B.Comp. Dissertation

A BERT-Based Framework for Targeted Sentiment Analysis

By

Xiang Pan

Department of Computer Science

School of Computing

National University of Singapore

2020/04

B.Comp. Dissertation

A BERT-Based 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

Sentiment Analysis In the report, we distinguish the differences between the general sentiment analysis and Targeted Sentiment Analysis. Future more, we analyze the existing problems of Targeted Sentiment Analysis. To address these 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 make 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 behave better than 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 the future work in targeted sentiment analysis.

Subject Descriptors:
C5 Computer System Implementation
G2.2 Graph Algorithms

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 be able to complete this project.

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.

The sentiment analysis was

1.2 The Problem

In different papers, the problem have different name. The target sentence analysis is task with several subtask, such as target term extraction, target term sentiment classification. In this work, we focus on the target term sentiment classification.

(pei2019targeted) proposed a detailed classification for the target sentiment analysis: We denote the f_{cls} as the classification model.

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)

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

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

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 for 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 typcial sequence data format, various sequence models have been applied to the text classification problem.

In order to perform the downstream tasks of natural language processing, it is usually necessary to reexpress 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 (devlinBERTPretrainingDeep2019) as a typical and successful pre-trained model for various nlp tasks, can be seen as a baselines. How to use the pre-trained language model to adapt to various downstream tasks and the efficient fine-tuning is focused by researchers.

2.2 Aspect Based Sentiment Analysis

We briefly review the related work in ABSA.

Using the Memory Network architecture (Tang2016), 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 (Tang2015) that used the left or right context alone, making full use of context words can improve its model.

The Attention Encoder Network (AEN) (ArxSong) 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 MHA inner layer and MHA inner layer in order to model the target words and context differently, which is lighter than the transformer architecture.

Graph Convolutional Neural Network (GCN) (ArxZhaoa2019) achieves another recent performance improvement, using graph convolutional neural network to explicitly establish the dependence between emotional words in sentences with multiple aspects mold. 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 are various for utilizing 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-etal-2014-semeval) subtask 2.

3.2 Design of Algorithm

3.3 A general framework for Targeted Sentiment Analysis(TSA)

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 sequence

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

To enhance the performance, (sun2019finetune) have 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 (Song2019) repeat the target words at the end of context sentence.

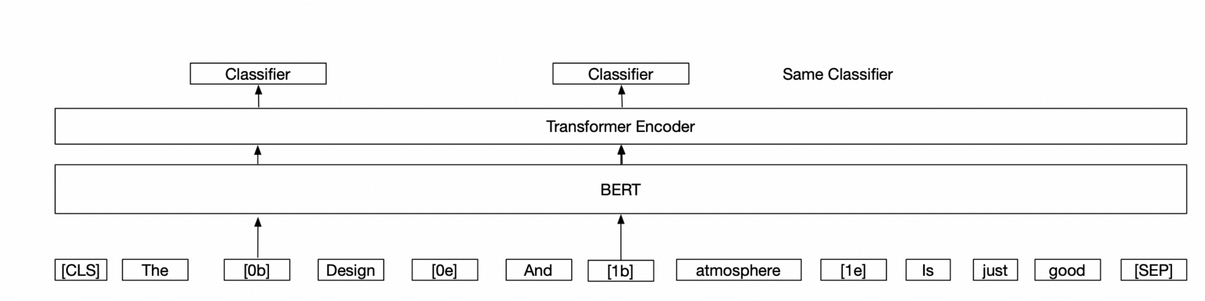

Figure 3.1: Our framework for TSA

For original bert, the most direct way to utilize bert

As show in the image 4.1, we add the begin token and end token around the target.

