Transformation Networks for Target-Oriented Sentiment Classification[1]

Xin Li[1], Lidong Bing[2], Wai Lam[1], Bei Shi[1]

[1]The Chinese University of Hong Kong

[2]Tencent AI Lab

ACL 2018

[1]Joint work with Tencent AI Lab

Outline

1. Target-Oriented Sentiment Classification
   - Introduction
   - Problem Formulation
2. Transformation Networks for Target-Oriented Sentiment Classification
   - Motivation
   - The proposed model
3. Experiment
   - Settings
   - Comparative Study

Outline

1. Target-Oriented Sentiment Classification
   - Introduction
   - Problem Formulation
2. Transformation Networks for Target-Oriented Sentiment Classification
   - Motivation
   - The proposed model
3. Experiment
   - Settings
   - Comparative Study

Introduction

Target-Oriented Sentiment Classification (TOSC) is to detect the overall opinions / sentiments of the user review towards the given opinion target.

- TOSC is a supporting task of Target / Aspect-based Sentiment Analysis [5].
- TOSC has been investigated extensively in other names:
  - Aspect-level Sentiment Classification [1, 7, 10, 11, 12].
  - Targeted Sentiment Prediction [6, 14].
  - Target-Dependent Sentiment Classification [2, 9].

Outline

Problem Formulation

- TOSC is a typical classification task but the input texts come from two sources:
  1. Target: explicitly mentioned phrase of opinion target, also called "aspect term" or "aspect".
  2. Context: the original review sentence or the sentence without target phrase.
- TOSC is to predict the overall sentiment of the context towards the target.

Example

- [**Boot time**] is super fast, around anywhere from 35 seconds to 1 minute.
  - This review conveys positive sentiment over the input **"Boot time"**.
- Great [**food**] but the [**service**] is dreadful.
  - Given the target **"food"**, the sentiment polarity is positive while if the input target is **"service"**, it becomes negative.

Outline

Motivation

1. Convolutional Neural Network (CNN) is more suitable for this task than Attention-based Models [1, 6, 7, 10, 11, 12, 13].
   - Sentiments towards the targets are usually determined by key phrases.
     - Example: This [dish] is my favorite and I always get it and never get tired of it.
     - CNN whose aim is to capture the most informative n-grams (e.g., "is my favorite") in the sentence should be a suitable model.
   - Attention-based weighted combination of the entire word-level features may introduce some noises (e.g., "never" and "tired" in above sentence).
   - We employ proximity-based CNN rather than attention-based RNN as the top-most feature extractor.

Motivation

2. CNN likely fails in cases where a sentence expresses different sentiments over multiple targets.
   - Example: great [food] but the [service] was dreadful!
   - CNN cannot fully explore the target information via vector concatenation.
   - Combining context information and word embedding is an effective way to represent a word in the convolution-based architecture [4]
   - Our Solution:
     (i) We propose a "Target-Specific Transformation" (TST) component to better consolidate the target information with word representations.
     (ii) We design two context-preserving mechanisms "Adaptive Scaling" (AS) and "Loseless Forwarding" (LF) to combine the contextualized representations and the transformed representations.

Motivation

3. Most of the existing works do not discriminate different words in the same target phrase
   - In the target phrase, different words would not contribute equally to the target representation.
   - For example, in **“amd turin processor”**, phrase head **“processor”** is more important than **“amd”** and **“turin”**.
   - Our TST solves this problem in two steps:
     (i) Explicitly calculating the importance scores of the target words.
     (ii) Conducting word-level association between the target and its context.

Outline

Model Overview

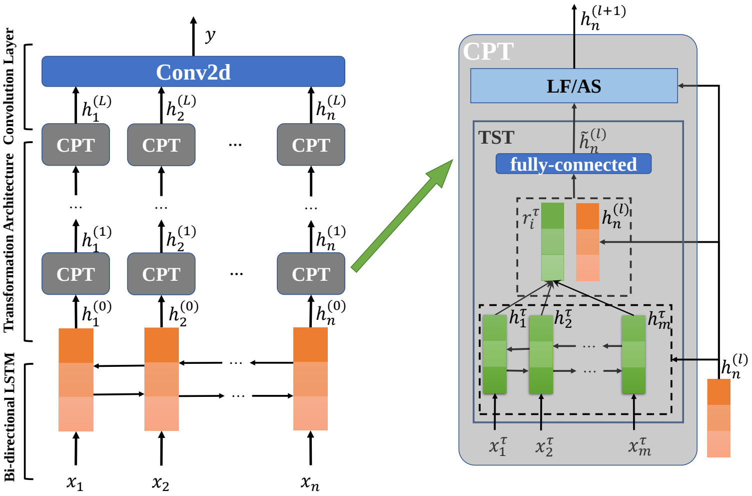


Figure: Architecture of TNet.

