Aspect Sentiment Triplet Extraction Using Reinforcement Learning

Samson Yu Bai Jian
SUTD, Singapore
samson_yu@sutd.edu.sg

Tapas Nayak
IIT KGP, India
tnk02.05@gmail.com

Navonil Majumder
SUTD, Singapore
navonil_majumder@sutd.edu.sg

Soujanya Poria
SUTD, Singapore
sporia@sutd.edu.sg

ABSTRACT

Aspect Sentiment Triplet Extraction (ASTE) is the task of extracting triplets of aspect terms, their associated sentiments, and the opinion terms that provide evidence for the expressed sentiments. Previous approaches to ASTE usually simultaneously extract all three components or first identify the aspect and opinion terms, then pair them up to predict their sentiment polarities. In this work, we present a novel paradigm, **ASTE-RL**, by regarding the aspect and opinion terms as arguments of the expressed sentiment in a hierarchical reinforcement learning (RL) framework. We first focus on sentiments expressed in a sentence, then identify the target aspect and opinion terms for that sentiment. This takes into account the mutual interactions among the triplet's components while improving exploration and sample efficiency. Furthermore, this hierarchical RL setup enables us to deal with multiple and overlapping triplets. In our experiments, we evaluate our model on existing datasets from laptop and restaurant domains and show that it achieves state-of-the-art performance. The implementation of this work is publicly available at https://github.com/declare-lab/ASTE-RL.

ACM Reference Format:
Samson Yu Bai Jian, Tapas Nayak, Navonil Majumder, and Soujanya Poria. 2021. Aspect Sentiment Triplet Extraction Using Reinforcement Learning. In *Proceedings of the 30th ACM Int'l Conf. on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Aspect-based sentiment analysis (ABSA) or target-based sentiment analysis (TBSA) is an important research area in natural language processing (NLP) [13]. It consists of various fine-grained sentiment analysis tasks [9–11], with the three most fundamental being aspect/target term extraction, opinion term extraction and aspect/target term sentiment classification. Aspect Sentiment Triplet Extraction (ASTE) is a relatively new subtask of ABSA introduced by Li et al. [7], Peng et al. [12]. In ASTE, the task is to extract triplets containing aspect terms, their associated sentiment polarities, and the opinion terms that express those sentiments. A sentence may contain multiple such triplets where the aspect terms or opinion terms across triplets may overlap with each other. We include an example of such triplets in Table 1.

Table 1: An Example of ASTE Triplets Present in a Sentence

Sentence	Appetizers are excellent ; you can make a great (but slightly expensive) meal out of them .
Triplets	[Aspect ; Opinion ; Sentiment] (1) Appetizer ; excellent ; positive (2) meal ; great ; positive (3) meal ; slightly expensive ; negative

Existing methods, such as, CMLA+ [20], RINANTE+ [2], Li-unified-R [7], WhatHowWhy [12], OTE-MTL [25], GTS [23], JET [24], TOP [4] and BMRC [1] are mainly divided into simultaneous and sequential methods. Early works [2, 7, 12, 20, 23, 25] usually employ a two-staged approach where they simultaneously extract aspect terms with sentiments and opinion terms. These triplets are subsequently decoded through triplet classification or pairwise matching. Recent works [1, 4] have shifted towards a more multi-stage, restrictive and sequential process during the extraction stage that can potentially capture more mutual dependencies and correlations among the triplet's components while forgoing the triplet decoding stage.

In this work, we tackle the ASTE task using a novel paradigm **ASTE-RL** where we consider the aspect and opinion terms as arguments of the sentiments expressed in a sentence. Previous approaches usually simultaneously extract all three components or first identify the aspect and opinion terms, then pair them up to predict their sentiment polarities. Unlike previous approaches, we propose a hierarchical reinforcement learning (RL) framework [19] where we first consider the sentiment polarities, then identify their associated opinion and aspect terms using separate RL processes. This process is repeated to extract all triplets present in a sentence. With this hierarchical RL setup, the model handle multiple triplets and overlapping triplets, and model interactions between the three components effectively. Inspired by the recent success of the multi-turn machine reading comprehension (MRC) framework [1, 8], we incorporate ideas to further improve mutual interactions.

