Transformation Networks for Target-Oriented Sentiment Classification

Xin Li[1], Lidong Bing[2], Wai Lam[1] and Bei Shi[1]
[1]Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong, Hong Kong
[2]Tencent AI Lab, Shenzhen, China
{lixin,wlam,bshi}@se.cuhk.edu.hk
lyndonbing@tencent.com

ACL2018読み会@名大

豊田工業大学　M2　辻村 有輝

Transformation Networks for Target-Oriented　Sentiment Classification

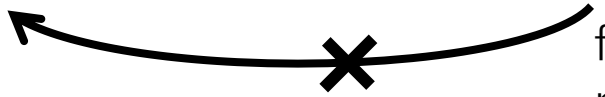
- 対象タスク：フレーズレベルのSentiment classification
  - 「great **food** but the **service** was dreadful!」
    ➡ food:Positive　service:Negative
- 新規性：ターゲットフレーズごとに表現を特化させて処理
  - 途中の層からターゲットフレーズの表現ベクトルを追加で毎層入力
    - アテンションベースの構造
    - 割とシンプルに見える
- 3つのベンチマークでSotA
  - 2つはSemEval ABSAのドメイン違いのデータセット
    残り1つはTwitterから収集されたデータセット

Target-oriented Sentiment Analysis

文中の特定フレーズに　入力：great **food** but the **service** was dreadful!
対する極性分類　　正解：　food:Positive　service:Negative

アテンションベース：

- 単語間に間違った結合を導入してしまうことで**性能が劣化**している
  - This **dish** is my favorite and I always get it and never get tired of it.



favoriteに対応する表現ベクトルの計算時に
neverの表現ベクトルに重みが付与されることがある

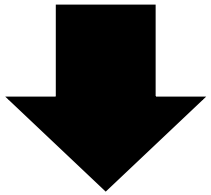
CNNベース：

- 極性を表す言及はフレーズとして現れることが多い
  - 例：“is my favorite”
  - CNNはフレーズを纏めて捉えやすい分RNNやアテンションより**タスクに合っているはず**
- しかし文中に複数のターゲットとキーフレーズのペアがあると
  ターゲットと関係ないフレーズのせいで間違った予測を行いがち
  - Great **food** but the **service** was dreadful!

foodの極性分類時にdreadfulのせいでNegativeと判断されてしまうことがある

モチベーションとアイディア

- Target-oriented Sentiment Analysisにはフレーズに強そうなCNNがふさわしいと考えられる
- しかしターゲットフレーズに関係ない部分に反応してしまう



- 各単語の表現ベクトルを現在注目しているターゲットごとに特化させて作り直すことで解決を図る
  - 具体的には，途中の層からターゲットの表現ベクトルを毎層追加入力として与えることで特化させる
  - 似た手法は以前（Wang et al., 2016）にもあったが，本論文の手法の方がより性能が高くなっている

TNet: Target-Specific Transformation Network

- Bi-LSTM層
  - 二方向をconcatして出力とする普通のBi-LSTM
- **CPT層（L層スタック）**
  - **ターゲットに合わせて各単語の表現を特化させる**
  - **本論文のメイン部分**
  - 出力ベクトルはターゲットとの相対位置に応じてスケーリングされる (Chen et al., 2017)（position relevanceと言っている部分）
  - Residual or Highway結合有り
- CNN層
  - Max-poolingでまとめ上げる
- 出力層
  - 全結合→softmax

