Aspect sentiment triplet extraction incorporating syntactic constituency parsing tree and commonsense knowledge graph

Zhenda HU

Zhaoxia WANG
Singapore Management University, zxwang@smu.edu.sg

Yinglin WANG
Shanghai University of Finance and Economics

Ah-hwee TAN
Singapore Management University, ahtan@smu.edu.sg

Cognitive Computation
https://doi.org/10.1007/s12559-022-10078-4

Aspect Sentiment Triplet Extraction Incorporating Syntactic Constituency Parsing Tree and Commonsense Knowledge Graph

Zhenda Hu[1] · Zhaoxia Wang[2] · Yinglin Wang[1] · Ah-Hwee Tan[2]

Received: 16 July 2022 / Accepted: 15 November 2022

Abstract
The aspect sentiment triplet extraction (ASTE) task aims to extract the target term and the opinion term, and simultaneously identify the sentiment polarity of target-opinion pairs from the given sentences. While syntactic constituency information and commonsense knowledge are both important and valuable for the ASTE task, only a few studies have explored how to integrate them via flexible graph convolutional networks (GCNs) for this task. To address this gap, this paper proposes a novel end-to-end model, namely GCN-EGTS, which is an enhanced Grid Tagging Scheme (GTS) for ASTE leveraging syntactic constituency parsing tree and a commonsense knowledge graph based on GCNs. Specifically, two types of GCNs are developed to model the information involved, namely span GCN for syntactic constituency parsing tree and relational GCN (R-GCN) for commonsense knowledge graph. In addition, a new loss function is designed by incorporating several constraints for GTS to enhance the original tagging scheme. The extensive experiments on several public datasets demonstrate that GCN-EGTS outperforms the state-of-the-art approaches significantly for the ASTE task based on the evaluation metrics. The outcomes of this research indicate that effectively incorporating syntactic constituency parsing information and commonsense knowledge is a promising direction for the ASTE task.

Keywords Aspect sentiment triplet extraction · Syntactic constituency parsing tree · Commonsense knowledge graph · Graph convolutional network

Introduction

Aspect-based sentiment analysis (ABSA) is an aggregation of several fine-grained sentiment analysis tasks [1]. The aspect sentiment triplet extraction (ASTE) task [2] is the most recently proposed subtask of ABSA, aiming to extract the target term and decide its associated sentiment polarity, and simultaneously extract the option term to explain the reason for the sentiment polarity. As a result, the corresponding sentiment triplets (target term, option term, sentiment) can be obtained for the given sentences. Sentiment triplets make the results of sentiment analysis more complete and more interpretable: the target term provides information on the evaluation target, the sentient gives the sentiment polarity of the target, and the opinion term explains the reason why the sentiment is given. For instance, for the sentence "The coffee is great but hot dogs are so so", it aims to extract two triplets (coffee, great, positive) and (hot dogs, so so, negative).

Due to the necessity of extracting these three elements simultaneously, the ASTE task is promising, but challenging. The initial method designed for the ASTE task was a two-stage pipeline model. In the first stage, target terms and opinion terms are extracted, and then in the second stage, the extracted target and opinion terms are paired to identify the sentiment [2]. However, such pipeline approaches usually suffer from the error propagation problem and cannot capture the relationship of the three elements accurately. Recently, end-to-end models become mainstream for the

✉ Yinglin Wang
wang.yinglin@shufe.edu.cn

Zhenda Hu
huzhenda2020@gmail.com

Zhaoxia Wang
zxwang@smu.edu.sg

Ah-Hwee Tan
ahtan@smu.edu.sg

1 School of Information Management and Engineering, Shanghai University of Finance and Economics, 777 Guoding Road, Shanghai 200433, China

2 School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, Singapore 178902, Singapore

Published online: 28 December 2022

ASTE task [3–6], which can jointly extract the sentient triplets. Among these models, an innovative tagging scheme, Grid Tagging Scheme (GTS), was proposed by [4], which has the capability of extracting all sentiment triplets simultaneously via one unified grid tagging task. The compatibility and effectiveness of GTS have been demonstrated [7].

However, there is a gap between a GTS model's performance on triplets of single and multiple words [6]. The term "multi-word triplet" refers to having at least one target word or opinion term that is made up of more than one word, as opposed to the term "single-word triplet" [6]. Therefore, improving the performance on such difficult multi-word triples is a promising direction. Recently, there have been successful studies for enhancing the performance of ABSA using syntax information [8–10]. Instead of constituent trees, they employed syntactic dependency representations [7]. The dependency edges represent the inter-relations between arguments and predicates, while the constituency structure locates more about phrase boundaries of argument spans, and then directs the paths to the predicate globally. Therefore, the constituency structures which locate more about phrase boundaries of argument spans have the potential to contribute to multi-word triples.

Some researchers tried to introduce external sentiment knowledge to provide additional information for sentiment analysis tasks, for improving the performance and enhancing the generalization ability of the models [11–14]. Specifically, external sentiment knowledge mainly includes explicit sentiment lexicons [15], syntactic information [7, 16, 17], structured knowledge base [12, 18] and commonsense knowledge graph [11, 16]. It has been proved that the utilization of external knowledge can improve the performance of sentiment analysis systems. Moreover, meta-based self-training for ABSA [19], detecting neutrality in sentiment analysis [20] and sentiment sensing with ambivalence handling [21] are also challenging topics, which represent good examples of utilization of external knowledge as reflected in the good results obtained. Especially for commonsense knowledge graph, it can provide additional information about sentiment domain knowledge which is not contained in the given sentences [22, 23].

However, only a few studies have explored how to simultaneously integrate such syntactic constituency information and commonsense knowledge graph via graph convolutional networks (GCNs) for ASTE. GCNs are extensively applied in several NLP tasks such as text classification [24], machine translation [25] and semantic role labelling [26], which have been demonstrated to be flexible and effective. Because both syntactic constituency parsing tree and commonsense knowledge are based on graph-structured data, GCNs can be used to encode both of them.