Model Overview

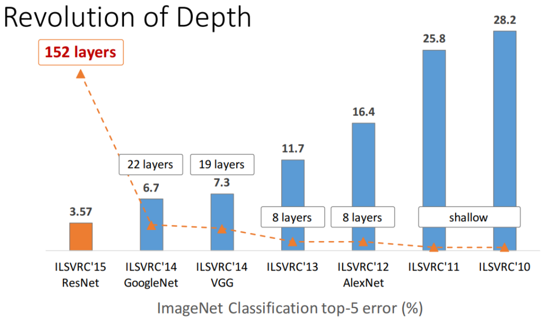
The proposed TNet consists of the following three components:

1. (BOTTOM) Bi-directional LSTM for memory building
   - Generating contextualized word representations.
2. (MIDDLE) Deep Transformation architecture for learning target-specific word representations
   - Refining word-level representations with the input target and the contextual information.
3. (TOP) Proximity-based convolutional feature extractor.
   - Introducing position information to detect the most salient features more accurately.

Deep Transformation Architecture

Deep Transformation Architecture stacks multiple Context-Preserving Transformation (CPT) layers

– Deeper network helps to learn more abstract features (He et al., CVPR 2016; Lecun et al., Nature 2015).



CPT Layer

The functions of the CPT layer are two folds:

1. Incorporating opinion target information into the word-level representations.
   - Generating context-aware target representations r_i^τ conditioned on the i-th word representation $h_i^{(l)}$ fed to the l-th layer:

$$r_i^\tau = \sum_{j=1}^{m} h_j^\tau * \mathcal{F}(h_i^{(l)}, h_j^\tau),$$

$$\mathcal{F}(h_i^{(l)}, h_j^\tau) = \frac{\exp(h_i^{(l)\top} h_j^\tau)}{\sum_{k=1}^{m} \exp(h_i^{(l)\top} h_k^\tau)},$$

   - Obtaining target-specific word representations $\tilde{h}_i^{(l)}$:

$$\tilde{h}_i^{(l)} = g(W^\tau[h_i^{(l)} : r_i^\tau] + b^\tau),$$

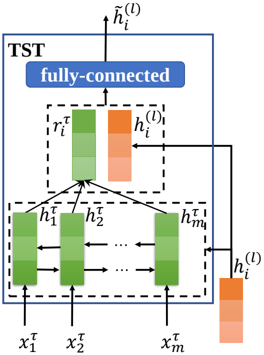


Figure: Target-Specific Transformation (TST) component

CPT Layer

2. Preserving context information for the upper layers
   - We design two Context-Preserving Mechanisms to add context information back to the transformed word features $\tilde{h}_i^{(l)}$
     (i) Adaptive Scaling (**AS**) (Similar to Highway Connection [8]):

$$t_i^{(l)} = \sigma(W_{trans} h_i^{(l)} + b_{trans}),$$

$$h_i^{(l+1)} = t_i^{(l)} \odot \tilde{h}_i^{(l)} + (1 - t_i^{(l)}) \odot h_i^{(l)}.$$

     (ii) Lossless Forwarding (**LF**) (Similar to Residual Connection [3]):

$$h_i^{(l+1)} = h_i^{(l)} + \tilde{h}_i^{(l)}.$$

Proximity-based Convolutional Feature Extractor

This component aims to capture the most salient feature w.r.t. the current target for sentiment prediction.

- As observed in (Chen et al., 2017; Li and Lam, 2017), distance information is effective for better locating the salient features.
  - Basic idea: Up-weighting the words close to the target and down-weighting those far away from the target.
- Convolutional neural network (Kim, 2014) is used to extract features from the weighted word representations.