2 PROPOSED FRAMEWORK

2.1 Overview

We divide our framework **ASTE-RL** into three components: 1) aspect-oriented sentiment classification, 2) opinion term extraction and 3) aspect term extraction. For the sentiment classification component, the sentiment is expressed towards the aspect term, and has four possible labels: $L^C = \{\text{none}, \text{positive}, \text{negative}, \text{neutral}\}$. Our opinion and aspect extraction components are sequence labeling models with a BIO tagging scheme [15]. With this BIO

CIKM '21, November 1–5, 2021, Virtual Event, Australia.
© 2021 Association for Computing Machinery.
ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

scheme, we have three different labels to tag an input sequence for the opinion/aspect terms: $L^{O,A} = \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$. For a given sentence $X = \{x_1, x_2, ..., x_J\}$ with J tokens, ASTE-RL aims to output a set of labels $\{(Y^O, Y^A, C)^{|K|}\}$ where $|K|$ is the number of labels, $Y^O = \{y_1^O, y_2^O, ..., y_J^O\}$ represents the tagging labels for the opinion term in a predicted triplet, $Y^A = \{y_1^A, y_2^A, ..., y_J^A\}$ represents the tagging labels for the aspect term, and $C = \{c\}$ represents the sentiment polarity.

The three components are structured in a two-level hierarchy [19]. In the higher level, we have the sentiment indicator. During the sequential scan of a sentence, an agent will decide at each position j in a sentence at the token x_j if it has gathered sufficient information to mark the position as indicative of a sentiment that is expressed towards an aspect term. If not, the agent will mark it as none. Otherwise, it will mark it as either positive, negative or neutral. In the latter case, the agent launches two subtasks in the lower level for the opinion and aspect extractions to identify the terms as arguments of the sentiment and engages in sequence labeling. Upon completion, the agent will return to the high-level sentiment indication process and continue the sequential scan of the sentence. This process is well-suited to be formulated as a semi-Markov decision process [18]: 1) a high-level RL process that detects a sentiment indicator in a sentence; 2) two low-level RL processes that identify the opinion and aspect terms separately for the corresponding sentiment.

2.2 Aspect-Oriented Sentiment Classification with High-Level RL

The high-level RL policy π_u aims to detect the aspect-oriented sentiments in a sentence. This can be seen as a RL policy over options, where options are high-level actions [18].

Option: The option $\mathbf{o}_t$ is selected from $L^C = \{\text{none}, \text{positive}, \text{negative}, \text{neutral}\}$ where none indicates no sentiment indicated towards any aspect term.

State: The state $\mathbf{s}_t^u$ at each time step t is represented by: 1) the current hidden state $\mathbf{h}_t$, 2) the current part-of-speech (POS) tag $\mathbf{p}_t$, 3) the sentiment polarity vector $\mathbf{v}_t^c$, and 4) the high-level state for the previous time step $\mathbf{s}_{t-1}^u$. To obtain $\mathbf{p}_t$ for each token in a sentence, we pass the sentence into the spaCy (https://spacy.io/) POS tagger. The sentiment polarity vector $\mathbf{v}_t^c$ is the embedding of the latest option $\mathbf{o}_{t'}$ where $\mathbf{o}_{t'} \neq$ none. Both the POS tag and sentiment embeddings are learned parameters in the model. Hence, the state $\mathbf{s}_t^u$ is formally represented by:

$$\mathbf{s}_t^u = f_s^u(\mathbf{W}_s^u[\mathbf{h}_t;\mathbf{p}_t;\mathbf{v}_t^c;\mathbf{s}_{t-1}^u]), \tag{1}$$

where $f_s^u(\cdot)$ is a non-linear function implemented by a MLP. The hidden state $\mathbf{h}_t$ is obtained from a pre-trained BERT model [3] with Whole Word Masking and fine-tuned on the SQuAD v1.1 training set [14]. Specifically, we first combine the query "*Which tokens indicate sentiments relating pairs of aspect spans and opinion spans?*" and the review sentence X into the BERT tokenizer to get a final input $I = \{[CLS], q_1^u, ..., q_{|q|}^u, [SEP], x_1, x_2, ..., x_J\}$. We then pass this input into the BERT model, and $\mathbf{h}_t$ represents the output vector from the BERT model that corresponds to the token x_t. The initial state $\mathbf{s}_0^u$ is initialized as: $\mathbf{s}_0^u = \mathbf{0}$.

