Knowledge-Based Systems 257 (2022) 109943

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Logic tensor network with massive learned knowledge for aspect-based sentiment analysis

Hu Huang [a], Bowen Zhang [b,*], Liwen Jing [b], Xianghua Fu [b], Xiaojun Chen [c], Jianyang Shi [d]

[a] School of Cyberspace Science and Technology, University of Science and Technology of China, Hefei, China
[b] College of Big Data and Internet, Shenzhen Technology University, Shenzhen, China
[c] College of Computer Science and Software, Shenzhen University, Shenzhen, China
[d] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

ARTICLE INFO

Article history:
Received 6 June 2022
Received in revised form 8 September 2022
Accepted 25 September 2022
Available online 3 October 2022

Keywords:
Aspect-based sentiment analysis
Attention mechanism
Syntax-based method
Logic tensor network

ABSTRACT

Aspect-based sentiment analysis assists service providers to better understand users' opinions expressed in massive amounts of online posts, because it automatically infers users' sentiments towards the aspect terms of interest. Recently, several researchers have attempted to apply first-order logic (FOL) rules to deep neural networks via the posterior constraint method. However, existing methods simply apply a priori constraints to represent the FOL with coefficients selected by hand, which requires improvements in incorporating and adapting abstract knowledge in data. In this study, we propose a novel logic tensor network with massive rules (LTNMR) for aspect-based sentiment analysis, which is constructed by incorporating FOL. Specifically, we integrate two types of knowledge into the logic tensor network: (1) dependency knowledge, which improves the efficiency of the capture of aspect-related words and (2) the human-defined knowledge rule, which helps the classifier understand the sentiment of the extracted aspect-related words. Furthermore, to achieve high inferring accuracy, we propose a mutual distillation structure knowledge injection (MDSKI) strategy. MDSKI transfers dependency knowledge from teacher Bert to LTNMR, which acts as the student network. Experiments demonstrate that the proposed LTNMR, combined with the MDSKI strategy, substantially outperforms state-of-the-art results for aspect-based sentiment analysis.

1. Introduction

With the rapid growth of opinion-rich online data collection, much attention has been given to automatically extracting sentiment polarity towards aspect terms of interest in a sentence, which is termed aspect-based sentiment analysis (ABSA) [1]. For example, given an online post, "delicious *food* but *service* terrible", the task of ABSA is to infer the sentiment polarity towards the aspect terms, for which a positive polarity for "*food*" and a negative polarity for "service" is expected. ABSA is one of the most popular research fields because of its broad applications in brand monitoring, customer service, market research, politics, and social science. For example, companies can develop strategies based on data-mining results from users' attitudes toward product reviews.

ABSA approaches can be classified into two categories according to the features and knowledge they utilize: rule- and corpus-based methods. The rule-based method mainly consists of two steps. First, hand-crafted rules (e.g., dependency relation) are designed to match aspect-related words to their corresponding aspect terms. Second, with the help of the sentiment lexicon, negative and positive aspect-related words in the sentence are counted, and the sentiment polarity towards the aspect term is assigned as positive if positive words outnumber negative words [2,3]. Rule-based methods provide a clear explanation of the ABSA process. However, such methods are heavily dependent on feature engineering, which is time consuming and expensive. Inspired by the rapid advancement of deep learning in a wide range of natural language processing tasks [4], deep neural networks (DNNs), from long short-term memory networks (LSTM) and attention-based networks (Atts) to the recent pre-trained language models (PLMs), have become dominant in the field owing to their outstanding performance. Nevertheless, a powerful end-to-end deep neural network is typically used as a "black box", which has limitations, including the requirement of a massive amount of labeled data, uninterpretability of prediction results, and difficulty incorporating human intentions and domain knowledge [5].

Several techniques have recently been developed for ABSA tasks to address the interpretability issue of neural classifiers. Conventional research has focused on model-agnostic explanation methods such as data augmentation methods [6]. Another

* Corresponding author.
E-mail address: zhang_bo_wen@foxmail.com (B. Zhang).

https://doi.org/10.1016/j.knosys.2022.109943
0950-7051/

solution is to build syntax-aware neural networks that train the DNN with extra handcrafted features or knowledge rules [7]. Recently, several studies proposed the introduction of first-order logic (FOL) rules into neural networks. FOL can formulate flexible declarative and predicative rules to communicate high-level cognition and express structured knowledge. Thus, integrating logic rules into DNNs, which may transmit human purpose and domain knowledge to DNNs and manage the learning process, has great promise. In general, these methods integrate FOL constraints via posterior regularization into DNNs [8–10].

Despite the intuitiveness of FOL rules and performance improvements in various tasks, aspect-based sentiment analysis remains a challenge in practice. The posterior regularization approach has traditionally been restricted to a priori fixed constraints with manually determined weights and no capacity to infer and adapt abstract knowledge from the data. This issue is exacerbated when it comes to controlling DNNs that directly map raw input sequences to the label space, where the semantic gap between inputs and outputs is large, with no intermediary abstract concepts for encoding complex human knowledge. To achieve the interpretability of ABSA, it would be extremely desirable to develop DNNs for ABSA that can be explained by FOL because this would be agreeable for humans and valuable in situations where explanations are necessary.

In this study, we propose a generalized framework that enables a learning procedure for knowledge rules with FOL-based DNNs, which is widely applicable to massive amounts of knowledge in diverse formats, facilitating incorporation into rich domain expertise by practitioners. To achieve this aim, we first propose an interpretable DNN as a sentiment classifier called a **L**ogic **T**ensor **N**etwork with **M**assive **R**ules (LTNMR). LTNMR is constructed based on a logic tensor network, neurosymbolic formalism, and computational model that supports learning and reasoning about data with a differentiable first-order logic language [11]. We introduce two types of knowledge that are incorporated into LTNMR to improve the performance and interpretability of the ABSA: (1) *dependency knowledge* is leveraged to enhance the performance of acquiring the connections between aspect terms and their corresponding contextual words, which is widely utilized in most of the recent attention-based models; and (2) the human-defined *knowledge rule* that helps the classifier to understand the sentiment of the extracted aspect-related words. Finally, to realize LTNMR with high inferring accuracy and effectively integrate massive knowledge rules, we propose a mutual distillation structure knowledge injection (MDSKI) strategy. The MDSKI contains a teacher–student network (LTNMR). For the teacher model, we utilized the PLM model (e.g., Bert) to induce dependency knowledge, which was then injected into the LTNMR.

The following is a summary of our contributions:

- We propose the LTNMR method, which is constructed following FOL, and provides a flexible declarative language for integrating both dependency knowledge and manual-defined rules.
- We propose a mutual distillation structure knowledge injection (MDSKI) strategy that enhances the capability of the LTN-based model by exploiting the induced syntax knowledge from PLM.
- Extensive experiments were performed to demonstrate the effectiveness of the ABSA model. The results show that the proposed model significantly outperforms all the compared methods by a great extent.

The rest of this paper is organized in the following structure. Section 2 extensively reviews related work, including methods with first-order logic and ABSA methods. Section 3 describes the proposed LTNMR network in detail. In Section 4, the experimental setup and quantitative evaluation results and analysis are explained. Conclusions and future research directions are given in Section 5.

2. Related work

2.1. Methods with first-order logic

Logic rules supply flexible declarative language for communicating concept-level knowledge. Recently, several studies attempted to integrate first-order logic rules into DNNs. For instance, a general distillation framework (DNNLR) was proposed by Hu et al. in which knowledge expressed as FOL is transferred into neural networks via posterior regularization [8]. Subsequently, the DNNLR framework was extended by integrating various types of knowledge [9], and the training strategy was further extended based on posterior regularization [10]. In general, these methods integrate the knowledge expressed in FOL constraints via posterior regularization.

