Knowledge Graph Enhanced Language Models for Sentiment Analysis

Jie Li[1], Xuan Li[1], Linmei Hu[2(✉)], Yirui Zhang[1], and Jinrui Wang[1]

[1] Beijing University of Posts and Telecommunications, Beijing, China
{Jli,lixuan20000530,zhangyirui,wangjr}@bupt.edu.cn
[2] Beijing Institute of Technology, Beijing, China
hulinmei@bit.edu.cn

Abstract. Pre-trained language models (LMs) have been widely used in sentiment analysis, and some recent works have focused on injecting sentiment knowledge from sentiment lexicons or structured commonsense knowledge from knowledge graphs (KGs) into pre-trained LMs, which have achieved remarkable success. However, these works often only obtain knowledge from a single source in either the sentiment lexicon or the KG, and only perform very shallow fusion of LM representations and external knowledge representations. Therefore, how to effectively extract multiple sources of external knowledge and fully integrate them with the LM representations is still an unresolved issue. In this paper, we propose a novel knowledge enhanced model for sentiment analysis (KSA), which simultaneously incorporates commonsense and sentiment knowledge as external knowledge, by constructing a heterogeneous Commonsense-Senti Knowledge Graph. Additionally, a separate global token and global node are added to the text sequence and constructed knowledge graph respectively, and a fusion unit is used to enable global information interaction between the different modalities, allowing them to perceive each other's information and thereby improving the ability to perform sentiment analysis. Experiments on standard datasets show that our proposed KSA significantly outperforms the strong pre-trained baselines, and achieves new state-of-the-art results on most of the test datasets.

Keywords: Knowledge Graph · Knowledge Fusion · Sentiment Analysis

1 Introduction

Sentence-level sentiment analysis strives to extract the overall sentiment, which has garnered considerable attention in natural language processing (NLP) [1,2]. Recently, pre-trained language models (LMs) [3–6] have shown their power in learning general semantic representations, leading to significant advancements in most NLP tasks, including sentiment analysis. These models learn encoders on large-scale corpora via well-designed pre-training tasks [7]. However, the application of general purposed pre-trained LMs in sentiment analysis is limited, because they neglect to consider the importance of external knowledge [8].

T. R. Payne et al. (Eds.): ISWC 2023, LNCS 14265, pp. 447–464, 2023.
https://doi.org/10.1007/978-3-031-47240-4_24

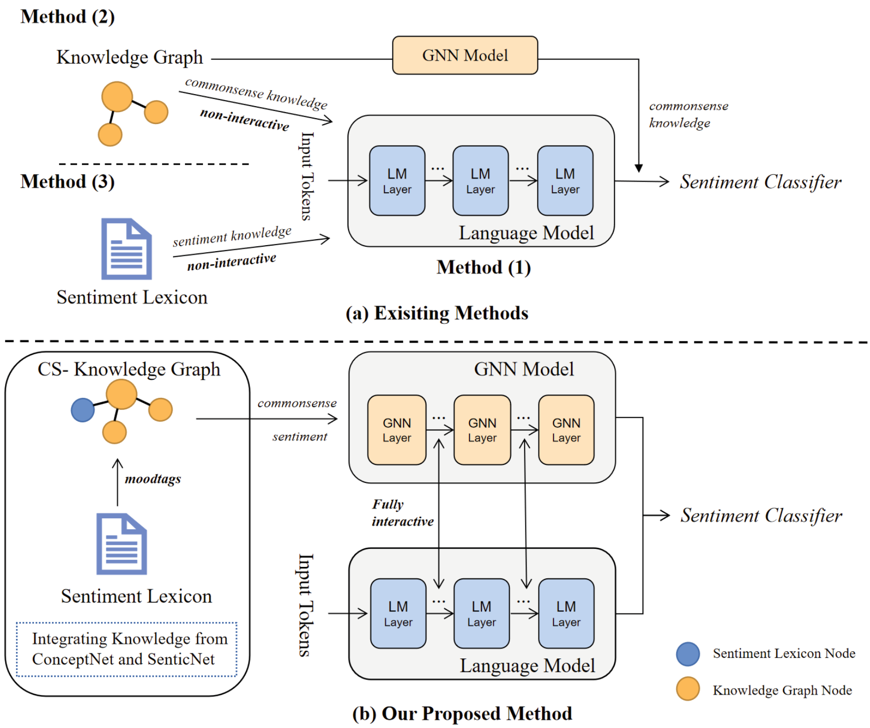


Fig. 1. Illustration of comparison of existing methods and our proposed method. Method (1) directly performs sentiment analysis using pretrained LMs. Method (2) injects commonsense knowledge from the knowledge graph into LMs in a non-interactive manner. It can either encode LM representations and graph representations separately, where graph representations directly participate in the final sentiment analysis, or enhance text encoding representations using commonsense knowledge during the LM encoding stage. Method (3) enhances text encoding representations using sentiment knowledge from the sentiment lexicon. Our proposed method utilizes knowledge from both sources to construct a CS-knowledge graph and effectively integrates graph representations and LM representations for sentiment analysis.

Some recent works attempt to integrate various knowledge into pre-trained LMs (see Fig. 1). On the one hand, some researches [9,10] have infused commonsense knowledge using massive knowledge graphs (KG), such as ConceptNet [11], Freebase [12]. On the other hand, sentiment lexicons, such as SenticNet [13], SentiWordNet [14], have been injected into pre-trained LMs [15–18]. These researches have demonstrated the significant role of KGs or sentiment lexicons in sentiment polarity prediction. However, these methods still have two problems: 1) Using only one kind of knowledge. Prior methods typically inject one kind of knowledge into LMs, but whether the KG that provides a rich source of background concepts or the sentiment lexicon that provides specific moodtags is helpful to enhance the sentiment polarity prediction. For example, in the case of a movie review, KG can provide commonsense information such as director, cast, plot, and themes, while sentiment lexicon can offer corresponding positive or negative moodtags to the words in the review. Therefore, there are still challenges in how to acquire knowledge from both sources. 2) How to effectively fuse the external knowledge. The existing methods can only fuse the external knowledge representations and LM representations in a shallow and non-interactive manner, which constrain the performance of the model. Exploring how to effectively integrate two representations in a truly unified manner is still an underexplored area.