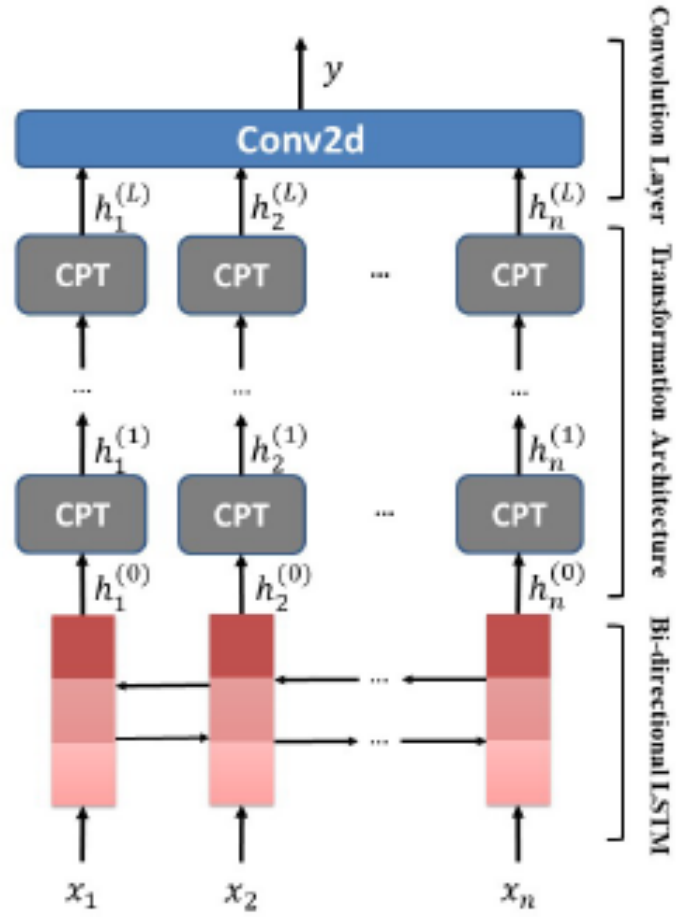


Figure 1: Architecture of TNet.

CPT: Context-Preserving Transformation

本論文のオリジナルなモジュール

各単語ベクトルをターゲットフレーズに特化した表現に変える

- 位置iごとに，その位置の単語ベクトルをもとに
  ターゲットフレーズをアテンションで纏め上げる

$$r_i^\tau = \sum_{j=1}^{m} h_j^\tau * \mathcal{F}(h_i^{(l)}, h_j^\tau) \qquad \mathcal{F}(h_i^{(l)}, h_j^\tau) = \frac{\exp\left(h_i^{(l)\top} h_j^\tau\right)}{\sum_{k=1}^{m} \exp\left(h_i^{(l)\top} h_k^\tau\right)}$$

- アテンションで出来たターゲットの表現と元の単語ベクトルを
  concatしたものを全結合層に入力したものがCPT層の出力

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

・CPT内では周りが（ターゲット以外）見えない

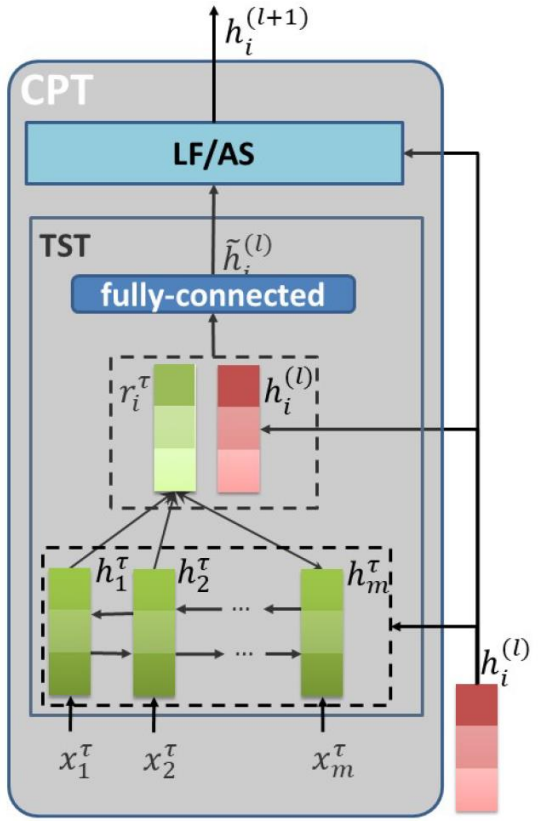


Figure 2: Details of a CPT module

6. I am pleased with the fast [log on]P , speedy [wifi connection]P and the long [battery life]P (> 6 hrs) .

TNet: Target-Specific Transformation Network

- Bi-LSTM層
  - 二方向をconcatして出力とする普通のBi-LSTM
- **CPT層（L層スタック）**
  - **ターゲットに合わせて各単語の表現を特化させる**
  - **本論文のメイン部分**
  - 出力ベクトルはターゲットとの相対位置に応じてスケーリングされる (Chen et al., 2017)（position relevanceと言っている部分）
  - Residual or Highway結合有り
- CNN層
  - Max-poolingでまとめ上げる
- 出力層
  - 全結合→softmax

