Knowledge-Based Systems 236 (2022) 107723

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Reasoning over multiplex heterogeneous graph for Target-oriented Opinion Words Extraction

Yaqing Dai [a], Pengfei Wang [a], Xiaofei Zhu [b,*]

[a] School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China
[b] College of Computer Science and Engineering, Chongqing University of Technology, Chongqing 400054, China

ARTICLE INFO

Article history:
Received 3 June 2021
Received in revised form 7 November 2021
Accepted 9 November 2021
Available online 23 November 2021

Keywords:
Target-oriented Opinion Word Extraction
Reinforcement learning
Syntactic dependency tree

ABSTRACT

Target-oriented Opinion Word Extraction (TOWE) is a new emerging subtask of Aspect Based Sentiment Analysis (ABSA), which aims to extract fine-grained opinion terms for a given aspect term from a sentence. In this task, the key point is how to find the correct opinion that is far away from its corresponding aspect. Ideally, reinforcement learning (RL) seems to be a promising approach due to its delayed reward mechanism. However, as aspect-opinion interaction data is likely to be complicated, it is not easy to directly apply RL techniques to improve the performance. In this paper, we propose a novel **P**adding-**E**nhanced **R**einforcement learning model (PER) to address this issue. Specifically, PER first designs a multiplex heterogeneous graph to cover both sequential structure and syntactic structure in order to enrich their interactions and alleviate the long distance issue. By formulating the extraction task as a Markov Decision Process (MDP), PER then walks on the designed graph to infer corresponding opinions for each aspect. In addition, a padding module is further designed to aggregate rich information from distant nodes to guide the exploration process. Extensive experimental results on four widely used datasets illustrate that our proposed model consistently outperforms the state-of-the-art methods.

1. Introduction

Target-oriented Opinion Word Extraction (TOWE) is a new emerging subtask of Aspect Based Sentiment Analysis (ABSA). TOWE has attracted considerable attention because it can provide fine-grained opinion information rather than just user rating scores when analyzing user reviews [1–5]. It has been widely used in identifying users' attitude towards an aspect of a product, and explaining why they like or dislike it. Given an aspect, the goal of TOWE aims to extract its corresponding opinion terms from a sentence. As shown in Fig. 1, given the example sentence "*It absolutely is more expensive than most PC laptops, but the ease of use, security, and minimal problems that have arisen make it well worth the price tag.*", we could find that "use" and "price tag" are aspect terms, "ease" and "well worth" are opinion terms. In TOWE task, given the aspect term "use", the goal is to extract its corresponding opinion term "ease", and given the aspect term "price tag", the goal is to extract its corresponding opinion term "well worth".

Due to its valuable benefits to many practical applications (i.e., sentiment classification [6–9], opinion summarization [10, 11]), various models are recently designed to make a correct extraction. Fan et al. [1] firstly explicitly brought out TOWE task, and released a benchmark corpus including four datasets. They formulate it as a sequence labeling task. After that, a few methods were proposed for improvement. Wu et al. [2] leverage transfer learning in order to get more opinion knowledge from sentiment classification task. Though effective, these models still cannot perform well once the aspect term is far away from its corresponding opinion.

Syntactic dependency trees contain rich linguistics information. Many works [12–14] attempt to leverage syntactic dependency trees for the aspect term extraction task. For example, Yin et al. [14] learn new word embeddings based on the dependency path between words. Some researchers propose to introduce syntactic structures into the TOWE task. Meanwhile, graph neural networks (GNNs) [15–18] seem to be helpful to learn the representations of words from syntactic structures. Veyseh et al. [3] find that syntactic dependency trees are critical for the TOWE task. Some related opinion terms, which are far from aspect term in sequential structure, are usually close to aspect term based on the syntactic structure. Therefore, they put the distance information of syntactic structure into graph convolutional networks (GCNs) [19] to alleviate this long distance issue. Zhou et al. [4] also leverage GCNs and calculate an important weight for each edge in order to learn the difference of dependency relations

* Corresponding author.
E-mail addresses: daiyaqing97@bupt.edu.cn (Y. Dai), wangpengfei@bupt.edu.cn (P. Wang), zxf@cqut.edu.cn (X. Zhu).

https://doi.org/10.1016/j.knosys.2021.107723
0950-7051/

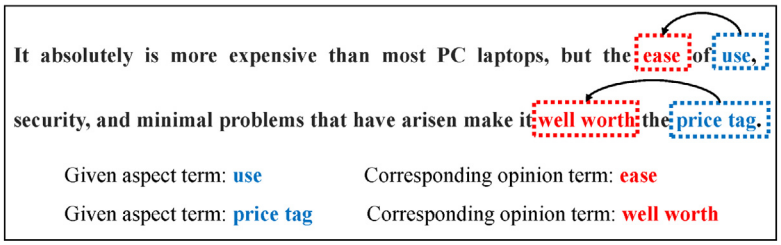

Fig. 1. An example of TOWE task. The aspect terms are in blue and the opinion terms are in red. TOWE task aims to extract corresponding opinion terms for an aspect term in the sentence. We use arrows to connect the aspect and its corresponding opinion term in this figure.

between words. Jiang et al. [5] propose a novel attention-based relational graph convolutional neural network (ARGCN) to exploit syntactic information over dependency graphs.

These methods mentioned above leverage supervised learning to capture partial features to identify opinion terms based on sequential structure or syntactic structure during training. However, they could not effectively explore all relationships between aspect terms and opinion terms in the two structures, especially when the opinion terms are far from the aspect terms. Recently, reinforcement learning (RL) [20–22] has achieved great success in many real world applications, such as information retrieval [23,24], machine translation [25,26], recommendation [27–29], and question answering [30,31], etc. Randomly walking on a graph with RL approach is widely used for finding explainable paths or reasoning over knowledge graph [32,33]. Due to its delayed reward mechanism, it can well explore several steps to inquire the result in order to maximize the cumulative reward. Therefore, RL seems to be a promising solution for the TOWE task by leveraging its strong reasoning capability to identify the relationship between aspect terms and opinion terms.

In this paper, we propose a novel Padding-Enhanced Reinforcement learning model (PER). Specifically, we construct a multiplex heterogeneous graph which captures the information from both sequential and syntactic structures in order to alleviate the long distance issue. We formulate the TOWE task as a Markov Decision Process (MDP), and apply RL on the multiplex heterogeneous graph for word extraction. By using this method, we can explore a path from aspect term to opinion term. To the best of our knowledge, it is the first time that RL approach has been explicitly explored and applied in the TOWE task. Although it is appealing in theory, it is a non-trivial problem to optimize a long-term reward for opinion extraction in practice. First, aspect-opinion interaction data is usually complicated. For a given aspect term, it is challenging to recognize all opinion terms by reasoning over such a multiplex heterogeneous graph. To deal with this issue, we design a reward function which can better guide the agent to explore on the graph. Second, as the agent can only choose actions based on information from their one-hop nodes, and it is a challenge to capture rich information from other distant nodes. To address this issue, we introduce a padding module which can aggregate the rich information from distant nodes to help guide the agent when choosing actions.