To test the token add methods, in our initial experiments, we add the same token for different targets in the context sentence. However, the results 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

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

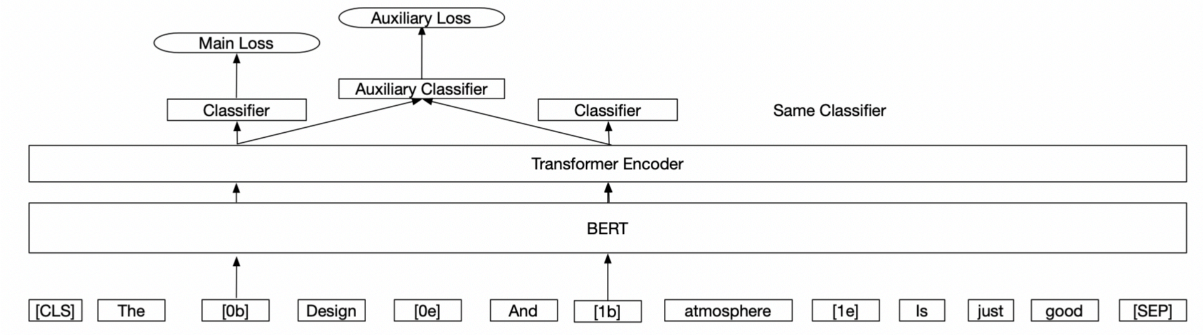

Figure 3.2: auxiliary training

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

(karimi2020adversarial) use adversarial training methods from (miyato2016adversarial) to enhance BERT-PT(xu2019bert) model's performance. We utilize the similar methods in our framework.

From the 3.3 we can see that our model is still have some dependence on the bottom layer of BERT. Such dependence may contribute to higher score in reappearing target sentiment analysis. However, when a target is never seen before, the dependence may decrease the robustness. We would like to make the framework to be less rely on the target tokens.

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

3.6.1 Robustness Test Settings

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

3.6.2 Adversarial training

The adversarial training method is searching the worst perturbations which can make the largest classification error. Towards the main optimization function, the adversarial training target is maximize the main loss function. For maximize 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:

