Target Sentiment Analysis: Extraction, Classification, and Sentiment-Aware Embeddings

BING Lidong (邴立东)
http://www.cs.cmu.edu/~lbing/

R&D Center Singapore
Machine Intelligence Technology
Alibaba DAMO Academy

Define Smarter Tomorrow.

ICDM-SENTIRE 2018 Nov. Singapore

Alibaba DAMO Academy

Rooted in Science, Innovate for Applications

"Must outlive Alibaba", "Serve at least 2 billion people worldwide", "Future-oriented and use technology to solve the challenges of the future".

5 Research Areas | 14 Laboratories

- **Machine Intelligence Technology**
  - Speech Lab
  - Language Technology Lab
  - Vision Lab
  - Decision Intelligence Lab
  - City Brain Lab
- **Data Computing**
  - Computing Technology Lab
  - Data Analytics & Intelligence Lab
  - Database & Storage Lab
- **Robotics**
  - Intelligence Transportation Lab
- **Financial Technology**
  - Financial Intelligence Lab
  - Blockchain Lab
  - Biometrics Lab
- **X Laboratory**
  - Quantum Lab
  - AI Labs

Machine Intelligence Technology at DAMO

Hundreds of Researchers and Engineers
in Hangzhou, Beijing, Seattle, Silicon Valley and Singapore

Speech Processing

- Speech Recognition
- Speech Synthesis
- Voice Biometrics
- Human-Machine Interaction

Natural Language Processing

- Semantic Analysis
- Sentiment Analysis
- Text Classification
- Question and Answering, Chatbot
- Machine Translation

Image/Video Analytics

- Product Identity & Search
- Face Recognition
- Object Recognition
- Scene Recognition
- Video Search

Optimization & Decision Making

- Predictive Inventory Optimization
- Delivery Assignment Optimization
- Manufacturing Scheduling
- Predictive Maintenance

NLP R&D at Alibaba

NLP research has made great progress from using complex sets of human rules, statistical natural language processing techniques to deep learning nowadays

Missions of Alibaba's NLP R&D:

1. Support all the demands of NLP techniques and applications in Alibaba's eco-system (new-retail, finance, logistics, entertainment etc.)
2. Enable Alibaba's business partners with NLP solutions
3. Advance the State-of-the-Art NLP research with colleagues from both academia and industries

Alibaba-DAMO-NLP: 100 employees (e.g., former tenured Professors and senior researchers) in 6 locations all over the world.

R&D Center Singapore

An international R&D team with the focus on developing cutting edge speech and language processing technologies, including **ASR, TTS, NLP, and MT**.

Paying special attention to the areas of **multilingual speech and language processing**, including:

- Speech recognition and synthesis of multiple languages
- NLP technology for multiple languages
- Machine translation systems for Southeast Asian languages

AliNLP

AliNLP is a large-scale NLP platform for the entire Alibaba Eco-system. The platform covers major aspects of NLP such as data collecting/processing techniques and multilingual algorithms for lexical, syntactic, semantic, document analysis, and distributed representation of text

Used in 350+ business scenarios (Oct, 2018) with more than 1000Billion+ API calls per day.

Some key characteristics:

- Utilizing behavior data instead of demanding human annotations for NLP algorithms
- Utilizing multiple correlated tasks for improving effectiveness of individual tasks of the complex Alibaba eco-system

AliNLP

NLP Applications

- Search
- Recommendation
- Intelligent HCI
- Translation
- Advertising
- Opinion Monitoring
- Logistics
- Finance
- Customer Service

NLP Techniques

- Content Search
- Content Recommendation
- Comment Analysis
- Question Answering
- Text Understanding
- Anti-Spam
- Social Media Analysis
- Dialog
- Sentiment Analysis
- Language Generation
- Discourse Analysis
- Image to Text
- Address Parsing
- Language Identification
- Knowledge Reasoning
- Topic Detection
- Product Understanding
- Intention Detection

NLP Foundations

Lexical Analysis

- Tokenization
- POS
- NER
- Spelling Check

Syntactical Analysis

- Constituency Parsing
- Dependency Parsing
- Semantic Representation

Semantic Analysis

- WSD
- SRL
- Topic Model
- Behavior Model

Text Analysis

- Clustering
- Classification
- Event Mining
- Hierarchical Clustering
- Intention Classification

Deep Learning

- Word2Vec
- Graph2Vec
- LSTM/ C-DSSM

NLP Data

- Language Dictionary
- Token & POS Corpus
- Entity Corpus
- Treebank (CN, EN…)
- Sentiment Corpus
- News Category Corpus
- Word Relation Graph
- Knowledge Graph

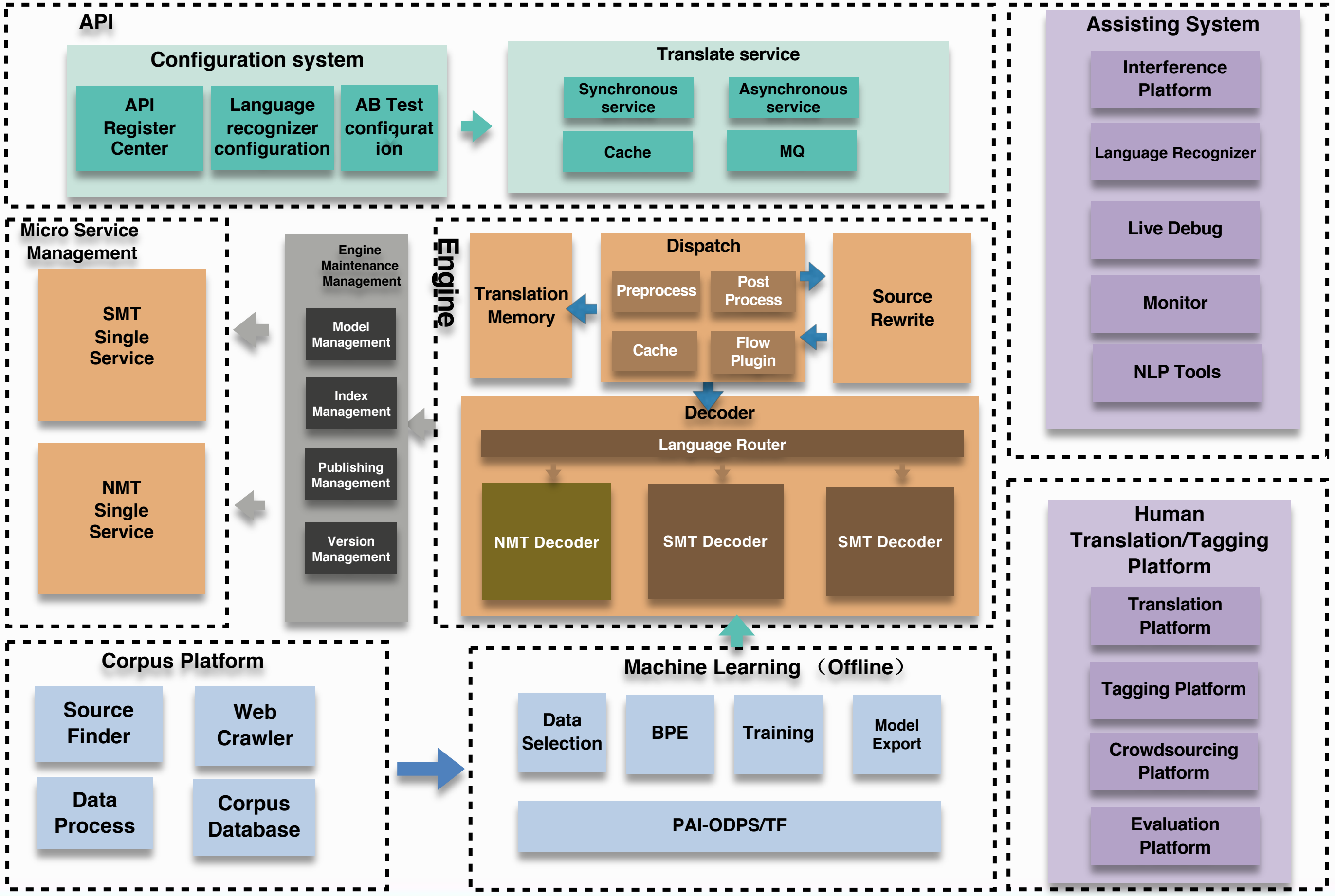
NLP Engineering

- Visualization
- Requirements Management
- User Management
- Monitoring
- System Maintenance
- Annotation Center
- Training Center
- Evaluation Center