In summary, the main contributions of our work are as follows:

- We formulate the TOWE task into a Markov Decision Process (MDP). To the best of our knowledge, it is the first time that RL approach has been explicitly explored and applied in the TOWE task.
- We propose a novel multiplex heterogeneous graph to capture both sequential structure and syntactic structure information, which is designed to alleviate the long distance problem.
- We introduce a padding module to capture rich information from distant nodes, which can help to guide the exploration process of RL.
- Experimental results on four real-world datasets show that our model consistently outperforms all state-of-the-art baselines in terms of all evaluation metrics.

2. Related work

2.1. Target-oriented opinion words extraction

Target-oriented Opinion Words Extraction (TOWE) is a new emerging subtask of Aspect Based Sentiment Analysis (ABSA). The early related works mainly focus on aspect term extraction (also called opinion target extraction) which is a subtask in ABSA. Hu and Liu [10] leverage association mining to extract aspect terms. Yin et al. [14] leverage syntactic dependency information to learn new word embeddings and use CRF for aspect term extraction. Xu et al. [34] employ two types of pre-trained embeddings: general-purpose embeddings and domain-specific embeddings, and design a simple CNN model for aspect term extraction.

Later, some works extract both aspect terms and opinion terms at the same time, but they fail to capture the corresponding relationships. For example, Qiu et al. [35] propose to utilize syntactic relation rules and double propagation. They attempt to propagate information between opinion words and aspect words. Liu et al. [36] employ different pre-trained word embeddings and RNNs to extract terms. Wang et al. [37] apply multiple attention layers to learn interactively to propagate information between aspect terms and opinion terms. This method can exploit indirect relations between terms and boost the results considerably.

Recently, Fan et al. [1] firstly propose the TOWE task and formulate it as a sequence labeling problem. Due to lack of annotated data, Wu et al. [2] transfer sentiment classification task into TOWE task to gather more opinion knowledge via an auxiliary learning signal. Though these approaches achieve the state-of-the-art performance, they still suffer challenges when extracting opinions that are far from the corresponding aspects. Veyseh et al. [3] further leverage graph convolutional networks (GCNs) [38] to aggregate the information from neighbors based on the syntactic structure. Zhou et al. [4] leverage GCNs to generate syntactic representations of words. As not all words are equally important to the TOWE task, they calculate an important weight for each edge to distinguish effects of different neighbors. Jiang et al. [5] propose a novel attention-based relational graph convolutional neural network (ARGCN) to exploit syntactic information over dependency graphs, while they also introduce a target-aware representation to fully exploit opinion target information in a concise way.

Differ with existing works, we propose a novel complex multiplex heterogeneous graph to capture both sequential structure and syntactic structure information, and explicitly apply RL to reason over this graph in order to effectively extract corresponding opinion terms for a given aspect.

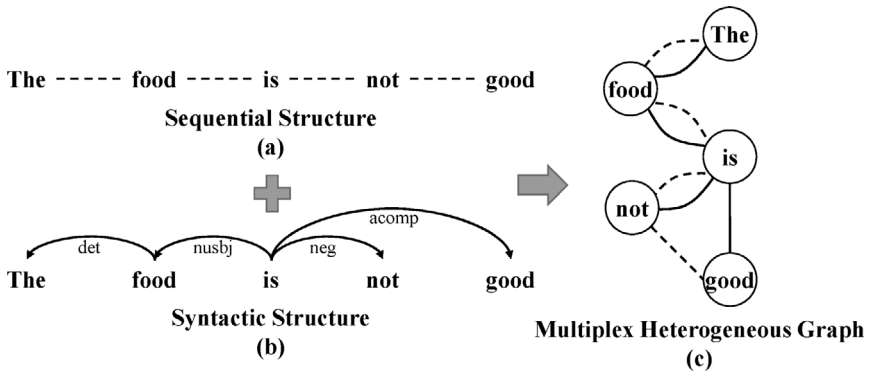

Fig. 2. The process of building our multiplex heterogeneous graph for the example sentence - "The food is not good". The dashed lines denote sequential edges, and the solid lines denote syntactic edges. Notice that we do not draw self-loop edge of every word for clarity.

2.2. Reinforcement learning

Reinforcement learning (RL) [20–22] is an area of machine learning concerned with how agents take actions in an environment to maximize the cumulative reward. Due to its advantage of considering long-term feedback, RL achieves substantial success in various areas, such as information retrieval [23,24], machine translation [25,26], recommendation [27–29,39], and question answering [30,40], etc. For example, Feng et al. [41] utilize a Monte Carlo tree search strategy to alleviate the greedy selection in diverse ranking. Ranzato et al. [25] introduce reinforcement learning algorithm for text generation to avoid exposure bias. Xian et al. [42] design an RL framework to directly optimize the long-term user engagement in online recommendation. Wang et al. [43] apply reinforcement learning to open-domain QA to deal with the lack of annotation in the passage selection.

Reasoning over a graph with RL approach also has been used widely in many areas such as knowledge graph reasoning [44–46], explainable recommendation [33,42], and conversation system [47], etc. For example, Hildebrandt et al. [48] infer the missing triple in the graph based on debate dynamics, where two agents explore over the graph to prove the query is right or wrong. Zhao et al. [33] try to reason over an item knowledge graph and find a path to explain why we recommend an item to a user. By reasoning over the graph to get correlative entities, Moon et al. [47] generate better response in conversation system.

To the best of our knowledge, our work is the first time to explicitly apply reinforcement learning approach for the TOWE task.

3. Graph construction

To model both the sequential structure and syntactic structure effectively in order to alleviate the long distance issue, in this section, we design a novel multiplex heterogeneous graph which can capture the information of these two different structures in a single graph. After that, a Markov Decision Process (MDP) will be applied on this graph for word extraction. The multiplex heterogeneous graph has two types of edges, i.e., sequential edge and syntactic edge, within a node pair. Specifically, for a given sentence $\mathcal{W} = \{w_1; w_2; \cdots; w_n\}$ where w_i represents the ith word in the sentence and n is the number of words in the sentence, we transform $\mathcal{W}$ into a multiplex heterogeneous graph $\mathcal{G}$, which consists of two subgraphs, i.e., a sequential subgraph $\mathcal{G}^{seq}$ and a syntactic subgraph $\mathcal{G}^{syn}$.

The sequential subgraph $\mathcal{G}^{seq} = \{\mathcal{V}; \mathcal{E}^{seq}\}$ is built based on the sequential structure shown in Fig. 2(a), where $\mathcal{V}$ is the set of words and $\mathcal{E}^{seq}$ is the set of sequential edges between two words, each edge (dashed line) represents a sequential relation between two consecutive words. We define the triple $(w_i; e_{ij}^{seq}; w_j) \in \mathcal{G}^{seq}$ which means word w_i and w_j are connected by an edge e_{ij}^{seq} in graph $\mathcal{G}^{seq}$. The syntactic subgraph $\mathcal{G}^{syn} = \{\mathcal{V}; \mathcal{E}^{syn}\}$ is built based on the syntactic structure shown in Fig. 2(b), where $\mathcal{V}$ is the same word set as we defined in $\mathcal{G}^{seq}$ and $\mathcal{E}^{syn}$ is the set of syntactic edges between the words, each edge (solid line) denotes a syntactic dependency relation between two words. Similarly, we define the triple $(w_i; e_{ij}^{syn}; w_j) \in \mathcal{G}^{syn}$ where $w_i; w_j \in \mathcal{V}; e_{ij}^{syn} \in \mathcal{E}^{syn}$. In addition, we also extend $\mathcal{E}^{syn}$ by adding a self-loop edge e_{ii}^{loop}, $i \in (1; 2; \cdots; n)$.