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

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

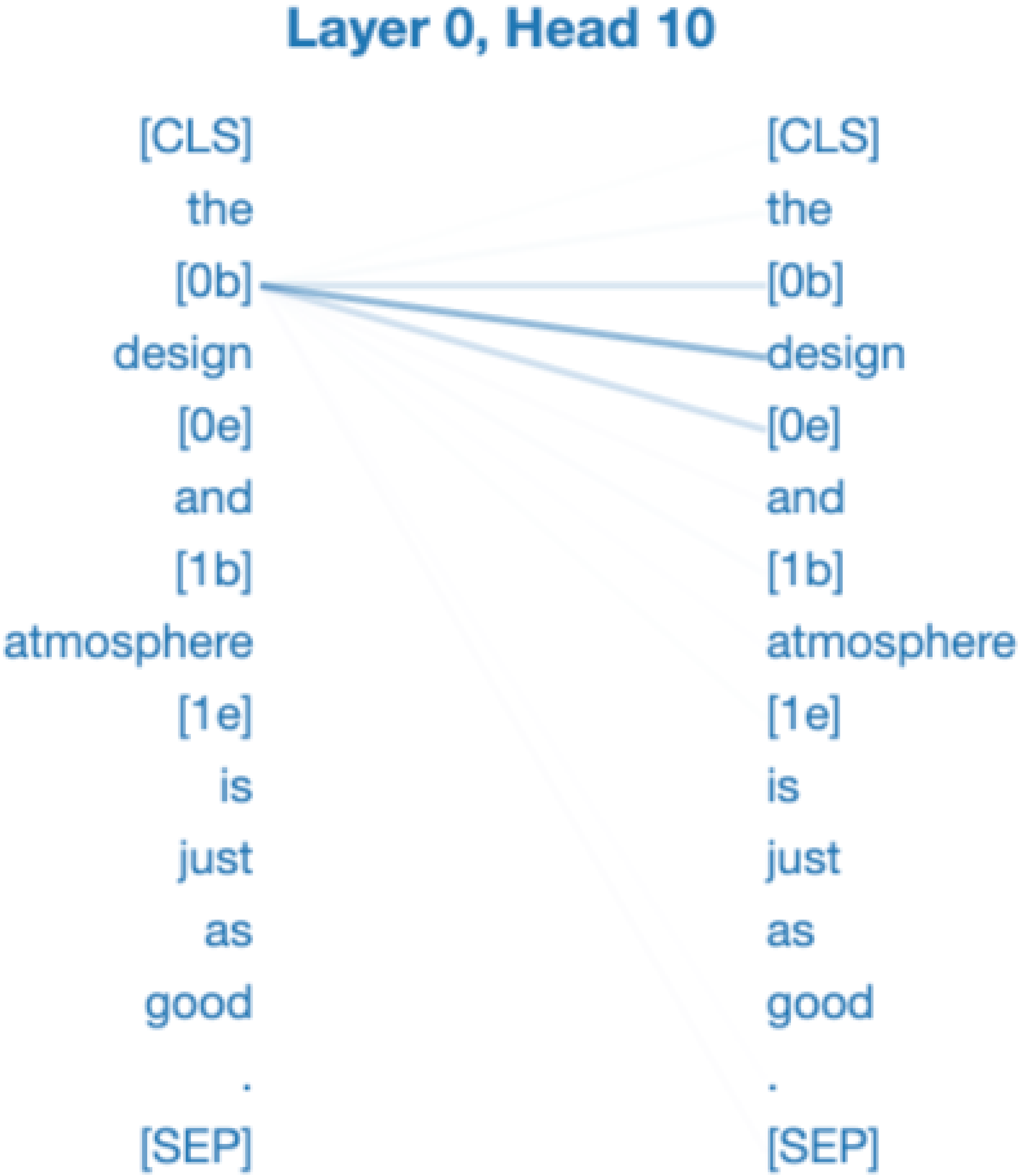

Figure 3.3: Bottom layer of BERT visualization

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 libraries. 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, adversarial training usually takes up to 10)

4.2.1 Targeted Sentiment Analysis

For our framework, the number of fine-tuning epochs is less than 10.

