.7451	0.843	0.780	0.7633	0.7291
	Framework+aux1.0	-	-	**0.8554**	**0.7962**	**0.7806**	**0.7523**
	Framework+aux0.1	-	-	0.8464	0.788	0.7759	0.7370

4.4.2 Domain Adaption Test Results

The result is shown on 4.2. The domain bert is post-trained on same domain unlabeled articles. We use the domain bert from BERT-ADA(Rietzler, Stabinger, Opitz, & Engl, 2019).

4.4.3 Robustness Test Results

The results show that our framework work well on the domain adapted bert and own robustness towards the unseen targets. In the original data set, our framework can reach SOTA's performance.

Table 4.2: Test results on three domain bert

		Twitter		Restaurant		Laptop	
Domain	Method	acc	f1	acc	f1	acc	f1
joint	BERT-ADA Joint	-	-	0.8635	0.7889	0.7896	0.7418
rest	BERT-ADA Rest	-	-	0.8714	0.8005	0.7860	0.7409
lapt	BERT-ADA Lapt	-	-	0.8551	0.7809	0.7919	0.7418
	Framework	0.7673	0.7451	0.843	0.7808	0.7633	0.7291
	aux 1	-	-	0.8554	0.7962	0.7806	0.7523
	aux 0.1	-	-	0.8464	0.7883	0.7759	0.737
joint	Framework	-	-	0.8491	0.7697	0.7821	0.7421
	aux 1	-	-	0.8768	0.8153	0.7978	0.7506
	aux 0.1	-	-	0.8625	0.8017	0.7915	0.745
rest	Framework	-	-	0.86911	0.8061	0.7931	0.7461
	aux 1	-	-	0.8786	0.8139	0.7978	0.7571
	aux 0.1	-	-	0.87589	0.81696	0.7978	0.7538
lapt	Framework	-	-	0.8562	0.7928	0.79	0.7477
	aux 1	-	-	0.8786	0.8139	0.8088	0.7627
	aux 0.1	-	-	0.87589	0.81696	0.8025	0.7653

Table 4.3: Test results on three re-split data sets.

	Models	Twitter		Restaurant		Laptop	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
	BERT-SPC	0.7052	0.6898	0.7857	0.6570	0.7524	0.6875
Our	Framework	0.7197	0.7054	0.8420	0.7740	0.7696	0.7282
	Framework+adv1	0.7673	0.7513	0.8545	0.7954	0.7821	0.7420
	Framework+aux1	-	-	0.8420	0.7735	0.7774	0.7466
	Framework+adv1+aux1	-	-	0.8500	0.7742	0.7884	0.7466

Chapter 5

Conclusion and Future Work

5.1 Conclusion

In our work, we analyze the characteristics of the targeted sentiment analysis problem. Then we proposed a generic framework for TSA. Based on the framework, we introduce the auxiliary training methods to better fine-tune BERT. To test the robustness of our framework, we construct a new robustness data set based on the original data set. We compare our method in the robustness data set. To enhance the model's robustness towards unseen or few-seen targets, we utilize the adversarial training to make the model less rely on the target tokens(words) but make more use of the context information.

5.2 Future Work

5.2.1 Domain adaptation for BERT(Post-training BERT)

The post-training bert can achieve better performance in the specific domain's sentiment analysis. In previous research, many people tried to use the within-domain corpus to do the post-training of bert. And utilize the But how to do the post-training is an interesting question. For a general answer, we can follow the within-domain post-training methods in text classification. But it is obviously not a good idea for a specific fine-grained problem. In other areas, some people utilize corpus based knowledge graphs to construct special features self-supervised

training methods. A similar idea can be applied in TSA problem.

5.2.2 Data Augmentation

Another way of solving the problem of lack of labeled training data is to use the data augmentation. According to previous research, transfer learning usually cab be done cross closely related areas such as reviews on laptops and restaurants. However, the data set size is still limited. To make the model more widely applicable and enhance the model's performance, we can consider some data argumentation methods. Actually, adversarial learning is a general transformation for Data Augmentation. There are other data argumentation methods to be further considered.

References

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, , May, 2019.

Gao, Z., Feng, A., Song, X., & Wu, X. (2019). Target-dependent sentiment classification with bert. *IEEE Access*, *7*, 2019, 154290–154299.

Karimi, A., Rossi, L., Prati, A., & Full, K. (2020). Adversarial training for aspect-based sentiment analysis with bert.

Mitchell, M., Aguilar, J., Wilson, T., & Van Durme, B. (2013). Open domain targeted sentiment. *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing* (pp. 1643–1654), 2013.

Miyato, T., Dai, A. M., & Goodfellow, I. (2016). Adversarial training methods for semi-supervised text classification.

Pei, J., Sun, A., & Li, C. (2019). Targeted sentiment analysis: A data-driven categorization.