Machine Translation at Alibaba

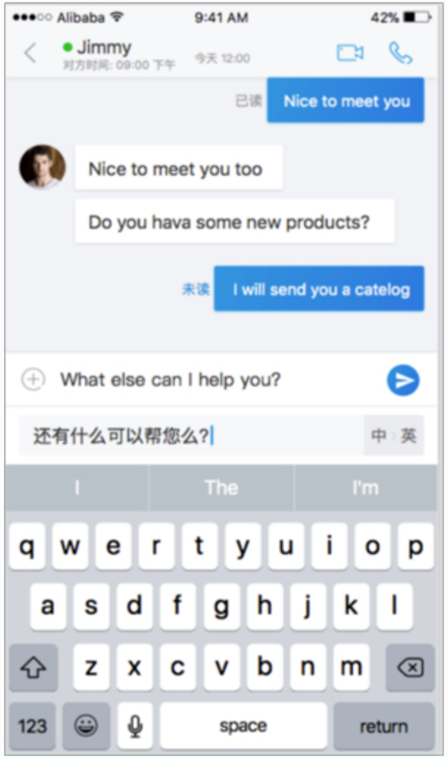
2017- 2018 :

- Support AliExpress, Alibaba.com and Lazada. Processing 250 billion requests in the whole year (60% increase)
- Translating 20 trillion words in the whole year ($2 billion if using Google)
- In WMT'18 got No. 1 in 5 MT tasks for automatic evaluation



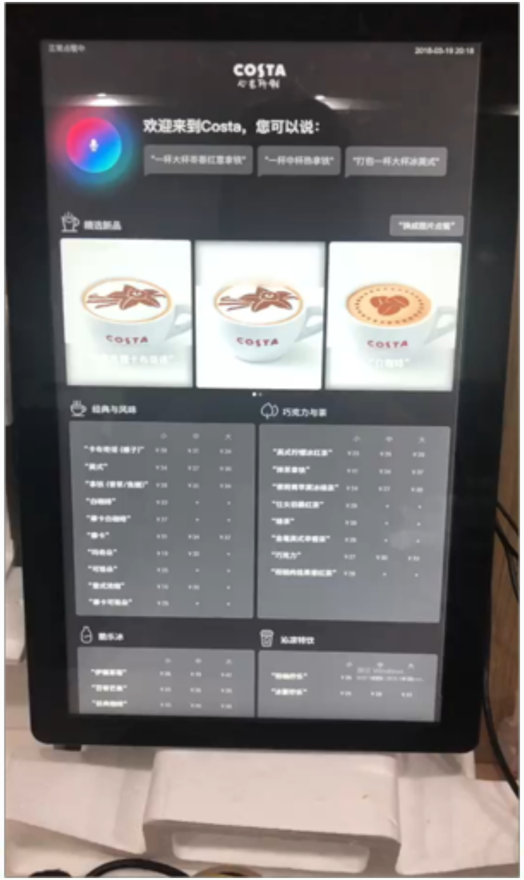
Machine Translation at Alibaba

Real-time Machine Translation that supports the instant communication Between wholesale buyer and seller.



Voice-enabled Ticket Machine at Shanghai Subway (video)

Voice-enabled Coffee-Order Machine at COSTA (video)



Target Sentiment Analysis: Extraction, Classification, and Sentiment-Aware Embeddings

SHI Bei（石贝）

LI Xin（李昕）

CHEN Peng（陈鹏）

Target/Aspect Oriented Sentiment Analysis

- Sentiment classification at both the document and sentence (or clause) levels are useful, but they do not find what people liked and disliked.

- We need to go to the entity and aspect levels, or target level.

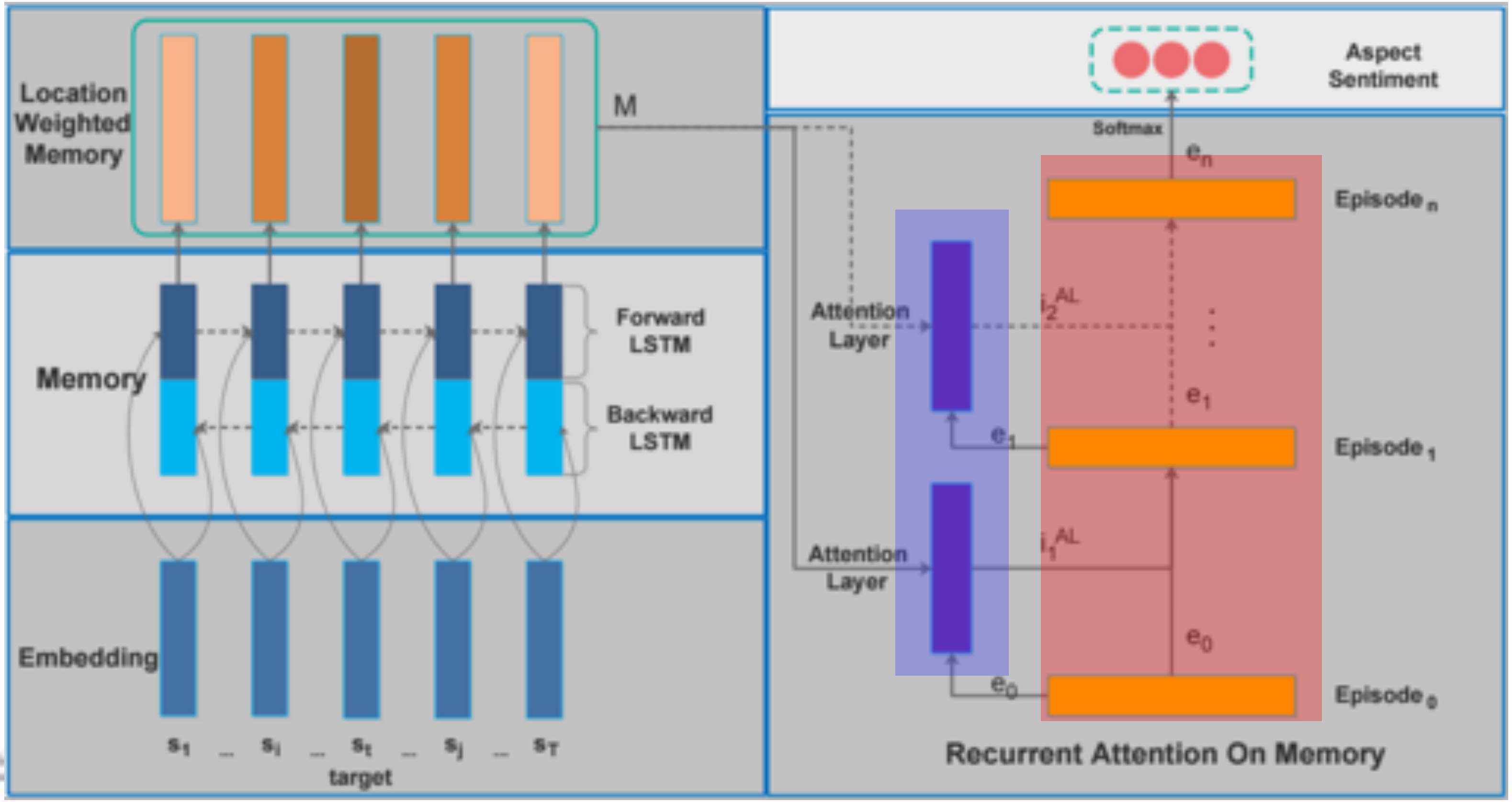
- Problems (E.g. "*Apple is doing very well in this poor economy*.")
  - Target extraction: identify the mentioned sentiment target in a sentence.
    - E.g. "*Apple*" *and* "*economy*"
  - Sentiment prediction: predict the sentiment polarity over the target.
    - E.g. Positive on "*Apple*", but negative on "*economy*"

RAM: Recurrent Attention Memory Network for Aspect Sentiment Prediction [EMNLP 2017]

- Task: predict sentiment polarity over an aspect
  - E.g. predict sentiment over “*battery*” in “The *battery* of the laptop lasts quite long”.
  - A classification problem, given a target and its sentence.