2.1.1. Logic tensor network

Logic tensor networks (LTN) combine deep neural networks with symbolic knowledge that efficiently realizes the process of learning and reasoning. For example, an LTN has been used in semantic image interpretation, where relational object information is integrated into deep networks to discover object relationships [11]. Bianchi et al. implemented an LTN for ontological knowledge reasoning in [12], and they further developed an LTN for modeling concepts into a latent space by utilizing ontological knowledge [13]. In summary, the LTN employs end-to-end differentiable FOL as the language representation in deep neural networks, which satisfies the interpretability requirement in certain tasks.

The semantics of logic in an LTN (called Real Logic) are different from the standard FOL. In real logic, every object denoted by a constant, variable, or term is encoded by a real-valued tensor. Predicates are interpreted as a function or tensor operation that projects values in the interval [0, 1]. For real logic, functions are usually implemented using neural networks. Grounding in Real Logic, denoted by $\mathcal{G}$, associates a real-valued tensor within interval [0, 1].

2.2. Aspect-based sentiment analysis

In both academia and industry, sentiment classification is gradually becoming increasingly important [14–16]. Recently, the ABSA has drawn increasing attention from researchers. There are two types of reported techniques: rule-based and DNN-based techniques. Rule-based methods require manual feature extraction [17]. For example, Poria et al. first extracted aspect-related words using the dependency rule they proposed, followed by a machine-learning approach to automatically analyze sentiment polarity [2]. With deep learning flourishing in recent years, DNN-based models have been widely used in ABSA tasks. Tang et al. proposed a memory network that develops a multi-hop attention mechanism for modeling the contexts [18]. They further developed TD-LSTM, in which the standard LSTM was extended to two individual LSTMs, modeling the left contexts towards the target word and the right contexts towards the target word separately [19]. Li et al. employed a hierarchical attention network to recognize sentiment-bearing words [20]. An IAN was proposed by Ma et al. where two attention networks were implemented to learn the target and context representations interactively [21]. Sun et al. conducted an ABSA by performing a sentence-pair classification task instead of constructing

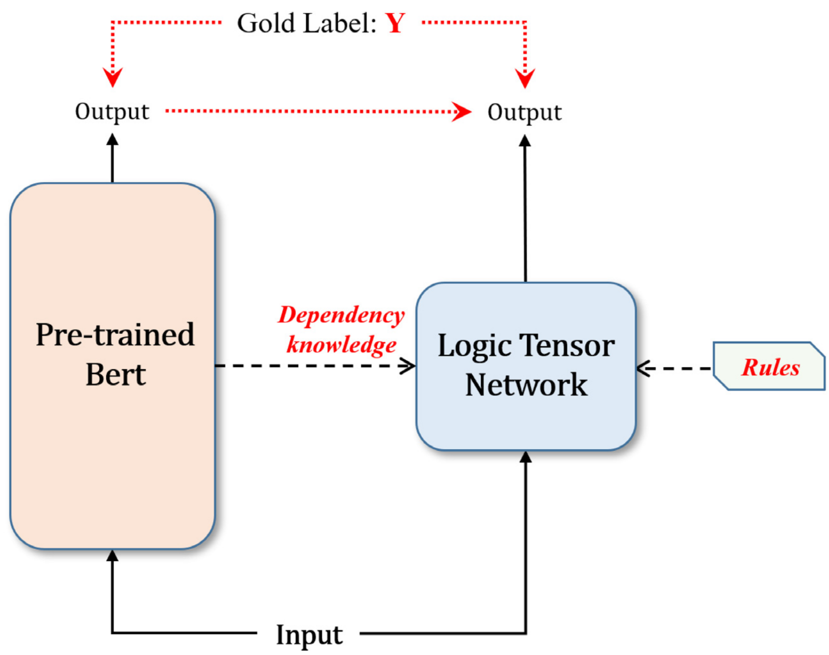


Fig. 1. Overall framework of our method.

auxiliary sentences [22]. Subsequently, several new graph convolutional networks (CGNs) were proposed with more appropriately allocated sentiment words for aspect categories, and they obtained better performance [23–26]. More recently, pretrained language models have been proposed for ABSA tasks, such as Bert. For example, [27] utilized Bert as a text representation model and fine-tuned it using the ABSA dataset. Furthermore, [28,29] achieved state-of-the-art performance by combining the attention mechanism with a pretrained Bert model.

2.3. External knowledge used in ABSA

Enhancing ABSA capabilities by incorporating external knowledge has garnered much attention because models with such knowledge focus on informative features such that the sentiment inference process is interpretable. First, researchers utilized sentiment lexicons in ABSA tasks, where word polarity score, emotion knowledge, and semantic knowledge were developed to enhance sentiment-inferring ability. Subsequently, syntax knowledge is incorporated into the DNN models to assist the learning of aspect terms and their related words. For instance, the ASGCN method proposed by Zhang et al. integrated syntactic dependency knowledge into DNN and then employed GCN for modeling the text for the first time [30]. Subsequently, several extensions of ASGCN were reported, where the aspect-oriented dependency tree structure was fine-tuned by reconstructing the standard dependency tree using carefully designed rules [31–34]. Similarly, phrase knowledge was incorporated into the tree structure to improve performance [35–37]. Furthermore, human-defined knowledge rules, such as local n-gram rules and double-negative rules, are also exploited for ABSA tasks [7,38–40].

3. Method

3.1. Framework overview

The overall framework of our proposed method is given in Fig. 1. The right branch is the LTNMR, which serves as a basic prediction model. LTNMR is constructed following FOL, where each *Grounding* $\mathcal{G}$ is constructed by a simple trainable neural network structure. We implement two types of knowledge expressed by FOL and integrated into the logic tensor network. The first type is dependency relation which helps to extract the aspect-related word, while the second type of knowledge identifies the sentiment-aware word. To realize LTNMR with high inferring accuracy, we train LTNMR with the MDSKI strategy. Specifically, in MDSKI, we use Bert (the left branch) as the teacher and LTNMR as the student.

In Section 3.2, we first describe the problem definition. Then, we elaborate on the tree-inducing method in Section 3.3. To fully descried the proposed LTNMR, we first introduce the basic LTN in Section 3.4 and the knowledge integration method in Section 3.5 in detail. In the final Section 3.6, we introduce the MDSKI training strategy of our model.

3.2. Problem definition

First, the general formulation of the ABSA task is described as follows. Given a sentence $x = \{w_1^c, \ldots, w_a^a, \ldots, w_{a+m}^a, \ldots, w_n^c\}$ containing aspect-term words $w_a^a, \ldots, w_{a+m}^a$, where each word in the sentence is denoted as w and m represents the number of words in the aspect-term. The superscripts “c” and “a” indicate the word to be a context word or an aspect-term word, respectively. The sentiment label of each sentence is denoted as y. ABSA aims to predict the sentiment label y for the input sentence x towards the aspect-term of interest.

3.3. Inducing tree structure from PTMs

To leverage knowledge from large-scale corpora, the perturbed masking method (PMM) induces a dependency tree structure from a pre-trained model without requiring additional training parameters.

The goal of PMM is to discover syntactic knowledge from pre-trained language models. Formally, given a text x, Bert first transform each w_i into a hidden representation $H_\theta(w_i)$. Then, PMM computes the value $f(w_i, w_j)$, which represent the impact between w_j and w_i. Specifically, to acquire this value, PMM first

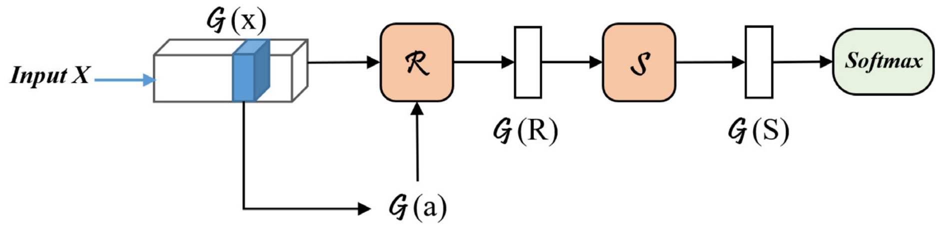


Fig. 2. Computational graph of logic tensor network for aspect-based sentiment analysis.

use a special token "[MASK]" to replaces w_i. Then Bert predicts the hidden state of w_i ($H_\theta(w_i)$) when inputs sentence with "[mask]". Subsequently, it further masked both w_i and w_j to obtain the hidden state $H_\theta(w_i, w_j)$. The impact value $f(w_i, w_j)$ can be computed by the euclidean distance:

$$f(w_i, w_j) = \|H_\theta(w_i)_i - H_\theta(w_i, w_j)_i\|_2 \quad (1)$$

After that, we can iteratively compute the impact value between each i and j, which denotes as $M_{i,j} = f(w_i, w_j)$. Finally, the tree-structure can be decoded from M by utilizing tree decoding algorithm, say Eisner algorithm [41] and etc. In this paper, the dependency-tree is denoted as *Adj*.