Policy: The stochastic policy for sentiment detection π_u specifies a probability distribution over the options:

$$\mathbf{o}_t \sim \pi_u(\mathbf{o}_t|\mathbf{s}_t^u) = softmax(\mathbf{W}_\pi^u\mathbf{s}_t^u). \tag{2}$$

Reward: At every time step, when $\mathbf{o}_t$ is executed, the **intermediate reward $\mathbf{r}_t^u$** provided by the environment follows this:

$$\mathbf{r}_t^u = \begin{cases} 1, & \text{if } \mathbf{o}_\mathrm{t} \text{ in } X \\ 0, & \text{if } \mathbf{o}_\mathrm{t} = \text{none} \\ -1, & \text{if } \mathbf{o}_\mathrm{t} \text{ not in } X. \end{cases} \tag{3}$$

If a sentiment that is expressed towards an aspect term is detected at a time step (i.e. $\mathbf{o}_t \neq$ none), the agent will launch two subtasks as low-level RL processes. When the subtasks are completed, the agent will return to the high-level RL process. Otherwise, the agent continues its sequential scan of X until the last option $\mathbf{o}_J$ about the last word x_J of X is sampled. When all options are sampled (i.e. at the end of the combined hierarchical RL process), there is a **final reward $\mathbf{r}_{final}^u$** for the high-level process: $\mathbf{r}_{final}^u = F_1(X)$ where F_1 is the harmonic mean of the precision and recall in terms of the sentiment(s) in a sentence X.

2.3 Opinion and Aspect Extractions with Low-Level RL

Every time the high-level policy detects an aspect-oriented sentiment, two low-level policies π_l^O, π_l^A will extract the corresponding opinion and aspect terms respectively and separately for the sentiment. In this subsection, we will generalize the RL elements such that they apply for both low-level RL processes, unless otherwise stated.

Action: The action at every time step t is to assign a tag to the current word. The action $\mathbf{a}_t$ is selected from $L^{O,A} = \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$, following a BIO tagging scheme. The B/I symbols represent the beginning and inside of an opinion/aspect term respectively, while the O symbol represents the unmarked label.

State: Similar to the high-level policy, the state $\mathbf{s}_t^l$ at each time step t is represented by: 1) the current hidden state $\mathbf{h}_t$, 2) the current POS tag $\mathbf{p}_t$, 3) the opinion/aspect tag vector $\mathbf{v}_t^{O,A}$, 4) the low-level state for the previous time step $\mathbf{s}_{t-1}^l$. To enhance the interactions between the sentiment and its associated opinion/aspect terms, we add a context vector $\mathbf{v}_{t'}^{ctx}$ to the state $\mathbf{s}_t^l$ at each time step t, using the sentiment state representation assigned to the latest option $\mathbf{o}_{t'}$:

$$\mathbf{v}_{t'}^{ctx} = g^{ctx}(\mathbf{W}_{v_{t'}}^{ctx}\mathbf{s}_{t'}^u). \tag{4}$$

We also add the output vector from the BERT model for the [CLS] token, while computing the output vectors for the hidden states. Hence, the state $\mathbf{s}_t^l$ is formally represented by:

$$\mathbf{s}_t^l = f_s^l(\mathbf{W}_s^l[\mathbf{h}_t;\mathbf{p}_t;\mathbf{v}_t^{O,A};\mathbf{s}_{t-1}^l;\mathbf{v}_{t'}^{ctx};\mathbf{h}_t^{CLS}]). \tag{5}$$

Note that the representations used to compute the first low-level states for the opinion and aspect extractions are different. The representation used to compute the first low-level state for opinion term extraction $\mathbf{s}_1^{l,O}$ is initialized using $\mathbf{s}_{t'}^u$ as: $\mathbf{s}_0^{l,O} = g^l(\mathbf{W}_{s_0}^l\mathbf{s}_{t'}^u)$. The representation used to compute the first low-level state for aspect term extraction $\mathbf{s}_1^{l,A}$ is initialized using $\mathbf{s}_J^{l,O}$ as: $\mathbf{s}_0^{l,A} = \mathbf{s}_J^{l,O}$. These initializations help us capture interactions between the triplet's components. $f_s^l(\cdot)$ is a non-linear function implemented by a MLP, while $g^{ctx}(\cdot)$ and $g^{l}(\cdot)$ are linear functions that are implemented by a single linear layer. The hidden state $\mathbf{h}_t$ is obtained in the same way as in the high-level RL process. However, the queries are changed. The query is "*What is the opinion span for the* $\mathbf{o}_{t'}$ *sentiment indicated at* $x_{t'}$*?*" for opinion term extraction and "*What is the aspect span for the* $\mathbf{o}_{t'}$ *sentiment indicated at* $x_{t'}$*?*" for aspect term extraction.

