Detecting Spoilers in Movie Reviews with External Movie Knowledge and User Networks

Anonymous ACL submission

Abstract

Online movie review platforms are providing crowdsourced feedback for the film industry and the general public, while spoiler reviews greatly compromise user experience. Although preliminary research efforts were made to automatically identify spoilers, they merely focus on the review content itself, while robust spoiler detection requires putting the review into the context of facts and knowledge regarding movies, user behavior on film review platforms, and more. In light of these challenges, we first curate a large-scale network-based spoiler detection dataset **LCS** and a comprehensive and up-to-date movie knowledge base **UKM**. We then propose **MVSD**, a novel spoiler detection model that takes into account the external knowledge about movies and user activities on movie review platforms. Specifically, **MVSD** constructs three interconnecting heterogeneous information networks to model diverse data sources and their multi-view attributes, while we design and employ a novel heterogeneous graph neural network architecture for spoiler detection as node-level classification. Extensive experiments demonstrate that **MVSD** advances the state-of-the-art on two spoiler detection datasets, while the introduction of external knowledge and user interactions help ground robust spoiler detection.

1 Introduction

Movie review websites such as IMDB and Rotten Tomato have become popular avenues for movie commentary, discussion, and recommendation (Cao et al., 2019). Among user-generated movie reviews, some of them contain *spoilers*, which reveal major plot twists and thus negatively affect people's enjoyment (Loewenstein, 1994). As a result, automatic spoiler detection has become an important task to safeguard users from unwanted exposure to potential spoilers.

Existing spoiler detection models mostly focus on the textual content of the movie review. Chang et al. (2018) propose the first automatic spoiler detection approach by jointly encoding the review text and the movie genre. Wan et al. (2019) extend the hierarchical attention network with item (i.e., the subject to the review) information and introduce user bias and item bias. Chang et al. (2021) propose a relation-aware attention mechanism to incorporate the dependency relations between context words in movie reviews. Combined with several open-source datasets (Boyd-Graber et al., 2013; Wan et al., 2019), these works have made important progress toward curbing the negative impact of movie spoilers.

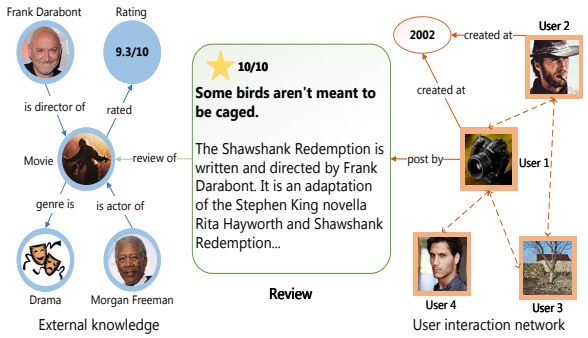


Figure 1: An example of a movie review and its context. The review mentions Tim Robbins and Morgan Freeman, which are the names of the actors. Guided by external movie knowledge, the names can be recognized as the roles in the movie. Moreover, by incorporating user networks, it is discovered that User 1 likes to post spoilers on some specific genres of movies such as drama and comedy. Thus the review is more likely to be a spoiler.

However, robust spoiler detection requires more than just the textual content of movie reviews, and we argue that two additional information sources are among the most helpful for reliable and well-grounded spoiler detection. Firstly, **external knowledge** of films and movies (e.g. director, cast members, genre, plot summary, etc.) are essential in putting the review into the movie context. Without knowing what the movie is all about,

Table 1: Statistics of LCS and existing dataset Kaggle.

KB	# Review	# Cast	# Metadata	Year
KAGGLE	573,913	0	5	2018
LCS (Ours)	1,860,715	494,221	15	2022

Table 2: Statistics of UKM and existing movie KBs.

KB	# Entity	# Relation	# Triple	Year
MOVIELENS	14,708	20	434,189	2019
RIPPLENET	182,011	12	1,241,995	2018
UKM (Ours)	641,585	15	1,936,710	2022

it is hard, if not impossible, to accurately assess whether the reviews give away major plot points or surprises and thus contain spoilers. Secondly, **user activities** of online movie review platforms help incorporate the user- and movie-based spoiler biases. For example, certain users might be more inclined to share spoilers and different movie genres are disproportionally suffering from spoiler reviews while existing approaches simply assume the uniformity of spoiler distribution. As a result, robust spoiler detection should be guided by external film knowledge and user interactions on movie review platforms, putting the review content into context and promoting reliable predictions.