Therefore, this paper proposes an enhanced GTS model, namely GCN-EGTS, which utilizes GTS as our basic tagging scheme, and fuses syntactic constituency information and commonsense knowledge via GCN. First, GCN-EGTS adopts two frequently used deep learning models including CNN and LSTM, to encode the input sentences, and contextual semantic embeddings for each word in the given sentence can be obtained. Next, GCN-EGTS applies span GCN to model the syntactic constituency parsing tree constructed for each sentence. Then, GCN-EGTS utilizes relational GCN (R-GCN) to model the extracted subgraph from ConceptNet. Finally, an inference strategy and an updated loss function are designed for extracting more accurate triplets. To verify the effectiveness of GCN-EGTS, the extensive experiments are also conducted on four public datasets designed for ASTE.

In summary, this paper contributes to the following areas:

- This research proposes an enhanced GTS model, named GCN-EGTS, which incorporates syntactic constituency information and commonsense knowledge through GCN for the ASTE task.
- The proposed model fuses the syntactic constituency parsing information about boundaries and utilizes information about the word's neighbourhood in the constituent structure as well as syntactic labels of constituents for enhancing word representations via span GCN.
- The proposed model leverages an external knowledge graph, ConceptNet, to learn domain-specific commonsense knowledge features of concepts via R-GCN, and adds the commonsense knowledge to enhance the ASTE task.
- The results of extensive experiments demonstrate that GCN-EGTS outperforms the state-of-the-art approaches significantly.

Related Work

ASTE is one of the most recent sub-tasks of ABSA, first defined by [2], which aims to extract all targets and corresponding opinion terms in the given sentences, and simultaneously identify associated sentiments and complete the matching of the three parts to form a triple (target, sentiment, opinion). Peng et al. [2] adopted a two-stage pipeline model. The first stage is to predict all target words and associated sentiment polarities, as well as to predict all sentiment words that may describe the target word. The second stage is to pair the target words with sentiment and the sentiment words. One problem of these pipeline methods is that they cannot take full advantage of the relationship between the three elements in the ternary combination. The other problem is the error propagation issue.

To address these problems, several end-to-end models were developed [5, 6]. Xu et al. [5] designed an end-to-end model, namely JET, which first designed a new set of tags with a stronger expression ability and directly avoided the disadvantages of extracting incomplete features caused by using the original model in stages. The model mainly calculated the label of each word according to the relationship between the three elements to obtain the best label order using the conditional random field (CRF). Span ASTE [6] explicitly considered the interaction between phrases in ASTE task, which overcame the issue of incomplete extraction and sentiment conflicts in the existing methods. Chen et al. [7] designed a graph-sequence dual representation and modelling paradigm for the ASTE task using graph neural networks (GNNs) and also demonstrated feasible. Trueman and Cambria [27] designed a convolutional stacked bidirectional LSTM with a multiplicative attention mechanism. Mao and Li [28] proposed a novel gating mechanism for the bridging of multi-task learning towers. Their method has been evaluated based on ABSA and sequential metaphor identification tasks.

Different from the work above, Balazs and Velásquez found that through the integration of information from various sources, fusion of knowledge and information could markedly strengthen sentiment analysis models by increasing the availability of data [29]. Such methods have been widely explored in various research fields, which can enhance semantic representation to improve the performance and generalization ability of sentiment analysis systems [29, 30]. Specifically, external sentiment knowledge mainly includes explicit sentiment lexicons [15], syntactic information [7, 16, 17], knowledge base [12, 18] and the missing information contained in the commonsense knowledge graph [11, 16, 22], which can provide useful sentiment clues.

For example, Sun et al. [17] proposed a convolution over a dependency tree (CDT) model for the ABSA task. The model introduced the dependency tree information parsed by Stanford parser to help identify the evaluation words related to the evaluation target, and combined the syntactic information features learned by GCN on the dependency tree. SenticLSTM [22] fused external knowledge from SenticNet to enhance the performance of targeted ABSA. Similarly, a domain adaptative model was designed by leveraging commonsense-related information for ABSA [11]. Dragoni et al. proposed OntoSenticNet 2 [23], a conceptual model for the structuring of emotion analysis from multimodal resources.

In addition, Zhou et al. proposed a new syntax- and knowledge-based GCN (SK-GCN) model for ABSA, utilizing the syntactic dependency tree and commonsense knowledge [16]. Zhao and Yu [12] proposed a knowledge-enabled BERT for ABSA. Specifically, they injected sentiment domain knowledge into the language representation model to take advantage of the additional information from the sentiment graph. Lu et al. [31] designed a syntactic knowledge adapter and commonsense knowledge adapter-based network to handle position information, syntactic structure and external knowledge.

Among earlier approaches that incorporated syntax into neural networks, some research [32, 33] utilized recursive neural networks to model constituency trees. However, there are few studies which utilize syntactic constituency information for the ASTE task. Inspired by the work for ASTE as well as ABSA above, this paper proposes a new method named GCN-EGTS, which incorporates syntactic constituency information and commonsense knowledge through GCN for the ASTE task.

Proposed Method

In this section, we propose an end-to-end model, namely GCN-EGTS, which is designed to accomplish the ASTE task. This section includes two sub-sections. Firstly, in the "Task Definition and Preliminaries" sub-section, we describe the ASTE task definition and preliminaries, which include GTS, the triplets decoding strategy and the GCN model. Next, in the "GCN enhanced-GTS (GCN-EGTS)" sub-section, we present our proposed GCN-EGTS model in detail, which utilizes constituency parsing tree and commonsense knowledge, followed by the inference strategy and an enhanced loss function. The overall architecture of GCN-EGTS is presented in Fig. 1, in which constituency parsing tree encoding is on the right part and commonsense knowledge encoding is on the left part. Before describing the details of the proposed model, we introduce the task definition and preliminaries first.

Task Definition and Preliminaries

ASTE Task Definition Here we define the ASTE task. Given a sentence $s = \left\{w_1, w_2, \cdots, w_n\right\}$ consisting of n words, the objective of ASTE is to extract a set of opinion pairs $\{a, o\}$ from s, and identify the sentiment polarity c of the opinion pair at the same time. As a result, several aspect sentiment triplets $\{a, o, c\}$ can be obtained for the given sentence.