Outline

1. Target-Oriented Sentiment Classification
   - Introduction
   - Problem Formulation
2. Transformation Networks for Target-Oriented Sentiment Classification
   - Motivation
   - The proposed model
3. Experiment
   - Settings
   - Comparative Study

Settings

Datasets

- LAPTOP, REST: datasets from SemEval14 ABSA challenge, containing the user reviews from laptop domain and restaurant domain respectively.
- TWITTER: a dataset built in (Dong et al., 2014), containing twitter posts and the opinion targets are annotated.

Compared Models

- Traditional Models:
  - SVM (Kiritchenko et al., 2014).
- Attention-based Models:
  - ATAE-LSTM (Wang et al., 2016), MemNet (Tang et al., 2016), IAN (Ma et al., 2017), BILSTM-ATT-G (Liu and Zhang, 2017), RAM (Chen et al., 2017).
- Other Neural Models:
  - AdaRNN (Dong et al., 2014), TD-LSTM (Tang et al., 2016), AE-LSTM (Wang et al., 2016), CNN-ASP

Outline

1. Target-Oriented Sentiment Classification
   - Introduction
   - Problem Formulation
2. Transformation Networks for Target-Oriented Sentiment Classification
   - Motivation
   - The proposed model
3. Experiment
   - Settings
   - Comparative Study

Main Results

	Models	LAPTOP		REST		TWITTER	
		ACC	Macro-F1	ACC	Macro-F1	ACC	Macro-F1
TNet variants	TNet-LF	76.01†,‡	71.47†,‡	**80.79**†,‡	70.84‡	74.68†,‡	73.36†,‡
	TNet-AS	**76.54**†,‡	**71.75**†,‡	80.69†,‡	**71.27**†,‡	**74.97**†,‡	**73.60**†,‡
Baselines	SVM	70.49$^{\natural}$	-	80.16$^{\natural}$	-	63.40*	63.30*
	AdaRNN	-	-	-	-	66.30$^{\natural}$	65.90$^{\natural}$
	AE-LSTM	68.90$^{\natural}$	-	76.60$^{\natural}$	-	-	-
	ATAE-LSTM	68.70$^{\natural}$	-	77.20$^{\natural}$	-	-	-
	IAN	72.10$^{\natural}$	-	78.60$^{\natural}$	-	-	-
	CNN-ASP	72.46	65.31	77.82	65.11	73.27	71.77
	TD-LSTM	71.83	68.43	78.00	66.73	66.62	64.01
	MemNet	70.33	64.09	78.16	65.83	68.50	66.91
	BILSTM-ATT-G	74.37	69.90	80.38	70.78	72.70	70.84
	RAM	75.01	70.51	79.79	68.86	71.88	70.33

- The proposed TNet-LF and TNet-AS consistently outperform the baselines.
  - TNet variants perform well on both user reviews (LAPTOP & REST) and twitter posts (TWITTER).

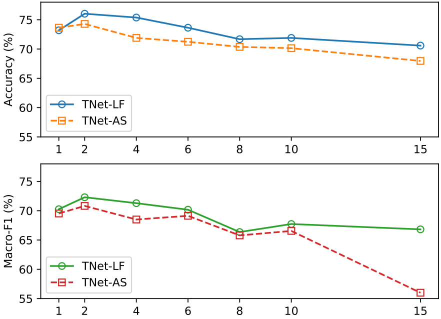
Ablation Experiment

	Models	LAPTOP		REST		TWITTER	
		ACC	Macro-F1	ACC	Macro-F1	ACC	Macro-F1
TNet variants	TNet-LF	76.01[†,‡]	71.47[†,‡]	**80.79**[†,‡]	70.84[‡]	74.68[†,‡]	73.36[†,‡]
	TNet-AS	**76.54**[†,‡]	**71.75**[†,‡]	80.69[†,‡]	**71.27**[†,‡]	**74.97**[†,‡]	**73.60**[†,‡]
CPT Alternatives	LSTM-ATT-CNN	73.37	68.03	78.95	68.71	70.09	67.68
	LSTM-FC-CNN-LF	75.59	70.60	80.41	70.23	73.70	72.82
	LSTM-FC-CNN-AS	75.78	70.72	80.23	70.06	74.28	72.60
Ablated TNet	TNet w/o transformation	73.30	68.25	78.90	65.86	72.10	70.57
	TNet w/o context	73.91	68.87	80.07	69.01	74.51	73.05
	TNet-LF w/o position	75.13	70.63	79.86	69.69	73.83	72.49
	TNet-AS w/o position	75.27	70.03	79.79	69.78	73.84	72.47

- Using attention (ATT) and fully-connected layer (FC) to replace CPT layer makes the performance worse.
- Each component / element in TNet contributes to the overall performance improvement.