3.4. Logic tensor network for ABSA

LTN can be utilized for the ABSA task, which infers the sentiment polarity toward the given aspect-term. LTN is built based on FOL rules, which are consistent with human cognitive processes. We first give the definition in Section 3.4.1. Details of the neural network structure of LTN are described in Section 3.4.2.

3.4.1. LTN definition for ABSA

Important terms are defined for the formulation of logic tensor network for ABSA:

- **Domains**:
  texts, representing the examples from dataset.
  labels, representing the class labels.
- **Variables**:
  x_+, $x_\circ$ and x_- denote the text with "positive", "neutral" and "negative" sentiment, respectively.
  $D(x_+) = D(x_\circ) = D(x_-) = D(x) =$ *texts*.
- **Constants**:
  l_+, $l_\circ$ and l_-, denoting the sentiment classes/labels for "positive", "neutral" and "negative", respectively.
  $D(l_+) = D(l_\circ) = D(l_-) =$ *labels*.
- **Predicates**:
  $R(x, a)$, denoting the aspect-related words towards the aspect-term a.
  $S(x)$, denoting the informative sentiment-baring words of the given input x.
  $P(R(x, a), l)$ denotes the fact that text x is classified as l when targeting to aspect-term a.
- **Axioms**:

$$\forall x_+ \; P(S(R(x_+, a)), l_+) \quad (2)$$

$$\forall x_\circ \; P(S(R(x_\circ, a)), l_\circ) \quad (3)$$

$$\forall x_- \; P(S(R(x_-, a), l_-)) \quad (4)$$

  It is worth noting that, rules about exclusiveness such as $\forall x(P(x, l_+) \to (\neg P(x, l_\circ) \wedge \neg P(x, l_-)))$ are not included since such constraints are already imposed by the grounding of P below, say the Softmax function.
- **Grounding**:
  $\mathcal{G}(l)$ is the one-hot vector where $\mathcal{G}(l_+) = [1, 0, 0]$, $\mathcal{G}(l_n) = [0, 1, 0]$ and $\mathcal{G}(l_-) = [0, 0, 1]$
  $\mathcal{G}(x)$ is a word matrix of x. $\mathcal{G}(a)$ and $\mathcal{G}(c)$ are word matrix of aspect-term a and content words c, respectively.
  $\mathcal{G}(R(x, a))$ is a vector sequence that computed by $\mathcal{G}(x)$ and $\mathcal{G}(a)$.
  $\mathcal{G}(P|\theta)$: (x, a), $l \mapsto$ softmax$(\mathrm{LTN}_\theta(x, a))$, where the LTN has three output neurons corresponding to the sentiment polarity "positive, neural or negative", and each neurons gives the probability corresponding to the class l.

3.4.2. Neural network structure of logic tensor network

Follow the LTN definition, we aim to construct $\mathcal{G}(P|\theta)$: (x, a), $l \mapsto$ softmax$(\mathrm{LTN}_\theta(x, a))$ by DNN structure. Specifically, following FOL form, $\mathcal{G}(P)$ can be decomposed by each *grounding*, which is constructed by the neural network layer. Specifically, the computational graph of LTN is given in Fig. 2.

Since *grounding* can be freely constructed by different DNNs, in this paper, we build the network for different groundings by considering the interpretability from low to high and construct four network structures (see Table 1). Formally, we give a detailed introduction to two types of models: (i) LTN-1 is the most explanatory but shallowest model, where each grounding of function or predicate is computed by tensor operations (e.g., tensor Product). (ii) LTN-3 is the complex attention-based model, where each grounding of function or predicate is constructed with an attention-based structure that fully exploits the learning ability of DNNs.

To realize $\mathcal{G}(P|\theta)$, we decompose the ABSA task into two stages: (1) obtaining aspect-related words; (2) learning the sentiment-aware word from aspect-related words.

LTN-1: To achieve $\mathcal{G}(P|\theta)$, the first layer *Grounding* is $\mathcal{G}(x)$, where input text x first goes through the embedding layer and comes out with an embedded word vector. Intuitively, $\mathcal{G}(x)$ of LTN-1 is the word vector matrix. Subsequently, to obtain aspect-related words, we represent the relationship between $\mathcal{G}(a)$ and $\mathcal{G}(x)$. Here, $\mathcal{G}(a)$ is obtained by selecting the corresponding vector through the index $\{w^a_a, \ldots, w^a_{a+m}\}$ from $\mathcal{G}(x)$, while the remaining w^c_i is set to 0. Thus the matrix dimension is the same as $\mathcal{G}(x)$. Then the representation of aspect-related words $\mathcal{G}(R(x, a))$ can be computed by summing up the product of $\mathcal{G}(x)$ and $\mathcal{G}(a)$ (denoted as $\mathcal{G}(R)$):

$$\mathcal{G}(R) = \sum_i (\mathcal{G}(x) \times \mathcal{G}(a)) \quad (5)$$

Then, to learn the sentiment-aware word from aspect-related words, we simply send $\mathcal{G}(R)$ into a fully-connect neural network to achieve $\mathcal{G}(S(R(x, a)))$ (denoted as $\mathcal{G}(S)$):

$$\mathcal{G}(S) = W\mathcal{G}(R) + b \quad (6)$$

Finally we take softmax computation of $\mathcal{G}(S)$ for sentiment polarity prediction:

$$\mathcal{G}(P) = softmax(\mathcal{G}(S)) \quad (7)$$

Table 1
Variants of logic tensor network structure.

Grounding	LTN-1	LTN-2	LTN-3	LTN-4
$\mathcal{G}(x)$	glove	glove-LSTM	glove-BiLSTM	Bert
$\mathcal{G}(R(x,a))$	$\sum_i(\mathcal{G}(x)\times\mathcal{G}(a))$	$\sum_i(\mathcal{G}(x)\times\mathcal{G}(a))$	$c=\mathcal{G}(x)\mathcal{G}(a)^{\mathrm{T}}$ $\mathcal{G}(x)_t softmax(\sum_t c_t)$	$c=\mathcal{G}(x)\mathcal{G}(a)^{\mathrm{T}}$ $\mathcal{G}(x)_t softmax(\sum_t c_t)$
$\mathcal{G}(S)$	$W\mathcal{G}(R(x,a))+b$	$W\mathcal{G}(R(x,a))+b$	$\lambda LN(\mathcal{G}(x)\mathcal{G}(a)^{\mathrm{T}}\mathcal{G}(x))+\mathcal{G}(x)$	$\lambda LN(\mathcal{G}(x)\mathcal{G}(a)^{\mathrm{T}}\mathcal{G}(x))+\mathcal{G}(x)$
$\mathcal{G}(P)$			Softmax	

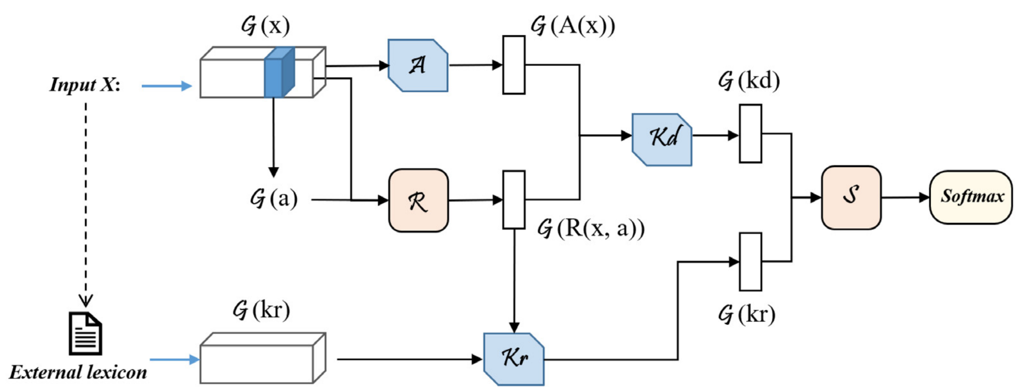


Fig. 3. Computational graph of logic tensor network with massive knowledge (LTNMR).