In this work, we present KSA, a novel knowledge enhanced model for sentiment analysis that addresses the two problems mentioned above. Our KSA has two key insights: (i) Constructing a heterogeneous Commonsense-Senti knowledge graph(CS-knowledge graph). We construct a CS-knowledge graph by integrating knowledge from the commonsense KG ConceptNet and the sentiment lexicon SenticNet to represent external knowledge that may contribute to sentiment polarity prediction. Based on the input, we first retrieve the corresponding entities from ConceptNet. Then, we iterate through these entities and search for their moodtags in SenticNet. The retrieved entities and moodtags are then used as nodes in the CS-knowledge graph. Corresponding edges are retrieved from ConceptNet to connect these nodes, resulting in the CS-knowledge graph that encompasses both commonsense and sentiment knowledge. (ii) Deep fusion of LM representations and graph representations. Our proposed KSA includes multiple stacked fusion layers, each of which is composed of a LM layer, a GNN layer, and a fusion unit. We encode the input text and the CS knowledge graph separately using LM and GNN. Additionally, the text sequence and CS knowledge graph are equipped with a global token and a global node, respectively, to capture global information. After each layer of LM and GNN encoding, the global token and global node are input into a special fusion unit, where a deep fusion of the two modalities is performed. The fused global token and global node then enter the next round of representation update, integrating global information from each other into their own modality representations, bridging the gap between the two sources of information.

Our contributions are outlined below.

- To the best of our knowledge, we propose for the first time to enhance sentiment classification using both commonsense and sentiment knowledge. Specifically, we construct a Commonsense-Senti knowledge graph for each input and employ GNN layers to learn the rich external knowledge in the graph. By integrating external knowledge representations with text representations, we effectively improve the performance of sentiment classification.
- In our KSA model, we designed a specialized fusion mechanism. The representations of the global token and global node are extracted, concatenated, and fed into the fusion unit to mix their representations. In subsequent layers, the mixed information from the global elements is combined with their respective modality representations. Through this mechanism, our model fully and effectively integrates knowledge.

- We conduct extensive experiments and achieve new state-of-the-art results on most of the test datasets, which proves the effectiveness of the KSA fusion approach and the significance of simultaneously injecting commonsense and sentiment knowledge.

2 Related Work

2.1 Incorporating External Knowledge for NLP

Various works have incorporated knowledge to augment NLP systems [19–23]. For example, ERNIE 3.0 [19] augments the original input sentence with triples, such as (Andersen, Write, Nightingale), which are then used as the basis for designing tasks that aim to predict the relationship between the entities in the triple, in this case the relation "Write". K-BERT [20] attaches triples to entities in the input sentence to create a sentence tree, and uses soft-position and visible matrix to reduce knowledge noise. SenseBERT [21] integrates word-supersense knowledge by predicting the supersense of masked words in the input, where candidates are nouns and verbs, and ground truth is derived from WordNet. KnowBERT [22] integrates knowledge bases into BERT by employing Knowledge Attention and Recontextualization mechanisms. The knowledge sources used are derived from synset-synset and lemma-lemma relationships in WordNet, as well as entity linking information extracted from Wikipedia. K-Adapter [23] develops adapters and treats them as add-ons with knowledge representations. These adapters are separated from the backbone pre-trained LMs and are trained from scratch through self-designed task. The above-mentioned methods for knowledge fusion are often unidirectional. To be more specific, while their fusion units empower the LMs with external knowledge, they miss out on the potential benefits of integrating contextual information from the LMs into the KG. A more comprehensive and bidirectional knowledge fusion process could be highly advantageous.

2.2 Incorporating External Knowledge for Sentiment Analysis

Analogously, external knowledge can be typically used as a source for enhancing the sentiment feature representations in the task of sentiment analysis (i.e., structured commonsense knowledge and sentiment knowledge). On the one hand, a line of works utilize commonsense knowledge from KGs to enhance sentiment analysis. Some of them [10,24] encode structured knowledge representations and language representations respectively, where graph representations directly participate in the final sentiment analysis. For example, KinGDOM [10] concatenates the graph feature representations learned through graph convolutional autoencoder and the language representations learned through DANN autoencoder to perform sentiment classification task. KGAN [24] integrates the knowledge graph into the embedding space, which is then fed into a hierarchical fusion module to fuse the learned multiview representations. Others [9,25,26] seek to use encoded representations of a linked KG to augment the textual representations. SEKT [9] uses the external knowledge to construct a knowledge graph, which is then fed into a graph convolutional network to learn graph representations, and it is fully integrated into the bidirectional long short-term memory (BiLSTM) stance classifier to enhance the text representations. SAKG-BERT [25] model constructs an SAKG in which triples are injected into sentences as domain knowledge to improve the interpretability of the deep learning algorithm. KG-MPOA [26] proposes a matching and filtering method to distill useful knowledge in the ConceptNet, and a bi-directional long-short term memory model with multipolarity orthogonal attention is adopted to fuse the distilled knowledge with the semantic embedding, effectively enriching the representations of sentences. On the other hand, sentiment lexicons are usually injected into LMs by designing sentiment-aware tasks [8,27–30]. For example, SKEP [27] integrates sentiment information at the word, polarity, and aspect levels into pre-trained sentiment representations. SentiLARE [28] incorporates linguistic knowledge at the word-level, such as part-of-speech tags and sentiment polarity (derived from SentiWordNet) into pre-trained LMs.

However, the existing methods still fall short in exploring the knowledge to augment the sentiment analysis. One main reason for this is that the interaction between external knowledge and LMs is limited as information between them only flows in one direction, often relying on external knowledge to enhance text representation. In addition, existing methods often choose one of sentiment lexicon or KG as external knowledge, but we think it is possible to combine the two to enrich the feature representations of external knowledge.