- Motivation:
  - Single attention is usually not enough to capture complicated features, such as *transitive sentences, and comparative sentence*
  - For using multiple attention, the main issue is *how to make them attend different information, and how to combine the attended features.*

RAM Model

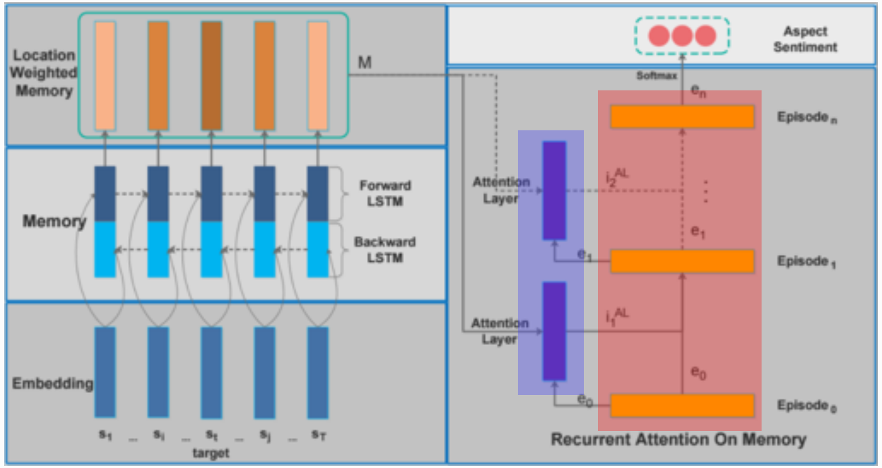


Location Weighted Memory
M
Memory
Forward LSTM
Backward LSTM
Embedding
s1 ... si ... st ... sj ... sT
target
Aspect Sentiment
Softmax
en
Episode n
Attention Layer
i2AL
e1
e1
Episode 1
Attention Layer
i1AL
e0
e0
Episode 0
Recurrent Attention On Memory

RAM Model

- Episode update, a GRU

$$r = \sigma(W_r i_t^{AL} + U_r e_{t-1})$$
$$z = \sigma(W_z i_t^{AL} + U_z e_{t-1})$$
$$\tilde{e}_t = \tanh(W_x i_t^{AL} + W_g(r \odot e_{t-1}))$$
$$e_t = (1 - z) \odot e_{t-1} + z \odot \tilde{e}_t$$



- Attention computation

$$i_t^{AL} = \sum_{j=1}^{T} \alpha_j^t m_j \qquad \alpha_j^t = \frac{\exp(g_j^t)}{\sum_k \exp(g_k^t)} \qquad g_j^t = W_t^{AL}(m_j, e_{t-1}[, v_\tau]) + b_t^{AL}$$

RAM Case Studies

✓ **<u>Using multiple attentions</u>**

INPUT: "Supplied software: the software comes with this machine is greatly welcomed compared to what windows comes with."

TARGET: **windows**

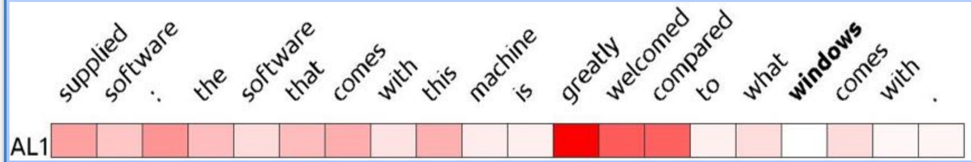
◆ Two attentions

➢ Firstly attend "welcomed", and then "compared"

➢ Combine them non-linearly, and generate a negative sentiment

◆ One attention

➢ More weight on "greatly", make a wrong prediction

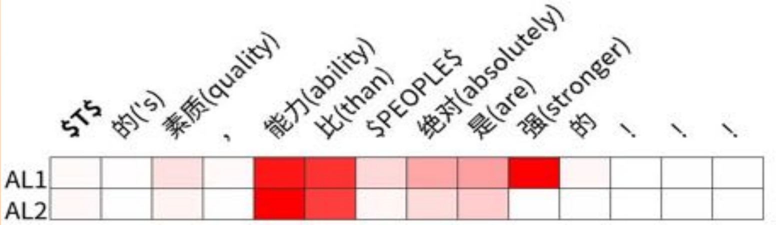


✓ **<u>Multiple target in one sentence</u>**

INPUT: "甲的素质，能力比乙绝对是强的!!!"

◆ For target "**甲**"

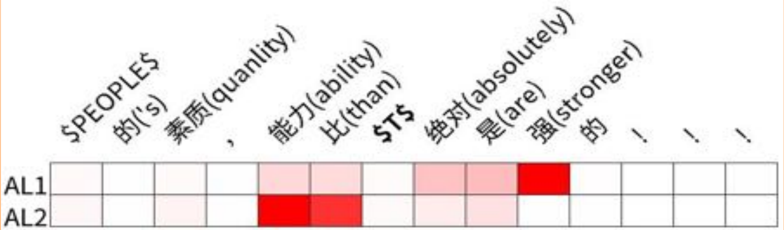
➢ Predict a positive by attending "ability", "stronger"



◆ For target "**乙**"

➢ Attend "stronger" after T, then "than" before T

➢ Inverse sentiment of "stronger" with GRU



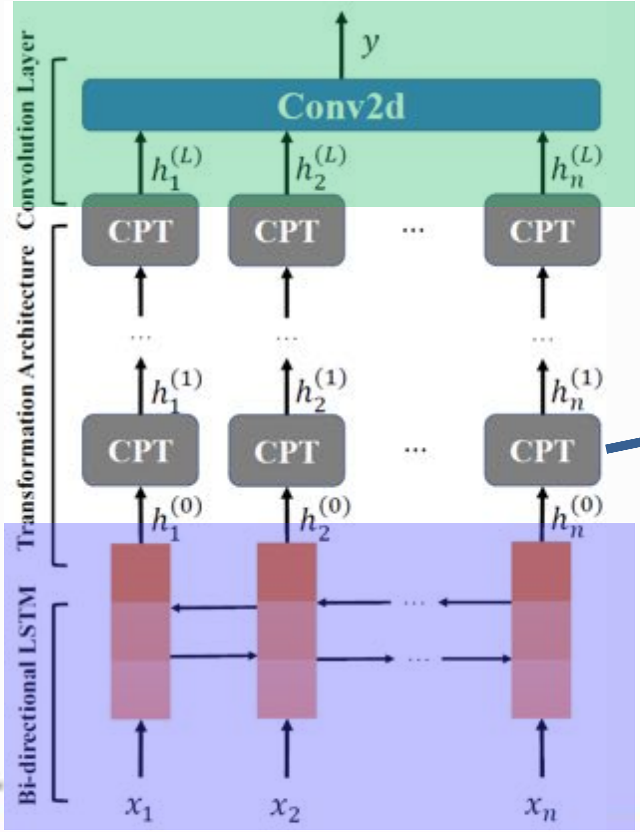
TNet: Transformation Networks for Target-Oriented Sentiment Classification [ACL 2018]