The multiplex heterogeneous graph $\mathcal{G}$ is formulated as $\mathcal{G} = \{\mathcal{V}; \mathcal{E}\} = \{\mathcal{G}^{seq}; \mathcal{G}^{syn}\}$, where $\mathcal{V}$ denotes the same word set as defined above, and $\mathcal{E} = \mathcal{E}^{seq} \cup \mathcal{E}^{syn}$. $\mathcal{G}$ is an undirected graph, i.e., if $(w_i; e_{ij}^{r}; w_j) \in \mathcal{G}$, then $(w_j; e_{ji}^{r}; w_i) \in \mathcal{G}$, where $r \in \{seq; syn\}$. Fig. 2(c) demonstrates an example of the multiplex heterogeneous graph.

4. Methodology

In this section, we introduce our proposed Padding-Enhanced Reinforcement learning model (PER) for the TOWE task. Denote the parameters of the model as , we formulate the TOWE task into a Markov Decision Process (MDP). Our goal is to find an optimal policy $_*$ to walk and automatically extract opinion words on the multiplex heterogeneous graph. To this end, we first randomly initialize and use it as a prior knowledge, which will be used to search a better policy $_e$ by leveraging the Monte Carlo tree search (MCTS). Then we update the policy by minimizing the difference between $_e$ and . We repeat the above two operations interchangeably until convergence or a specified number of iterations is reached.

4.1. MDP formulation of TOWE

The goal of the TOWE task is to extract opinion terms for a given aspect term from a sentence $\mathcal{W}$. An aspect or opinion term may contain several words while the RL agent needs to find all opinion words and label them correctly.

We use a MDP to formulate the TOWE task with our proposed multiplex heterogeneous graph and utilize RL to learn the exploring rules on graph, which can explore a path to extract opinion terms by using both sequential and syntactic information.

Assume $\mathcal{L}^a$ and $\mathcal{L}^o$ are the aspect label sequence and the opinion label sequence, respectively. Following Ramshaw and Marcus [49], $\mathcal{L}^a$ is obtained by labeling every word and pointing out the positions of terms in the sentence by utilizing BIO schema ($\mathcal{L} = \{B$: beginning, I: inside, O: other$\}$) for the aspect term. $\mathcal{L}^o$ is obtained in a similar way.

State $\mathcal{S}$: Let $s_t \in \mathcal{S}$ denote the state at time step t, where $\mathcal{S}$ is the state space. The agent needs the sentence $\mathcal{W}$, the graph $\mathcal{G} = \{\mathcal{V}; \mathcal{E}\}$, the given aspect term which is represented by the sequence $\mathcal{L}^a$ and history path $\mathcal{P}_t$ to make a correct decision. We design the state as a tuple $s_t = (\mathcal{W}; \mathcal{G}; \mathcal{L}^a; \mathcal{P}_t)$, where $\mathcal{P}_t = \{(e_i; w_{(i)}; l_i^o) | i \in [0; t]\}$ records the action information (chosen edge e_i, chosen neighbor word $w_{(i)}$, the predicted opinion label l_i^o of $w_{(i)}$) for each time step i. The agent has reached word $w_{(t)}$ at time step t, where the (t) is the index of word $w_{(t)}$ in the sentence $\mathcal{W}$. The agent starts from the first word of aspect term, thus the aspect label of $w_{(0)}$ must be B, and $s_0 = (\mathcal{W}; \mathcal{G}; \mathcal{L}^a; \mathcal{P}_0)$ where $\mathcal{P}_0 = \{(\varnothing; w_{(0)}; \varnothing)\}$. When the agent chooses **STOP** action or the maximum explore time step has been reached, the exploring process will stop at time step T and we obtain the terminal state s_T which includes the whole path $\mathcal{P}_T$.

Action $\mathcal{A}$: At each time step t, $\mathcal{A}_t = \{(e; w; l^o) | e \in \mathcal{E}; l^o \in \mathcal{L}; (w_{(t)}; e; w) \in \mathcal{G}\} \cup \{\textbf{STOP}\}$ is a set of possible actions according to s_t. And the whole action space is $\mathcal{A} = \cup \mathcal{A}_t$. It is worth noting that we do not allow the agent to go back to the word which already exists in history path $\mathcal{P}_t$ in order to make the agent focus on exploring new words. If the agent performs action $a_t = (e_{t+1}; w_{(t+1)}; l_{t+1}^o) \in \mathcal{A}_t$, it moves through an edge e_{t+1} from current word $w_{(t)}$ to a neighbor word $w_{(t+1)}$, and predicts the opinion label l_{t+1}^o for $w_{(t+1)}$. When the agent thinks all opinion words have been explored, it will choose the **STOP** action to stop exploring new words.

Reward $\mathcal{R}$: As there is no intermediate reward (i.e., $\mathcal{R}_t = 0$) available during the path exploring process, we employ the delayed reward strategy (i.e., use the terminal reward $\mathcal{R}_T$) to evaluate the whole path $\mathcal{P}_T$. A high quality path should be able to label opinion words correctly based on a few actions. However, since sequential edge and syntactic edge are not equally important, we still need to balance them.

Based on all above-mentioned intuitions, we introduce three kinds of rewards, i.e., exploration reward $\mathcal{R}_e$, accuracy reward $\mathcal{R}_a$ and efficiency reward $\mathcal{R}_f$, which are defined as follows:

$$\mathcal{R}_e = \frac{\hat{n}_o}{n_o}; \tag{1}$$

$$\mathcal{R}_a = \frac{n_c}{T}; \tag{2}$$

$$\mathcal{R}_f = -\frac{n_{syn} + (1 - \ )n_{seq}}{n}; \tag{3}$$

where n_o is the number of opinion words in the ground truth, $\hat{n}_o$ is the number of hit opinion words in path $\mathcal{P}_T$, n_c is the correct predicted label count in $\mathcal{P}_T$, n_{syn} and n_{seq} denote the number of syntactic edges and sequential edges in $\mathcal{P}_T$, respectively, n represents the length of the sentence $\mathcal{W}$, $\in (0; 1)$ is a hyper parameter which controls the edge selection preference of the agent, e.g., a higher will encourage the agent to select a sequential edge.

The exploration reward $\mathcal{R}_e$ means the path $\mathcal{P}_T$ should cover all opinion words and the accuracy reward $\mathcal{R}_a$ means the agent should predict correct labels for every word in the path $\mathcal{P}_T$. Meanwhile, the path should be shorter in order to reduce useless information. Therefore, the efficiency reward $\mathcal{R}_f$ controls the agent to explore the path based on a few actions and balance the ratio of syntactic edges and sequential edges in path $\mathcal{P}_T$. Considering about the three kinds of rewards, the terminal reward is written as:

$$\mathcal{R}_T = \mathcal{R}_e \cdot \mathcal{R}_a + \mathcal{R}_f; \tag{4}$$

As we mentioned before, a good path should cover all the opinion words (i.e. $\mathcal{R}_e$) with all correct labels (i.e. $\mathcal{R}_a$). Therefore, we multiply reward $\mathcal{R}_e$ and $\mathcal{R}_a$ in order to satisfy two conditions at the same time. As longer paths would get more useless information and lead to more negative reward, we introduce reward $\mathcal{R}_f$ to let the agent learn to explore faster. Please note that when $n_{syn} + n_{seq} = 0$, the agent will refuse to explore on the graph, therefore we define $\mathcal{R}_T = -1$ for this special case.