In contrast to prior works, we synergistically combine the LMs with both KG and sentiment knowledge by constructing CS-knowledge graph to obtain richer feature representations and effectively boost the performance of sentiment analysis. Additionally, both the text representations sent to LM and the CS-knowledge graph representations sent to GNN are attached with a global information extraction section, which can fuse the two modalities after each layer of LM and GNN, so that both modalities can reflect specific aspects of the other.

3 Methodology

3.1 Task Definition and Model Overview

We aim to determine the sentiment polarity of sentences by leveraging knowledge from a pre-trained LM and a structured KG. In the task of sentence-level sentiment classification (SSC), the dataset is typically composed of examples of a text sentence s and a digital label l. In particular, in this work, we will convert the numerical labels in the examples into textual labels, denoted by a. For example, in binary classification problems, label “1” corresponds to “It is positive” and label “0” corresponds to “It is negative”, thus connecting textual label a with the sentence s to form a pure text input (s, a). Note that we link all text-based label a options in the dataset with s in turn as input to judge the polarity of the sentence by scores. Furthermore, the external knowledge graph that we access is referred to as $\mathcal{G}$, which offers background knowledge relevant to the sentences being analyzed.

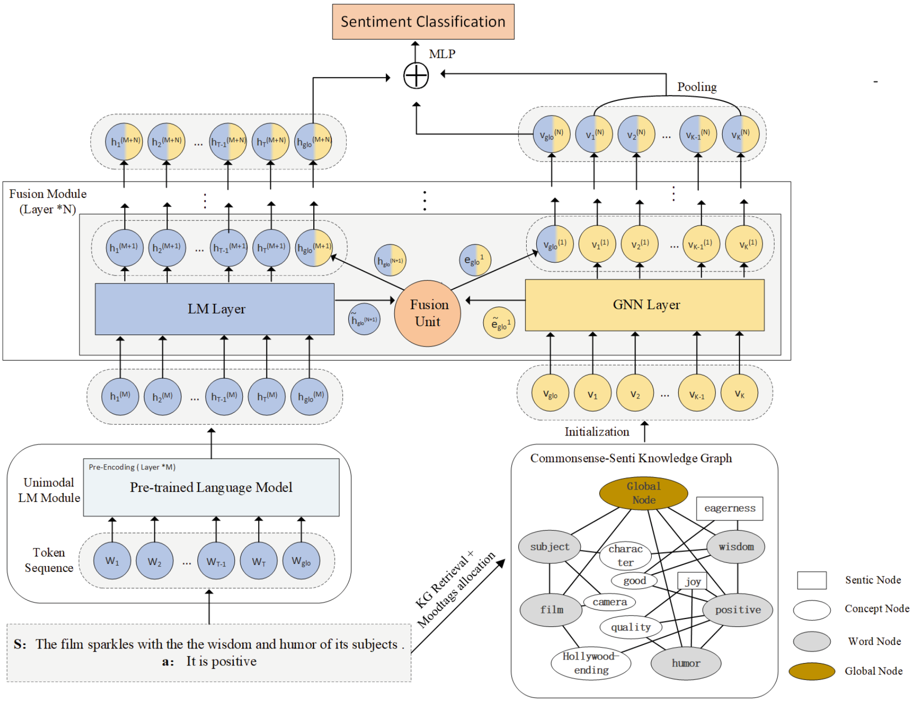


Fig. 2. Overview of our approach. The input tokens are attached with a special global token to extract the global information of the LM representations, and pre-encoding is performed through LM layers. At the same time, the corresponding CS-Knowledge Graph is extracted based on the input, and word nodes are connected to the global node to capture the global information of the graph. Then, both modalities enter the Fusion Module, with the language representations continuing to be updated through the LM layers, and the KG being processed using GNN Layers for information propagation between nodes. In each layer, after the representations of both modalities are updated, global token and node are extracted to exchange global information through the Fusion Unit. In subsequent layers, the mixed global token allows knowledge from the KG to influence the representations of other tokens, while the mixed global node allows language context to influence the node representations in the GNN.

As illustrated in Fig. 2, our model consists of three primary components: (1) a Commonsense-Senti knowledge graph (CS-knowledge graph) building module, (2) a unimodal LM module that learns an initial representation of the input tokens, and (3) a fusion module which learns to update representations of the input sequence and retrieved CS-knowledge graph, enabling the mixing of the textual representations derived from the underlying LM module with the graph representations. The unimodal module is composed of M stacked LM layers, while the fusion module is composed of N stacked layers, each includes a LM layer, a GNN layer and a fusion unit.

Given a textual input (s, a), first, we build a CS-knowledge graph (denoted $\mathcal{G}^{cs}$) from the KG ConceptNet and the sentiment lexicon SenticNet (Sect. 3.2). Meanwhile, we tokenize the combined sequence into $\{w_1, ..., w_T\}$, where T is the total number of tokens, which are then fed into the LM to obtain a pre-encoding representations (Sect. 3.4). Then the pre-encoding token representations denoted as $\{h_1^M, ..., h_T^M\}$, where M is the total number of layers in the LM unimodal module, and the set of nodes denoted as $\{v_1, ..., v_K\}$, where K is the total number of nodes, are fed into the fusion module. Within this module, LM is utilized to update textual representations, and GNN is employed to learn graph representations that capture semantic connections between nodes. Additionally, we introduce a special global token w_{glo} and a special global node v_{glo} to propagate global information from both modalities. The global token is added to the token sequence, and the global node is connected to all the KG entities mentioned in the given input sequence (Sect. 3.5). Finally, we utilize the w_{glo} token representation, v_{glo} node representation, and a pooled $\mathcal{G}^{cs}$ representation to make the final prediction (Sect. 3.6).