GTS We adopt an effective tagging scheme, GTS [4], as our basic scheme to complete the task. The overall structure of GTS can be seen in Fig. 2. In total, six tags $\{A, O, Pos, Neu, Neg, N\}$ are used for the relation of each word-pair $\left(w_i, w_j\right)$. A tagging example is given in Fig. 3. Specifically, A represents two words belonging to the same aspect term, while O represents two words belonging to the same opinion term. The three tags, Pos, Neu and Neg, denote the positive, neutral and negative sentiment polarities between a target-opinion pair, respectively. N represents no relation between the word-pair. Note that using GTS, all pairs of words are tagged in accordance with their

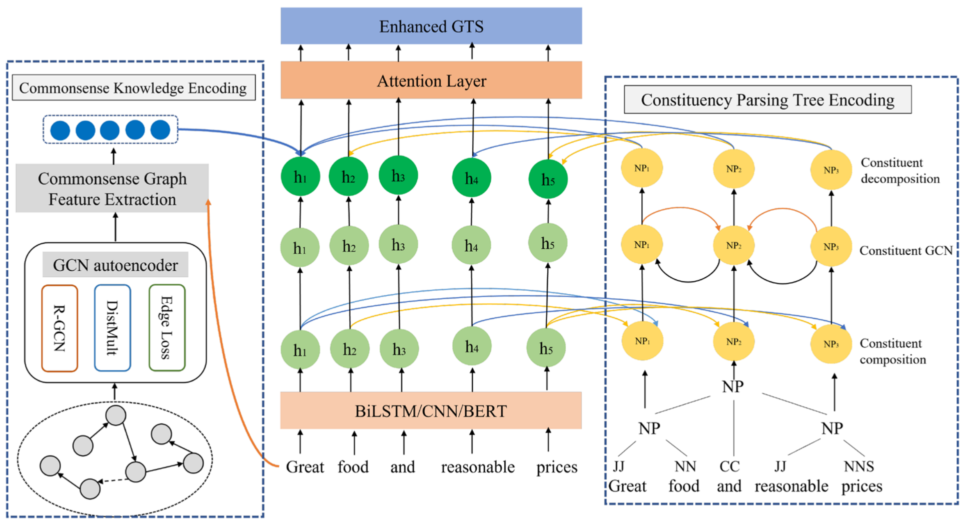

Fig. 1 The overall architecture of GCN-EGTS. Constituency parsing tree encoding is on the right part and commonsense knowledge encoding is on the left part. The new representation of each token with constituency parsing tree information and commonsense knowledge would be sent to enhanced GTS

relations by accomplishing end-to-end fine-grained sentiment extraction.

Triplets Decoding We adopt the decoding algorithm designed by [4]. First, the predictive tags of all word pairs (w_i, w_j) on the main diagonal are used to identify both aspects and opinions. It is taken as a complete aspect when the span contains continuous A, while it is considered a complete opinion when the span contains continuous O. Then, the predicted tags of all word pairs (w_i, w_j) are counted when $w_i \in a$ and $w_j \in o$. The sentiment label $s \in S$ with the maximum probability is considered as sentiment polarity for a triplet (a, o, s). If they are all predicted to be label N, it can be considered that a and o cannot form a triplet.

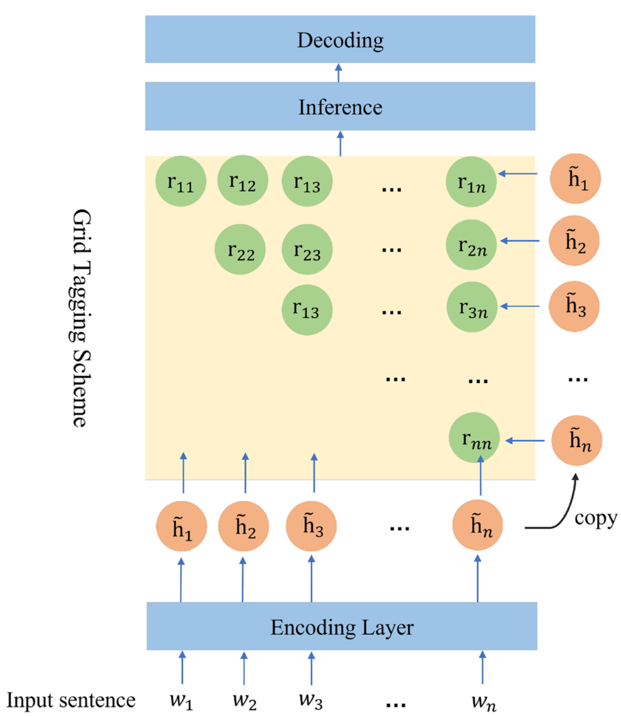

Fig. 2 The overall structure of GTS

Graph Convolutional Network (GCN) GCN was first developed by [34], which can be used to compute the node representation conditioned on the neighbouring nodes. The input to GCN is an undirected graph $G = (V, E)$, where V denotes the set of nodes and E denotes the set of edges. The node representation is calculated by the following equation:

$$h_v = ReLU\left(\sum_{u \in N(v)} Wh_u + b\right), \tag{1}$$

where $N(v)$ represents neighbours of v; $ReLU$ is the activation function. As mentioned in [35], a variant of GCN is adopted. The node representation is calculated by the update function:

$$h_v = ReLU\left(LayerNorm\left(\sum_{u \in N(v)} g_{v,u}(W_{T_f(u,v)}h_u + b_{T_f(u,v)})\right)\right), \tag{2}$$

	The	coffee	is	great	but	hot	dogs	are	so	so
The	N	N	N	N	N	N	N	N	N	N
coffee		A	N	Pos	N	N	N	N	N	N
is			N	N	N	N	N	N	N	N
great				O	N	N	N	N	N	N
but					N	N	N	N	N	N
hot						A	A	N	Neg	Neg
dogs							A	N	Neg	Neg
are								N	N	N
so									O	O
so										O

Fig. 3 A tagging case for ASTE. The aspect terms are highlighted in blue and opinion terms are in green

where $T_c(u, v)$ and $T_f(u, v)$ are fine-grained and coarse-grained versions of edge labels, and $g_(v, u)$ is the scalar gate to weight the contribution of each node in the neighbourhood.