4.2.2 Domain Adaption' effect on Targeted Sentiment Analysis

4.2.3 Robustness of Targeted Sentiment Analysis Algorithms

4.3 Visualization

4.3.1 Pre-trained BERT Visualization

From the vi

4.3.2 BERT-SPC Visualization

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

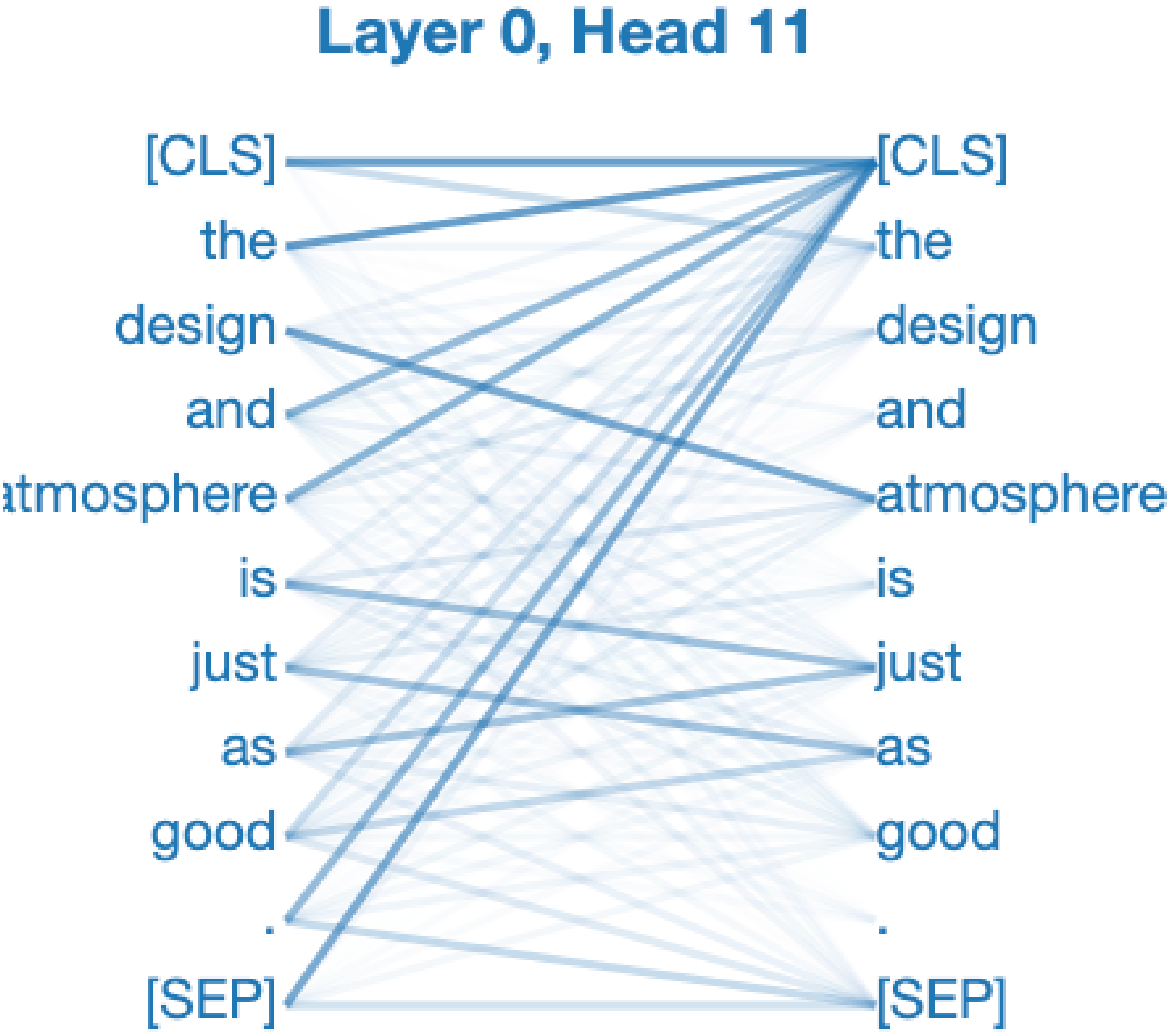

Figure 4.1: Layer 0 Head 11 of original BERT

4.4 Results

4.4.1 TSA results

For the test results, we experiment on the original semeval dataset and twitter dataset. We use the reported results from TD-BERT (8864964)

4.4.2 Robustness Test results

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	-	-	75.29	71.91	81.54	71.94
	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	76.96	0.7807	0.7508
Our	Framework	0.7673	0.7451	0.8536	0.778	0.8009	0.7673

Table 4.2: Test results on three re-split data sets.

	Models	Twitter		Restaurant		Laptop	
		Accuracy	Macro-F1	Accuracy	Macro-F1	Accuracy	Macro-F1
AEN-BERT	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	76.96	0.7807	0.7508
	TD-BERT	-	-	0.8495	76.96	0.7807	0.7508
Our	Framework	0.7673	0.7451	0.8536	0.778	0.7915	0.76

New re-split dataset

Chapter 5

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 instead to make more use of the context information.

5.1 Future Work

5.1.1 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 area, some people utilize corpus based knowledge graphs to construct special features self-supervised training methods. Similar idea can be applied in TSA problem.

5.1.2 Data Argumentation

Another way of solving the problem of lack of labeled training data is to use the data Argumentation. According to previous research, the transfer learning usually done cross closely related area such as reviews on laptop and restaurant. 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, the adversarial learning is a general transformations for Data Augmentation. There are other data argumentation methods to be further considered.