3.2 Commonsense-Senti Knowledge Graph Construction

In addition to the general knowledge provided by the LM, sentiment analysis often requires external knowledge such as world knowledge and specific sentiment knowledge. While language models excel at understanding human-like text, they may lack the contextual understanding necessary to accurately determine sentiment. By incorporating external knowledge sources, such as knowledge graphs, sentiment lexicons, we can enhance the LM's sentiment analysis capabilities. World knowledge helps the LM interpret language nuances and understand the connotations associated with certain words or phrases. For example, understanding that certain events or cultural references may have a positive or negative sentiment can greatly improve the accuracy of sentiment analysis. Meanwhile, by incorporating specific sentiment knowledge into the LM, it can better understand the sentiment orientation of words and phrases, enabling more precise sentiment analysis. Therefore, we construct a Commonsense-Senti knowledge graph (CS-knowledge graph) to represent the external knowledge that may contribute to sentiment polarity prediction.

Algorithm 1 describes the details of constructing the CS-Knowledge Graph. Given each input sequence (s, a), we retrieve the knowledge graph $\mathcal{G}^{cs}$ from $\mathcal{G}$. First, we perform entity linking to $\mathcal{G}$ based on (s, a) in order to identify the word nodes. We iterate through each word in the input, and if it appears in ConceptNet, we select it as an entity in our CS-knowledge graph. Next, we consider all entities that appear in any two-hop paths between the mentioned entity pairs as nodes in the CS knowledge graph. Then we attempt to assign moodtags to the entities retrieved above by looking for the sentiment lexicon

SenticNet. These moodtags are referred to as sentic nodes. For example, for a word "wisdom" in word entity, its corresponding moodtag from Senticnet is "#eagerness". Then we prune the set of nodes retrieved above. We calculate the relevance score by combining the node name with the context of the input example and feeding it through a pre-trained LM. We consider the output score of the node name as the relevance score and only keep the top 200 nodes with the highest scores while discarding the rest. Afterwards, we retrieve the edges in $\mathcal{G}$ based on the prunned nodes to form the retrieved knowledge graph. In this way, we construct a CS-knowledge graph that contains both commonsense knowledge and specific sentiment knowledge at the same time. Additionally, a global node v_{glo} (the dark yellow node in Fig. 2) is added to connect to all the word nodes to capture the global information in $\mathcal{G}^{cs}$ and fully integrate it with the contextual knowledge from LM (Sect. 3.5).

Algorithm 1. Construct CS-knowledge graph

Input: Sentence sequence (s, a)
Output: knowledge graph $\mathcal{G}_{cs}$
$\mathcal{G}_{cs} \leftarrow$ create empty graph
Initialize a set of nodes N_{pruned} to store pruned nodes based on relevance scores
entities $\leftarrow$ entity_linking$(s, a, \text{ConceptNet})$
$\mathcal{G}_{cs}$.add_nodes(*entities*)
for each entity in $\mathcal{G}_{cs}$ **do**
    moodtag $\leftarrow$ senticnet_lookup(*entity*)
    if *moodtag* is not None **then**
        $\mathcal{G}_{cs}$.add_nodes(*moodtag*)
for each entity in $\mathcal{G}_{cs}$ **do**
    context $\leftarrow$ get_context(*entity*, (s, a))
    score $\leftarrow$ pre_trained_LM(*context*)
    N_{pruned}.add(entity, score)
N_{pruned}.sort_by_score()
top_200_nodes $\leftarrow N_{pruned}[: 200]$
$\mathcal{G}_{cs} \leftarrow$ empty existing entities
$\mathcal{G}_{cs}$.add_nodes(*top_200_nodes*)
for each head-entity in $\mathcal{G}_{cs}$ **do**
    for each tail-entity in $\mathcal{G}_{cs}$ **do**
        if check_relation(entity1, entity2) **then**
            $\mathcal{G}_{cs}$.add_edge(entity1, entity2)
global_node $\leftarrow$ create_global_node($\mathcal{G}_{cs}$)
$\mathcal{G}_{cs}$.add_node(*global_node*)
for each entity in $\mathcal{G}_{cs}$ **do**
    if *entity* is in (s, a) **then**
        $\mathcal{G}_{cs}$.add_edge(*global_node*, *entity*)
return $\mathcal{G}_{cs}$

3.3 CS-Knowledge Graph Initialization

Generating initial node embeddings for entities in the CS-knowledge graph is a crucial step in our model. To begin with, we convert knowledge triples in $\mathcal{G}^{cs}$ into sentences using pre-defined templates for each relation. These templates ensure that the resulting sentences preserve the semantics of the original triples. Then, we feed these sentences into a BERT-large LM, which computes embeddings for each sentence. These sentence embeddings capture the semantic meaning of the knowledge triples in $\mathcal{G}^{cs}$. Subsequently, we extract all token representations of the entity's mention spans in sentences containing that entity, mean pool over these representations, and project the resulting mean-pooled representation. Finally, we initialize the embedding of the global node $v_{glo}^{(0)}$ randomly.

3.4 Textual Pre-embedding

In the M-Layer unimodal LM module, we provide the input sequence of tokens as $\{w_{glo}, w_1, ..., w_T\}$. The representation of a token w_t in the ℓ-th layer of the model is donated as ${h_t}^{(\ell)}$. The input representation for layer ℓ=0 is computed by summing the token, segment, and positional embeddings for each token, resulting in $\{h_{glo}^{(0)}, h_1^{(0)}, ..., h_T^{(0)}\}$. For each subsequent layer ℓ+1, the input representation is updated using the following process:

$$\left\{\boldsymbol{h}_{glo}^{(\ell+1)}, \boldsymbol{h}_1^{(\ell+1)}, \ldots, \boldsymbol{h}_T^{(\ell+1)}\right\} = \text{LM-Layer}\left(\left\{\boldsymbol{h}_{glo}^{(\ell)}, \boldsymbol{h}_1^{(\ell)}, \ldots, \boldsymbol{h}_T^{(\ell)}\right\}\right) \quad \text{for } \ell = 1, \ldots, M-1 \tag{1}$$

where LM-Layer $(\cdot)$ represents a single encoder layer of the LM, and its parameters are initialized with a pre-trained model.