In light of these challenges, this work greatly advances spoiler detection research through both resource curation and method innovation. We first propose a large-scale spoiler detection dataset **LCS** and an extensive movie knowledge base (KB) **UKM**. **LCS** is 114 times larger than existing datasets (Boyd-Graber et al., 2013) and is the first to provide user interactions on movie review platforms, while **UKM** presents an up-to-date movie KB with entries of modern movies compared to existing resources (Misra, 2019). In addition to resource contributions, we propose **MVSD**, a graph-based spoiler detection framework that incorporates external knowledge and user interaction networks. Specifically, **MVSD** constructs heterogeneous information networks (HINs) to jointly model diverse information sources and their multi-view features while proposing a novel heterogeneous graph neural network (GNN) architecture for robust spoiler detection.

We compare **MVSD** against three types of baseline methods on two spoiler detection datasets. Extensive experiments demonstrate that **MVSD** significantly outperforms all baseline models by at least 2.01 and 3.22 in F1-score on the Kaggle (Misra, 2019) and LCS dataset (ours). Further analyses demonstrate that **MVSD** empowers external movie KBs and user networks on movie review platforms to produce accurate, reliable, and well-grounded spoiler predictions.

2 Resource Curation

We first curate a large-scale spoiler detection dataset **LCS** based on IMDB, providing rich information such as review text, movie metadata, user activities, and more. Motivated by the success of external knowledge in related tasks (Hu et al., 2021; Yao et al., 2021; Li and Xiong, 2022), we construct a comprehensive movie knowledge base **UKM** with important movie information and up-to-date entries.

2.1 The LCS Dataset

We first collect the user id of 259,705 users from a user list presented in the Kaggle dataset (Misra, 2019). We then retrieve the most recent 300 movie reviews by each user and collect the information of users, movies, and cast members based on the IMDB website. Since IMDB allows users to self-report whether its review contains spoilers, we adopt these labels provided by IMDB as annotations. We provide the comparison of our dataset to the Kaggle dataset in Table 1. As illustrated in Table 1, the LCS dataset has a much larger scale, more up-to-date information, and more comprehensive data. [1]

2.2 The UKM Knowledge Base

Based on the LCS dataset, we then curate **UKM**, a comprehensive knowledge base of movie knowledge. We first assign each movie in the LCS dataset as an entity in the KB. We then collect all cast members and directors of these movies, de-duplicating them, representing each individual as an entity, and connecting movie entities with cast members based on their roles in the movie. After that, we further represent years, genres, and ratings as entities, connecting them to movie and cast member entities according to the information in the dataset.

We compare **UKM** against two existing movie knowledge bases (RippleNet (Wang et al., 2018) and MoviesLen-1m (Cao et al., 2019)) and present the results in Table 2, which demonstrates that

[1] Details and statistics of the LCS datasets are presented in Appendix D.

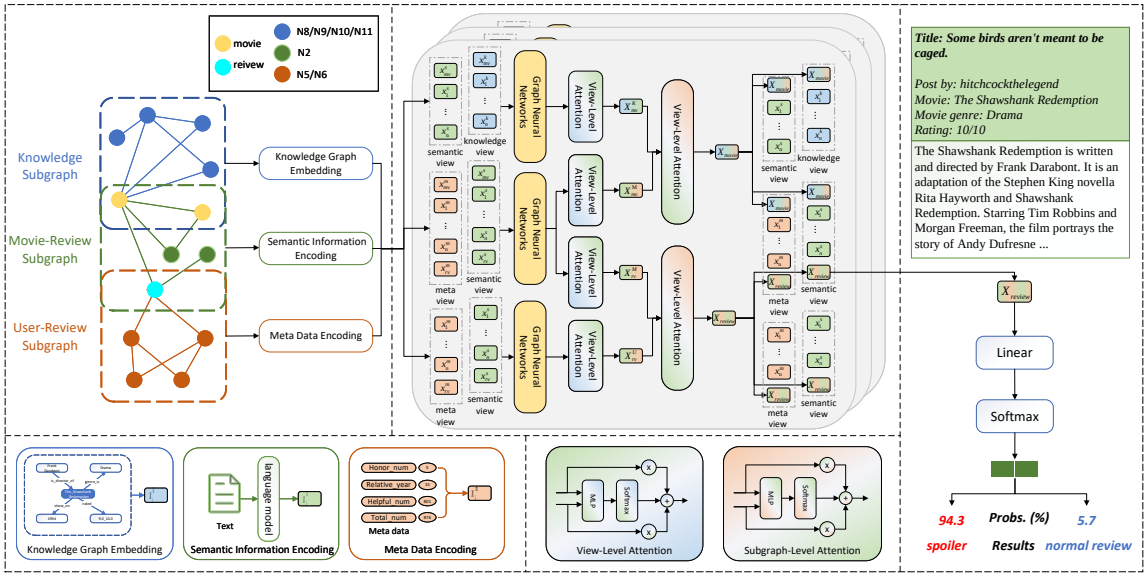


Figure 2: The architecture of MVSD, which incorporates external knowledge and social network interactions, leverages multi-view data and facilitates interaction between multi-view data.