LTN-3: To improve inferring accuracy, we sacrifice some interpretability and use an attention-based model for the grounding constructions. Specifically, given x, we first convert each word in x into a low-dimensional vector (e_i for the ith word) by the embedding layer. Then, we send the embedded vectors into the BiLSTM layer, and $\mathcal{G}(x)$ in LTN-3 is represented by the hidden states of BiLSTM.

Here, $\mathcal{G}(a)$ is the same as LTN-1, and the representation of aspect-related words $\mathcal{G}(R)$ can be constructed by the self-attention style mechanism by utilizing $\mathcal{G}(x)$ and $\mathcal{G}(a)$:

$$\mathcal{G}(R)=\lambda LayerNorm(\mathcal{G}(x)\mathcal{G}(a)^{\mathrm{T}}\mathcal{G}(x))+\mathcal{G}(x), \tag{8}$$

where *LayerNorm* denotes the layer normalization strategy.

Subsequently, to learn the sentiment-aware word from aspect-related words, we send $\mathcal{G}(R(x,a))$ into another attention method to calculate the informative words from $R(x,a)$:

$$\begin{aligned} c &= \mathcal{G}(R)\mathcal{G}(a)^{\mathrm{T}}, \\ \mathcal{G}(S) &= \mathcal{G}(R)_t softmax(\sum_t c_t), \end{aligned} \tag{9}$$

Finally, $\mathcal{G}(S)$ goes through a softmax layer for sentiment polarity distribution prediction of LTN-3.

Other network structure: Since LTN can be constructed flexibly, each grounding can be constructed by different network layers. For example, $\mathcal{G}(x)$ can be denoted as the output of Bert (LTN-4 in Table 1). Table 1 summarizes the variants of LTN, in which LTN 1–4 denote model complexity increases (interpretability decreases).

3.5. Knowledge integration into LTN

3.5.1. Knowledge definition

In this paper, two types of knowledge are utilized to further improve the capability and interpretability of ABSA.

We summarize the implementation details of two knowledge as follows:

Dependency relation (kd): The first type of knowledge aims to improve the ability to capture aspect-related words by utilizing *dependency knowledge*. The problem of incorrectly focusing on aspect unrelated words can be effectively addressed by using dependency knowledge. For instance, given the dependency relation *Adj* (Induced from Bert, see Section 3.3), $R(x,a)$ can be integrated by:

$$\mathcal{A}(x)\rightarrow R(x,a), \tag{10}$$

where $\mathcal{A}$ is the prior knowledge that use *Adj* to help enhance the ability to extract aspect-related words.

Knowledge rules (kr): The second type of knowledge is a human-defined rule that helps the classifier in comprehending the sentiment of the extracted aspect-related words. Specifically, one difficulty for the neural network is to identify informative words from the aspect-related words. We thus consider utilizing the prior knowledge from emotional lexicons to enhance the ability to learn sentiment-baring words in aspect-related words. Emotional features have been wildly used in rule-based methods and shown effective for ABSA. For example, given the emotional feature of the word, we expect the sentiment of the whole sentence to be consistent with the implicit emotional polarity of the aspect-related words. The logic rule is written as:

$$(R(x,a)\wedge emo_{+})\rightarrow P(R(x,a),l_{+}), \tag{11}$$

where emo_{+} denotes the emotion tags related to each word from SenticNet lexicon [42–45].

3.5.2. Knowledge grounding

As described in Section 3.5.1, this study considers two types of knowledge. The computational graph is given in Fig. 3.

(1) **Dependency knowledge**: Follow operations of FOL in [11], with the logic rule $kd:\mathcal{A}(x)\rightarrow R(x,a)$, $\mathcal{G}(kd)$ can be computed by:

$$\mathcal{G}(kd):=1-\mathcal{G}(\mathcal{A}(x))+\mathcal{G}(\mathcal{A}(x))\cdot\mathcal{G}(R(x,a)) \tag{12}$$

Specifically, $\mathcal{G}(\mathcal{A}(x))$ is obtained by:

$$\mathcal{G}(\mathcal{A}(x))=\sigma(\sum_{i=0}^{m}Adj_i\mathcal{G}(x)_iW_i'), \tag{13}$$

where W' are trainable parameters and σ denotes the activate function.

(2) **Knowledge Rule**: Given the knowledge rule: $kr : (R(x, a) \wedge emo_{+}) \rightarrow P(S, l_{+})$, where emo_{+} denotes the emotional knowledge of each word in aspect-related words. Emotional knowledge is wildly used in the conventional method that helps the predictor to learn the informative words toward the aspect-term. In this paper, we denote $\mathcal{G}(emo)$ as the word embedding vectors for the emotional words from sentiment lexicon. Thus $\mathcal{G}(kr)$ is defined as:

$$\begin{aligned} \gamma &= \mathcal{G}(R) \times \mathcal{G}(emo) \\ \mathcal{G}(kr) &:= 1 - \gamma + \gamma \times \mathcal{G}(S) \end{aligned} \tag{14}$$

3.5.3. Knowledge integration

In order to integrate a variety of different knowledge into the LTN network, we expect such knowledge to be dynamically integrated into the LTN based on its importance. Inspired by the KGCapsAN [7], we conduct a knowledge-guided dynamic routing mechanism (KGDRM). In KGDRM, knowledge groundings are upper-level capsules, and the text representations belong to lower level. The attention weight of KGDRM is considered to be the coupling coefficients between upper and lower layers. Notice that, different knowledge shares the same coupling coefficients, so the knowledge can be better integrated in the forward propagation process.

Formally, after acquiring the representation of $\mathcal{G}(kd)$ and $\mathcal{G}(kr)$, the knowledge rule can be integrated into LTN by the carefully designed attention mechanism. Specifically, we use $\mathcal{G}(kd)$ and $\mathcal{G}(kr)$ as the two attention queries (as upper-level capsules) to compute the attention distribution based on the representation of the aspect-related words $\mathcal{G}(R)$ (as lower-level capsules), in which the attention distribution reflects the coupling coefficients. Formally, given $\mathcal{G}(kd)$ and $\mathcal{G}(kr)$, $\mathcal{G}(S)$ is computed by the attention structure, defined as:

$$\begin{aligned} & C_{kr} = \mathcal{G}(kr)\mathcal{G}(R)^{\mathrm{T}}, C_{kd} = \mathcal{G}(kd)\mathcal{G}(R)^{\mathrm{T}} \\ & q = \beta LayerNorm(\sigma((C_{kr} + C_{kd})\mathcal{G}(R))) + \mathcal{G}(R) \\ & \mathcal{G}(S) = q(softmax \sum_{i}(C_{kr} + C_{kd})), \end{aligned} \tag{15}$$

After obtaining the representation of $\mathcal{G}(S)$, we send it into the softmax layer for obtaining the sentiment probability distribution $\mathcal{G}(P)$.

3.6. MDSKI training strategy

Knowledge distillation has proven a promising way to compress large networks while maintaining accuracy, in which the training objection of the student model is the output logits of the teacher model.