Pontiki, M., Galanis, D., Pavlopoulos, J., Papageorgiou, H., Androutsopoulos, I., & Manandhar, S. (2014). SemEval-2014 task 4: Aspect based sentiment analysis. *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)* (pp. 27–35), Dublin, Ireland, August, 2014: Association for Computational Linguistics.

Rietzler, A., Stabinger, S., Opitz, P., & Engl, S. (2019). Adapt or get left behind: Domain adaptation through bert language model finetuning for aspect-target sentiment classification.

Soares, L. B., FitzGerald, N., Ling, J., & Kwiatkowski, T. (2019). Matching the blanks: Distributional similarity for relation learning.

Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019a). Attentional encoder network for targeted sentiment classification. *arXiv preprint arXiv:1902.09314*, , 2019.

Song, Y., Wang, J., Jiang, T., Liu, Z., & Rao, Y. (2019b). Targeted sentiment classification with attentional encoder network. *Lecture Notes in Computer Science*, , 2019, 93–103.

Sun, C., Qiu, X., Xu, Y., & Huang, X. (2019). How to fine-tune bert for text classification?

Tang, D., Qin, B., Feng, X., & Liu, T. (2016a). Effective LSTMs for target-dependent sentiment classification. *COLING 2016 - 26th International Conference on Computational Linguistics, Proceedings of COLING 2016: Technical Papers* (pp. 3298–3307), 2016.

Tang, D., Qin, B., & Liu, T. (2016b). Aspect Level Sentiment Classification with Deep Memory Network. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 214–224), 2016.

Xu, H., Liu, B., Shu, L., & Yu, P. S. (2019). Bert post-training for review reading comprehension and aspect-based sentiment analysis.

Zhaoa, P., Houb, L., & Wua, O. (2019). Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification. *arXiv preprint arXiv:1906.04501*, , 2019.