Transition function $\mathcal{T}$: The transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ is defined that: $s_{t+1} = \mathcal{T}(s_t; a_t) = (\mathcal{W}; \mathcal{G}; \mathcal{L}^a; \mathcal{P}_t \cup \{(e_{t+1}; w_{(t+1)}; l_{t+1}^o)\})$, where means state records the information of action $a_t = (e_{t+1}; w_{(t+1)}; l_{t+1}^o)$.

Value function V : The value function $V : \mathcal{S} \rightarrow \mathbb{R}$ is a scalar evaluation. It is learned to approximate the terminal reward $\mathcal{R}_T$ for judging the quality of the whole path (an episode) based on the input state s_t. So we calculate state representation $\mathbf{s}_t$ by encoding the path $\mathcal{P}_t$. Therefore the state representation $\mathbf{s}_t$ and value $V(s_t)$ calculated by $\mathcal{P}_t = \{(e_i; w_{(i)}; l_i^o) | i \in [0; t]\}$ can be written as:

$$\mathbf{h}_t = \text{MLP}(\mathbf{e}_t \oplus \mathbf{w}_{(t)} \oplus \mathbf{l}_t^o \oplus \mathbf{l}_t^a; \ _{step}); \tag{5}$$

$$\mathbf{s}_t = \text{BiGRU}(\mathbf{s}_{t-1}; \mathbf{h}_t; \ _{gru}); \tag{6}$$

$$V(s_t) = \text{MLP}(\mathbf{s}_t; \ _v); \tag{7}$$

where $\mathbf{e}_t$ is the embedding of edge e_t, $\mathbf{w}_{(t)}$ is the pre-trained embedding of word $w_{(t)}$, $\mathbf{l}_t^o$ is the embedding of predicted opinion label of $w_{(t)}$, $\mathbf{l}_t^a$ is the embedding of aspect label of $w_{(t)}$, $\oplus$ denotes the concatenation operator, $_{gru}$ denotes all the related parameters of the Bi-directional Gated Recurrent Unit (BiGRU) network, in addition, $_{step}$ and $_v$ are parameters of two Multi-Layer Perceptrons (MLPs). Please note that here we use $\mathbf{l}_t^a$ in order to add more information of the given aspect term.

Policy : Our goal is to learn a path-exploring policy that calculates the probability distribution of all actions $a \in \mathcal{A}_t$ based on the state s_t. To this end, we need to calculate the representation $\mathbf{h}_a$ of each action. As we mentioned above, the agent will select **STOP** action based on path $\mathcal{P}_t$. We leverage $\mathbf{s}_t$ to calculate $\mathbf{h}_a$ of the **STOP** action:

$$\mathbf{h}_a = \text{MLP}(\mathbf{s}_t; \ _{stop}); \tag{8}$$

where $_{stop}$ represents the parameters of MLP.

The remaining actions are all about exploring on the graph. The representation $\mathbf{h}_a$ of action $a = (e; w; l^o)$ calculated with the new word embedding which is generated by padding module can be written as:

$$\mathbf{h}_a = \text{MLP}(\mathbf{e} \oplus \mathbf{w}' \oplus \mathbf{l}^o \oplus \mathbf{l}^a; \ _{step}); \tag{9}$$

where $\mathbf{e}$ and $\mathbf{l}^o$ are the corresponding embeddings of the each element e and l^o of action a, $\mathbf{l}^a$ is the embedding of aspect label of word w and $\mathbf{w}'$ is the new embedding of word w which generated by the padding module. We share the same parameters $_{step}$ which is used in the value function. Since the probability of an action should consider both the state and action, for any action $a \in \mathcal{A}_t$, we define its probability as:

$$(a|s_t) = \frac{\exp\{\text{MLP}(\mathbf{s}_t \oplus \mathbf{h}_a; \ \ )\}}{\sum_{a' \in \mathcal{A}_t} \exp\{\text{MLP}(\mathbf{s}_t \oplus \mathbf{h}_{a'}; \ \ )\}}; \tag{10}$$

where denotes as the parameters in the MLP.

4.2. Padding module

As the agent can only choose actions based on information from their one-hop nodes, while the rich information of distant nodes will be ignored. To utilize the rich information of distant nodes in state s_t and better guide the exploration process, we design a padding module to capture extra information that is ignored by the agent, and feed it into the action representations to better guide the exploration process. It is worth noting that there exists two type structures in our multiplex heterogeneous graph

$\mathcal{G}$. To fully capture their unique semantics, we introduce two different paddings for every word to help aggregate semantics from $\mathcal{G}^{seq}$ and $\mathcal{G}^{syn}$ respectively.

Firstly, we get syntactic paddings by a graph neural network (GNN). We define a simple GNN to consider effects of different kinds of syntactic edge. For a word w_i in the syntactic subgraph $\mathcal{G}^{syn}$, there is a neighbor word set $\mathcal{N}_{w_i} = \{w_j | (w_i; e_{ij}^{syn}; w_j) \in \mathcal{G}^{syn}\}$. In the kth layer GNN, we get the word embedding $\mathbf{w}_i^k$ of w_i by:

$$_{ij} = \frac{exp\{\mathbf{W}^k \mathbf{e}_{ij}\}}{\sum_{w_{j'} \in \mathcal{N}_{w_i}} exp\{\mathbf{W}^k \mathbf{e}_{ij'}\}}; \tag{11}$$

$$\mathbf{w}_i^k = \sum_{w_j \in \mathcal{N}_{w_i}} {}_{ij} \mathbf{w}_j^{k-1}; \tag{12}$$

where $\mathbf{e}_{ij}$ is the embedding of edge e_{ij}^{syn} between words w_i and w_j, $\mathbf{W}^k$ is a parameter matrix in the kth layer GNN. We define $\mathbf{w}_i^0 = \mathbf{w}_i$, where $\mathbf{w}_i$ is the pre-trained embedding of word w_i. As the maximum layer is $\mathcal{K}$, we define the $\mathcal{K}$-layer word embedding $\mathbf{w}_i^{\mathcal{K}}$ as the syntactic padding $\mathbf{w}_i^{syn}$ of word w_i.

Then we get sequential padding of word w_i by a BiGRU:

$$\mathbf{w}_i^{seq} = \text{BiGRU}(\mathbf{w}_{i-1}^{seq}; \mathbf{w}_i; \ _{seq}); \tag{13}$$

where $_{seq}$ are parameters to learn.

Final, we combine the two kinds of padding by averaging:

$$\mathbf{w}_i' = \text{Average}(\mathbf{w}_i^{syn}; \mathbf{w}_i^{seq}); \tag{14}$$

where the new word embedding $\mathbf{w}_i'$ contains further information for guiding the agent better.

4.3. Searching policy by MCTS

If we only leverage the policy to greedily select for sampling and training, we may get a suboptimal path. Because the agent has no idea about reasoning over the multiplex heterogeneous graph at first, while it is impossible to explore the whole space for the optimal paths. Therefore we leverage MCTS to make a heuristic search in the whole space for a good policy $_e$ as a prior knowledge. We employ MCTS by following Silver [50] with four steps.