3.5 Fusion Module

After initializing and pre-encoding respectively, the structured CS-knowledge graph and the input text are fed into the fusion module. The fusion module draws inspiration from previous work which focused on enhancing text representations using external knowledge. However, these approaches only implemented one-way information propagation. In order to enrich the representations further, our work introduces a bidirectional fusion mechanism in the fusion module. The fusion module facilitates the interaction of global information between textual representations and KG representations, enabling the two modalities to integrate information from each other. Specifically, pre-encoded tokens are further encoded through the LM layers in the fusion module, while the initialized $\mathcal{G}^{cs}$ is fed into the GNN network for node information propagation. It is worth noting that after each LM and GNN layer, an additional fusion operation is performed on the h_{glo} and the v_{glo} to exchange global information between the two modalities. Therefore, our fusion module mainly consists of the LM layers, GNN layers, and fusion units, and we will introduce the specific operations of each.

The fusion module is specifically designed to separately encode information for both modalities. Specifically, in a N-Layer fusion module, the input textual embeddings from the M-th unimodal LM layer are further processed by additional transformer LM encoder blocks. The textual embedding of the $(M+\ell+1)$-th layer is updated using the following process:

$$\left\{\tilde{\boldsymbol{h}}_{glo}^{(M+\ell+1)}, \boldsymbol{h}_1^{(M+\ell+1)}, \ldots, \boldsymbol{h}_T^{(M+\ell+1)}\right\} = \text{LM-Layer}\left(\left\{\boldsymbol{h}_{glo}^{(M+\ell)}, \boldsymbol{h}_1^{(M+\ell)}, \ldots, \boldsymbol{h}_T^{(M+\ell)}\right\}\right) \quad \text{for } \ell = 1, \ldots, N-1 \tag{2}$$

where $\tilde{h}_{glo}$ refers to the global token that have not undergone interaction in the current layer. As $\tilde{h}_{glo}$ interacts with the nodes $\tilde{v}_{glo}$ in the knowledge graph representation and encodes the received global graph information, this allows for token representations to mix with $\mathcal{G}^{cs}$ representations in the later LM layers, which will be explained in detail later.

Meanwhile, in the fusion module, the graph representations that are initialized above are fed into GNN layers to perform information propagation over the CS-knowledge graph, aiming to fully exploit the commonsense knowledge and emotional connections between the nodes. Our GNN network utilizes the graph attention network (GAT) framework [31] to learn the graph representations. This framework uses iterative message passing to induce the learning of node representations among graph neighbors. Specifically, in the fusion module, for each layer, we update the representation of each node ${v_k}^{(\ell)}$ by

$$\left\{\tilde{\boldsymbol{v}}_{glo}^{(\ell+1)}, \boldsymbol{v}_1^{(\ell+1)}, \ldots, \boldsymbol{v}_K^{(\ell+1)}\right\} = \text{GNN}\left(\left\{\boldsymbol{v}_{glo}^{(\ell)}, \boldsymbol{v}_1^{(\ell)}, \ldots, \boldsymbol{v}_K^{(\ell)}\right\}\right) \quad \text{for } \ell = 1, \ldots, N-1 \tag{3}$$

where $\tilde{v}_{glo}$ refers to the global node that have not undergone interaction in the current layer. Importantly, since our GNN layer performs message passing on the graph, it will simultaneously utilize the representations of both the textual context and CS-knowledge graph through global node ${v_{glo}}^{(\ell)}$. Further elaboration on this will be provided later.

After updating token embeddings and node embeddings using a LM layer and a GNN layer respectively, we use a fusion unit (FU) to enable the two modalities to exchange information through h_{glo} that captures text global information and v_{glo} that captures knowledge graph global information. We concatenate the unmixed embeddings of $\tilde{h}_{glo}^{(\ell)}$ and $\tilde{v}_{glo}^{(\ell)}$, apply a mixing operation (FU) to the joint representation, and then separate the fused embeddings into $h_{glo}^{(\ell)}$ and $e_{glo}^{(\ell)}$.

$$\left[\boldsymbol{h}_{glo}^{(\ell)}; \boldsymbol{v}_{glo}^{(\ell)}\right] = \text{FU}\left(\left[\tilde{\boldsymbol{h}}_{glo}^{(\ell)}; \tilde{\boldsymbol{v}}_{glo}^{(\ell)}\right]\right) \tag{4}$$

where FU adopts a two-layer MLP operation. Of course, besides the global token h_{glo} and the global nodes v_{glo}, other tokens and nodes do not directly participate in fusion. Instead, in the next layer when they are separately encoded, they obtain information from each other through their respective h_{glo} and v_{glo}. For the textual representations, the fused h_{glo} and the rest of them are sent to the next layer of the LM, where they undergo the next round of modal propagation (i.e., Eqs. 2), which allows for the fusion with commonsense knowledge and specific sentiment knowledge from the knowledge graph. For the graph representations, the fused v_{glo} and the rest of the nodes are fed into the next GNN layer, where its propagation mechanism enables each node to integrate contextual information from the LM (i.e., Eqs. 3). As a result, LM and KG form a bidirectional information propagation mechanism that reinforces each other.

3.6 Inference and Learning

Given a sentence s and a textualized label a, we leverage the information from both the context and the external knowledge to calculate the probability of it being the correct polarity $p(a \mid q) \propto \exp(\mathrm{MLP}(\boldsymbol{h}_{glo}^{(M+N)}, \boldsymbol{v}_{glo}^{(N)}, \boldsymbol{g}))$, where $\boldsymbol{g}$ denotes the pooling of $\{\boldsymbol{v}_k^{(N)} \mid v_k \in \mathcal{G}^{cs}\}$. The cross entropy loss is utilized to optimize the entire model in an end-to-end manner.