Policy: The stochastic policy for opinion/aspect extraction π_l specifies a probability distribution over the actions given the low-level state $\mathbf{s}_t^l$ and the high-level option $\mathbf{o}_{t'}$ that launches the current subtask:

$$\mathbf{a}_t \sim \pi_l(\mathbf{a}_t|\mathbf{s}_t^l;\mathbf{o}_{t'}) = softmax(\mathbf{W}_\pi^l[\mathbf{o}_{t'}]\mathbf{s}_t^l). \tag{6}$$

Reward: At every time step, when $\mathbf{a}_t$ is executed, the **intermediate reward** r_t^l is computed as the prediction error over the gold labels:

$$r_t^l = \begin{cases} \lambda, & \text{if tag } y_t \text{ is predicted correctly} \\ -0.5, & \text{otherwise,} \end{cases} \tag{7}$$

where $\lambda \geq 0$ and λ depends on the aspect/opinion tag type. This enables the model to learn a policy that emphasizes the prediction of B and I tags and avoids only predicting O tags in a trivial manner. When all actions are sampled, there is a **final reward** r_{final}^l for the low-level processes, represented by:

$$r_{final}^l = \begin{cases} 1, & \text{if all tags are predicted correctly} \\ -1, & \text{otherwise.} \end{cases} \tag{8}$$

There will also be negative rewards in the cases where the low-level processes produce impossible predictions, namely cases where there are no or more than one B tag present, and no or more than one opinion/aspect term identified for each predicted triplet. Note that the low-level rewards are non-zero only in the case where the option $\mathbf{o}_{t'}$ from the high-level process is correctly predicted.

2.4 Hierarchical Policy Learning

We learn the high-level policy by maximizing the expected total reward at each time step t as the agent samples trajectories following the high-level policies π_u. Likewise, we learn the low-level policies by maximizing the expected total reward at each time step t as the agent samples trajectories following the low-level policies π_l^A, π_l^O. We then optimize all policies using policy gradient methods [17] with the REINFORCE algorithm [19, 21].

2.5 Training Procedure

We pre-train our ASTE-RL models for 40 epochs with a learning rate of 2e-5. During pre-training, we give our model the ground-truth options or actions at every time step to limit the exploration of the agent due to the high-dimensional state space in our setup. This prevents the agent from exploring too many unreasonable cases, e.g. an I tag preceding a B tag, and learning too slowly. We then fine-tune the best model (chosen based on the Dev F_1 score) with RL policy for 15 epochs with a learning rate of 5e-6. We sample 5 trajectories for each data point during RL fine-tuning.

We initialize the BERT parameters from pre-trained weights [3] and update them during training for this task. We set the dimension of sentiment polarity and opinion/aspect tag embeddings at 300. For POS embeddings, we set the dimension at 25. We randomly initialize these embeddings and update them during training. We set the state vector dimension for $\mathbf{s}_t^u$ and $\mathbf{s}_t^l$ at 300. We apply dropout [16] after the non-linear activations in $f_s^u(\cdot)$ and $f_s^l(\cdot)$ during training and set the dropout rate at 0.5. We train our models in mini-batches of size 16 and optimize the model parameters using the Adam optimizer [5].