- Task: predict sentiment polarity over an aspect

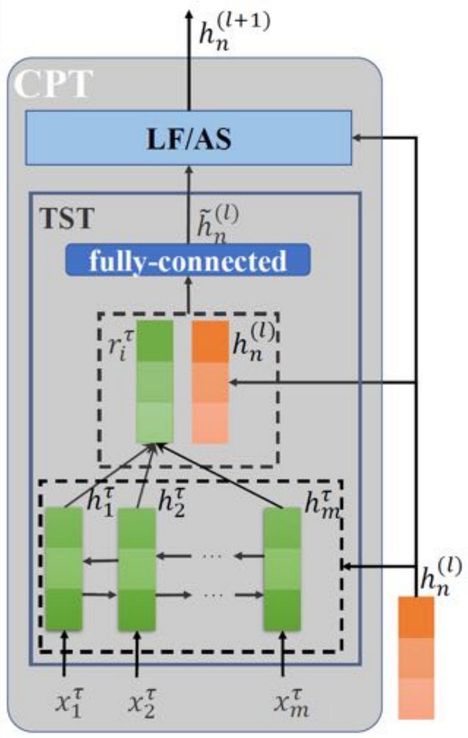
- Motivation
  - Attention usually attends irrelevant information
  - Is there an alternative way to keep its advantage but overcome the limitation?

- Our approach
  - Perform aspect specific transformation on hidden states from RNN
  - Apply highway or residual like method to keep the context information of the original hidden state
  - Using a CNN layer to extract n-gram features

TNet Model



Convolution Layer
Transformation Architecture
Bi-directional LSTM
y
Conv2d
$h_1^{(L)}$ $h_2^{(L)}$ $h_n^{(L)}$
CPT CPT ... CPT
$h_1^{(1)}$ $h_2^{(1)}$ $h_n^{(1)}$
CPT CPT ... CPT
$h_1^{(0)}$ $h_2^{(0)}$ $h_n^{(0)}$
x_1 x_2 x_n



$h_n^{(l+1)}$
CPT
LF/AS
TST
$\tilde{h}_n^{(l)}$
fully-connected
r_i^τ $h_n^{(l)}$
h_1^τ h_2^τ h_m^τ
$h_n^{(l)}$
x_1^τ x_2^τ x_m^τ

TST Component

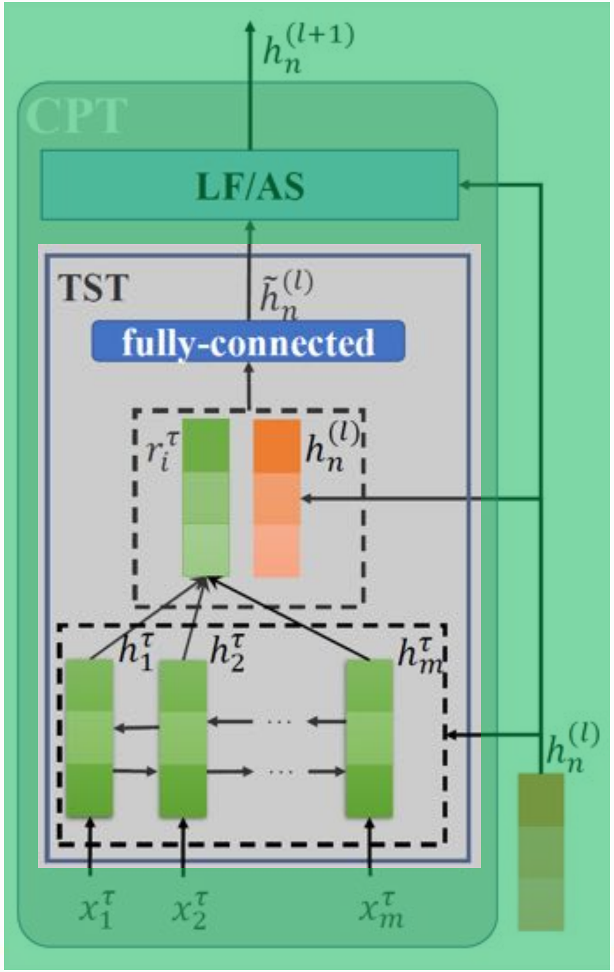
- Incorporating opinion target information into the context word representations
  - Generate the target representation, conditioned on a context word.

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

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

  - A fully-connected layer to obtain the target specific representation of the i-th context word

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



LF/AS Context Preserving

- The context information from the LSTM layer will be lost after TST, so we design context preserving mechanisms to contextualize the generated target-specific representations of context word.

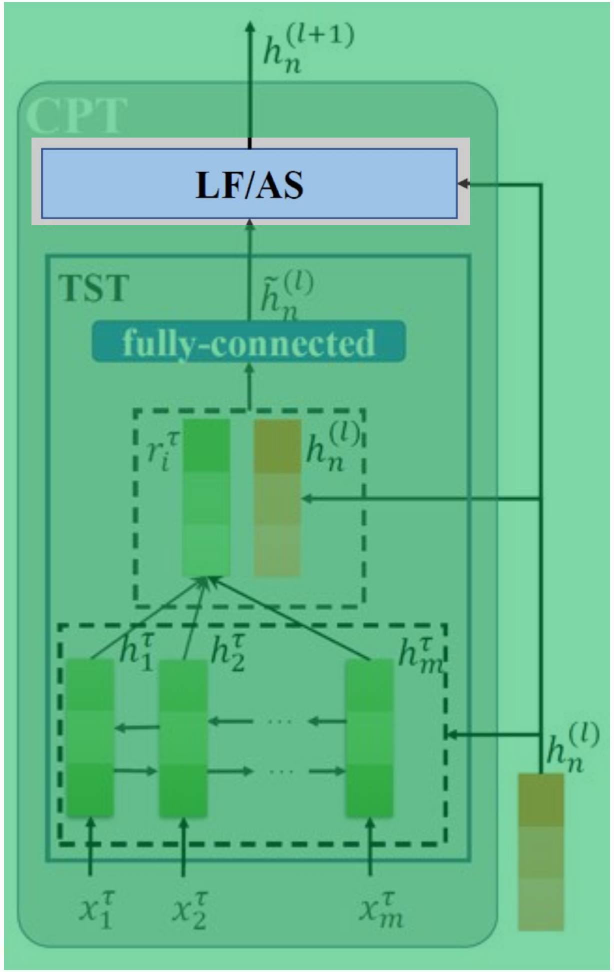
- Lossless Forwarding

$$h_i^{(l+1)} = h_i^{(l)} + \tilde{h}_i^{(l)}, i \in [1, n], l \in [0, L]$$

- Adaptive Scaling

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

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



Result Comparisons

	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[♮]	-	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 performs well for different kinds of UGC, such as product reviews and tweets.