In MDSKI, Bert denotes a teacher, and LTNMR denotes a student. Thus, the LTNMR aims to achieve two goals. One is carrying out sentiment analysis for Bert; and another is for LTNMR with knowledge distillation. Formally, the loss function is:

$$\begin{aligned} \mathcal{L} = \alpha_1 \cdot CE(y_s, y) + \alpha_2 \cdot CE(\mathcal{G}(p), y) \\ + \alpha_3 \cdot MSE(y_s, \mathcal{G}(p)), \end{aligned} \tag{16}$$

where $\alpha_{\{1,2,3\}}$ is the hyper-parameter which are summed to 1, y is the gold label. Here, *CE* denotes the standard *Cross-Entropy* function, and *MSE* represents the *Mean Squared Error*. Finally, parameter update is achieved by ADADELTA optimization algorithm in our model.

Table 2
Statistics of the datasets.

Dataset		Positive	Neutral	Negative
SpABSA	Train	1792	1559	1375
	Test	433	412	337
Twitter	Train	1561	3127	1560
	Test	173	346	173
Lap14	Train	994	464	870
	Test	341	169	128
Rest15	Train	912	36	256
	Test	326	34	182
Rest14	Train	2164	637	807
	Test	728	196	196

4. Experiments

4.1. Datasets

To evaluate our proposed method, we conducted extensive experiments on datasets listed in Table 2.

- **Twitter** dataset is obtained from Twitter webpages.[1] [46] Each sentence has labeled aspect terms and a corresponding sentiment label, including "positive", "neutral", or "negative". There were 1561 positive, 3127 neutral, and 1560 negative tweets for training and 692 for the test.
- **Lap14** & **Rest14** are two datasets from the SemEval-14 Task 4 [47]. Lap14 denotes the reviews of laptops, which have 2328 texts with three sentiment polarity classes for training (994 for positive, 464 for neutral and 870 for negative), and 638 samples in the test set. Rest14 is a collection of restaurant reviews that includes 2164 positive, 637 neutral, and 807 negative texts for training, and 1120 test examples.
- **Rest15** is obtained from SemEval-2015 task 12 [48], where 1204 training texts with three sentiment labels and 542 testing data are available.
- **SpABSA** is obtained from [7]. The sentences in SpABSA all contain unique structures (e.g. subjunctive or conditional statements). Most text in SpABSA has a special sentence structure. SpABSA contains 4726 training samples, and the test set includes 1182 samples.

The "conflict sentences" are discarded as in [18], which refers to sentences with any aspect-term with multiple sentiment labels.

4.2. Baselines and experimental setting

Various sentiment classification methods are tested for baseline comparisons.

- **LSTM** [19] conducts the Vanilla LSTM for ABSA. Specifically, Softmax function is applied to the last hidden state of LSTM to acquire the polarity distribution.
- **MemNet** [18] proposes memory network for ABSA, in which multi-hop attention method is proposed.
- **IAN** [21] models the representations of the aspect and text with two independent Vanilla LSTMs. Specifically, IAN conducts the interactive attention to learn the relations between target and text.
- **AOA** [49] jointly learns the aspects and sentences by explicitly modeling the interaction attention methods.

[1] https://twitter.com/

Table 3
Experimental setting.

Parameters	LTNMR (1–3)	LTNMR 4
Word2vec	GloVe "*glove.840B.300d.zip*"	Bert "*bert-base-uncased*"
Hidden size	128	256
Learning rate	0.001	$2e^{-5}$
β from Eq. (15)	–	0.1
λ from Eq. (8)	0.01	
Knowledge utilized	SpaCy for *kd*; SenticNet for *kr*	
Batch size	128	
Optimizer	Adam optimizer	
GPU	Tesla A100 40G	
Initial weight	uniform distributed range from −0.01 to 0.01 randomly	
Experiment platform	*PyTorch* with *cuda-10*	

- **TNet-LF** [50] utilizes transformer structure for model the relation between aspect-term and text.
- **ASGCN** [30] creates GCN to represent the dependency graph, which construct by external dependency tree from SpaCy.
- **R-GAT** [32] re-constructs the syntax tree of ASGCN that is rooted in an aspect-term by reshaping the dependency tree from ASGCN.
- **MIMLLN** [51] defines sentences and words as bags and instances, and the words representing the aspect category as key instances.
- **PhraseRNN** [52] takes both dependency and constituent trees into LSTM. **SynAtten** further integrates attention mechanism with PhraseRNN.
- **Bert** [53] is a pre-trained model to assist ABSA. In particular, we construct the input context and target into a "[CLS] + aspect-term + [SEP] + context" structure.
- **Bert-PT** [53] develops a post-training strategy based on Bert. Here, the input structure is constructed to be "[CLS] + sentence + [SEP] + aspect+ [SEP]".
- **LCF-Bert** [28] introduces local features from contexts into Bert for ABSA.
- **AEN-Bert** [17] combines attention encoder network with the pre-trained Bert, which adapts the Bert-based models into learning target-specific phrases.
- **KGCapsAN** [7] simultaneously introduces multiple external knowledge integrated by a capsule network guided by knowledge. KGCapsAN-B denotes KGCapsAN combined pre-trained Bert model.

Experimental Setting. In all the experiments, the input words are converted into 300-dimensional vectors (given by pre-trained GloVe) in the word embedding layer for LTN/LTNMR (1–3). The initialization of other parameters are obtained by sampling from a uniform distributed range from −0.01 to 0.01 randomly. The Adam optimization method is utilized to optimize the model (batch size = 64, learning rate = 0.001). We select Bert-base[2] in our experiments with the learning rate as $2e^{-5}$. The detail of the experimental setting is shown in Table 3.

Evaluation Metrics. As in [30], widely adopted accuracy and macro-averaged F1 score is used as the evaluation criteria. We first compute the F1 score independently for each class and then take average for all classes as the final performance, hence all classes are treated equally.

4.3. Task setup and quantitative results

Following conventional methods [7,30,51], we executed the procedure thrice to evaluate the model stably, and the best results for the non-Bert and Bert-based models are presented in Tables 4 and 5, respectively. In addition, the t-test was adopted to confirm the significance of differences between methods, with a p-value of 0.05.

Compared with the non-Bert models (from Table 4), the results indicate that LSTM performs the worst because it does not utilize aspect information to predict sentiment. Attention-based methods (methods 3–8) perform much better than LSTM because such approaches directly model the relationship between aspect terms and content. For example, these methods improve by 1.17%, 1.78%, 1.72%, and 5.06% in the F1 of Rest14 dataset, respectively. This illustrates that the attention mechanism helps the model to capture aspect-specific information. Syntax-aware networks perform better than conventional attention-based methods. For example, the ASGCN improves by 4.58% on Twitter and 5.31% on Rest14 for the F1 score. ASGCN improves the learning capacity of dependencies among words within a long range by using a syntactical dependency tree.

According to the results, our model (LTNMR-3) outperformed the baseline approaches by a large margin. First, LTNMR-3 outperforms all nonsyntax baselines (No. 1–8) on all datasets, indicating that LTNMR-3 has a better ability to infer sentiment polarities by utilizing dependency knowledge. Taking the comparison with the best competitor of methods 1–8 as an example, LTNMR-3 improves by 2.87% on Lap14, 4.03% on Rest14 and 3.71% on Rest15 in terms of F1 score. The performance improvement comes from the dependency knowledge distilled from the big model, which is utilized in LTNMR-3, and enriches the learning ability between the word and aspect terms. Second, compared with the syntax-aware models (No. 9–12), LTNMR-3 improves by 2% on Lap14, 2.55% on Rest14, and 4.04% on Rest15 for F1 score. This is because the syntax dependency tree from the SpaCy tool[3] utilized by the conventional method may introduce additional errors, particularly when the text is short and informal. Third, we can observe from LTN (NO. 13–15) and LTNMR (NO. 17–19), the method with full knowledge, LTNMR-3, performs the best. This is because our LTNMR framework incorporated richer knowledge.