4 Experiment

4.1 Datasets and External Knowledge

To verify the effectiveness of KSA, we examine the model on four popular sentence-level sentiment analysis datasets. Table 1 summarizes the statistics of the datasets used in the experiments, which contain the amount of training/validation/test sets and the number of classes. The datasets include Stanford Sentiment Treebank (SST2 and SST5) [32], IMDB [33] and Movie Review (MR) [34]. For MR and IMDB, there is no validation set in the original dataset, so we randomly select a subset from the training set for validation. The accuracy of the model is used as the metric to assess its performance.

Table 1. Statistics of datasets used in our experiments.

Dataset	Amount(Train/Dev/Test)	classes
SST-2	67,349/872/1,821	2
IMDB	22,500/2,500/25,000	2
MR	8,534/1,078/1,050	2
SST-5	8,544/1,101/2,210	5

Given that our approach is adaptable to all pre-trained models of BERT-style, we opt to utilize RoBERTa [35] as the foundational framework to construct Transformer blocks in this paper. And we utilize two external sources of knowledge in our study: ConceptNet [11], a general-domain knowledge graph with 799,273 nodes and 2,487,810 edges, and moodtags from the sentiment lexicon SenticNet [13] which provides us with specific sentiment knowledge. After receiving each textual input (sentence s and textual label a), we follow the pre-processing step outlined in §3.2 to construct the CS-knowledge graph from $\mathcal{G}$, with a hop size of k = 2. We subsequently prune the CS-knowledge graph, retaining only the top 200 nodes based on their node relevance scores.

4.2 Comparison Methods

In order to demonstrate the effectiveness of the proposed method for sentence-level sentiment analysis, we compare our model with both general pre-trained models and knowledge-aware pre-trained models. For general pre-trained models, we use vanilla BERT [36], XLNet [6] and RoBERTa [35] as our baselines, which are knowledge-agnostic. For knowledge-aware pre-trained models, we adopt some methods focusing on leveraging external knowledge as baselines, i.e., SentiX [8], SentiLARE [28], SentiBERT [37], KESA [38] and SentiWSP [39]. The key difference between our model and these baseline methods is that they only incorporate one type of knowledge, while our model integrates both structured knowledge and sentiment knowledge. Additionally, they do not integrate the representations of both modalities across multiple fusion layers, thus not enabling the representations of the two modalities to affect each other (Sect. 3.5).

4.3 Implementation Details

We implement our model utilize RoBERTa as the foundational framework to construct Transformer blocks and graph attention network(GAT) as the foundational framework to construct GNN blocks. According to the experiments, the GNN module for the IMDB dataset and other datasets was set to 5 layers and 7 layers, respectively, with a dimensionality of 200. And we applied a dropout rate of 0.3 to each layer of GNN module. We train the model with the RAdam optimizer using two GPUs (NVIDIA A30). The batch size is set to 64 and 128 for IMDB and other datasets. The learning rate for the LM module is set to 1e-5 for SST2 and MR, and 5e-5 for SST5 and IMDb. And the learning rate for the GNN module is set to 1e-3. To ensure coverage of over 90% of the samples, we set the input sequence length to 512 for the IMDb dataset and 128 for other datasets. The experimental results were reported as the mean values averaged over 5 runs.

In addition, capturing multi-hop semantic relationships is one of the crucial aspects of our model's overall performance. Therefore, we conducted experimental research to investigate the impact of the number of hops. We evaluated the model's performance by varying the number of hops from 1 to 4. We found that the optimal results were obtained when using 2 or 3 hops. This could be attributed to the fact that graph neural networks with intermediate hops can effectively capture the semantic relationships between words while preventing the introduction of unnecessary noise.

4.4 Overall Results

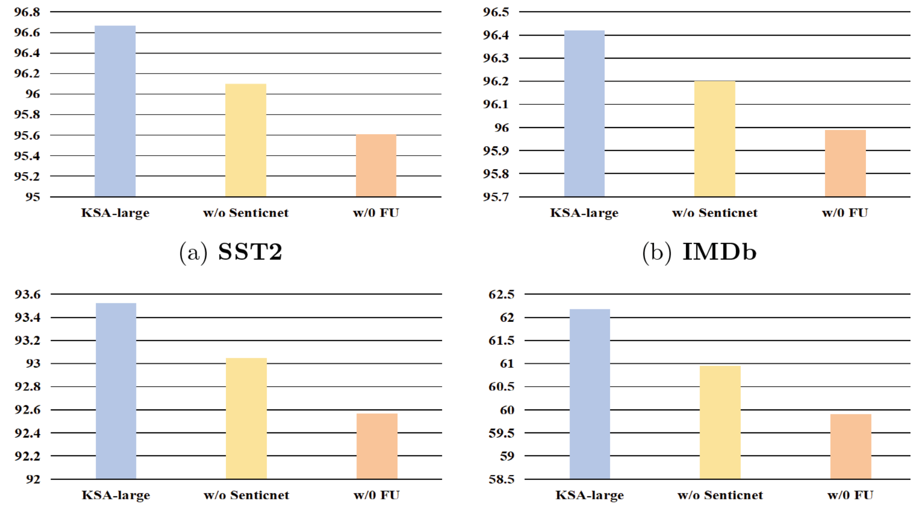
Table 2 shows that our results improved across four datasets. We find that our base model's test performance improves by (1.11%, 0.81%, 2.64%, 2.32%) on (SST2, IMDB, MR, SST5) over LMs which are knowledge-agnostic. Furthermore, on most of the datasets, KSA has also shown improvements over the best previous model equipped with external knowledge. These results demonstrate the effectiveness of incorporating two external knowledge sources, i.e., structured KGs and sentiment lexicons. Meanwhile, it also verifies the effectiveness of the proposed global information fusion approach between LM representations and graph representations. Additionally, our proposed KSA model is highly sensitive to parameter size. Compared to the basic version, significant improvements can be achieved by increasing the model layers. Moreover, KSA-large can achieve competitive results on dataset leaderboards compared to systems with similar or even larger parameter sizes, particularly demonstrating state-of-the-art performance on SST5.