UKM presents the largest and most up-to-date collection of movie and film knowledge to the best of our knowledge. **UKM** has great potential for numerous related tasks such as spoiler detection, movie recommender systems, and more.

3 Methodology

We propose MVSD, a **M**ulti-**V**iew **S**poiler **D**etection framework. To leverage external movie knowledge and user activities that are essential in robust spoiler detection, MVSD constructs heterogeneous information networks to jointly represent diverse information sources. Specifically, we build three subgraphs: movie-review subgraph, user-review subgraph, and knowledge subgraph, each modeling one aspect of the spoiler detection process. MVSD first separately encodes the multi-view features of these subgraphs through heterogeneous GNNs, then fuses the learned representations of the three subgraphs through subgraph interaction. MVSD conducts spoiler detection with a node classification setting based on the learned representations of review nodes.

3.1 Heterogeneous Graph Construction

Graphs and graph neural networks have become increasingly involved in NLP tasks such as misinformation detection (Hu et al., 2021) and question answering (Yu et al., 2022). In this paper, we construct heterogeneous graphs to jointly model textual content, metadata, and external knowledge in spoiler detection. Specifically, we first construct the three subgraphs modeling different information sources: movie-review subgraph $\mathcal{G}^K = \{\mathcal{V}^K, \mathcal{E}^K\}$, user-review subgraph $\mathcal{G}^M = \{\mathcal{V}^M, \mathcal{E}^M\}$, and knowledge subgraph $\mathcal{G}^U = \{\mathcal{V}^U, \mathcal{E}^U\}$. We mainly explain the compositions of the graph in the following and elaborate on the details about all the nodes and relations in Appendix C.

Movie-Review Subgraph The movie-review subgraph models the bipartite relation between movies and user reviews. We first define the nodes denoted as $\mathcal{V}^M$, which include *movie* nodes, *rating* nodes, and *review* nodes.

User-Review Subgraph The user-review subgraph is responsible for modeling the heterogeneity of user behavior on movie review platforms. The nodes in this subgraph, denoted as $\mathcal{V}^U$, include *review* nodes, *user* nodes, and *year* nodes.

Knowledge Subgraph The knowledge subgraph is responsible for incorporating movie knowledge in external KBs. Nodes in this subgraph, denoted as $\mathcal{V}^K$, include *movie* nodes, *genre* nodes, *cast* nodes, *year* nodes, and *rating* nodes.

Note that the most vital nodes, movie nodes and review nodes, both appear in two subgraphs. These shared nodes then serve as bridges for information exchange across subgraphs, which is enabled by the MVSD model architecture in Section 3.3.

3.2 Multi-View Feature Extraction

The entities in the heterogeneous information graph have diverse data sources and multi-view attributes. In order to model the rich information of these entities, we propose a taxonomy of the views, dividing them into three categories.

Semantic View The semantic view reflects the semantics contained in the text. We pass movie review documents, movie plot descriptions, user bio, and cast bio to pre-trained RoBERTa, averaging all tokens, and produce node embeddings v^s as the semantic view.

Meta View The meta view is the numerical and categorical feature. We utilize metadata of user accounts, movie reviews, movies, and cast, and calculate the z-score as node embeddings v^m to get the meta view. Details about metadata can be found in Appendix D.2.

Knowledge View The knowledge view captures the external knowledge of movies. Following previous works (Hu et al., 2021; Zhang et al., 2022), we use TransE (Bordes et al., 2013) to train KG embeddings for the **UKM** knowledge base and use these embeddings as node features v^k for the external knowledge view.

Based on these definitions, each subgraph has two feature views, thus nodes in each subgraph have two sets of feature vectors. Specifically, the knowledge subgraph $\mathcal{G}^K$ has the external knowledge view and the semantic view, the movie-review subgraph $\mathcal{G}^M$ and the user-review subgraph $\mathcal{G}^U$ has the meta view and the semantic view. We then employ one MLP layer for each feature view to encode the extracted features and obtain the initial node features x_i^s, x_i^m, x_i^k for the semantic, meta, and knowledge view.