Appendix A

Visualization Results

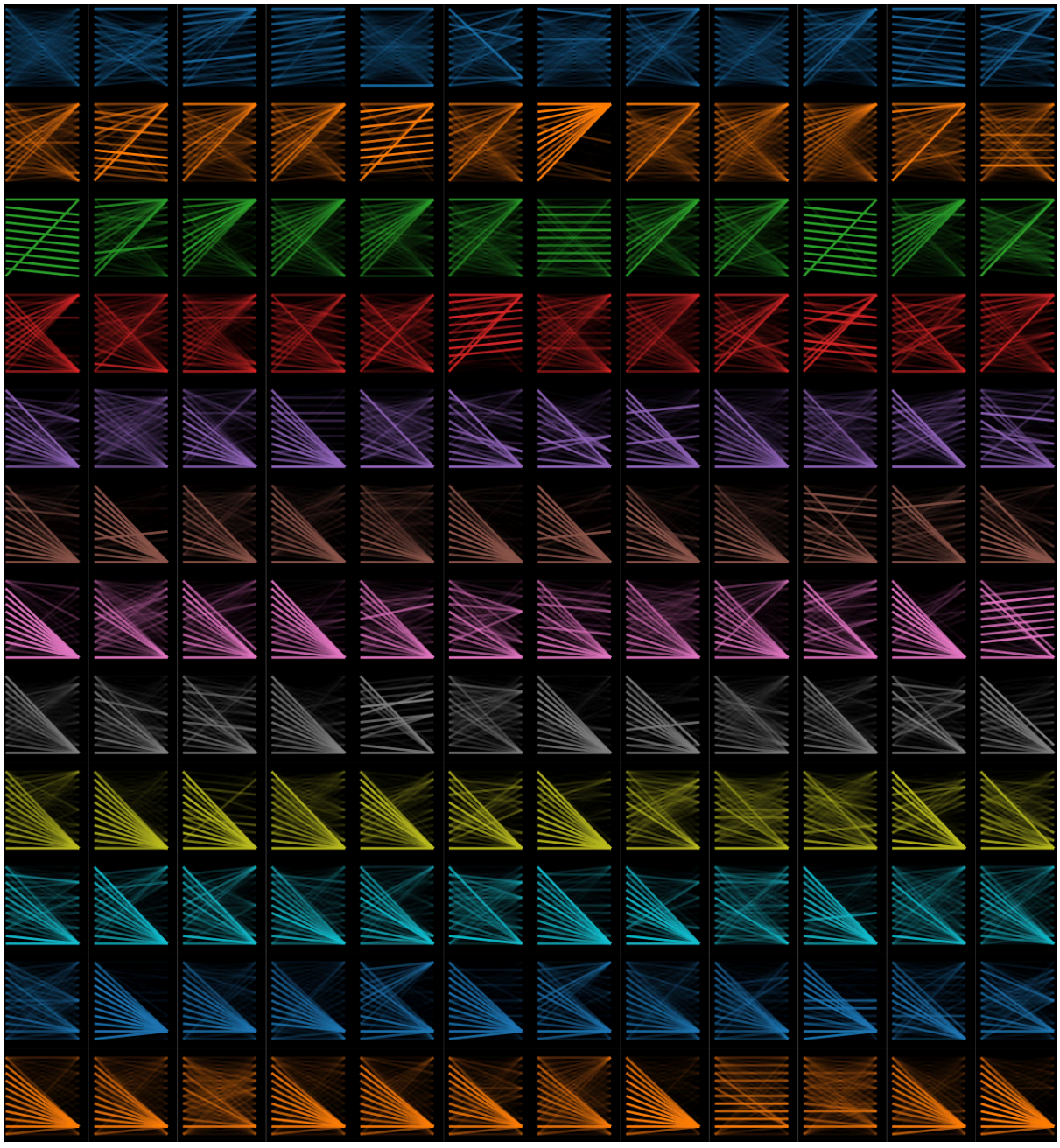

Figure A.1: BERT ALL Layers(From top to bottom, is layer 0 to layer 11. From left to right, is the head 0 to head 8)

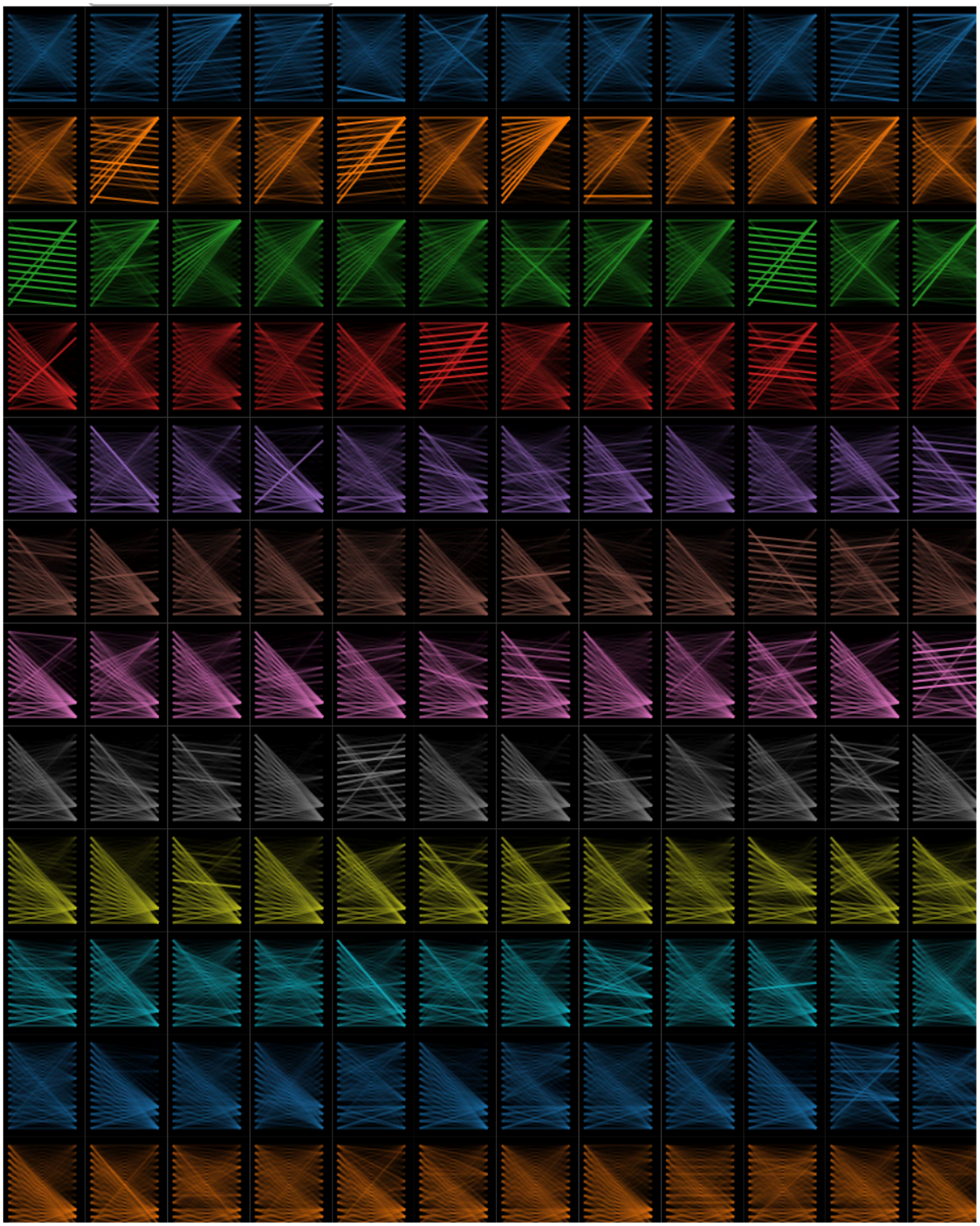

Figure A.2: BERT-SPC ALL Layers(From top to bottom, is layer 0 to layer 11. From left to right, is the head 0 to head 8)

Appendix B

Code

Our code is available on TSA-Pytorch.