- TST captures the correlation between context word and aspect term
- CNN-based feature extractor can extract accurate features

Case Study

Sentence	RAM	TNet
1. Air has higher [resolution]$_P$ but the [fonts]$_N$ are small .	(N✗, N)	(P, N)
2. Great [food]$_P$ but the [service]$_N$ is dreadful .	(P, N)	(P, N)
3. Sure it ' s not light and slim but the [features]$_P$ make up for it 100% .	N✗	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, P, O✗, P)	(P, P, P, P)
5. [startup times]$_N$ are incredibly long : over two minutes .	P✗	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)
7. The [staff]$_N$ should be a bit more friendly .	P✗	P✗

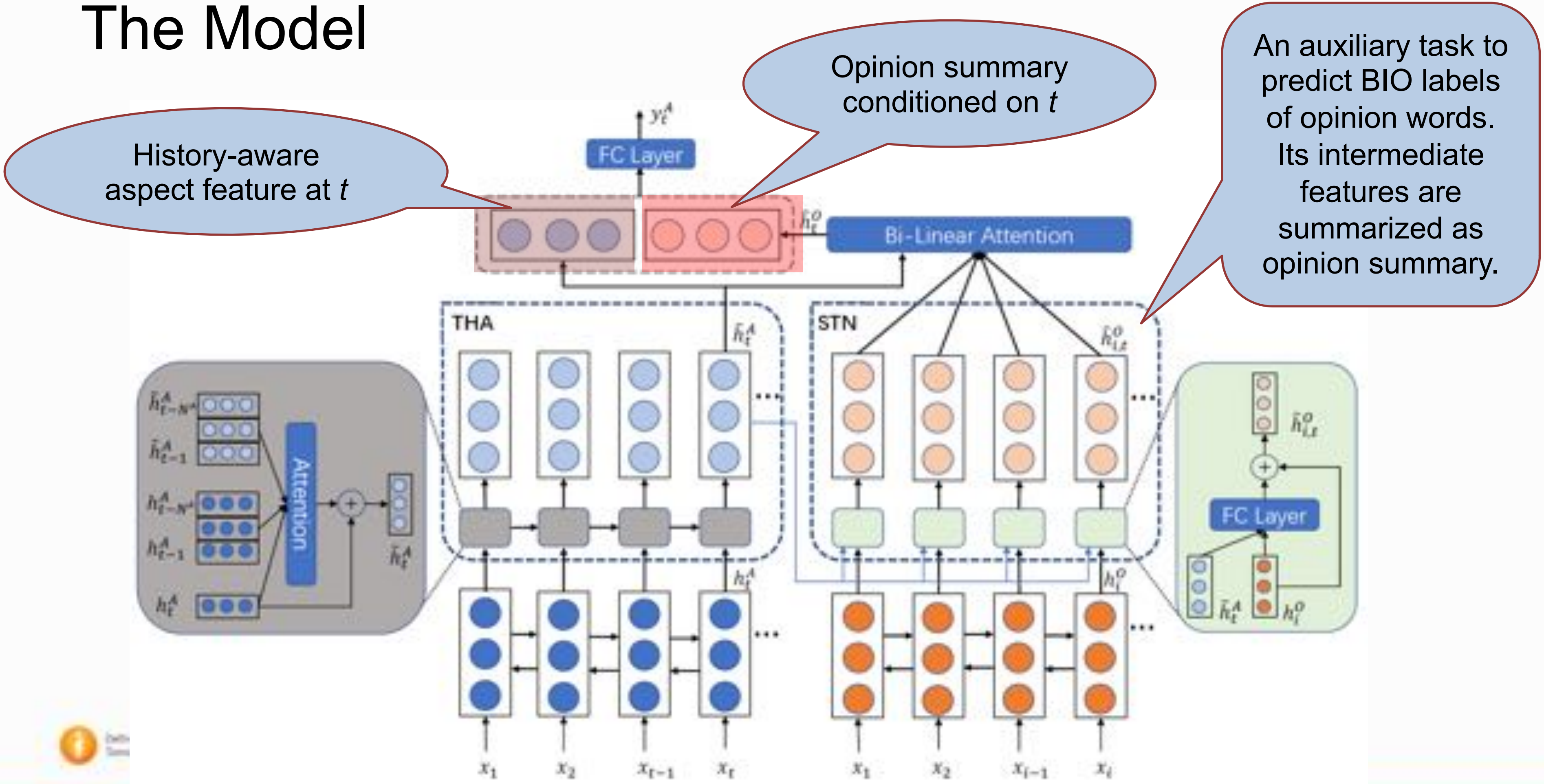
[An aspect is underlined with a particular color, and its corresponding most informative n-gram feature captured by TNet is in the same color]

Aspect Term Extraction with History Attention and Selective Transformation [IJCAI 2018]

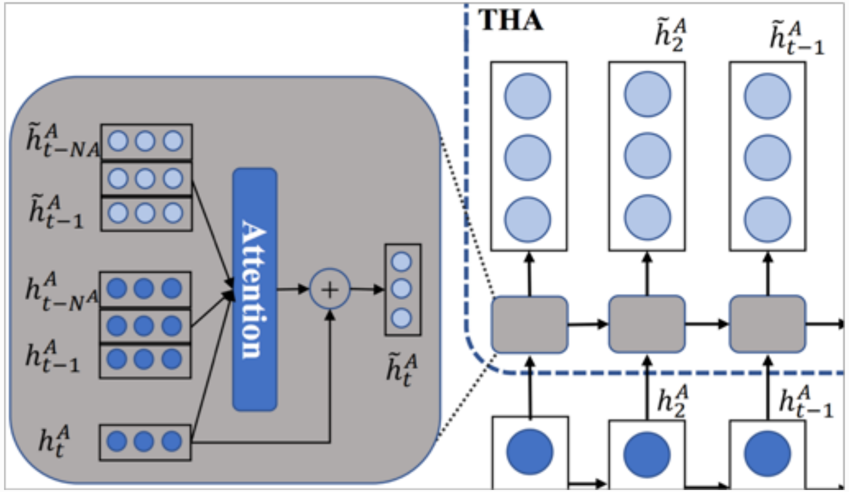
- Task: extract the target that *carrying* sentiment in a sentence
  - E.g., "I love the *operating system* and *preloaded software*"
  - A token level sequence labeling problem

- Intuition
  - Aspect terms should co-occur with opinion words, according to the task definition of aspect sentiment analysis
  - Introduce an auxiliary opinion word detection task to improve the performance of aspect extraction
  - Model the coordinate structure, e.g. "We love the *food, drinks, and atmosphere*!"

The Model


History-aware aspect feature at t
Opinion summary conditioned on t
An auxiliary task to predict BIO labels of opinion words. Its intermediate features are summarized as opinion summary.
FC Layer
Bi-Linear Attention
THA
STN
Attention
FC Layer

The Model: THA

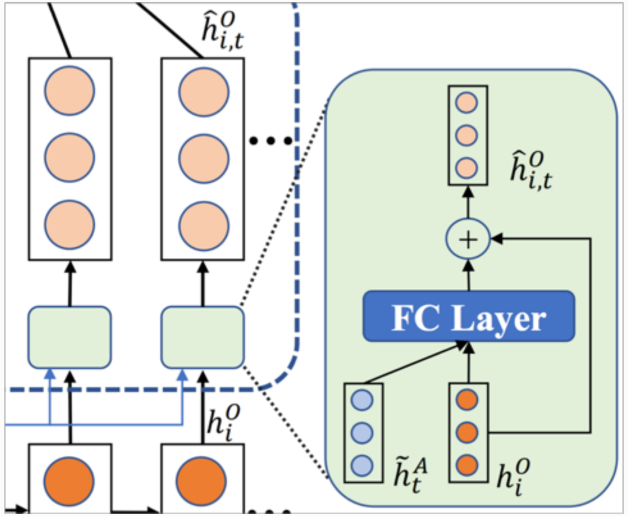


- *Truncated history-attention* (THA): explicitly exploit the relation between the previous predictions and the current prediction in RNN.
  - Reduce error in predicting the current label by considering the B-I-O definition
  - Improve the prediction accuracy for multiple aspects in one coordinate structure
- History-aware aspect representations: $\tilde{h}_t^A = h_t^A + \mathrm{ReLU}(\hat{h}_t^A)$
- Aspect history: $\hat{h}_t^A = \sum_{i=t-N^A}^{t-1} s_i^t \times \tilde{h}_i^A$
- Importance score for each

$$a_i^t = \mathbf{v}^\top \tanh(\mathbf{W}_1 h_i^A + \mathbf{W}_2 h_t^A + \mathbf{W}_3 \tilde{h}_i^A) \qquad s_i^t = \mathrm{Softmax}(a_i^t)$$

The Model: STN

- The auxiliary task
  - Predict BIO labels for opinion words to explore the cooccurrence of aspect terms and opinion words.
  - The aim is to distill the intermediate features as opinion summary conditioned on *t*



- *Selective transformation network* (STN) to *highlight opinion features* with respect to the aspect candidate at *t* so as to *suppress the noise*.
  - For “the fish is unquestionably fresh”, opinion feature of “fresh” is useful for predicting “fish” as an aspect term.