3 EXPERIMENTS AND ANALYSIS

3.1 Datasets & Evaluation Metrics

Table 2: ASTE-Data-V2 Dataset Statistics

	14Lap			14Rest			15Rest			16Rest		
	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test	Train	Dev	Test
#sentence	906	219	328	1266	310	492	605	148	322	857	210	326
#positive	817	169	364	1692	404	773	783	185	317	1015	252	407
#negative	517	141	116	480	119	155	205	53	143	329	76	78
#neutral	126	36	63	166	54	66	25	11	25	50	11	29

We use the ASTE-Data-V2 dataset[1] curated by Xu et al. [24] to show the effectiveness of ASTE-RL in two different domains of English reviews, namely the laptop and restaurant domains. 14Rest, 15Rest, 16Rest are the datasets of the restaurant domain and 14Lap is of the laptop domain. We include the statistics of the four datasets in ASTE-Data-V2 in Table 2, where #sentence represents the number of sentences, and #positive, #negative, and #neutral represent the numbers of triplets with positive, negative, and neutral sentiment polarities respectively.

We process the sentences with BERT's WordPiece tokenizer [22] to make them work for ASTE-RL. Since the WordPiece tokenization may break down the tokens in the original dataset into subwords, we need to align the opinion/aspect term annotations and our BIO tagging scheme. We tag every token that corresponds to the opinion/aspect term tokens in the original annotations with I, except for the first token, which we tag with B.

We follow the evaluation metrics of Xu et al. [24] for our experiments. An extracted triplet is correct if the entire aspect term, opinion term, and sentiment polarity match with a ground-truth triplet. We report **precision**, **recall** and $\mathbf{F_1}$ score based on this.

3.2 Baselines

We compare the performance of ASTE-RL against the following baselines: (i) **WhatHowWhy**: Peng et al. [12] proposed a multi-layer LSTM neural architecture for co-extraction of aspect terms with sentiments, and opinion terms, with a Graph Convolutional Network [6] component to capture dependency information to enhance the co-extraction. (ii) **OTE-MTL**: Zhang et al. [25] proposed a multi-task learning framework to jointly extract aspect and opinion terms while parsing word-level sentiment dependencies, before conducting a triplet decoding process. We use results from Huang et al. [4] for OTE-MTL's performance on ASTE-Data-V2. (iii) **GTS**: Wu et al. [23] proposed an end-to-end grid tagging framework and a grid inference strategy to exploit mutual indication between opinion factors. We use results from Huang et al. [4] for GTS' performance on ASTE-Data-V2, and report them for two variants: bidirectional LSTM (BiLSTM) and BERT. (iv) **JET**: Xu et al. [24] proposed a position-aware tagging scheme for triplet extraction. They encode information about sentiment polarities and distances between the start position of aspect term and the

[1] https://github.com/xuuuluuu/SemEval-Triplet-data

Table 3: Results of ASTE-RL and Previous Methods on the ASTE-Data-V2 Dataset

	14Lap				14Rest				15Rest				16Rest			
Model	Dev F_1	Prec	Rec	F_1	Dev F_1	Prec	Rec	F_1	Dev F_1	Prec	Rec	F_1	Dev F_1	Prec	Rec	F_1
WhatHowWhy [12]	-	37.38	50.38	42.87	-	43.24	63.66	51.46	-	48.07	57.51	52.32	-	46.96	64.24	54.21
OTE-MTL [25]	-	54.26	41.07	46.75	-	63.07	58.25	60.56	-	60.88	42.68	50.18	-	65.65	54.28	59.42
$\text{GTS}_{\text{BiLSTM}}$ [23]	-	58.02	40.11	47.43	-	71.41	53.00	60.84	-	64.57	44.33	52.57	-	70.17	55.95	62.26
$\text{JET}^{t}_{\text{BiLSTM}}$ [24]	48.26	54.84	34.44	42.31	53.14	66.76	49.09	56.58	55.06	59.77	42.27	49.52	58.45	63.59	50.97	56.59
$\text{JET}^{o}_{\text{BiLSTM}}$ [24]	45.83	55.98	35.36	43.34	53.54	61.50	55.13	58.14	60.97	64.37	44.33	52.50	60.90	70.94	57.00	63.21
GTS_{BERT} [23]	-	57.12	53.42	55.21	-	71.76	59.09	64.81	-	54.71	55.05	54.88	-	65.89	66.27	66.08
$\text{JET}^{t}_{\text{BERT}}$ [24]	50.40	53.53	43.28	47.86	56.00	63.44	54.12	58.41	59.86	68.20	42.89	52.66	60.67	65.28	51.95	57.85
$\text{JET}^{o}_{\text{BERT}}$ [24]	48.84	55.39	47.33	51.04	56.89	70.56	55.94	62.40	64.78	64.45	51.96	57.53	63.75	70.42	58.37	63.83
TOP [4]	-	57.84	59.33	58.58	-	63.59	73.44	68.16	-	54.53	63.30	58.59	-	63.57	71.98	67.52
BMRC [1]	56.08	65.91	52.15	58.18	62.83	72.17	65.43	68.64	72.47	62.48	55.55	58.79	70.91	69.87	65.68	67.35
ASTE-RL	58.14	64.80	54.99	**59.50**	64.40	70.60	68.65	**69.61**	74.01	65.45	60.29	**62.72**	72.11	67.21	69.69	**68.41**
- Pre-training only	57.35	62.00	55.84	58.73	64.50	69.70	69.23	69.47	72.84	63.31	61.61	62.44	71.50	64.76	70.74	67.57