Appendix A

Visualization Results

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

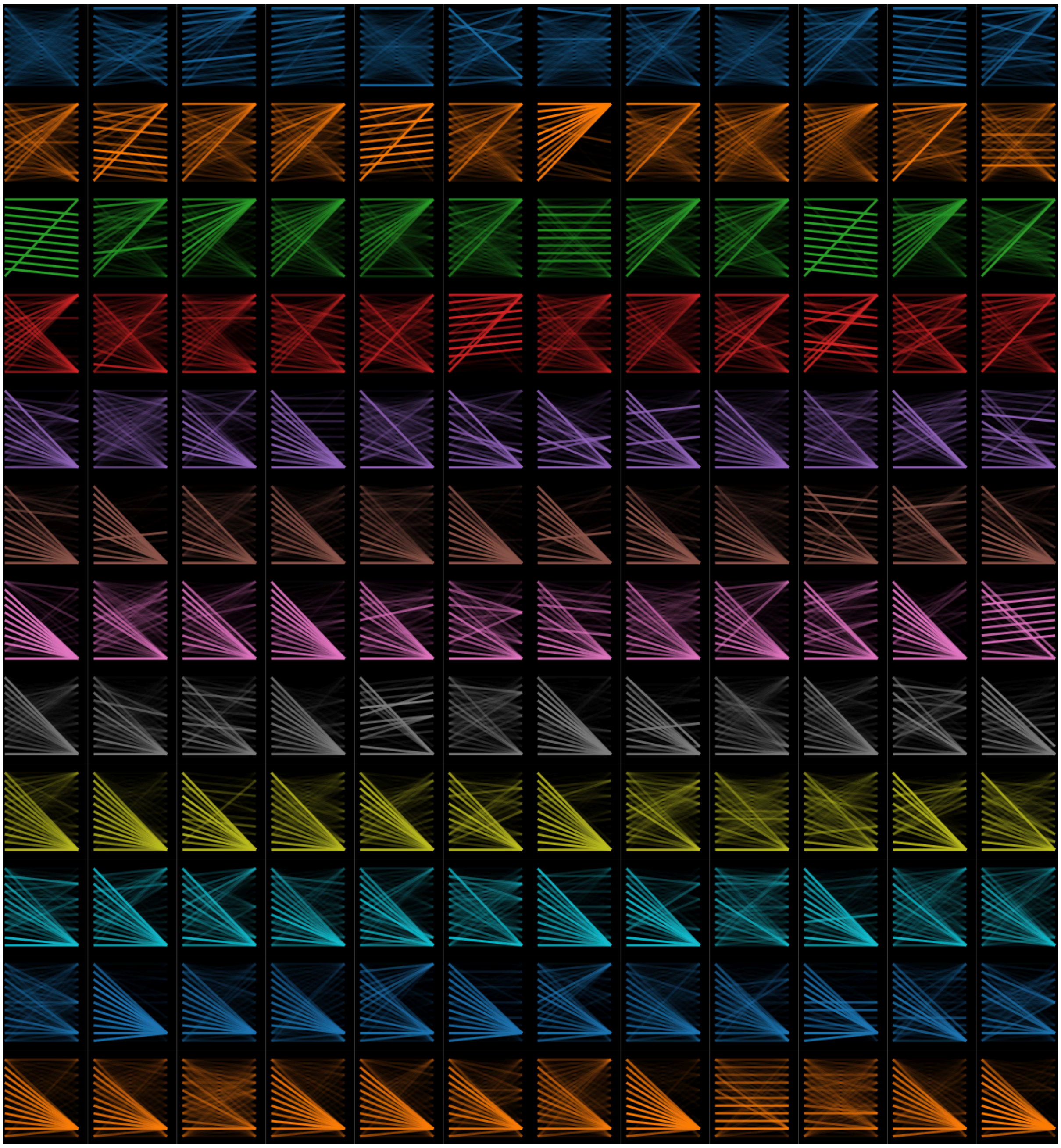

Figure A.1: BERT ALL Layers

(From top to bottom, is layer 0 to layer 11. From left to right, is the head 0 to head 8)