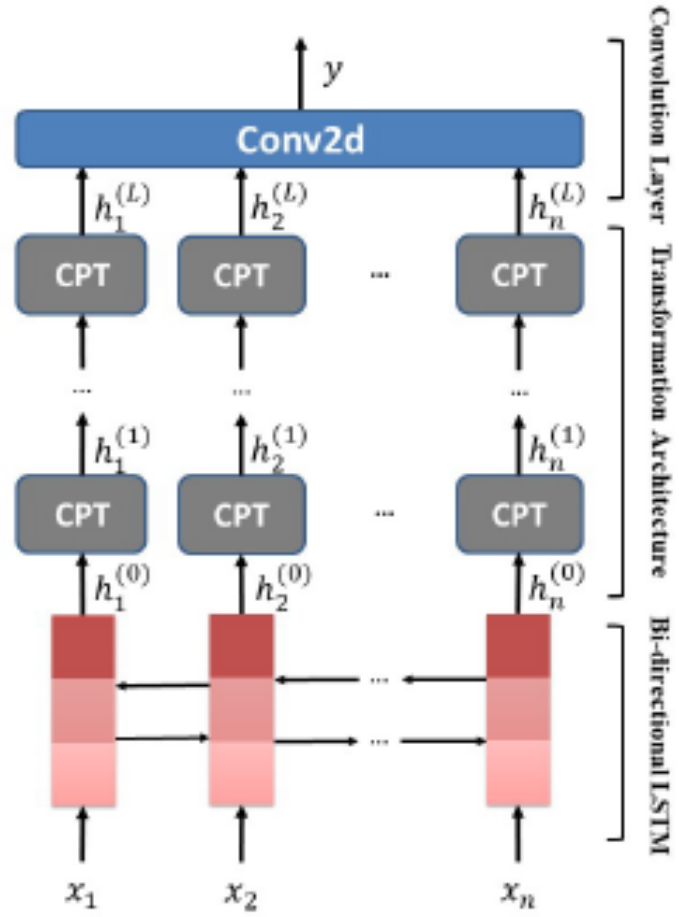


Figure 1: Architecture of TNet.

出力ベクトルのスケーリング（論文中でposition relevanceと言っている部分）

- 各CPT層の出力はターゲットから遠いほど小さくスケーリング

$$v_i = \begin{cases} 1 - \frac{(k+m-i)}{C} & i < k+m \\ 1 - \frac{i-k}{C} & k+m \le i \le n \\ 0 & i > n \end{cases}$$

k：ターゲットフレーズの先頭インデックス

m：ターゲットフレーズの長さ

C：定数（30～40）

$$\hat{h}_i^{(l)} = h_i^{(l)} * v_i \qquad i \in [1, n], l \in [1, L]$$

- フレーズの前後で非対称・フレーズ内で不均一だがいいのか？
  - 直前 i=k-1で 1-(m+1)/C，直後 i=k+mで 1-m/C
  - おそらく元にした (Chen et al., 2017) では対称・内部で均一になっている

TNet: Target-Specific Transformation Network

- Bi-LSTM層
  - 二方向をconcatして出力とする普通のBi-LSTM
- **CPT層（L層スタック）**
  - **ターゲットに合わせて各単語の表現を特化させる**
  - **本論文のメイン部分**
  - 出力ベクトルはターゲットとの相対位置に応じてスケーリングされる (Chen et al., 2017)（position relevanceと言っている部分）
  - Residual or Highway結合有り
- CNN層
  - Max-poolingでまとめ上げる
- 出力層
  - 全結合→softmax