GCN Enhanced-GTS (GCN-EGTS)

This sub-section presents the details of our proposed GCN-EGTS. First, GCN-EGTS adopts two frequently used deep learning models including convolutional neural networks (CNN) and long short-term memory (LSTM) to encode the input sentences, so that contextual semantic embeddings for each word in the given sentence can be obtained. In order to capture more abundant features, GCN-EGTS next applies span GCN to model the syntactic constituency parsing tree constructed for each sentence. Then, GCN-EGTS utilizes R-GCN to model the extracted sub-graph from ConceptNet. Finally, an inference strategy and an updated loss function are designed for extracting more accurate triplets.

Input Sentence Encoding

Given a sentence $s = \{w_1, w_2, \cdots, w_n\}$, CNN and LSTM [36] can be used as the encoder of GCN-EGTS to obtain the representation h_i of each word and then generate the representation r_{ij} of the word-pair (w_i, w_j).

CNN: Following the design of DE-CNN [33], an effective aspect term extraction model, we employ a stack of 4 CNN layers to encode the sentences and get the feature representation h_i for each word w_i.

LSTM: We adopt BiLSTM, which employs a standard forward LSTM and a backward LSTM, to encode the sentences, and then concatenate the hidden states in two LSTMs as the representation h_i for each word w_i.

BERT: We adopt BERT [37], which employs a multi-layer bidirectional transformer [38], to generate the contextual represenation h_i for each word w_i.

Syntactic Constituency Parsing Encoding

Adding syntactic constituency parsing information to the word representation can supplement the boundary information of multi-word spans. Marcheggiani and Titov [35] have demonstrated that using span GCN to fuse constituent trees information into word representations is an effective way. Following [35], we utilize span GCN to inject constituency syntax into word representations. Span GCN consists of three parts in total: constituent composition, constituent GCN and constituent decomposition. The details are as follows:

1. Constituent composition

   After obtaining the word representation from input sentence encoder, syntactic constituency information would be added to each word. Firstly, we conduct the syntactic constituency parsing for each sentence to get the constituency tree as shown in the right part of Fig. 1. For the sentence "Great food and reasonable prices", we can obtain the constituency tree "(([Great]JJ [food] NN)NP [and]CC ([reasonable]JJ [prices]NNS)NP)NP" after syntactic constituency parsing. A constituency tree is made up of words and constituents in the sentence. We need to encode each constituent (not including leaf nodes) and add the constituent representation to the words related to the constituent. For each constituent, there are a start token and an end token of its span (blue and yellow arrows in Fig. 1, respectively). For the example, for the constituent NP_1, "great" is the start token and "food" is the end token.
2. Constituent GCN

   This layer is for constituent nodes to exchange information which ensures the children nodes can get information from their parent node and vice versa. The constituents in the constituency tree correspond to the nodes in the graph. As shown in Fig. 1, orange and black arrows denote parent-to-children and children-to-parent information, respectively.
3. Constituent decomposition

   This layer is the inverse of the constituent composition. In this layer, the constituents transmit information to the tokens at the start and at the end of their spans.

   Through syntactic constituency parsing encoding, we can obtain a new representation of each word, $\hat{h}_i$, which has fused the syntactic constituency information.

Commonsense Knowledge Encoding

The process of commonsense knowledge encoding includes three parts: Commonsense Graph Construction, Knowledge Graph Pre-training and Commonsense Graph Feature Extraction as shown in Fig. 4. The details are presented as follows:

1. Commonsense Graph Construction

   For the construction of commonsense graph, we utilize ConceptNet [39] as our source knowledge graph. ConceptNet is a directed labelled graph and we use $G = (V, E, R)$ to represent it. Each triplet in ConceptNet is represented as $(v_i, r_{ij}, v_j) \in \mathcal{E}$, for example (coffee, AtLocation, restaurant). ConceptNet has about

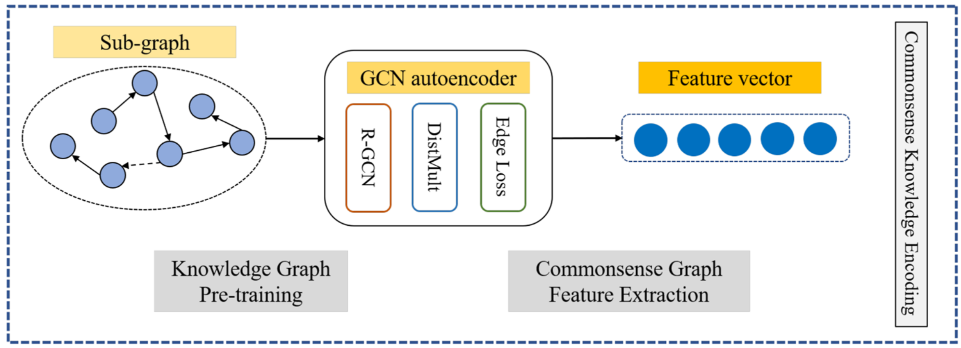

Fig. 4 The process of commonsense knowledge encoding

34 million triplets and we only extract a subset of the whole graph in our experiment. Firstly, we extract the set of all the unique nouns, adjectives and adverbs in the review sentences as the seeds. Then, a sub-graph can be obtained from ConceptNet through the seeds. Particularly, if the distance between the triplet and these seeds is within 1, the triplet would be extracted. As a result, we obtain a sub-graph $\bar{G} = (\bar{V}, \bar{\mathcal{E}}, \bar{R})$. Concepts from all domains would be included in this sub-graph $\bar{G}$, as well as links between them.