Table 2. Overall accuracy on sentence-level sentiment classification benchmarks(%). The note * means our model significantly outperforms the reproduced baselines based on t-test ($p < 0.01$).

Model	SST-2	IMDB	MR	SST-5
BERT	91.38	93.45	86.62	53.52
XLNet	92.75	94.99	88.83	54.95
RoBERTa	94.00	95.13	89.84	57.09
SentiX	92.23	94.62	86.81	55.59
SentiLARE	94.58	95.73	90.50	58.54
SentiBERT	94.72	94.04	88.59	56.87
KESA	94.96	95.83	91.26	59.26
SentiWSP	–	96.26	92.41	59.32
KSA (ours)	95.11	95.94	92.48	59.41
KSA-large (ours)	**96.67***	**96.42***	**93.52***	**62.17***

4.5 Ablation Results

To investigate the impact of each component on our KSA model, we perform the ablation test by separately removing the sentiment lexicon knowledge within the CS-knowledge graph (denoted as w/o Senticnet), and the modality interaction fusion unit (denoted as w/o FU). Specifically, for the w/o Senticnet model, the external knowledge is only obtained from the ConceptNet graph, and is combined with the text representations to perform sentiment polarity prediction,



(a) **SST2** (b) **IMDb**

(c) **MR** (d) **SST5**

Fig. 3. Ablation study of our model components.

with a single source of knowledge. For the w/o FU model, we remove the connection of the LM to the GNN, there is not the global token h_{glo} and the global node v_{glo}, and the two modalities no longer exchange information. The external knowledge representations was encoded by GNN and combined with the LM representations for sentiment polarity analysis at the end, rather than performing multi-layers fusion. The ablation results are shown in Fig. 3. We observe that both the Senticnet and FU make great improvements to our KSA method. On the one hand, compared to simply fusing external commonsense knowledge, incorporating specific sentiment knowledge to construct heterogeneous graphs can help KSA better capture multi-hop semantic correlations between words and sentiment labels, significantly improving the model performance. On the other hand, FU helps to fully integrate external knowledge with the LM representations, promoting the learning of consistent representations between the LM and GNN, which allows the model representations to better adapt to sentiment analysis.

Based on our empirical observations, the information exchange between the LM-encoded text representations and the GNN-encoded graph representations in the fusion module is one of the most crucial parts for the overall performance of KSA. Therefore, we also studied the impact of the number of fusion module layers for this process in KSA. Specifically, we evaluated the performance of KSA by increasing the number of fusion module layers from 3 to 8 with a step size of 1. We find that the best results are achieved when N = 5 for SST2, SST5 and MR. For IMDB, the best performance was achieved when N = 7. This may be because such fusion layers strike a good balance between the complexity and efficiency of the model while achieving effective information interaction between different modalities.

4.6 Advantages and Drawbacks of Combining KGs and LMs

Combining KGs with LMs brings significant advantages, allowing for comprehensive knowledge representation. Specifically, by incorporating the structured information from KGs into LMs, we can enhance their understanding and reasoning abilities, resulting in more accurate and contextually appropriate responses. As shown in Table 2, our KSA model demonstrates significant improvement compared to LMs without integrated KGs. However, along with these benefits, it also presents challenges related to data quality and increased complexity. In terms of data quality, ensuring the accuracy and reliability of the information stored in KGs becomes paramount. Since KGs are built upon the amalgamation of data from various sources, there is a high likelihood of encountering conflicting or outdated information. This can lead to potential inaccuracies in the knowledge presented by the KG, which can then be propagated to the LMs during the fusion process. Another challenge lies in the increased complexity introduced by the integration of KGs with LMs. LMs and KGs have different ways of representing knowledge, where LMs understand natural language by learning language patterns in texts, while KGs express structured knowledge using entity-relation-entity triples. During the fusion process, it is necessary to establish an effective bridge to transform the knowledge in KGs into a format suitable for processing in LMs, or to map the language representations in LMs to entities and relations in KGs. This introduces additional complexity to the integration process. Therefore, when adopting this approach, it is essential to consider these factors. In this study, we used ConceptNet, which offers extensive and accurate knowledge, and integrated the two modalities in an appropriate manner. Although it may introduce some complexity, the overall impact is positive.

5 Conclusion

In this paper, we propose KSA, an end-to-end sentiment analysis model that leverages LMs and external knowledge. Our key innovations include (i) Constructing CS knowledge graph, which incorporates both commonsense knowledge from ConceptNet and sentiment knowledge from SenticNet, and (ii) Deep fusion of LM representations and graph representations, where we utilize fusion units to enable comprehensive interaction between the two modalities, bridging the gap between them for improving sentiment polarity prediction. Our experiments demonstrate the usefulness of both external knowledge sources in sentiment analysis task, as well as the significance of the fusion approach used in KSA.

Supplemental Material Statement: Source code and other supplemental materials for KSA are available from https://github.com/lll111000/KSA.

Acknowledgments. This work was supported by the National Science Foundation of China (No. U22B2019, 62276029), Beijing Academy of Artificial Intelligence (BAAI) and CCF-Zhipu.AI Large Model Fund (No. 202217).