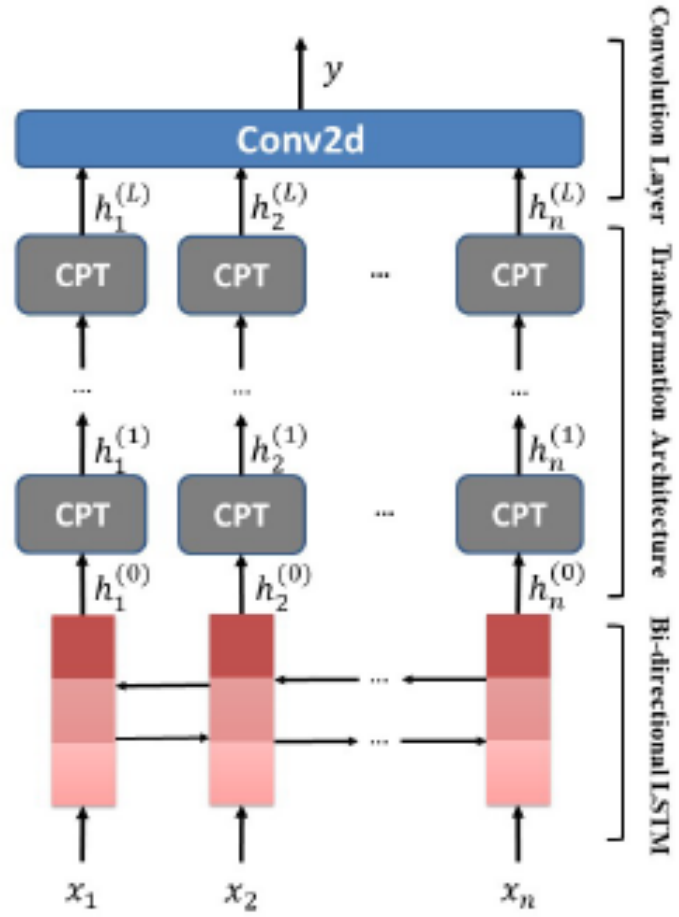


Figure 1: Architecture of TNet.

他のモデルとの比較実験

- SVM
- AdaRNN
  - 構文木を使うモデル．RNNはバニラなbinary RecursiveNN
- AE-LSTM, ATAE-LSTM
  - 単層双方向LSTMの入力にターゲットの表現をconcatしたモデル．元論文ではターゲットフレーズではなくターゲット分野の表現を使っており異なるように思える
- CNN-ASP
  - 著者らが比較のために作った，ターゲットの表現を入力にconcatした単純なCNN
- IAN
  - ターゲットフレーズと全体のコンテキストを別々のLSTMで処理し，あとでconcatする
- TD-LSTM，BILSTM-ATT-G
  - ターゲットより左，ターゲット，ターゲットより右に分けてそれぞれ別個でLSTM
- MemNet，RAM
  - アテンションベース．RAMではターゲットからの相対位置に応じてスケーリングを行う

他のモデルとの比較実験

	Models	LAPTOP		REST		TWITTER	
		ACC	Macro-F1	ACC	Macro-F1	ACC	Macro-F1
Baselines	SVM	70.49♮	-	80.16♮	-	63.40*	63.30*
	AdaRNN	-	-	-	-	66.30♮	65.90♮
	AE-LSTM	68.90♮	-	76.60♮	-	-	-
	ATAE-LSTM	68.70♮	-	77.20♮	-	-	-
	IAN	72.10♮	-	78.60♮	-	-	-
	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
TNet variants	TNet w/o context	73.91	68.87	80.07	69.01	74.51	73.05
	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**†,‡

LF：Residual　AS：Highway

- 各データセットでSotA
  - 2つはSemEval ABSAのドメイン違いのデータセット
    残り1つはTwitterから収集されたデータセット
- LSTMのモデルはTwitterで弱い傾向
  - 構文があまり厳密でないため，時系列に沿って処理してもそれだけでは
    あまりうまくいかない