Impact of CPT layer number

We conduct experiments on the held-out training data of LAPTOP and vary layer number L from 2 to 10, increased by 2.



- Increasing the layer number can increase the performance but the results will go down when $L \geq 4$ due to the limited training data.

Case Study

Sentence	BILSTM-ATT-G	RAM	TNet-LF	TNet-AS
1. Air has higher [**resolution**]$_P$ but the [**fonts**]$_N$ are small .	(N✗, N)	(N✗, N)	(P, N)	(P, N)
2. Great [**food**]$_P$ but the [**service**]$_N$ is dreadful .	(P, N)	(P, N)	(P, N)	(P, N)
3. Sure it ' s not light and slim but the [**features**]$_P$ make up for it 100% .	N✗	N✗	P	P
4. Not only did they have amazing , [**sandwiches**]$_P$, [**soup**]$_P$, [**pizza**]$_P$ etc , but their [**homemade sorbets**]$_P$ are out of this world !	(P, O✗, O✗, P)	(P, P, O✗, P)	(P, P, P, P)	(P, P, P, P)
5. [**startup times**]$_N$ are incredibly long : over two minutes .	P✗	P✗	N	N
6. I am pleased with the fast [**log on**]$_P$, speedy [**wifi connection**]$_P$ and the long [**battery life**]$_P$ ($>$ 6 hrs) .	(P, P, P)	(P, P, P)	(P, P, P)	(P, P, P)
7. The [**staff**]$_N$ should be a bit more friendly .	P✗	P✗	P✗	P✗

- Our TNet can make correct predictions when the opinion is target specific, e.g., “long” in the 5th and the 6th example.
- TNet can capture the salient features for target sentiment prediction accurately.

Summary

- Our TNet employs CNN as feature extractor to detect the salient features, avoiding introducing the noises.
- Armed with target-specific word representation and proximity information, the TNet variants can predict the sentiment w.r.t. the target more accurately.

References:

[1] P. Chen, Z. Sun, L. Bing, and W. Yang. Recurrent attention network on memory for aspect sentiment analysis. In *Proceedings of EMNLP*, pages 463–472, 2017.

[2] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, and K. Xu. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of ACL*, pages 49–54, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of CVPR*, pages 770–778, 2016.

[4] S. Lai, L. Xu, K. Liu, and J. Zhao. Recurrent convolutional neural networks for text classification. In *Proceedings of AAAI*, volume 333, pages 2267–2273, 2015.

[5] B. Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[6] J. Liu and Y. Zhang. Attention modeling for targeted sentiment. In *Proceedings of EACL*, pages 572–577, 2017.

[7] D. Ma, S. Li, X. Zhang, and H. Wang. Interactive attention networks for aspect-level sentiment classification. In *Proceedings of IJCAI*, pages 4068–4074, 2017.

[8] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, 2015.

[9] D. Tang, B. Qin, X. Feng, and T. Liu. Effective lstms for target-dependent sentiment classification. In *Proceedings of COLING*, pages 3298–3307, 2016a.

[10] D. Tang, B. Qin, and T. Liu. Aspect level sentiment classification with deep memory network. In *Proceedings of EMNLP*, pages 214–224, 2016b.

[11] Y. Tay, A. T. Luu, and S. C. Hui. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. *arXiv preprint arXiv:1712.05403*, 2017.

[12] Y. Wang, M. Huang, x. zhu, and L. Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of EMNLP*, pages 606–615, 2016.

[13] M. Yang, W. Tu, J. Wang, F. Xu, and X. Chen. Attention based lstm for target dependent sentiment classification. In *Proceedings of AAAI*, pages 5013–5014, 2017.

[14] M. Zhang, Y. Zhang, and D.-T. Vo. Gated neural networks for targeted sentiment analysis. In *Proceedings of AAAI*, pages 3087–3093, 2016.