Selection: Starting from the root node s_r, MCTS recursively selects the child nodes until a leaf node is reached. At each node s_t, MCTS selects an action a_t based on both the action value $Q(s_t; a)$ and the uncertain estimate $U(s_t; a)$, which can be written as:

$$a_t = \text{argmax}_a(Q(s_t; a) + U(s_t; a)); \tag{15}$$

$$U(s_t; a) = cP(a|s_t)\frac{\sqrt{\sum_{a' \in \mathcal{A}_t} N(s_t; a')}}{1 + N(s_t; a)}; \tag{16}$$

where c is a hyper parameter to control the level of exploration of MCTS, $P(a|s_t)$ is a prior probability, and $N(s_t; a)$ is the visit count.

Evaluation: When reaching a leaf node s_t, the value $V(s_t)$ of the node is estimated either with the value function $V\ (s_t)$ or with the terminal reward $\mathcal{R}_T$ depending on whether or not the node is a terminal node (i.e., the end of an episode). Formally, the value $V(s_t)$ is:

$$V(s_t) = \begin{cases} \mathcal{R}_T; & \text{when } s_t \text{ is a terminal node;} \\ V\ (s_t); & \text{otherwise.} \end{cases} \tag{17}$$

Expansion: If s_t is a non-terminal leaf node, we will expand the tree by adding all child nodes (corresponding to every action $a \in \mathcal{A}_t$) of node s_t. We initialize each new child node with $P(a|s_t) = \ (a|s_t); Q(s_t; a) = 0; N(s_t; a) = 0$.

Backup: We recursively backup the element $Q; N$ of all tree nodes according to the path $\mathcal{P}_t$ by:

$$Q(s; a) \leftarrow \frac{V(s) + Q(s; a) \times N(s; a)}{N(s; a) + 1}; \tag{18}$$

$$N(s; a) \leftarrow N(s; a) + 1. \tag{19}$$

After reaching the maximum simulation time, we randomly choose the action according to searching policy $_e$. A softmax function with temperature is used to get probabilities for every action based on the visit count $N(s; a)$.

$$_e(a_t|s_t) = \frac{\exp\{N(s_t; a_t)^{1=}\}}{\sum_{a' \in \mathcal{A}_t} \exp\{N(s_t; a')^{1=}\}}. \tag{20}$$

4.4. Learning and prediction

Our policy is learned to mimic the searching policy $_e$ by a cross entropy loss and value function V is learned to predict the final reward $\mathcal{R}_T$ by a mean square error.

$$\mathcal{L} = (\mathcal{R}_T - V\ (s_t))^2 - \ _e(s_t)^\top log \quad (s_t) + \quad \| \ \|^2; \tag{21}$$

where is a parameter controlling the level of l_2 weight regularization. Please note that we optimize and $_e$ interchangeably, i.e., we fix to optimize $_e$, and vice versa.

During test time, we select action $a_t = \text{argmax}_{a \in \mathcal{A}_t} \ (a|s_t)$ for each step. When reaching a terminal state s_T, we generate a predicted opinion label sequence $\mathcal{L}_T^o$ based on $\mathcal{P}_T = \{(e_i; w_{\ (i)}, l_i^o); i \in [0; T]\}$, where the (i)-th label in $\mathcal{L}_T^o$ is l_i^o. For the words in sentence $\mathcal{W}$ which not exists in the path $\mathcal{P}_T$, we assign the label O to these words.

5. Experiment

5.1. Datasets and metrics

We evaluate our model on four widely used datasets generated by Fan et al. [1]. The 14res and 14lap are derived from the SemEval challenge 2014 Task4 [51], 15res is from SemEval challenge 2015 Task12 [52] and 16res is from SemEval challenge 2016 Task5 [53]. The suffixes "res" and "lap" mean that the reviews are from restaurant domain and laptop domain, respectively. Each sample in these datasets consists of a review sentence, a given aspect term in the sentence and its corresponding opinion terms.

We use the metrics precision, recall, and F1 score to measure the performance of baselines and our model. An opinion term is considered to be a correct prediction when the position (i.e., the beginning and the ending offset) of the term as well as the label of the term are both predicted correctly.

5.2. Settings

We initialize word embedding vectors with 300 dimension Glove [54] vectors which are pre-trained on 840 billion words. We fixed the word vectors during training. We randomly initialize the edge embeddings and label embeddings, and learn these embeddings during training. For GNN, we empirically set the number of layers as 3. The l_2 weight regularization is $1e^{-5}$, hyper parameter in reward function is 0.3 and c in MCTS is 5.

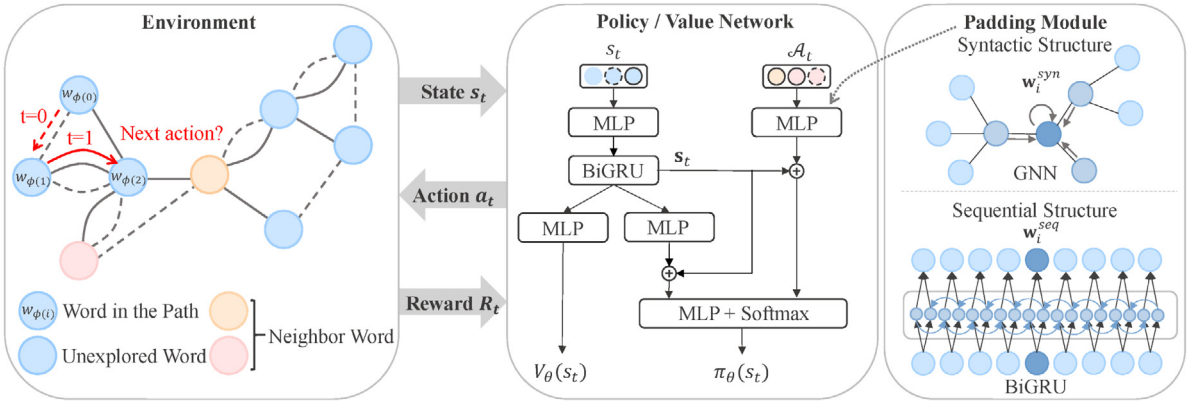

Fig. 3. The pipeline of our model to explore a path on the graph. The goal is to find a good path from aspect term to opinion term and predict correct label for every word to distinguish if it is an opinion word. The left part of the figure shows that the agent first chooses a sequential edge (dashed line) at $t = 0$ and a syntactic edge (solid line) at $t = 1$. After that, the agent will take the next action by choosing an edge, going to a neighbor word through it and predicting the opinion label of the neighbor word. The right part of the figure shows the framework of our model, we also use dashed border and solid border of circles to distinguish the word is connected by a sequential edge or a syntactic edge.

Table 1
Main experiment results(%). Best results are in bold (for P, R, and F1 score, the larger is the better).