The Model: details

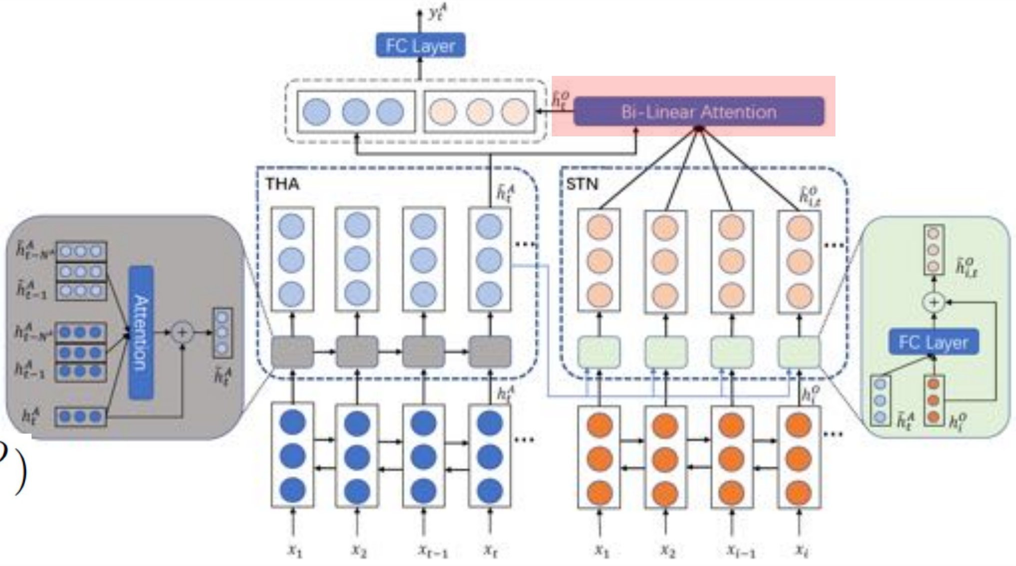
- New opinion representation conditioned on time *t*

$$\hat{h}^O_{i,t} = h^O_i + \mathrm{ReLU}(\mathbf{W}_4 \tilde{h}^A_t + \mathbf{W}_5 h^O_i)$$

- Distilled opinion summary

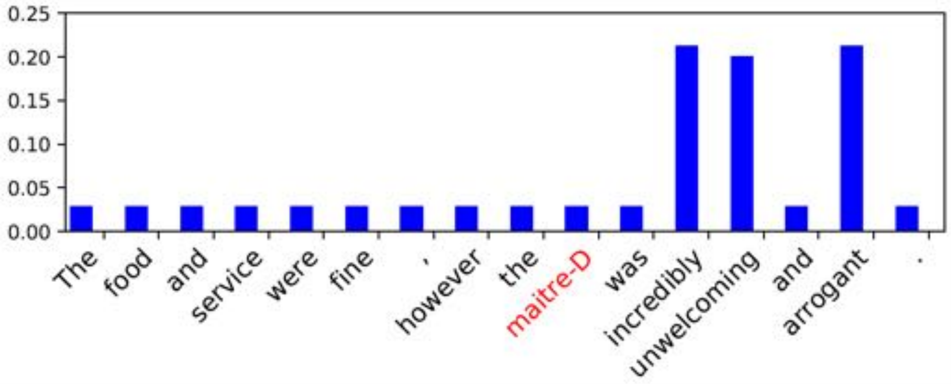
$$\hat{h}^O_t = \sum_{i=1}^{T} w_{i,t} \times \hat{h}^O_{i,t}$$

$$w_{i,t} = \mathrm{Softmax}(\tanh(\tilde{h}^A_t \mathbf{W}_{bi} \hat{h}^O_{i,t} + \mathbf{b}_{bi}))$$

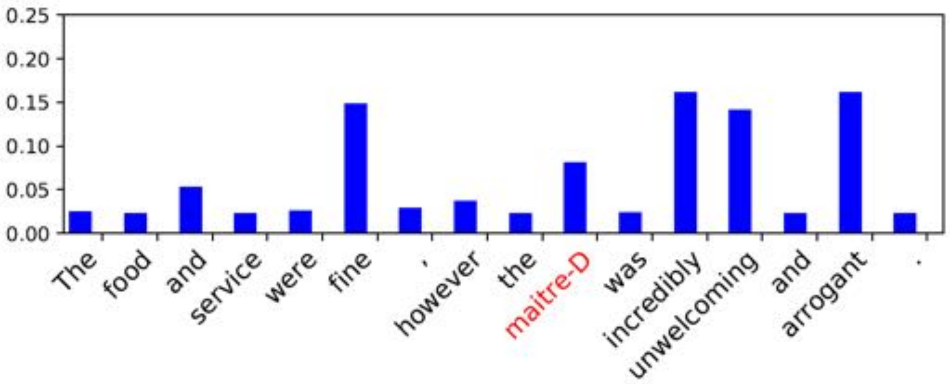


Case Study: attending better opinion words

- With STN



- Without STN



Case Study: extraction results

Input sentences	Output of LSTM	Output of our model
1. *the device speaks about it self*	*device*	**NONE**
2. *Great survice !*	**NONE**	*survice*
3. *Apple is unmatched in product quality, aesthetics, craftmanship, and custormer service*	*quality, aesthetics, custormer service*	*product quality, aesthetics, craftmanship, custormer service*
4. *I am pleased with the fast log on, speedy WiFi connection and the long battery life*	*WiFi connection, battery life*	*log on, WiFi connection, battery life*
5. *Also, I personally wasn't a fan of the portobello and asparagus mole*	*asparagus mole*	*portobello and asparagus mole*

A Unified Model for Opinion Target Extraction and Target Sentiment Prediction [AAAI 2019]

- Task: extract the target that carrying sentiment in a sentence, and predict the sentiment polarity
  - E.g., "I love the [*operating system*]$_{\mathbf{POS}}$, but the [*preloaded software*]$_{\mathbf{NEG}}$ is bad."