opinion term's start and end positions (JET^t) or vice versa (JET^o). We report the results for two variants: BiLSTM and BERT. (v) **TOP**: Huang et al. [4] proposed a two-stage method to enhance correlations between aspect and opinion terms. Aspect and opinion terms are first extracted with sequence labeling, and artificial tags are added to each pair to establish correlation. A sentiment polarity is then identified for each pair using the resulting representations. (vi) **BMRC**: Chen et al. [1] proposed a transformation of the ASTE task into a multi-turn MRC task and a bidirectional MRC framework to address it. They use non-restrictive, restrictive and sentiment classification queries in a three-turn process to extract triplets. We train and test BMRC on ASTE-Data-V2 over 5 runs with different random seeds.

Table 4: F_1 Scores for Multiple and Overlapping Triplets

	14Lap		14Rest		15Rest		16Rest	
	ASTE-RL	BMRC	ASTE-RL	BMRC	ASTE-RL	BMRC	ASTE-RL	BMRC
Single	62.46	63.04	67.44	66.51	61.33	59.59	66.31	67.83
Multiple	57.70	53.77	70.23	69.17	63.95	56.37	69.84	64.11
No Overlap	62.24	62.80	72.16	70.81	62.17	59.75	70.13	69.23
Overlap	55.94	50.18	67.23	63.96	63.83	55.48	65.20	58.59

3.3 Experimental Results

The experimental results are shown in Table 3. We observe that BERT-based models (their results are in the row above ASTE-RL's results in Table 3) generally perform better than the non-BERT models. Hence, we only experiment with BERT for our ASTE-RL model. We select our best model for each dataset based on its Dev F_1 score. For reproducibility, we report the testing results averaged over 5 runs with different random seeds. ASTE-RL outperforms existing baselines on all four datasets, and significantly outperforms existing baselines on the 15Rest dataset. When compared to the second-best performance for each dataset, we observe an average improvement of 1.68% F_1 score across all four datasets, and an improvement of 3.93% on 15Rest. We also observe that our model strikes a balance between the TOP and BMRC models in terms of precision and recall, and hypothesize that this balance can be flexibly shifted depending on β to fit dataset requirements, if we generalize $\mathbf{r}^{u}_{final} = F_{\beta}(X)$, where $F_{\beta}(X)$ is the weighted harmonic mean of precision and recall.

3.4 Effect of RL Fine-tuning

In Table 3, we report our results for ASTE-RL without the RL fine tuning step. In this setting, we pre-train our ASTE-RL for 40 epochs as usual and after that we run for another 15 epochs with a learning rate of 5e-6 (as used in RL fine-tuning step). As compared to the RL fine-tuning setting with multinomial sampling, this setting has lower F_1 scores with an average decrease of 0.51% over 5 runs with different random seeds. In this setting, our model achieves slightly higher recall, but precision is significantly lower across all four datasets. This might be because multinomial sampling encourages more exploration after the initial pre-training of 40 epochs.

3.5 Analysis on Multiple & Overlapping Triplet Extraction

We show the results of ASTE-RL and BMRC in complex situations where there are multiple and overlapping triplets in a sentence in Table 4. For the multiple triplet scenario, we observe that there is a performance increase for 14Rest, 15Rest and 16Rest and a decrease for 14Lap as compared to the case where only one triplet is present in a sentence. For the overlapping triplet scenario, we observe a performance increase for for 15Rest and a decrease for 14Lap, 14Rest and 16Rest.