As shown in Table 5, the LTN-based method is also capable of utilizing the external knowledge conveyed by the pretrained Bert model (LTNMR-4). Comparisons between our LTNMR-4 and comparable Bert models are presented in Table 5. The results clearly show that the proposed LTNMR-4 has the best performance in most cases, which justifies the usefulness of our method in utilizing Bert knowledge. In conclusion, the advantage of LTNMR stems from it providing sufficient syntax knowledge, allowing the neural network to combine prior knowledge effectively.

4.3.1. Computational efficiency vs. accuracy

Fig. 4 summarized the parameter size and the F1 score results. Our proposed method achieves the best trade-off between model performances and trainable parameter sizes (also reflects the

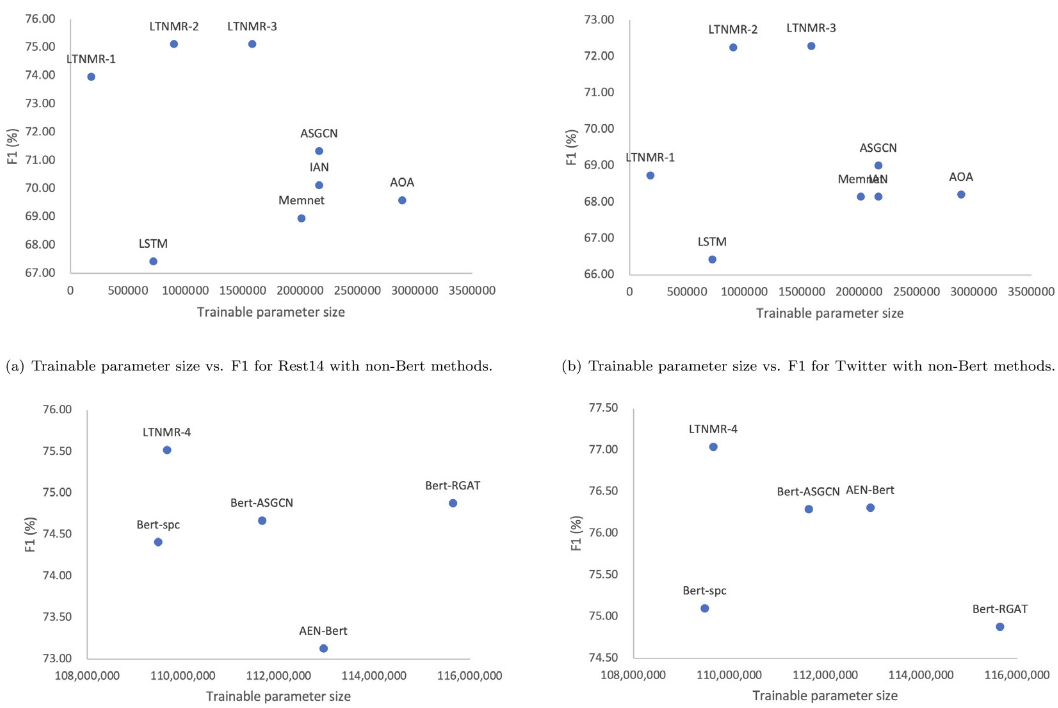
[2] Provided by Huggingface https://huggingface.co/.

[3] https://spacy.io/

Table 4
Evaluation results (%) of non-Bert model. The best result on each task is in bold.

	Model	Twitter		Lap14		Rest14		Rest15		SpABSA	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
1	SVM¶	63.40	63.30	70.49	–	80.16	–	–	–	–	–
2	LSTM†	69.56	66.42	69.28	63.21	78.13	67.42	77.37	52.97	61.16	59.17
3	Memnet¶	71.48	68.14	70.64	65.19	79.61	68.14	77.31	56.17	61.42	57.56
4	AOA†	72.30	68.20	72.62	66.97	79.97	69.59	78.17	57.21	63.76	64.41
5	IAN†	72.50	68.14	72.05	67.38	79.26	70.12	78.54	52.21	63.76	63.96
6	CapsNet¶	–	–	–	–	69.63	69.63	78.14	61.57	–	–
7	TNet-LF†	72.98	71.43	74.61	70.14	80.40	70.57	78.47	59.12	63.76	64.96
8	MIMLLN¶	–	- -	–	–	81.06	71.25	78.27	60.59	–	–
Syntax-aware Network											
9	PhraseRNN¶	–	–	–	–	66.20	59.32	–	–	–	–
10	SynAttn¶	–	–	72.57	69.13	80.45	71.26	–	–	–	–
11	ASGCN†	72.15	71.00	71.05	70.72	80.86	72.73	79.89	59.47	66.24	65.24
12	R-GAT†	71.56	71.07	72.49	71.01	73.83	72.14	78.92	61.24	66.87	65.14
LTN											
13	LTN-1	65.14	62.92	65.11	62.09	78.48	65.14	75.49	52.12	61.77	61.17
14	LTN-2	66.94	64.91	70.12	65.42	79.16	68.06	78.90	56.71	62.30	60.61
15	LTN-3	66.99	65.02	70.85	65.94	79.21	68.72	78.78	57.21	62.14	62.72
LTN-MR											
17	LTNMR-1	70.21	68.76	72.57	68.15	81.16	71.87	79.97	57.76	63.74	60.14
18	LTNMR-2‡	73.14	72.24	76.95	72.95	**83.13**	75.12	81.01	64.11	67.60	67.38
19	LTNMR-3‡	**73.21**	**72.29**	**77.49**	**73.01**	83.09	**75.28**	**81.22**	**65.28**	**67.92**	**67.41**

The mark ¶ refers to the results reported in the original papers, while † mark refers to the open implementation, ‡ mark refers to p-value < 0.05.



(a) Trainable parameter size vs. F1 for Rest14 with non-Bert methods.
(b) Trainable parameter size vs. F1 for Twitter with non-Bert methods.
(c) Trainable parameter size vs. F1 for Twitter with Bert-based methods.
(d) Trainable parameter size vs. F1 for Lap14 with Bert-based methods.

Fig. 4. Trainable parameter size vs. F1 score.

training cost). First, compare LTNMR with the non-Bert baselines, our method has fewer parameters but achieves state-of-the-art results. For instance, LTNMR-2 has 58.32% fewer parameters than ASGCN, but the performance improved by 4.46% on Rest15 and 2.23% on Lap14. Noticing that, even with the fewest model parameters (even fewer 74.91% than standard LSTM), LTNMR-1 can still produce comparable results to the conventional best attention mechanism model (ASGCN). Since the dependency knowledge of LTNMR-4 is produced from the pre-trained Bert, it needs almost the same computational complexity as Bert. However, other strong competitors usually require more trainable features, as well as GPU memory costs. In conclusion, the results demonstrate the utility of the MDSKI framework and demonstrate that our model is capable of striking a balance between computational efficiency and high inferring accuracy.

4.4. Ablation study

In order to better understand the contributions of each component of our proposed LTNMR, we conduct the ablation test

Model	Prediction	Label		Attention visualization	Aspect
MemNet	positive	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
IAN	positive	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
ASGCN	neutral	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
LTN-3	positive	negative	×	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	negative	positive	×	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion
LTNMR-3	negative	negative	√	but , the filet mignon was not very good at all cocktail hour includes free appetizers -LRB- nice non-sushi selection -RRB-.	filet mignon
	positive	positive	√	The Sashimi portion are big enough to appease most people , but I did n't like the fact they used artificial lobster meat .	Sashimi portion

Fig. 5. Attention visualization of two examples. The √ indicates a correct prediction while × indicates an incorrect prediction.
Source: Samples are selected from [7].

Table 5
Evaluation results (%) for Bert-based methods. The best result on each task is in bold.

	Model	Twitter		Lap14		Rest14	
		Acc	F1	Acc	F1	Acc	F1
1	Bert	74.42	72.89	78.84	75.10	85.45	78.42
2	Bert-PT	–	–	78.07	75.08	84.98	76.96
3	LCF-Bert	73.27	72.35	79.62	76.26	84.38	78.72
4	AEN-Bert	74.71	73.13	**79.93**	76.31	83.12	73.76
5	KGCapsAN	74.13	72.52	76.96	72.89	82.05	74.04
6	Ours (LTN-4)	74.53	72.94	79.26	75.63	**85.48**	78.66
7	Ours (LTNMR-4)	**76.02**	**73.34**	79.77	**76.51**	85.36	**80.96**

Table 6
Ablation study results (F1%) of LTNMR-3.