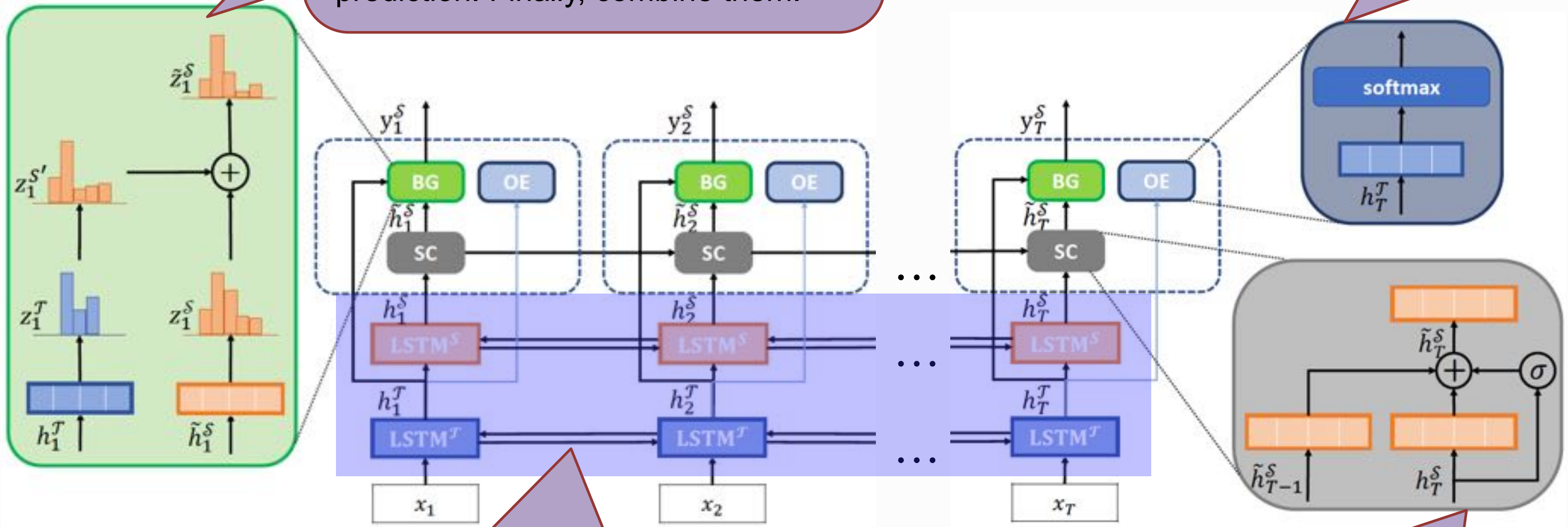
Input	The	AMD	Turin	Processor	seems	to	always	perform	much	better	than	Intel
Joint	O	B	I	E	O	O	O	O	O	O	O	S
	O	POS	POS	POS	O	O	O	O	O	O	O	NEG
Unified	O	B-POS	I-POS	E-POS	O	O	O	O	O	O	O	S-NEG

  - Sequence labeling problem with unified tagging scheme:
    - B-POS,I-POS,E-POS,S-POS; B-NEG,I-NEG,E-NEG,S-NEG; …
- Motivation: if two sub-tasks have strong couplings, a more integrated model is usually more effective than a pipeline.
- Intuition
  - The unified tagging and BIO tagging have the same boundary
  - The consistency of individual words' sentiment within the same target mention

The Model

1. Two stacked LSTM networks. The lower one predicts BIO tags, the upper one predicts the unified tags. The upper takes h_i^T from the lower as input.

2. Combine the predictions of the two networks. First, predict the BIO and unified tags with fully-connected layer. Then, transform the BIO predictions into the unified tag prediction. Finally, combine them.



3. Output *tilde h* that combines the current feature and the previous feature to maintain sentiment consistency in the same target, with a gate function.

4. To enhance the target boundary accuracy, we predict if there are opinion words appearing in the surrounding context of the current word.

The Model: BG component

- Transform the BIO predictions (softly)

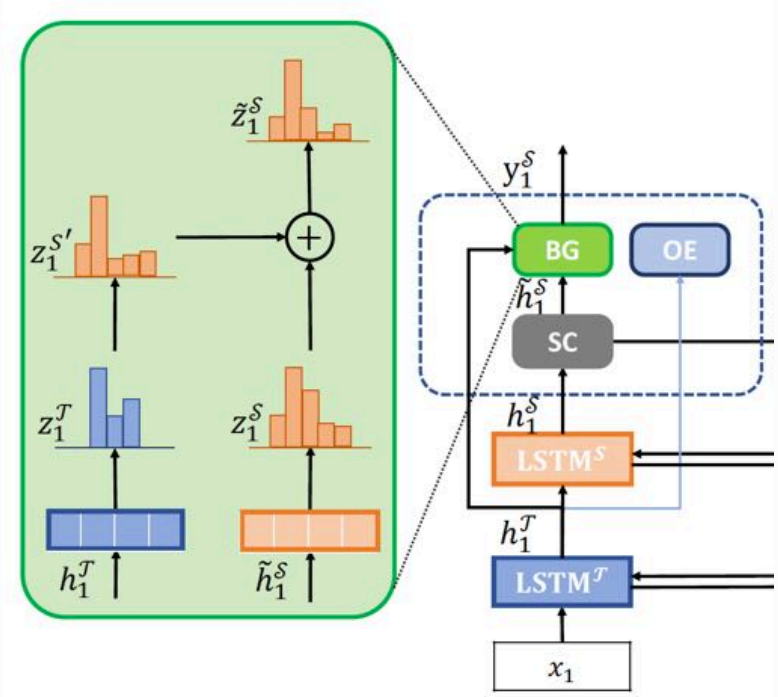
$$z_t^{\mathcal{S}'} = (\mathbf{W}^{tr})^\top z_t^{\mathcal{T}}$$

- Merge the predictions

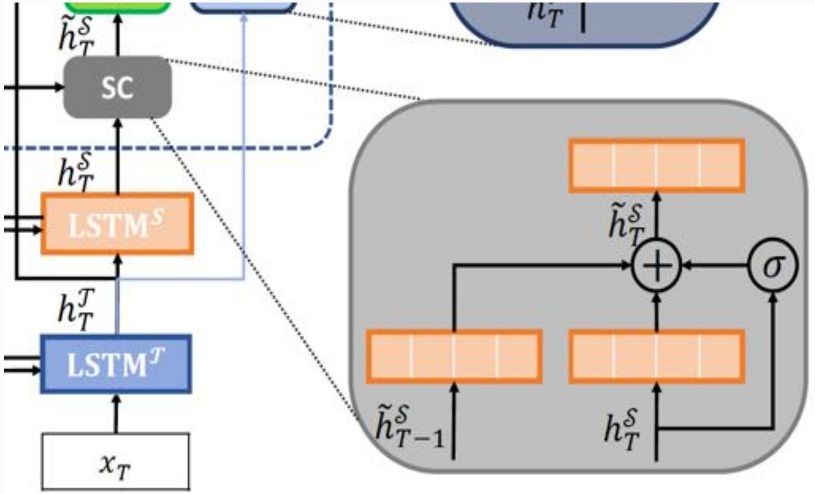
$$\tilde{z}_t^{\mathcal{S}} = \alpha_t z_t^{\mathcal{S}'} + (1 - \alpha_t) z_t^{\mathcal{S}}$$

alpha is derived based on the confidence of $z_t^{\mathcal{T}}$

- for confident boundary prediction, larger alpha
- otherwise, smaller alpha



The Model: SC component



- In one multi-word target, the sentiment polarities of words should be consistent.

- Use a gate to merge the features from the current and the previous time steps.

$$\tilde{h}_t^{\mathcal{S}} = g_t \odot h_t^{\mathcal{S}} + (1 - g_t) \odot \tilde{h}_{t-1}^{\mathcal{S}}$$

$$g_t = \sigma(\mathbf{W}^g h_t^{\mathcal{S}} + \mathbf{b}^g)$$

Result Comparisons

- The *Base model* (the stacked LSTMs) always outperforms *LSTM-unified*.
- Adding **BG** component (*Base model + BG*), the performances are improved a lot.
- Adding **SC or OE** into the "*Base model + BG*" does not bring in too much gains.
- But putting them together, i.e., "*Full model*", leads to the new state-of-the-art.