2. Knowledge Graph Pre-training

   In order to compute the representations of all nodes, a graph autoencoder model is trained to conduct link prediction. Following [40], our graph autoencoder model includes an entity encoder and a scoring decoder.

   For the encoder module, an effective variant of GCN, R-GCN [40] is adopted. Through multiple steps of inference, it has the ability that accumulates relational information from the neighbourhood of a given concept. For the specific structure of the encoder, two stacked R-GCN encoders are adopted. The two-step process is designed to generate the domain-aggregated feature vector $h_i \in R^d$.

   For the decoder module, we utilize DistMult factorization [41] as the scoring function, which has been demonstrated to be effective for training knowledge graph. For a triplet (c_i, r, c_j), the score can be computed through the following equation:

$$s(c_i, r, c_j) = l\left(h_{c_i}^T, R_r, h_{c_j}\right) \tag{3}$$

   where l is the logistic function; $h_{c_i}, h_{c_j} \in R^d$ are the R-GCN encoded feature vectors for concept c_i,c_j respectively. Each relation $r \in R$ is also associated with a diagonal matrix $R_r \in R^{d \times d}$.

   For the training process, the graph autoencoder model is trained through the way of negative sampling [40]. The task becomes a binary classification task to predict the triplets are positive or negative using the standard cross-entropy loss function. After training, the encoded node representations will contain the graph information.

3. Commonsense Graph Feature Extraction

   To obtain the sentence-specific commonsense graph features, we take the following steps. Firstly, we extract a set W of all seed words presented in the sentence. Next, we extract a subgraph $\bar{G}_W$ from $\bar{G}$. Then, we get feature vectors k_j for all unique nodes j in $\bar{G}_W$ and average all unique nodes. Finally, we can obtain the feature vector g for each sentence and concatenate the feature vector and the representation of each word which is nouns, adjectives or adverbs.

   Consequently, if the word in the given sentence is a noun, adjective or adverb, the new representation fusing commonsense knowledge is $\bar{h}_i = [\hat{h}_i; \mathrm{g}]$; if not, $\bar{h}_i = [\hat{h}_i; \mathbf{0}]$, where $\mathbf{0}$ is a zero vector and ; denotes the vector concatenation.

Attention Layer

Following GTS, using an attention layer to improve the interaction between w_i and w_j will have the effect of learning a robust representation for word-pair (w_i, w_j). The calculation steps are as follows:

$$o_{ij} = v^T \left(W_{a1}\bar{h}_i + W_{a2}\bar{h}_j + b_a\right), \tag{4}$$

$$\alpha_{ij} = \frac{\exp\left(o_{ij}\right)}{\sum_{k=1}^{n} \exp\left(o_{ik}\right)}, \tag{5}$$

$$\tilde{h}_\imath = \bar{h}_i + \sum_{j=1}^{n} \alpha_{ij}\bar{h}_j, \tag{6}$$

where W_{a1} and W_{a2} are weight matrices, and b_a is the bias. Finally, we concatenate the enhanced representations of w_i and w_j to represent the word-pair (w_i, w_j), i.e. $r_{ij} = [\tilde{h}_\imath; \tilde{h}_J]$.

Inference on GTS

Following the strategy of GTS, we use the following equations to inference the relation of word-pairs. However, different from the original inference strategy in GTS, we only adopt one turn inference strategy due to the effectiveness of one turn inference strategy and it can save computing costs [4]. The equations are as follows:

$$p_{ij}^0 = softmax\left(W_s r_{ij} + b_s\right), \tag{7}$$

$$p_i^0 = maxpooling\left(p_{i,:}^0\right), \tag{8}$$

$$p_j^0 = maxpooling\left(p_{j,:}^0\right), \tag{9}$$

$$q_{ij}^0 = \left[z_{ij}^0; p_i^0; p_j^0; p_{ij}^0\right], \tag{10}$$

$$z_{ij}^1 = W_q q_{ij}^{t-1} + b_q, \tag{11}$$

$$p_{ij}^1 = softmax\left(W_s z_{ij}^1 + b_s\right), \tag{12}$$

where W_s and b_s are trainable parameters, and p_{ij}^t denotes the prediction after the t-th turn. The t is a hyperparameter denoting the inference times.

Constraints in the Loss Function

Moreover, on the basis of the original loss function in GTS, we also define two intuitive constraints including diagonal constraint and implication constraint.

1. Diagonal constraint: It means that the label in the diagonal belongs to $\{A, O, N\}$.

$$L_{dia} = \sum_{i=1}^{n} \left(H(\max_{l \in C_s} \{p_{i,i,l}\}) \right), \tag{13}$$

   where C_s denotes the set of sentiment tags, including tags *Pos*, *Neu* and *Neg*; $C_{a/o}$ denotes the set of Aspect or Opinion term Tag; and $H(u) = \max(u, 0)$ is the hinge loss.
2. Implication constraint: A key intuition is that if a sentiment exists, then its aspect term and opinion term must also exist. In other words, it is impossible for a sentiment to exist without two corresponding terms. It implies that the probability of sentiment is not greater than the probability of each argument term. Since we model term and sentiment labels in a unified probability space, we use it in the model as the implication constraint.

   For each word in the diagonal, its maximum possibility over the term type space $C_{a/o}$ must not be lower than the maximum possibility for other words in the same row or column over the sentiment type space C_s.

$$L_{imp} = \sum_{i=1}^{n} \left(H(\max_{l \in C_s} \{p_{i,:,l}, p_{:,i,l}\} - \max_{l \in C_{a/o}} \{p_{i,i,l}\}) \right), \tag{14}$$

   where $H(u) = \max(u, 0)$ is the hinge loss.
3. Cross entropy loss: The original GTS calculates the cross-entropy loss between the true distribution and the predicted distribution p_{ij}^1 of the whole word-pairs. We also adopt this function.

$$L_{cross} = -\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k \in C} I\left(y_{ij} = k\right) \log\left(p_{i,j|k}^1\right), \tag{15}$$

where $I()$ is the indicator function and C denotes the label set, i.e. $\{A, O, Pos, Neu, Neg, N\}$.