Methods	Twitter (%)	Lap14 (%)	Rest14 (%)
LTNMR-3	72.29	73.01	75.28
w/o MDSKI	71.09	71.82	72.63
w/o K	71.07	72.46	73.04
-w/o kd	72.23	72.93	73.85
-w/o kr	72.14	72.87	74.49

of LTNMR-3 by discarding the MDSKI (denoted as w/o MDSKI), both two type knowledge (denoted as w/o K), the dependency knowledge (denoted as w/o kd), the knowledge rule (denoted as w/o kr). The ablation results are summarized in Table 6.

Specifically, we construct w/o MDSKI by utilizing the BiLSTM and the standard attention mechanism, and utilize the standard dependency tree as the syntax knowledge. According to Table 6, we can find that all the proposed components contribute a great improvement to LTNMR. In particular, the F1 score decreases sharply when discarding the MDSKI framework. This is within our expectations since the MDSKI injects the dependency knowledge from Bert into LTNMR. For example, the F1 score drops 1.12%, 1.19%, and 2.65% for Twitter, Lap14 and Rest14, respectively. In addition, both two types of knowledge also contribute to the effectiveness of LTNMR for Rest14. For example, the F1 score decreases 1.43% for Rest14 when discarding *kd*, and decreases 0.15% for Twitter when discarding *kr*. Not surprisingly, the full combination of all components achieves the best performance among all the experiments.

4.5. Case study

In order to provide an intuitive demonstration, we perform case study with two examples. Specifically, we visualize and compare the attention weights calculated by baseline methods with strong performance, and our model LTN-3 and LTNMR-3 in Fig. 5. The prediction and the ground truth labels are also provided for such instances.

Firstly, the first sentence contains the special structure "*not very good*". Modeling such a structure using traditional methods is difficult. For example, MemNet is unable to identify the sentiment-bare words. IAN and ASGCN focus on "good", which leads to wrong predictions. Since LTN did not use the dependency knowledge, where the structure is similar to the attention-based model, thus LTN predicts the wrong polarity. For the second sentence, which has two different aspect-terms with opposite emotional labels, such as *sashimi portion* and *artificial lobster meat*. In this case, the conventional attention model may incorrectly align the aspects with the aspect-related words. When it comes to the aspect "sashimi portion", the existing methods (IAN, MemNet and ASGCN) tend to focus on the words which in fact describe "artificial lobster meat". The LTNMR-3 can accurately predict these sentences. This may be the reason that the dependency knowledge produced by Bert enriches the learning ability of LTN, while knowledge rules integrate the external emotional feature, which can help identify the sentiment-bare words.

4.6. Error analysis

To better understand the limitations of LTNMR, we additionally carry out an analysis of the errors made by LTNMR. Specifically, we randomly select 50 instances that are incorrectly predicted by LTNMR. We revealed several reasons for the classification errors, which can be divided into the following categories. First, LTNMR fails to understand some sentences that the stance towards a given target is expressed in an implicit way. For example, the correct label for the sentence "*You can eat gourmet food at a fast food price.*" should be "positive" towards the aspect-term "price". LTNMR tends to classify the sentence as "neutral" label, because the model was unable to understand the internal relation between "fast food price" and "gourmet food". The second error category is caused by the external knowledge introduced in LTNMR. For example, for the "positive" sentence "*It has so much more speed and the screen is very sharp*", with the aspect-term "screen", LTNMR tends to predict an incorrect "negative" sentiment polarity. This is because the emotional words of "sharp" are "sadness" and "disgust" from SenticNet lexicon, which may lead to mispredictions. Additionally, some errors are caused by the complex and informal text structure, which poses a big challenge for the dependency knowledge (*kd*) utilized in LTNMR. It suggests that certain knowledge selection strategy needs to be devised in the future.

5. Conclusion

In this work, a novel LTNMR network is proposed for ABSA. LTNMR is interpretable because it is constructed using FOL. The sentiment inference process can be decomposed into different grounding, which is constructed using a trainable neural network. To achieve high inferring accuracy, we propose a MDSKI strategy. The performance of the experiments demonstrated that the proposed LTNMR with the MDSKI strategy significantly outperforms state-of-the-art Bert-based methods for aspect-based sentiment analysis. In the future, we intend to extend our proposed framework to similar tasks, including both natural language processing and computer vision applications.

CRediT authorship contribution statement

Hu Huang: Conception and design of study, Acquisition of data, Writing – original draft. **Bowen Zhang:** Conception and design of study, Writing – original draft, Writing – review & editing. **Liwen Jing:** Acquisition of data, Writing – review & editing. **Xianghua Fu:** Analysis and/or interpretation of data. **Xiaojun Chen:** Acquisition of data, Analysis and/or interpretation of data. **Jianyang Shi:** Conception and design of study.

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 research was supported in part by Stable Support Project for Shenzhen Higher Education Institutions (SZWD2021011); Shenzhen Research Foundation for Basic Research, China, under Grant JCYJ20210324093000002. All authors approved the version of the manuscript to be published.