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Distance-rule	58.39	43.59	49.92	50.13	33.86	40.42	54.12	39.96	45.97	61.90	44.57	51.83
Dependency-rule	64.57	52.72	58.04	45.09	31.57	37.14	65.49	48.88	55.98	76.03	56.19	64.62
LSTM	52.64	65.47	58.34	55.71	57.53	56.52	57.27	60.69	58.93	62.46	68.72	65.33
BiLSTM	58.34	61.73	59.95	64.52	61.45	62.71	60.46	63.65	62.00	68.68	70.51	69.57
Pipeline	77.72	62.33	69.18	72.58	56.97	63.83	74.75	60.65	66.97	81.46	67.81	74.01
TC-BiLSTM	67.65	67.67	67.61	62.45	60.14	61.21	66.06	60.16	62.94	73.46	72.88	73.10
IOG	82.85	77.38	80.02	73.24	69.63	71.35	76.06	70.71	73.25	82.25	78.51	81.69
LOTN	84.00	80.52	82.21	77.08	67.62	72.02	76.61	70.29	73.29	86.57	80.89	83.62
ONG(Glove)	82.36	79.81	81.07	74.95	70.69	72.76	81.34	71.60	76.16	86.53	83.24	84.85
PER(Our)	**86.43**	**80.39**	**83.30**	**80.68**	**70.72**	**75.38**	**81.50**	**75.05**	**78.14**	**90.00**	**84.00**	**86.90**

5.3. Baselines

We compare our model with the following baselines:

Distance-rule [10]: this method uses POS tags and distances to extract the opinion words, and it chooses the nearest adjective to the aspect term as the opinion term.

Dependency-rule [11]: it learns the dependency paths with POS tags from aspect word to opinion word, then uses the high frequency dependency templates to extract from the test data .

LSTM/BiLSTM [36]: this approach employs word embeddings to represent words, put them into a LSTM or BiLSTM, and makes a 3-class classification for every hidden state. It is a sentence-level opinion words extraction.

Pipeline [1]: it combines BiLSTM and distance rule. After getting the result of BiLSTM, it chooses the nearest opinion term to the aspect term as the final result.

TC-BiLSTM: this method follows the design of the work for target-oriented sentiment classification [7]. This method uses the average embedding of the aspect term as the aspect vector, concatenates it to every word embedding of the sentence, and puts them into BiLSTM to do sequence labeling.

IOG [1]: the authors use six different positional and directional LSTMs to extract opinion terms of the aspect term.

LOTN [2]: it transfers sentiment classification task into TOWE task to gather more opinion knowledge via an auxiliary learning signal.

ONG [3]: this method introduces distance information of syntactic structure into extraction. It employs BERT to get word embeddings in the paper and we reproduce the model by using Glove vectors for a fair comparison.

5.4. Results

Table 1 shows the performance of our model and baseline models. We can observe that our proposed model achieves the best performance among all methods on the four datasets.

The performance of Distance-rule is unsatisfactory due to the fact that it only finds the single word as the opinion word. Dependency-rule achieves a better performance than Distance-rule but it is still considerably worse than other sequence-labeling based approaches. LSTM and BiLSTM perform not well in the task because they will extract the same opinion terms for different aspect terms in the sentence. The Pipeline extracts the nearest opinion term of the aspect term and obtains a high performance as compared with LSTM/BiLSTM. TC-LSTM performs worse than Pipeline, as it concatenates the aspect vector to each word, while neglecting the position information of the aspect term. IOG is better than other baselines, but it suffers from the high model complexity and no supplementary information. LOTN transfers sentiment classification task into TOWE and get better results, but it relies on leveraging external information, which limits its applications in real-world scenario. ONG leverages the distance information in syntactic structure and it performs better than other baselines. However, ONG ignores the dependency relations of syntactic structure, which results in a sub-optimal performance. Our model PER achieves significant improvements over all the baselines. For example, on the dataset 14res, PER outperforms the state-of-the-art method ONG by 4.07%, 0.58%, 2.23% in terms of P, R, F1, respectively. Similar improvements can also be observed on the other three datasets. The results verify the effectiveness of introducing the two structures, i.e., sequential structure and syntactic structure, as well as employing RL in the task of TOWE.

Table 2
Experiment results(%) of analyzing the effects of two structures and padding module.

Model	14res			14lap			15res			16res		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
No sequential part	83.26	74.85	78.83	72.62	65.96	69.13	77.10	68.97	72.81	85.84	77.33	81.36
No syntactic part	83.22	72.72	77.62	74.60	66.31	70.21	73.65	66.33	69.80	83.93	75.62	79.56
no padding	81.49	75.24	78.24	71.29	65.26	68.14	71.80	72.82	72.31	83.06	77.52	80.20
No sequential padding	83.19	74.95	78.86	72.19	64.55	68.16	77.20	69.37	73.08	85.03	77.90	81.31
No syntactic padding	86.38	76.99	81.42	77.69	68.78	72.97	80.27	72.62	76.25	85.89	81.14	83.45
PER(Our)	**86.43**	**80.39**	**83.30**	**80.68**	**70.72**	**75.38**	**81.50**	**75.05**	**78.14**	**90.00**	**84.00**	**86.90**

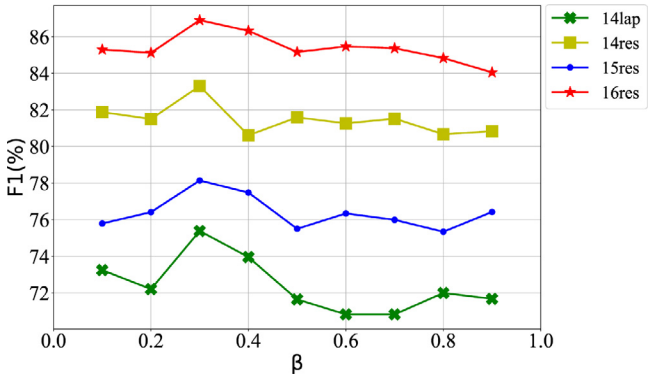

Fig. 4. The effect of parameter .

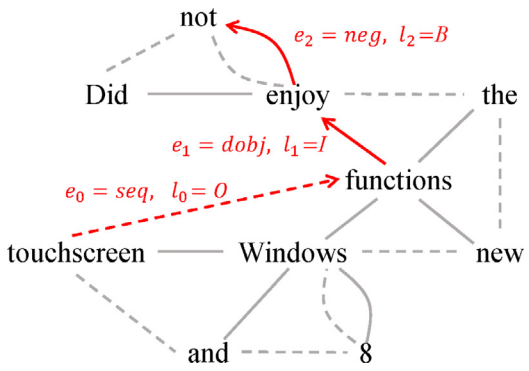

Fig. 5. A multiplex heterogeneous graph of the third sentence in Table 3. We show the reasoning path explored by the agent.

5.5. Ablation study

In order to learn the effects of different parts of our model, we compare PER with the following variations: (i) **no sequential part**: we remove the sequential structure (dashed line in Fig. 3) from our model, and also remove BiGRU from padding module because there is no sequence structure information, (ii) **no syntactic part**: we remove the syntactic structure (solid line in Fig. 3) from our model, and also remove GNN from padding module, (iii) **no padding**: we remove the padding module and let the agent make decision only by 1-hop neighbor information, (iv) **no sequential padding**: we remove the BiGRU from the padding module, (v) **no syntactic padding**: we remove the GNN from the padding module. From Table 2, we can know that the two structures are important and can compensate each other for extracting opinion words. For example, on the dataset 14res, if we remove sequential part, the performance decrease 3.17%, 5.54%, 4.47% in terms of P, R, F1, respectively. Similar trends can also be observed on other datasets when remove different parts. Meanwhile, both parts of padding module play an important role in guiding the exploration in the graph. And if we remove the whole padding module, the performance is worse than removing either the sequential padding or the syntactic padding.