As a result, the final training loss L of GCN-EGTS consists of three parts, including L_{cross}, L_{dia} and L_{imp}. We add them through the following equation:

$$L = L_{cross} + L_{dia} + L_{imp}. \tag{16}$$

Experiments and Results

Dataset

We adopt the same datasets as that in [4], which is available through the link given in [4]. The four datasets are all from SemEval Challenges [1, 42, 43]. The datasets are all split into three parts: training set, validation set and test set. As shown in Table 1, there may be multiple target terms or opinion terms in one sentence. Note that one target term can match more than one opinion term, and the reverse holds true too.

Experimental Settings

We initialize the word vectors by utilizing word embeddings from Glove and fastText. Specifically, 300-dimension GloVe embeddings and 100-dimension vectors trained by fastText are adopted as domain-general and domain-specific embeddings, respectively. To encode the input sentences, we set the CNN kernel size to 5 and set the dimension of LSTM cell to 50. The dropout layer with probability 0.5 is also applied. For BERT encoder, uncased BERT Base version is used. For training, the initial learning rate is 0.001 and the batch size is set to 32. The Adam optimizer [44] is adopted for optimizing networks.

Table 1 Statistics of the four datasets

Type	Res14 Train	Res14 Dev	Res14 Test	Lap14 Train	Lap14 Dev	Lap14 Test
#S	1259	315	493	899	225	332
#A	2064	487	851	1257	332	467
#O	2098	506	866	1270	313	478
#T	2356	580	1008	1452	383	547
#SW	1586	388	657	824	190	291
#MW	752	189	337	636	156	252
Type	Res15 Train	Res15 Dev	Res15 Test	Res16 Train	Res16 Dev	Res16 Test
#S	603	151	325	863	216	328
#A	871	205	436	1213	298	456
#O	966	226	469	1329	331	485
#T	1038	239	493	1421	348	525
#SW	678	165	297	918	216	344
#MW	335	84	188	476	123	170

#S, #A, #O, #T, #SW and #MW represent the number of sentences, target terms, opinion terms, triplets, single-word triplets and multi-word triplets respectively

For span GCN training, the Adam optimizer is adopted. The learning rate is set to 0.001 and the batch size is set to 64. For commonsense graph pre-training, the fraction of graph edges used in training is set to 0.5. The Adam optimizer is also adopted for training R-GCN and the learning rate is set to 0.01.

To evaluate the performance of different models, three classic evaluation metrics including precision, recall and F1-score are used. The best model weights are selected according to the F1 scores on the development set and the average results are reported on the test set of 5 runs.

Table 2 Results on the ASTE task

Model	Res14 P.	Res14 R.	Res14 F1	Lap14 P.	Lap14 R.	Lap14 F1
IMN+IOG[46]	59.57	63.88	61.65	49.21	46.23	47.68
JET [5]	61.50	55.13	58.14	53.03	33.89	41.35
GTS(BiLSTM) [4]	66.13	59.91	63.73	55.35	42.99	48.31
S^3E^2 [7]	69.08	64.55	66.74	59.43	46.23	52.01
GCN-EGTS(BiLSTM)	70.86	60.56	65.30	55.62	43.38	48.67
GCN-EGTS(CNN)	68.74	62.07	65.72	55.94	45.25	49.89
GCN-EGTS(BERT)	70.14	68.07	**69.20**	54.54	52.27	**53.64**
Model	Res15 P.	Res15 R.	Res15 F1	Res16 P.	Res16 R.	Res16 F1
IMN+IOG[46]	55.24	52.33	53.75	59.25	58.09	58.67
JET [5]	64.37	44.33	52.50	70.94	57.00	63.21
GTS(BiLSTM) [4]	60.10	47.89	53.65	64.28	60.56	62.79
S^3E^2 [7]	61.06	56.44	58.66	71.08	63.16	**66.87**
GCN-EGTS(BiLSTM)	60.64	49.61	54.55	67.31	62.39	64.76
GCN-EGTS(CNN)	61.54	51.29	55.97	63.73	63.86	63.77
GCN-EGTS(BERT)	59.23	58.15	**58.84**	66.89	65.86	66.28

We rerun GTS model [4] with BiLSTM encoder and report the results. The results of other three baseline models are directly obtained from the references [5, 7, 46]

Table 3 Analysis with different evaluation modes

Type	Model	Res14			Lap14		
		P.	R.	F1	P.	R.	F1
Single	GTS	73.33	61.11	66.67	62.50	54.42	58.18
	GCN-EGTS	75.67	62.29	68.28	64.06	56.95	60.24
	Δ	+2.34	+1.18	+1.61	+1.56	+2.53	+2.06
Multi	GTS	58.87	45.06	51.05	35.12	28.69	34.58
	GCN-EGTS	62.65	49.95	55.37	38.86	34.92	39.36
	Δ	+3.78	+4.89	+4.32	+3.74	+5.23	+4.78
		Res15			Res16		
Type	Model	P.	R.	F1	P.	R.	F1
Single	GTS	70.93	51.60	59.74	77.93	57.62	66.25
	GCN-EGTS	72.61	54.49	61.73	78.86	59.47	67.78
	Δ	+1.68	+2.98	+1.99	+0.93	+1.85	+1.53
Multi	GTS	48.25	31.07	37.80	70.78	53.29	60.80
	GCN-EGTS	48.89	37.29	42.31	72.99	56.12	63.25
	Δ	+0.64	+6.22	+4.51	+2.21	+2.83	+2.45

Results of Sentiment Triplet Extraction

The experiment results of ASTE are presented in Table 2. We compare GCN-EGTS with several state-of-the-art models in terms of precision (P.), recall (R.) and F1 score on the four datasets. When choosing the compared models, for pipeline models, we employ IMN [45] to extract the aspect-sentiment pair, and then use IOG [46] to make a combination. For end-to-end models, JET [5] is employed. In JET, three elements in a triplet are tagged using a position-aware tagging scheme for the ASTE task. In addition, we rerun the basic model GTS [4] with BiLSTM encoder. A more recent model named S^3E^2 [7] is also employed as a baseline model.