References

[1] S.M.J. Zafra, M.T.M. Valdivia, E. Martínez-Cámara, L.A.U. López, Studying the scope of negation for spanish sentiment analysis on Twitter, IEEE Trans. Affect. Comput. 10 (1) (2019) 129–141.
[2] S. Poria, N. Ofek, A.F. Gelbukh, A. Hussain, L. Rokach, Dependency tree-based rules for concept-level aspect-based sentiment analysis, 475, 2014, pp. 41–47.
[3] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, Target-dependent twitter sentiment classification, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 151–160.
[4] I. Goodfellow, Y. Bengio, A. Courville, Y. Bengio, Deep Learning, Vol. 1, MIT press Cambridge, 2016.
[5] H. Liu, I. Chatterjee, M. Zhou, X.S. Lu, A. Abusorrah, Aspect-based sentiment analysis: A survey of deep learning methods, IEEE Trans. Comput. Soc. Syst. 7 (6) (2020) 1358–1375.
[6] H. Chen, Y. Ji, Improving the explainability of neural sentiment classifiers via data augmentation, 2019, arXiv preprint arXiv:1909.04225.
[7] B. Zhang, X. Li, X. Xu, K. Leung, Z. Chen, Y. Ye, Knowledge guided capsule attention network for aspect-based sentiment analysis, IEEE ACM Trans. Audio Speech Lang. Process. 28 (2020) 2538–2551.
[8] Z. Hu, X. Ma, Z. Liu, E. Hovy, E. Xing, Harnessing deep neural networks with logic rules, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 2410–2420.
[9] Z. Hu, Z. Yang, R. Salakhutdinov, E. Xing, Deep neural networks with massive learned knowledge, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 1670–1679.
[10] B. Zhang, X. Xu, X. Li, X. Chen, Y. Ye, Z. Wang, Sentiment analysis through critic learning for optimizing convolutional neural networks with rules, Neurocomputing 356 (2019) 21–30.
[11] S. Badreddine, A. d'Avila Garcez, L. Serafini, M. Spranger, Logic tensor networks, Artificial Intelligence 303 (2022) 103649.
[12] F. Bianchi, P. Hitzler, On the capabilities of logic tensor networks for deductive reasoning, in: Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering, in: CEUR Workshop Proceedings, Vol. 2350, 2019.
[13] F. Bianchi, M. Palmonari, P. Hitzler, L. Serafini, Complementing logical reasoning with sub-symbolic commonsense, in: Rules and Reasoning - Third International Joint Conference, in: Lecture Notes in Computer Science, Vol. 11784, Springer, 2019, pp. 161–170.
[14] W. Tu, D.W. Cheung, N. Mamoulis, M. Yang, Z. Lu, Investment recommendation using investor opinions in social media, in: Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2016, pp. 881–884.
[15] E. Cambria, Affective computing and sentiment analysis, IEEE Intell. Syst. 31 (2) (2016) 102–107.
[16] H.T. Phan, N.T. Nguyen, D. Hwang, Convolutional attention neural network over graph structures for improving the performance of aspect-level sentiment analysis, Inform. Sci. 589 (2022) 416–439.
[17] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, 2019, arXiv preprint arXiv:1902.09314.
[18] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 214–224.
[19] D. Tang, B. Qin, X. Feng, T. Liu, Effective LSTMs for target-dependent sentiment classification, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.
[20] L. Li, Y. Liu, A. Zhou, Hierarchical attention based position-aware network for aspect-level sentiment analysis, in: Proceedings of the 22nd Conference on Computational Natural Language Learning, 2018, pp. 181–189.
[21] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, AAAI Press, 2017, pp. 4068–4074.
[22] C. Sun, L. Huang, X. Qiu, Utilizing BERT for aspect-based sentiment analysis via constructing auxiliary sentence, in: Proceedings of NAACL-HLT, 2019, pp. 380–385.
[23] B. Liang, H. Su, L. Gui, E. Cambria, R. Xu, Aspect-based sentiment analysis via affective knowledge enhanced graph convolutional networks, Knowl. Based Syst. 235 (2022) 107643.
[24] H. Wu, Z. Zhang, S. Shi, Q. Wu, H. Song, Phrase dependency relational graph attention network for aspect-based sentiment analysis, Knowl. Based Syst. 236 (2022) 107736.
[25] H. Chen, Z. Zhai, F. Feng, R. Li, X. Wang, Enhanced multi-channel graph convolutional network for aspect sentiment triplet extraction, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, Association for Computational Linguistics, 2022, pp. 2974–2985.
[26] W. Shi, F. Li, J. Li, H. Fei, D. Ji, Effective token graph modeling using a novel labeling strategy for structured sentiment analysis, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022, pp. 4232–4241.
[27] H. Xu, B. Liu, L. Shu, S.Y. Philip, BERT post-training for review reading comprehension and aspect-based sentiment analysis, 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, pp. 2324–2335.
[28] B. Zeng, H. Yang, R. Xu, W. Zhou, X. Han, Lcf: A local context focus mechanism for aspect-based sentiment classification, Appl. Sci. 9 (16) (2019) 3389.
[29] J. Dai, H. Yan, T. Sun, P. Liu, X. Qiu, Does syntax matter? A strong baseline for aspect-based sentiment analysis with roberta, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021, Association for Computational Linguistics, 2021, pp. 1816–1829.
[30] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, 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, pp. 4560–4570.
[31] X. Bai, P. Liu, Y. Zhang, Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network, IEEE ACM Trans. Audio Speech Lang. Process. 29 (2021) 503–514.
[32] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2020, pp. 3229–3238.
[33] T. Yang, Q. Yin, L. Yang, O. Wu, Aspect-based sentiment analysis with new target representation and dependency attention, IEEE Trans. Affect. Comput. (2019) 1.

[34] F. Chen, J. Xia, H. Gao, H. Xu, W. Wei, TRG-datt: The target relational graph and double attention network based sentiment analysis and prediction for supporting decision making, ACM Trans. Manag. Inf. Syst. 13 (1) (2022) 3:1–3:25.

[35] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, K. Dai, Enhancing Cross-target Stance Detection with Transferable Semantic-Emotion Knowledge, Association for Computational Linguistics, 2020.

[36] M. Zhang, T. Qian, Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 3540–3549.

[37] Y. Choi, J. Wiebe, R. Mihalcea, Coarse-grained +/-effect word sense disambiguation for implicit sentiment analysis, IEEE Trans. Affect. Comput. 8 (4) (2017) 471–479.

[38] R.K. Yadav, L. Jiao, O. Granmo, M. Goodwin, Human-level interpretable learning for aspect-based sentiment analysis, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, the Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021, 2021, pp. 14203–14212.

[39] B. Bhattarai, O.-C. Granmo, L. Jiao, Word-level human interpretable scoring mechanism for novel text detection using tsetlin machines, Appl. Intell. (2022) 1–25.

[40] R.K. Yadav, L. Jiao, O.-C. Granmo, M. Goodwin, Robust interpretable text classification against spurious correlations using AND-rules with negation, in: L.D. Raedt (Ed.), Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22, 2022, pp. 4439–4446.

[41] J. Hewitt, C.D. Manning, A structural probe for finding syntax in word representations, 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, pp. 4129–4138.

[42] E. Cambria, S. Poria, R. Bajpai, B. Schuller, Senticnet 4: A semantic resource for sentiment analysis based on conceptual primitives, in: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 2666–2677.

[43] E. Cambria, S. Poria, D. Hazarika, K. Kwok, Senticnet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[44] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, Senticnet 6: Ensemble application of symbolic and subsymbolic AI for sentiment analysis, in: CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020.

[45] E. Cambria, Q. Liu, S. Decherchi, F. Xing, K. Kwok, Senticnet 7: a commonsense-based neurosymbolic AI framework for explainable sentiment analysis, in: Proceedings of LREC 2022, 2022.

[46] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2014, pp. 49–54.

[47] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEVal 2014), 2014.

[48] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), 2015, pp. 486–495.

[49] B. Huang, Y. Ou, K.M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation, Springer, 2018, pp. 197–206.

[50] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 946–956.

[51] Y. Li, C. Yin, S. Zhong, X. Pan, Multi-instance multi-label learning networks for aspect-category sentiment analysis, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2020, pp. 3550–3560.

[52] T.H. Nguyen, K. Shirai, Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2509–2514.

[53] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 4171–4186.

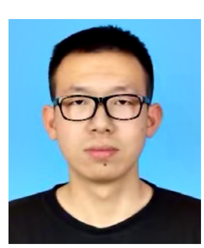
Hu Huang received the B.Eng. from Huazhong University of Science and Technology in 2004 and 2009, and received M.Sc. from Peking University. Once worked as a senior engineer in The Smart City Research Institute of China Electronics Technology Group Corporation. He is now a student at the School of Cyber Science and Technology at the University of Science and Technology of China. His research interests are in the areas of deep learning, graph mining, natural language processing and social computing.

Bowen Zhang is now an assistant professor in College of Big Data and Internet, Shenzhen Technology University. He received a Ph.D. in Computer Science from Harbin Institute of Technology in 2021, and M.Sc. degrees in computer science and application from Macau University of Science and Technology, Taipa, China. His research interests are in natural language processing, big data analysis, data mining.

Liwen Jing received her Ph.D. degree from the Hong Kong University of Science and Technology (HKUST), Hong Kong, China in 2018. She is currently an assistant professor in the College of Big Data and Internet at Shenzhen Technology University. Her research interests include natural language processing and wireless communications. She was a recipient of the best student paper award in MTT-S IWS 2016 Shanghai conference and the HKUST Overseas Research Award to visit Northeastern University in Boston as a visiting research scientist.

Xianghua Fu received the Ph.D. degree in Computer Science and Technology from Xi'an Jiaotong University, Shanxi, China. He is currently a Professor of the College of Big Data and Internet, and Vice Dean of the College of Big Data and Internet. His research interests include natural language processing, information retrieval, machine learning and data mining, etc. He has published more than 60 articles in lots of important journals and international conferences at home and abroad.

Xiaojun Chen (Member, IEEE) received the Ph.D. degree from the Harbin Institute of Technology in 2011. He is currently an Associate Professor with the College of Computer Science and Software, Shenzhen University. His research interests include multimodal learning, unsupervised/supervised/few-shot/weakly supervised learning.



Jianyang Shi received a B.Eng. degree in information science and engineering from Yunnan University, Kunming, China, in 2019. He is currently working toward a Ph.D. degree in computer science and technology from Harbin Institute of Technology, Shenzhen, China. His research interests include computer vision, image synthesis, and fashion intelligence.