5.6. The effect of parameter

In this section, we study the sensitivity of PER to the parameter , and explore how the different values of would affect our model performance. Fig. 4 shows the F1 score of PER by varying from 0 to 1 with a step size 0.1 on all datasets. From the figure, we can observe that with the increase of the value of , the F1 score first gradually increases until it reaches the highest performance at = 0:3, and then starts to decline.

This result shows that our propose model prefer a relative small (e.g., 0.3), which means the syntactic structure will provide more explainable information in our model and the agent prefers to choose more syntactic edges than sequential edges.

5.7. Case study

Table 3 presents the results of our method PER and the best performing baseline ONG on some cases and analyze the strengths and weaknesses of them. The first case shows that ONG predicts a wrong opinion word "anywhere", this is because ONG uses the distances between the words and the aspect term in the dependency trees to measure how likely a word is an opinion word. However, both words "fast" and "anywhere" have the same distance to the aspect term "Boot time" on dependency tree, which cause ONG predicts "anywhere" as an opinion word. The second case shows that ONG only extract the opinion word "easy" while neglect the opinion word "intuitive". The third case, ONG could not extract any opinion words. In contrast, our proposed approach PER correctly extracts all opinion terms.

5.8. Path analysis

In Table 4, we show the reasoning paths generated by PER for every sentence in Table 3. We can find that the agent can leverage both sequential information and syntactic information to explore a path which starts from the first word of aspect term and covers all opinion terms. Though we set = 0:3 to let the agent tend to choose syntactic edges, it can still find the relations from context and use sequential edges to explore a shorter path in the third case. We also show the multiplex heterogeneous graph and the reasoning path of the third sentence in Fig. 5. PER can get the correct result with a good quality path in such a complex situation.

As we shown in Fig. 5, the agent starts moving from "touchscreen" which is the first word of the given aspect term "touchscreen functions". The agent first chooses a sequential edge, moves to the neighbor word "functions", and predicts the label *O* for the word "functions" which indicates it is not an opinion word. Then the agent chooses a syntactic edge, moves to word "enjoy", and predicts the label *I* for it, indicating "enjoy" is a word in the opinion term, but not the beginning word of the

Table 3
Cases of the extracted results of our method and the best performing baseline method. The aspect terms are underlined and the corresponding opinion words are in bold. The "$\mathcal{NULL}$" represents that the prediction is empty.

SENTENCE	ONG	PER
Boot time is super **fast**, around anywhere from 35 s to 1 min.	fast, anywhere	fast
Everything is so **easy** and **intuitive** to setup or configure.	easy	easy, intuitive
Did **not enjoy** the new Windows 8 and touchscreen functions.	$\mathcal{NULL}$	not enjoy

Table 4
The paths explored by PER of the sentences in Table 3. The type of chosen edge is shown above the arrows between two words and the opinion label predicted by the agent is shown in the parentheses behind the word. The meaning of the dependency relations in the examples: *acomp: adjectival complement, compound: compound, conj: conjunct, dobj: direct object, neg: negation modifier, nmod: nominal modifier, nsubj: nominal subject, xcomp: open clausal complement*. And the relation *seq* is the sequential edge.

Boot $\xrightarrow{compound}$ time(O) $\xrightarrow{nusbj}$ is(O) $\xrightarrow{acomp}$ fast(B) $\longrightarrow$ **STOP**
configure $\xrightarrow{conj}$ setup(O) $\xrightarrow{xcomp}$ easy(B) $\xrightarrow{conj}$ intuitive(B) $\longrightarrow$ **STOP**
touchscreen $\xrightarrow{seq}$ functions(O) $\xrightarrow{dobj}$ enjoy(I) $\xrightarrow{neg}$ not(B) $\longrightarrow$ **STOP**

opinion term. In the next step, the agent chooses a syntactic edge, moves to word "not", and predicts the label *B* for it, i.e. "not" is the beginning word of the opinion term. The agent prefers to choose the syntactic edge when there exists two kinds of edges between words "not" and "enjoy", because we set a relative small . Finally, the agent thinks all opinion words have been explored. Therefore, it chooses **STOP** action to stop reasoning over the graph.

6. Conclusion

In this paper, we propose a novel deep reinforcement learning model for the TOWE task. We first propose a multiplex heterogeneous graph which captures both sequential structure and syntactic structure. Then, we formulate the TOWE task as a Markov Decision Process (MDP) to reason over the graph for inferring corresponding opinion for each aspect. To our knowledge, it is the first time that RL approach has been explicitly explored and applied in this task. And to fully use the rich information of distant nodes in the state, we design a padding module to aggregate them. Experimental results on four widely used datasets show that our model consistently outperforms all baselines.

CRediT authorship contribution statement

Yaqing Dai: Conceptualization, Methodology, Software, Writing – original draft. **Pengfei Wang:** Writing – original draft, Writing – review & editing, Supervision. **Xiaofei Zhu:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [No. 61802029] and the Federal Ministry of Education and Research, China [No. 01LE1806A].

References

[1] Z. Fan, Z. Wu, X.-Y. Dai, S. Huang, J. Chen, Target-oriented opinion words extraction with target-fused neural sequence labeling, in: NAACL-HLT, 2019, pp. 2509–2518.
[2] Z. Wu, F. Zhao, X.-Y. Dai, S. Huang, J. Chen, Latent opinions transfer network for target-oriented opinion words extraction, in: AAAI, 2020, pp. 9298–9305.
[3] A. Pouran Ben Veyseh, N. Nouri, F. Dernoncourt, D. Dou, T.H. Nguyen, Introducing syntactic structures into target opinion word extraction with deep learning, in: EMNLP, 2020, pp. 8947–8956.
[4] Y. Zhou, W. Jiang, P. Song, Y. Su, T. Guo, J. Han, S. Hu, Graph convolutional networks for target-oriented opinion words extraction with adversarial training, in: IJCNN, 2020, pp. 1–7.
[5] J. Jiang, A. Wang, A. Aizawa, Attention-based relational graph convolutional network for target-oriented opinion words extraction, in: EACL, 2021, pp. 1986–1997.
[6] B. Liu, Sentiment analysis and opinion mining, in: Synthesis Lectures on Human Language Technologies, 2012.
[7] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: COLING, 2016, pp. 3298–3307.
[8] M. Zhang, T. Qian, Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis, in: EMNLP, 2020, pp. 3540–3549.
[9] P. Zhu, Z. Chen, H. Zheng, T. Qian, Aspect aware learning for aspect category sentiment analysis, ACM Trans. Knowl. Discov. Data 13 (6) (2019) 55:1–55:21.
[10] M. Hu, B. Liu, Mining and summarizing customer reviews, in: ACM SIGKDD, 2004, pp. 168–177.
[11] L. Zhuang, F. Jing, X.-Y. Zhu, Movie review mining and summarization, in: CIKM, 2006, pp. 43–50.
[12] H. Ye, Z. Yan, Z. Luo, W. Chao, Dependency-tree based convolutional neural networks for aspect term extraction, in: J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, Y.-S. Moon (Eds.), PAKDD, 2017, pp. 350–362.
[13] Y. Yin, C. Wang, M. Zhang, PoD: Positional dependency-based word embedding for aspect term extraction, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 1714–1719.
[14] Y. Yin, F. Wei, L. Dong, K. Xu, M. Zhang, M. Zhou, Unsupervised word and dependency path embeddings for aspect term extraction, in: IJCAI, 2016, pp. 2979–2985.
[15] F. Scarselli, M. Gori, A.C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Trans. Neural Netw. 20 (1) (2009) 61.
[16] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, in: ICLR, 2018.
[17] J. Chen, M. Zhong, J. Li, D. Wang, T. Qian, H. Tu, Effective deep attributed network representation learning with topology adapted smoothing, IEEE Trans. Cybern. (2021) 1–12.
[18] G. Xue, M. Zhong, J. Li, J. Chen, C. Zhai, R. Kong, Dynamic network embedding survey, 2021.
[19] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: ICLR, 2017.
[20] D. Silver, A. Huang, C.J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, D. Hassabis, Mastering the game of go with deep neural networks and tree search, Nature (2016) 484–489.
[21] A. Heuillet, F. Couthouis, N.D. Rodríguez, Explainability in deep reinforcement learning, Knowl. Based Syst. 214 (2021) 106685.
[22] M. Chen, Y. Chen, Y. Du, L. Wei, Y. Chen, Heuristic algorithms based on deep reinforcement learning for quadratic unconstrained binary optimization, Knowl. Based Syst. 207 (2020) 106366.
[23] Y. Fan, J. Guo, Y. Lan, J. Xu, C. Zhai, X. Cheng, Modeling diverse relevance patterns in ad-hoc retrieval, in: SIGIR, 2018, pp. 375–384.
[24] L. Chen, Z. Tang, G.H. Yang, Balancing reinforcement learning training experiences in interactive information retrieval, in: SIGIR, 2020, pp. 1525–1528.
[25] M. Ranzato, S. Chopra, M. Auli, W. Zaremba, Sequence level training with recurrent neural networks, in: ICLR, 2016.
[26] X. Kang, Y. Zhao, J. Zhang, C. Zong, Dynamic context selection for document-level neural machine translation via reinforcement learning, in: EMNLP, 2020, pp. 2242–2254.