The experimental results show that GCN-EGTS consistently outperforms the first three baseline models with both BiLSTM and CNN sentence encoders. Especially compared to GTS(BiLSTM), GCN-EGTS(BiLSTM) can both obtain about 2% improvements on Res14 and Res16, which demonstrates the advantages when utilizing syntactic constituency parsing and commonsense knowledge. Compared to S^3E^2, GCN-EGTS(BERT) achieves an apparent absolute increase of F1 scores by 2.46% and 1.63% on Res14 and Lap15, respectively.

Comparison of Single-Word and Multi-Word Spans

For the two settings of single-word and multi-word spans, we compare the performance of our proposed GCN-EGTS with the original GTS model [4]. The results are shown in Table 3. For the single-word triplets, both the precision and recall score of our model consistently improve on the four datasets, resulting in an improved F1 score. It can be found that GCN-EGTS improves more significantly when comparing the performances on multi-word triplets for all the evaluation metrics. Because GCN-EGTS explicitly consider the syntactic constituency information to enhance the original GTS, the interactions of word-pair can be learnt more fully. In addition, for both the two models, multi-word triplets are more challenging and the F1 scores decline for over 10 points on all the datasets.

Ablation Study

To study the influences of syntactic constituency information and commonsense knowledge on our GCN-EGTS model, an ablation study is conducted and the experiment results of F1 scores are presented in Table 4.

After removing the syntactic constituency information encoding, the F1 score of GCN-EGTS declines over 1% both for the BiLSTM and CNN encoder, which indicates that the syntactic constituency information can contribute

Table 4 Results of ablation study for the ASTE task

Encoder	Model	Res14	Lap14	Res15	Res16
BiLSTM	GCN-EGTS	**65.30**	**48.67**	**54.55**	**64.76**
	w/o -constituency	64.03	46.98	53.21	63.20
	w/o -commonsense	64.23	47.16	53.53	63.64
CNN	GCN-EGTS	**65.72**	**49.89**	**55.97**	**63.77**
	w/o -constituency	64.29	48.11	54.50	62.52
	w/o -commonsense	64.60	48.26	54.73	62.90
BERT	GCN-EGTS	**69.20**	**53.64**	**58.84**	**66.28**
	w/o -constituency	68.60	52.96	57.74	65.01
	w/o -commonsense	68.07	52.46	57.35	65.03

to predict the relation between words. In terms of commonsense knowledge encoding, it is found that GCN-EGTS with the full setting outperforms GCN-EGTS without commonsense knowledge encoding significantly on all the datasets. It makes sense because the commonsense knowledge graph can contribute to obtaining the missing commonsense knowledge information not contained in the text and it can also make more complete predictions. Note that when BERT encoder is adopted, the improvement by utilizing constituency is limited. It shows that the value of the contributions of constituency would be influenced by a pre-trained contextualized model.

Conclusion

In this paper, a novel end-to-end model, namely GCN-EGTS is developed, which is an enhanced GTS for the ASTE task. Besides a new loss function designed by incorporating several constraints, GCN-EGTS incorporates syntactic constituency parsing tree and commonsense knowledge graph via two types of GCNs: (1) Using span GCN, the proposed model leverages the syntactic constituency parsing information about boundaries and utilizes information about the word's neighbourhood in the constituent structure as well as syntactic labels of constituents for enhancing word representations, and (2) using R-GCN, the proposed model leverages an external knowledge graph, ConceptNet, to learn domain-specific commonsense knowledge features of knowledge base concepts, and utilizes the commonsense knowledge. The results of experiments demonstrate that GCN-EGTS developed in this research outperforms the state-of-the-art approaches significantly for the ASTE task. The outcomes of this research indicate that effectively utilizing syntactic constituency parsing information and commonsense knowledge is a promising direction for the ASTE task.

Author Contribution Zhenda Hu: conceptualization, methodology, software, formal analysis, validation, writing — original draft preparation. Zhaoxia Wang: methodology, supervision, investigation, writing — reviewing and editing. Yinglin Wang: supervision, investigation, writing — reviewing and editing. Ah-Hwee Tan: supervision, investigation, writing — reviewing and editing.

Funding We would like to thank the National Natural Science Foundation of China (Grant No. 61375053) for part of the financial support of this research. This research was supported by the Fundamental Research Funds for the Central Universities and the graduate innovation fund of Shanghai University of Finance and Economics (under Project No. CXJJ-2020-428). This research was supported by A*STAR under its Advanced Manufacturing and Engineering (AME) Programmatic Grant (Award No.: A19E2b0098).

Data Availability All data generated or analysed during this study are included in this published article.

Declarations

Competing Interests The authors declare no competing interests.

References

1. Pontiki M, Galanis D, Papageorgiou H, Androutsopoulos I, Manandhar S, Al-Smadi M, Al-Ayyoub M, Zhao Y, Qin B, DeClercq O. Semeval-2016 task 5: aspect based sentiment analysis. In: International Workshop on Semantic Evaluation. 2016;19–30.
2. Peng H, Xu L, Bing L, Huang F, Lu W, Si L. Knowing what, how and why: a near complete solution for aspect-based sentiment analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2020;34:8600–8607.
3. Zhang C, Li Q, Song D, Wang B. A multi-task learning framework for opinion triplet extraction. In: Findings of the Association for Computational Linguistics: EMNLP. 2020;819–828.
4. Wu Z, Ying C, Zhao F, Fan Z, Dai X, Xia R. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In: Findings of the Association for Computational Linguistics: EMNLP. 2020;2576–2585.
5. Xu L, Li H, Lu W, Bing L. Position-aware tagging for aspect sentiment triplet extraction. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020;2339–2349.
6. Xu L, Chia YK, Bing L. Learning span-level interactions for aspect sentiment triplet extraction. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers) 2021;4755–4766.
7. Chen Z, Huang H, Liu B, Shi X, Jin H. Semantic and syntactic enhanced aspect sentiment triplet extraction. In: Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021;1474–1483.
8. Zhang M, Qian T. Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020;3540–3549.
9. Phan MH, Ogunbona PO. Modelling context and syntactical features for aspect-based sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics. 2020;3211–3220.
10. Gómez-Rodríguez C, Alonso-Alonso I, Vilares D. How important is syntactic parsing accuracy? An empirical evaluation on rule-based sentiment analysis. Artif Intell Rev. 2019;52(3):2081–97.
11. Ghosal D, Hazarika D, Roy A, Majumder N, Mihalcea R, Poria S. Kingdom: Knowledge-guided domain adaptation for sentiment analysis. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics 2020;3198–3210
12. Zhao A, Yu Y. Knowledge-enabled bert for aspect-based sentiment analysis. Knowl-Based Syst. 2021;227:107220.
13. Mao R, Liu Q, He K, Li W, Cambria E. The biases of pre-trained language models: an empirical study on prompt-based sentiment analysis and emotion detection. IEEE Trans Affect Comput. 2022.
14. Kumar JA, Trueman TE, Cambria E. Gender-based multi-aspect sentiment detection using multilabel learning. Inform Sci. 2022;606:453–68.
15. Khoo CS, Johnkhan SB. Lexicon-based sentiment analysis: Comparative evaluation of six sentiment lexicons. J Inf Sci. 2018;44(4):491–511.