In general, we observe that ASTE-RL can handle multiple and overlapping triplets in a sentence consistently well due to its hierarchical RL setup, as compared to BMRC. There is a total F_1 decrease of 4.76% for multiple triplet extraction for ASTE-RL across all four datasets as compared to 16.21% for BMRC, and a total F_1 decrease for overlapping triplet extraction of 16.16% for ASTE-RL as compared to 34.38% for BMRC.

4 CONCLUSION

In this work, we propose a novel ASTE-RL model based on hierarchical reinforcement learning (RL) paradigm for aspect sentiment triplet extraction (ASTE). In this paradigm, we treat the aspect and opinion terms as arguments of the sentiment polarities. We decompose the ASTE task into a hierarchy of three subtasks: high-level sentiment polarity extraction, and low-level opinion and aspect term extractions. This approach is good at modeling the interactions between the three tasks and handling multiple and overlapping triplets. We incorporate the multi-turn MRC elements in our model to further improve these interactions. Our proposed model achieves state-of-the-art performance on four challenging datasets for the ASTE task.

ACKNOWLEDGMENTS

This project is supported by the DSO grant no. RTDST190702 awarded to SUTD titled Complex Question Answering.

REFERENCES

[1] Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional Machine Reading Comprehension for Aspect Sentiment Triplet Extraction. *arXiv preprint arXiv:2103.07665* (2021).

[2] Hongliang Dai and Yangqiu Song. 2019. Neural Aspect and Opinion Term Extraction with Mined Rules as Weak Supervision. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. https://www.aclweb.org/anthology/P19-1520

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[4] Lianzhe Huang, Peiyi Wang, Sujian Li, Tianyu Liu, Xiaodong Zhang, Zhicong Cheng, Dawei Yin, and Houfeng Wang. 2021. First Target and Opinion then Polarity: Enhancing Target-opinion Correlation for Aspect Sentiment Triplet Extraction. *arXiv preprint arXiv:2102.08549* (2021).

[5] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[6] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[7] Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A Unified Model for Opinion Target Extraction and Target Sentiment Prediction. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI*. https://doi.org/10.1609/aaai.v33i01.33016714

[8] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 1340–1350. https://doi.org/10.18653/v1/P19-1129

[9] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* (2012).

[10] Bing Liu et al. 2010. Sentiment analysis and subjectivity. *Handbook of natural language processing* (2010).

[11] Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd international conference on Knowledge capture*.

[12] Haiyun Peng, Lu Xu, Lidong Bing, Fei Huang, Wei Lu, and Luo Si. 2020. Knowing what, how and why: A near complete solution for aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://aaai.org/ojs/index.php/AAAI/article/view/6383

[13] S. Poria, D. Hazarika, N. Majumder, and R. Mihalcea. 2020. Beneath the Tip of the Iceberg: Current Challenges and New Directions in Sentiment Analysis Research. *IEEE Transactions on Affective Computing* 01 (nov 2020), 1–29. https://doi.org/10.1109/TAFFC.2020.3038167

[14] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250* (2016).

[15] Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 157–176.

[16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[17] Richard S Sutton, David A McAllester, Satinder P Singh, Yishay Mansour, et al. 1999. Policy gradient methods for reinforcement learning with function approximation.. In *NIPs*, Vol. 99. Citeseer, 1057–1063.

[18] Richard S Sutton, Doina Precup, and Satinder Singh. 1999. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial intelligence* (1999).

[19] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*. https://doi.org/10.1609/aaai.v33i01.33017072

[20] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14441

[21] Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning* 8, 3-4 (1992), 229–256.

[22] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[23] Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid Tagging Scheme for Aspect-oriented Fine-grained Opinion Extraction. *arXiv preprint arXiv:2010.04640* (2020).

[24] Lu Xu, Hao Li, Wei Lu, and Lidong Bing. 2020. Position-Aware Tagging for Aspect Sentiment Triplet Extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/2020.emnlp-main.183

[25] Chen Zhang, Qiuchi Li, Dawei Song, and Benyou Wang. 2020. A Multi-task Learning Framework for Opinion Triplet Extraction. *arXiv preprint arXiv:2010.01512* (2020).