[27] L. Zou, L. Xia, Z. Ding, J. Song, W. Liu, D. Yin, Reinforcement learning to optimize long-term user engagement in recommender systems, in: KDD, 2019, pp. 2810–2818.
[28] X. Zhao, L. Xia, L. Zhang, Z. Ding, D. Yin, J. Tang, Deep reinforcement learning for page-wise recommendations, in: RecSys, 2018, pp. 95–103.
[29] L. Huang, M. Fu, F. Li, H. Qu, Y. Liu, W. Chen, A deep reinforcement learning based long-term recommender system, Knowl. Based Syst. 213 (2021) 106706.
[30] E. Choi, D. Hewlett, J. Uszkoreit, I. Polosukhin, A. Lacoste, J. Berant, Coarse-to-fine question answering for long documents, in: ACL, 2017, pp. 209–220.
[31] Y. Hua, Y. Li, G. Haffari, G. Qi, T. Wu, Few-shot complex knowledge base question answering via meta reinforcement learning, in: EMNLP, 2020, pp. 5827–5837.
[32] Y. Shen, J. Chen, P. Huang, Y. Guo, J. Gao, M-Walk: Learning to walk over graphs using Monte Carlo tree search, in: NeurIPS, 2018, pp. 6787–6798.
[33] K. Zhao, X. Wang, Y. Zhang, L. Zhao, Z. Liu, C. Xing, X. Xie, Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs, in: ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR, 2020, pp. 239–248.
[34] H. Xu, B. Liu, L. Shu, P.S. Yu, Double embeddings and CNN-based sequence labeling for aspect extraction, in: Proceedings of ACL, 2018, pp. 592–598.
[35] G. Qiu, B. Liu, J. Bu, C. Chen, Opinion word expansion and target extraction through double propagation, Comput. Linguist. 37 (1) (2011) 9–27.
[36] P. Liu, S. Joty, H. Meng, Fine-grained opinion mining with recurrent neural networks and word embeddings, in: EMNLP, 2015, pp. 1433–1443.
[37] W. Wang, S.J. Pan, D. Dahlmeier, X. Xiao, Coupled multi-layer attentions for co-extraction of aspect and opinion terms, in: Proceedings of AAAI, 2017, pp. 3316–3322.
[38] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, 2017.
[39] X. Zhao, L. Zhang, Z. Ding, L. Xia, J. Tang, D. Yin, Recommendations with negative feedback via pairwise deep reinforcement learning, in: KDD, 2018, pp. 1040–1048.
[40] R. Das, S. Dhuliawala, M. Zaheer, L. Vilnis, I. Durugkar, A. Krishnamurthy, A. Smola, A. Mccallum, Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning, in: ICLR, 2017.
[41] Y. Feng, J. Xu, Y. Lan, J. Guo, W. Zeng, X. Cheng, From greedy selection to exploratory decision-making: Diverse ranking with policy-value networks, in: SIGIR, 2018, pp. 125–134.
[42] Y. Xian, Z. Fu, S. Muthukrishnan, G. de Melo, Y. Zhang, Reinforcement knowledge graph reasoning for explainable recommendation, in: ACM SIGIR, 2019, pp. 285–294.
[43] S. Wang, M. Yu, X. Guo, Z. Wang, T. Klinger, W. Zhang, S. Chang, G. Tesauro, B. Zhou, J. Jiang, R^3: Reinforced Ranker-Reader for Open-Domain Question Answering, in: AAAI, 2018, pp. 5981–5988.
[44] W. Xiong, T. Hoang, W.Y. Wang, DeepPath: A reinforcement learning method for knowledge graph reasoning, in: EMNLP, 2017, pp. 564–573.
[45] G. Wan, S. Pan, C. Gong, C. Zhou, G. Haffari, Reasoning like human: Hierarchical reinforcement learning for knowledge graph reasoning, in: IJCAI, 2020, pp. 1926–1932.
[46] D. Lei, G. Jiang, X. Gu, K. Sun, Y. Mao, X. Ren, Learning collaborative agents with rule guidance for knowledge graph reasoning, in: EMNLP, 2020, pp. 8541–8547.
[47] S. Moon, P. Shah, A. Kumar, R. Subba, OpenDialKG: Explainable conversational reasoning with attention-based walks over knowledge graphs, in: Proceedings of the 57th ACL, 2019, pp. 845–854.
[48] M. Hildebrandt, J.A.Q. Serna, Y. Ma, M. Ringsquandl, M. Joblin, V. Tresp, Reasoning on knowledge graphs with debate dynamics, in: AAAI, 2020, pp. 4123–4131.
[49] L. Ramshaw, M. Marcus, Text chunking using transformation-based learning, in: Third Workshop on Very Large Corpora, 1995.
[50] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A.a. Bolton, Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359.
[51] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, SemEval (2014) 27–35.
[52] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, SemEval-2015 task 12: Aspect based sentiment analysis, in: SemeVal, 2015, pp. 486–495.
[53] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. de clercq, V. Hoste, M. Apidianaki, X. Tannier, N. Loukachevitch, E. Kotelnikov, N. Bel, S.M. Zafra, G. Eryiğit, SemEval-2016 Task 5: Aspect based sentiment analysis, in: SemeVal, 2016, pp. 19–30.
[54] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, 14, 2014, pp. 1532–1543, http://dx.doi.org/10.3115/v1/D14-1162,