References

1. Bing, L.: Sentiment analysis and opinion mining (synthesis lectures on human language technologies). University of Illinois, Chicago, IL, USA (2012)
2. Zhang, L., Wang, S., Liu, B.: Deep learning for sentiment analysis: a survey. Wiley Interdisc. Rev. Data Mining Knowl. Dis. **8**(4), e1253 (2018)
3. Peters, M., Neumann, M., Iyyer, M., et al.: deep contextualized word representations. In: The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 2227–2237 (2018)
4. Radford, A., Narasimhan, K., Salimans, T., et al.: Improving language understanding by generative pre-training (2018)
5. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. In: The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 4171–4186 (2019)
6. Yang, Z., Dai, Z., Yang, Y., et al.: Xlnet: generalized autoregressive pretraining for language understanding. Adv. Neural. Inf. Process. Syst. **32**, 5754–5764 (2019)
7. McCoy, R.T., Pavlick, E., Linzen, T.: Right for the wrong reasons: diagnosing syntactic heuristics in natural language inference. Annual Meeting Assoc. Comput. Ling. **57**, 3428–3448 (2019)
8. Zhou, J., Tian, J., Wang, R., et al.: Sentix: a sentiment-aware pre-trained model for cross-domain sentiment analysis. In: The 28th international conference on computational linguistics, pp. 568–579 (2020)
9. Zhang, B., Yang, M., Li, X., et al.: Enhancing cross-target stance detection with transferable semantic-emotion knowledge. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 3188–3197 (2020)
10. Ghosal, D., Hazarika, D., Roy, A., et al.: Kingdom: knowledge-guided domain adaptation for sentiment analysis. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 3198–3210 (2020)
11. Speer, R., Chin, J., Havasi, C.: Conceptnet 5.5: an open multilingual graph of general knowledge. In: The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, vol. 31(1) (2017)
12. Bollacker, K., Evans, C., Paritosh, P., et al.: Freebase: a collaboratively created graph database for structuring human knowledge. In: The Association for Computing Machinery Conference on Management of data, pp. 1247–1250 (2008)
13. Cambria, E., Poria, S., Hazarika. D, et al.: SenticNet 5: discovering conceptual primitives for sentiment analysis by means of context embeddings. In: The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, vol. 32(1) (2018)
14. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining. In: The 10th International Conference on Language Resources and Evaluation, pp. 2200–2204 (2010)
15. Gururangan, S., Marasovi, A., Swayamdipta, S, et al.: Don't stop pretraining: adapt language models to domains and tasks. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 8342–8360 (2020)

16. Gu, Y., Zhang, Z., Wang, X., et al.: Train no evil: Selective masking for task-guided pre-training. In: The Conference on Empirical Methods in Natural Language Processing, pp. 6966–6974 (2020)
17. Tian, Y., Chen, G., Song, Y.: Enhancing aspect-level sentiment analysis with word dependencies. In: The 16th Conference of the European Chapter of the Association for Computational Linguistics, pp. 3726–3739 (2021)
18. Li, C., Gao, F., Bu, J., et al.: Sentiprompt: sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. In: Reproducibility Challenge (2022)
19. Sun, Y., Wang, S., Feng, S., et al.: Ernie 3.0: Large-scale knowledge enhanced pre-training for language understanding and generation. In: Reproducibility Challenge (2022)
20. Liu, W., Zhou, P., Zhao, Z., et al.: K-bert: enabling language representation with knowledge graph. In: The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence, vol. 34(03), pp. 2901–2908 (2020)
21. Levine, Y., Lenz, B., Dagan, O., et al.: SenseBERT: driving some sense into BERT. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 4656–4667 (2019)
22. Peters, M.E., Neumann, M., Logan, I.V.R. L., et al.: Knowledge enhanced contextual word representations. In: The 9th Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing, pp. 43–54 (2019)
23. Wang, R., Tang, D., Duan, N., et al.: K-adapter: infusing knowledge into pre-trained models with adapters. In: The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pp. 1405–1418 (2021)
24. Zhong, Q., Ding, L., Liu, J., et al.: Knowledge graph augmented network towards multiview representation learning for aspect-based sentiment analysis. IEEE Trans. Knowl. Data Eng. (2023)
25. Yan, X., Jian, F., Sun, B.: SAKG-BERT: enabling language representation with knowledge graphs for Chinese sentiment analysis. IEEE Access **9**, 101695–101701 (2021)
26. Liao, J., Wang, M., Chen, X., et al.: Dynamic commonsense knowledge fused method for Chinese implicit sentiment analysis. Inform. Process. Manag. **59**(3), 102934 (2022)
27. Tian, H., Gao, C., Xiao, X., et al.: SKEP: sentiment knowledge enhanced pre-training for sentiment analysis. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 4067–4076 (2020)
28. Ke, P., Ji, H., Liu, S., et al.: SentiLARE: sentiment-aware language representation learning with linguistic knowledge. In: The Conference on Empirical Methods in Natural Language Processing, pp. 6975–6988 (2020)
29. Tian, Y., Chen, G., Song. Y.: Enhancing aspect-level sentiment analysis with word dependencies. In: The 16th Conference of the European Chapter of the Association for Computational Linguistics, pp. 3726–3739 (2021)
30. Li, C., Gao, F., Bu, J., et al.: Sentiprompt: sentiment knowledge enhanced prompt-tuning for aspect-based sentiment analysis. arXiv preprint arXiv:2109.08306 (2021)
31. Veličković, P., Cucurull, G., Casanova, A., et al.: Graph attention networks. In: International Conference on Learning Representations (2018)
32. Socher, R., Perelygin, A., Wu, J., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: The Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)

33. Maas, A., Daly, R.E., Pham, P.T., et al.: Learning word vectors for sentiment analysis. In: The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp. 142–150 (2011)
34. Pang, B., Lee, L.: Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: The 43rd Annual Meeting on Association for Computational Linguistics, pp. 115–124 (2005)
35. Liu, Y., Ott, M., Goyal, N., et al.: Roberta: a robustly optimized bert pretraining approach. In: The 20th Chinese National Conference on Computational Linguistics, pp. 1218–1227 (2019)
36. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. The Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 4171–4186 (2019)
37. Yin, D., Meng, T., Chang, K.W.: Sentibert: a transferable transformer-based architecture for compositional sentiment semantics. In: The 58th Annual Meeting of the Association for Computational Linguistics, pp. 3695–3706 (2020)
38. Zhao, Q., Ma, S., Ren, S.: KESA: a knowledge enhanced approach for sentiment analysis. In: The 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and The 12th International Joint Conference on Natural Language Processing, vol. 1, pp. 766–776 (2022)
39. Fan, S., Lin, C., Li, H., et al.: Sentiment-aware word and sentence level pre-training for sentiment analysis. In: The Conference on Empirical Methods in Natural Language Processing, pp. 4984–4994 (2022)