16. Zhou J, Huang JX, Hu QV, He L. Sk-gcn: Modeling syntax and knowledge via graph convolutional network for aspect-level sentiment classification. Knowl-Based Syst. 2020;205:106292.
17. Sun K, Zhang R, Mensah S, Mao Y, Liu X. Aspect-level sentiment analysis via convolution over dependency tree. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). 2019;5679–5688.
18. Cambria E, Liu Q, Decherchi S, Xing F, Kwok K. SenticNet 7: a commonsense-based neurosymbolic AI framework for explainable sentiment analysis. Proceedings of LREC. 2022.
19. He K, Mao R, Gong T, Li C, Cambria E. Meta-based self-training and re-weighting for aspect-based sentiment analysis. IEEE Trans Affect Comput. 2022.
20. Valdivia A, Luzón MV, Cambria E, Herrera F. Consensus vote models for detecting and filtering neutrality in sentiment analysis. Information Fusion. 2018;44:126–35.
21. Wang Z, Ho S-B, Cambria E. Multi-level fine-scaled sentiment sensing with ambivalence handling. Int J Uncertainty Fuzziness Knowledge Based Syst. 2020;28(04):683–97.
22. Ma Y, Peng H, Cambria E. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive LSTM. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2018;32.
23. Dragoni M, Donadello I, Cambria E. Ontosenticnet 2: Enhancing reasoning within sentiment analysis. IEEE Intelligent Systems. 2022;37(2):103–10.
24. Yao L, Mao C, Luo Y. Graph convolutional networks for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2019;33:7370–7377.
25. Marcheggiani D, Bastings J, Titov I. Exploiting semantics in neural machine translation with graph convolutional networks. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2018;2(Short Papers):486–492.
26. Marcheggiani D, Titov I. Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017;1506–1515.
27. Trueman TE, Cambria E. A convolutional stacked bidirectional lstm with a multiplicative attention mechanism for aspect category and sentiment detection. Cogn Comput. 2021;13(6):1423–32.
28. Mao R, Li X. Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification. In: Proceedings of the AAAI Conference on Artificial Intelligence. 2021;35:13534–13542.
29. Balazs JA, Velásquez JD. Opinion mining and information fusion: a survey. Information Fusion. 2016;27:95–110.
30. Mohammad A-S, Hammad MM, Sa'ad A, Saja A-T, Cambria E. Gated recurrent unit with multilingual universal sentence encoder for Arabic aspect-based sentiment analysis. Knowl-Based Syst. 2021;107540.
31. Lu G, Yu H, Xue Y, Qiu Z, Zhong W. Scan: Syntactic knowledge and commonsense knowledge adapter based network for aspect-level sentiment classification. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. 2021;393–399.
32. Tai KS, Socher R, Manning CD. Improved semantic representations from tree-structured long short-term memory networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). 2015;1556–1566.
33. Xu H, Liu B, Shu L, Philip SY. Double embeddings and CNN-based sequence labeling for aspect extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2018;592–598.
34. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. 2016. arXiv preprint arXiv:1609.02907.
35. Marcheggiani D, Titov I. Graph convolutions over constituent trees for syntax-aware semantic role labeling. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). 2020;3915–3928.
36. Hochreiter S, Schmidhuber J. Long short-term memory. Neural Comput. 1997;9(8):1735–80.
37. Kenton JDM-WC, Toutanova LK. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT. 2019;4171–4186.
38. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I. Attention is all you need. Adv Neural Inf Proces Syst. 2017;30.
39. Speer R, Chin J, Havasi C. ConceptNet 5.5: an open multilingual graph of general knowledge. In: Thirty-first AAAI Conference on Artificial Intelligence. 2017.
40. Schlichtkrull M, Kipf TN, Bloem P, Berg Rvd, Titov I, Welling M. Modeling relational data with graph convolutional networks. In: European Semantic Web Conference. 2018;593–607. Springer.
41. Yang B, Yih SW-t, He X, Gao J, Deng L. Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of the International Conference on Learning Representations (ICLR). 2015.
42. Pontiki M, Galanis D, Papageorgiou H, Manandhar S, Androutsopoulos I. Semeval-2015 task 12: aspect based sentiment analysis. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015) 2015;486–495.
43. Kirange D, Deshmukh RR, Kirange M. Aspect based sentiment analysis semeval-2014 task 4. Asian J Comput Sci Inf Technol (AJCSIT). 2014;4:72–75.
44. Kingma DP, Ba J. Adam: a method for stochastic optimization. In: ICLR (Poster). 2015.
45. He R, Lee WS, Ng HT, Dahlmeier D. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019;504–515.
46. Fan Z, Wu Z, Dai X, Huang S, Chen J. Target-oriented opinion words extraction with target-fused neural sequence labeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). 2019;2509–2518.