Ablation study

	Models	LAPTOP		REST		TWITTER	
		ACC	Macro-F1	ACC	Macro-F1	ACC	Macro-F1
	SVM	70.49♮	-	80.16♮	-	63.40*	63.30*
	CNN-ASP	72.46	65.31	77.82	65.11	73.27	71.77
	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
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
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**†,‡

LF：Residual　AS：Highway　w/o context：ResidualやHighway結合無し

- Residual/Highway結合は重要
  - ただしTwitterではあまり影響がない．構文が厳密でない分LSTMの情報が重要ではないため．
- 相対位置によるスケーリングがない（表中w/o position）と全データセットを通じて少し落ちる
- w/o transformationはLSTM＋スケーリング＋CNNのみのモデルだがこれだけでもある程度のスコアは出る

CPT-alternatives

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
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**†,‡

- LSTM-ATT-CNNはCPTを完全にATTに変えたモデル
  - 大きくスコアは落ちる
- LSTM-FC-CNNはアテンションによるターゲットフレーズの集約を外して代わりにフレーズ内の平均化にしたモデル
  - ターゲットがフレーズではなく単語の場合はTNetと同一
  - スコアは少し落ちる

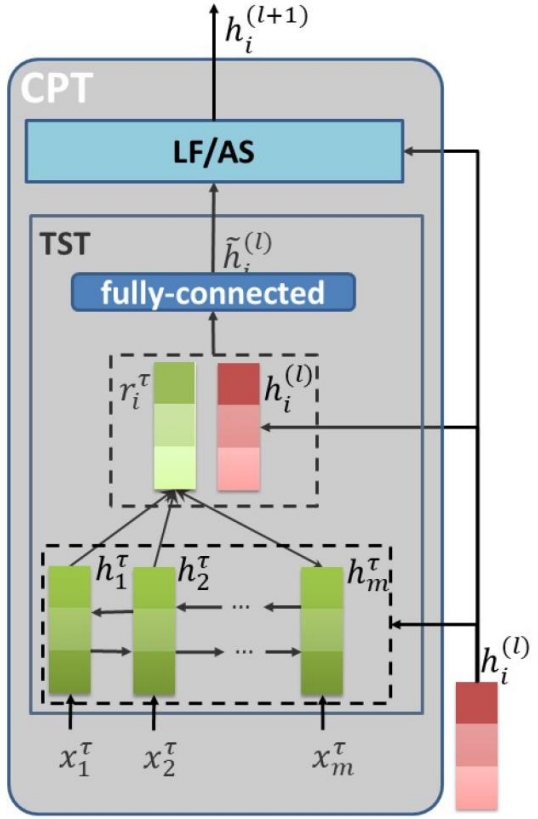


Figure 2: Details of a CPT module

CPTのスタック数

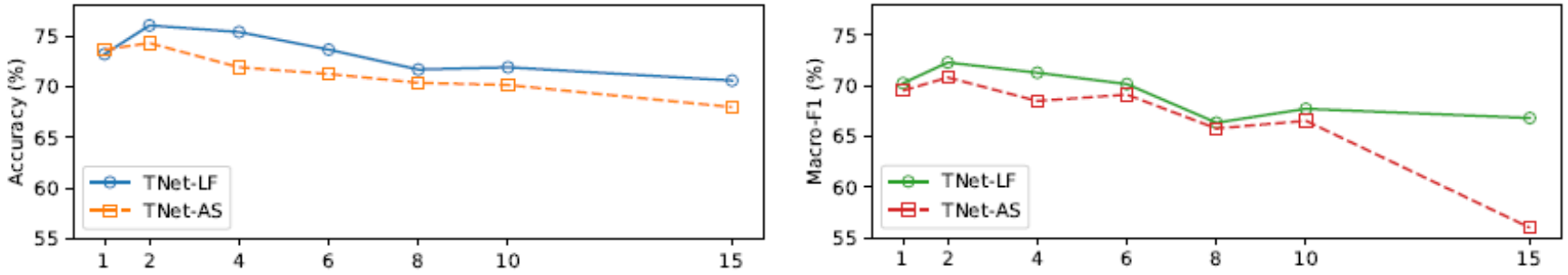


Figure 3: Effect of L.