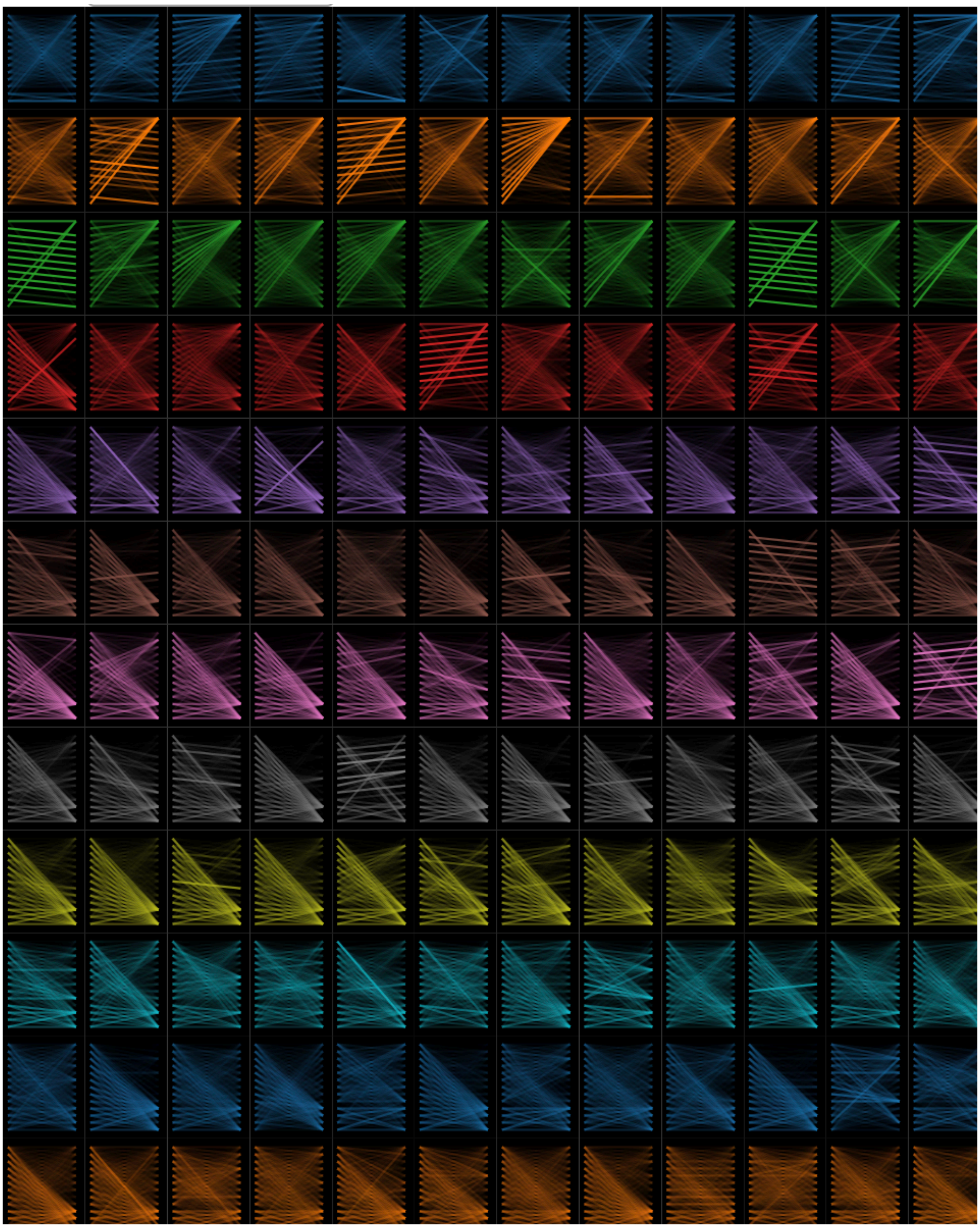

Figure A.2: BERT-SPC ALL Layers

(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.