Model	$\mathbb{D}_L$			$\mathbb{D}_R$			$\mathbb{D}_T$		
	P	R	F1	P	R	F1	P	R	F1
CRF-joint	57.38	35.76	44.06	60.00	48.57	53.68	43.09	24.67	31.35
CRF-unified	59.27	41.86	49.06	63.39	57.74	60.43	48.35	19.64	27.86
NN-CRF-joint	55.64	34.48	45.49	61.56	50.00	55.18	44.62	35.84	39.67
NN-CRF-unified	58.72	45.96	51.56	62.61	60.53	61.56	46.32	32.84	38.36
CRF-pipeline	59.69	47.54	52.93	52.28	51.01	51.64	42.97	25.21	31.73
NN-CRF-pipeline	57.72	49.32	53.19	60.09	61.93	61.00	43.71	37.12	40.06
HAST-TNet	56.42	54.20	55.29	62.18	73.49	67.36	46.30	49.13	47.66
LSTM-unified	57.91	46.21	51.40	62.80	63.49	63.14	51.45	37.62	43.41
LSTM-CRF-1	58.61	50.47	54.24	66.10	66.30	66.20	51.67	44.08	47.52
LSTM-CRF-2	58.66	51.26	54.71	61.56	67.26	64.29	53.74	42.21	47.26
LM-LSTM-CRF	53.31	59.4	56.19	68.46	64.43	66.38	43.52	52.01	47.35
Base model	60.00	46.85	52.61	61.48	66.16	63.73	53.02	41.47	46.50
Base model + **BG**	58.58	50.63	54.31	67.51	66.42	66.96	52.26	43.84	47.66
Base model + **BG** + **SC**	58.95	53.00	55.81	63.95	69.65	66.68	53.12	43.60	47.79
Base model + **BG** + **OE**	63.43	49.53	55.62	62.85	66.77	65.22	53.10	43.50	47.78
Full model	61.27	54.89	**57.90**$^{\natural,\sharp}$	68.64	71.01	**69.80**$^{\natural,\sharp}$	53.08	43.56	**48.01**$^{\sharp}$

Case Study

Input	Base model		Base model + BG		Full model	
	Target	Complete	Target	Complete	Target	Complete
1. And the fact that it comes with an [*i5 processor*]$_{POS}$ definitely speeds things up	*i5 processor*	[*processor*]$_{POS}$ (✗)	*i5 processor*	[*i5 processor*]$_{POS}$	*i5 processor*	[*i5 processor*]$_{POS}$
2. There were small problems with [*mac office*]$_{NEG}$.	*mac office*	[*mac*]$_{NEG}$ (✗)	*mac office*	[*mac office*]$_{NEG}$	*mac office*	[*mac office*]$_{NEG}$
3. The [*teas*]$_{POS}$ are great and all the [*sweets*]$_{POS}$ are homemade	*teas, sweets*	[*teas*]$_{POS}$, [*sweets*]$_{POS}$	*teas, sweets, homemade* (✗)	[*teas*]$_{POS}$, [*sweets*]$_{POS}$, [*homemade*]$_{POS}$ (✗)	*teas, sweets*	[*teas*]$_{POS}$, [*sweets*]$_{POS}$
4. I love the [*form factor*]$_{POS}$	NONE	NONE	NONE	NONE	*form factor*	[*form factor*]$_{POS}$
5. I blame the [*Mac OS*]$_{NEG}$.	*Mac OS*	[*Mac*$_{NEG}$ *OS*$_{NEU}$] (✗)	*Mac OS*	[*Mac*$_{NEG}$ *OS*$_{POS}$] (✗)	*Mac OS*	[*Mac OS*]$_{NEG}$
6. Also, I personally wasn't a fan of the [*portobello and asparagus mole*]$_{NEG}$.	*portobello and asparagus mole*	[*portobello*$_{NEG}$ *and*$_{NEG}$ *asparagus*$_{NEG}$ *mole*$_{NEU}$] (✗)	*portobello and asparagus mole*	[*portobello*$_{NEG}$ *and*$_{NEG}$ *asparagus*$_{NEU}$ *mole*$_{NEU}$] (✗)	*portobello and asparagus mole*	[*portobello and asparagus mole*]$_{NEG}$

- Stacking two LSTMs (*Base model*) may miss target words. E.g. Inputs 1 and 2 are lost.
- *Base model+BG* can solve Inputs 1 and 2, but fail for Inputs 3 and 4 since inaccurate target prediction
- *Base model and Base model +BG* can still predict inconsistent sentiments within the same target, e.g. Inputs 5 and 6.

Learning Domain-Sensitive and Sentiment-Aware Word Embeddings [ACL 2018]

- Task: generate domain-sensitive and sentiment-aware word embeddings

- Sentiment-Aware: Some words, especially sentiment words, have similar syntactic context but opposite sentiment polarity, such as the words “good” and “bad”

- Domain-Sensitive: The polarity of some sentiment words varies according to their domain.
  - E.g. “lightweight” has different polarity for Electronics and Movie

Our DSE Model

- For word *w*, appearing in two domains *p* and *q*:
  - One domain-common vector: U_w^c, two domain-specific vectors: U_w^p and U_w^q
  - A latent variable: $z_w=1$, *w* is common in both *p* and *q*; $z_w=0$, *w* is specific to *p* or *q*
- Context prediction

$$p(w_t|w, z_w = 1) = \frac{\exp(U_w^c \cdot V_{w_t})}{\sum_{w' \in \Lambda} \exp(U_w^c \cdot V_{w'})}$$

$$p(w_t|w, z_w = 0) = \begin{cases} \frac{\exp(U_w^p \cdot V_{w_t})}{\sum_{w' \in \Lambda} \exp(U_w^p \cdot V_{w'})}, \text{if } w \in \mathcal{D}^p \\ \frac{\exp(U_w^q \cdot V_{w_t})}{\sum_{w' \in \Lambda} \exp(U_w^q \cdot V_{w'})}, \text{if } w \in \mathcal{D}^q \end{cases}$$

Our DSE Model

- Sentiment prediction

$$p(y_w = 1 | w, z_w = 1) = \sigma(U_w^c \cdot \mathbf{s})$$

$$p(y_w = 1 | w, z_w = 0) = \begin{cases} \sigma(U_w^p \cdot \mathbf{s}) & \text{if } w \in \mathcal{D}^p \\ \sigma(U_w^q \cdot \mathbf{s}) & \text{if } w \in \mathcal{D}^q \end{cases}$$

- Objective Function

$$\mathcal{L} = \mathcal{L}^p + \mathcal{L}^q$$

$$\mathcal{L}^p = \sum_{r \in \mathcal{D}^p} \sum_{w \in r} \sum_{w_t \in c_w} \log p(w_t | w) + \sum_{r \in \mathcal{D}^p} \sum_{w \in r} \log p(y_w | w)$$

Case Study

<table>
<tr><th>Term</th><th>Domain</th><th>p(z = 1)</th><th>Sample Reviews</th></tr>
<tr><td rowspan="6">lightweight</td><td>B & D</td><td>0.999</td><td rowspan="6">- I find Seth Godin’s books incredibly lightweight. There is really nothing of any substance here. (B)

- I love the fact that it’s small and lightweight and fits into a tiny pocket on my camera case so I never lose track of it. (E)

- These are not ” lightweight” actors. (D)

- This vacuum does a pretty good job. It is lightweight and easy to use. (K)</td></tr>
<tr><td>B & E</td><td>0.404</td></tr>
<tr><td>B & K</td><td>0.241</td></tr>
<tr><td>D & E</td><td>0.380</td></tr>
<tr><td>D & K</td><td>0.013</td></tr>
<tr><td>E & K</td><td>0.696</td></tr>
<tr><td rowspan="3">die</td><td>B & E</td><td>0.435</td><td rowspan="3">- I’m glad Brando lived long enough to get old and fat, and that he didn’t die tragically young like Marilyn, JFK, or Jimi Hendrix. (B)

- Like many others here, my CD-changer died after a couple of weeks and it wouldn’t read any CD. (E)

- I had this toaster for under 3 years when I came home one day and it smoked and died. (K)</td></tr>
<tr><td>B & K</td><td>0.492</td></tr>
<tr><td>E & K</td><td>0.712</td></tr>
</table>

[B: books, D: DVDs, E: electronics items, K: kitchen appliances]

Open Questions

- The current researches may not be practically usable
  - A small number of domains
  - Small training and testing data

- Cross domain and cross lingual problems
  - Thousands of domains
  - Tens of languages

- UGC data is changing very rapidly, looks like this task cannot be completely solved.

达摩院
Alibaba DAMO Academy

Thanks

Define Smarter
Tomorrow.