- 2スタックが最高スコア，それ以上増やすと落ちる
  - CPTはターゲットフレーズ以外の周りが見えないので，そこまで層数がなくとも表現しきることが出来ると考えれば妥当？

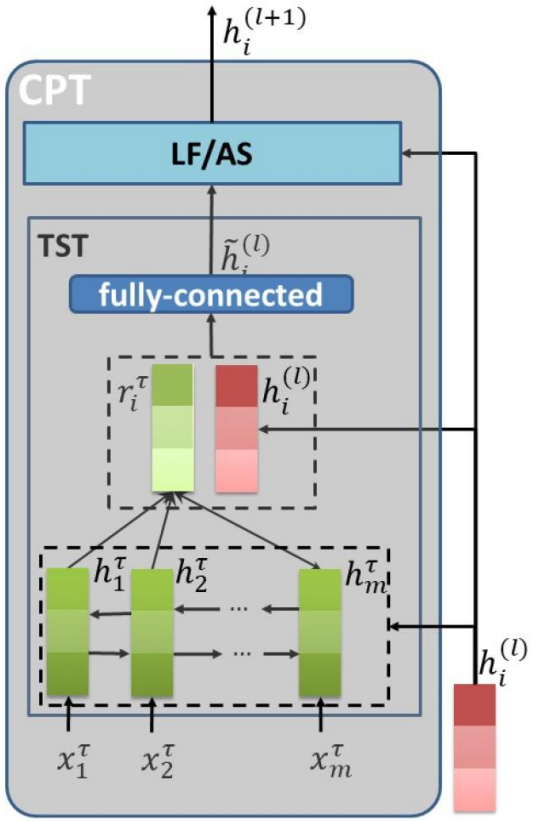


Figure 2: Details of a CPT module

サンプル事例

Sentence	BILSTM-ATT-G	RAM	TNet-LF	TNet-AS
1. Air has higher **[resolution]**$_P$ but the **[fonts]**$_N$ are small .	($N^{\times}$, N)	($N^{\times}$, 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% .	$N^{\times}$	$N^{\times}$	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, $O^{\times}$, $O^{\times}$, P)	(P, P, $O^{\times}$, P)	(P, P, P, P)	(P, P, P, P)
5. **[startup times]**$_N$ are incredibly long : over two minutes .	$P^{\times}$	$P^{\times}$	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 .	$P^{\times}$	$P^{\times}$	$P^{\times}$	$P^{\times}$

- 色がついている3-gramはmax-poolingで採用されたフィルタが最も多かった位置（CNNのwindow sizeが3のため．）

Transformation Networks for Target-Oriented　Sentiment Classification

- 対象タスク：フレーズレベルのSentiment classification
  - 「great **food** but the **service** was dreadful!」
    ➡ food:Positive　service:Negative
- 新規性：ターゲットフレーズごとに表現を特化させて処理
  - 途中の層からターゲットフレーズの表現ベクトルを追加で毎層入力
    - アテンションベースの構造
    - 割とシンプルに見える
- 3つのベンチマークでSotA
  - 2つはSemEval ABSAのドメイン違いのデータセット
    残り1つはTwitterから収集されたデータセット

感想

- かなり単純な工夫で性能が上がっている
- 重み付けが非対称じゃないのが気になる
  - 近接する単語についての仮説はまだまだうまく出来そう
  - 結局相対的な位置情報は何らかの形で必要？
- 構文木を使ったモデルはAdaRNNより強いものがありそうでは？
  - 結果も構文が一番効果の薄そうなTwitterだけなのはどうなのか？
- CPT内では周りが見えていないので，スタック数を増やすなら{CPTxL→Conv2D}のセットで増やすべき？
- コードは公開されているといいながら肝心のmain.pyが非公開
  - 論文中に載っているURLのGitHub上のissueで自分のメールアドレスを晒すと送られてくるようになっている模様
  - main.pyの使い方とデータセットをまとめたものは公開されている