3.3 MVSD Layer

After obtaining the three subgraphs and their initial node features under the textual, meta, and knowledge views, we employ MVSD layers to conduct representation learning and spoiler detection. Specifically, an MVSD layer first separately encodes the three subgraphs, then adopts hierarchical attention to enable feature interaction and the information exchange across various subgraphs.

Subgraph Modeling We first model each subgraph independently, fusing the two view features for each node. We then fuse node embeddings from different subgraphs to facilitate interaction between the three subgraphs. For simplicity, we adopt relational graph convolutional networks (R-GCN) (Schlichtkrull et al., 2018) to encode each subgraph. For the l-th layer of R-GCN, the message passing is as follows:

$$\mathbf{x}_i^{(l+1)} = \Theta_{self} \cdot \mathbf{x}_i^{(l)} + \sum_{r \in \mathcal{R}} \sum_{j \in \mathcal{N}_r(i)} \frac{1}{|\mathcal{N}_r(i)|} \Theta_r \cdot \mathbf{x}_j^{(l)}$$

where Θ_{self} is the projection matrix for the node itself while Θ_r is the projection matrix for the neighbor of relation r. By applying R-GCN, nodes in subgraph $\mathcal{G}^K$ get features from the knowledge and semantic view, denoting as $\mathbf{x}_k^K$ and $\mathbf{x}_s^K$, respectively. Nodes in subgraph $\mathcal{G}^M$ get features from the semantic and meta view, denoting as $\mathbf{x}_s^M, \mathbf{x}_m^M$, while nodes in subgraph $\mathcal{G}^U$ get the same views of feature, denoting as $\mathbf{x}_s^U, \mathbf{x}_m^U$.

Aggregation and Interaction Given the representation of nodes from different feature views, we adopt hierarchical attention layers to aggregate and mix the representations learned from different subgraphs. Our hierarchical attention contains two parts: view-level attention and subgraph-level attention. Considering movie node and review node are shared nodes of subgraphs and are of the most significance, we utilize these two kinds of nodes to implement our hierarchical attention.

We first conduct view-level attention to aggregate the multi-view information for each type of node. For each node in a specific subgraph, it has embeddings learned from two types of feature views. We first adopt our proposed view-level attention to fuse the information learned from different views for each node. We learn a weight for each view of features in a specific subgraph. Specifically, the learned weight for each view in a specific subgraph $\mathcal{G}$, $(\alpha_{v_1}^{\mathcal{G}}, \alpha_{v_2}^{\mathcal{G}})$ can be formulated as

$$(\alpha_{v_1}^{\mathcal{G}}, \alpha_{v_2}^{\mathcal{G}}) = \mathrm{attn}_v(\mathbf{X}_{v_1}^{\mathcal{G}}, \mathbf{X}_{v_2}^{\mathcal{G}}),$$

where attn_v denotes the layer that implements the view-level attention, and $\mathbf{X}_{v_i}^{\mathcal{G}}$ is the node embeddings from view v_i in subgraph $\mathcal{G}$. To learn the importance of each view, we first transform view-specific embedding through a fully connected layer, then we calculate the similarity between transformed embedding and a view-level attention vector $\mathbf{q}_{\mathcal{G}}$. We then take the average importance of all the view-specific node embedding as the importance of each view. The importance of each view,

denoted as w_{v_i}, can be formulated as:

$$w_{v_i} = \frac{1}{|\mathcal{V}_\mathcal{G}|} \sum_{j \in \mathcal{V}_\mathcal{G}} \mathbf{q}_\mathcal{G}^{\mathrm{T}} \cdot \tanh(\mathbf{W} \cdot \mathbf{x}_{v_i j}^{\mathcal{G}} + \mathbf{b}),$$

where $\mathbf{q}_\mathcal{G}$ is the view-level attention vector for each view of feature, $\mathcal{V}_\mathcal{G}$ is the nodes of subgraph $\mathcal{G}$, and $\mathbf{x}_{v_i j}^{\mathcal{G}}$ is the embedding of node j in subgraph $\mathcal{G}$ from view v_i. Then the weight of each view in subgraph $\mathcal{G}$ can be calculated by

$$\alpha_{v_i} = \frac{\exp(w_{v_i})}{\exp(w_{v_1}) + \exp(w_{v_2})}.$$

It reflects the importance of each view in our spoiler detection task. Then the fused embeddings of different views can be shown as:

$$\mathbf{X}^{\mathcal{G}} = \alpha_{v_1} \cdot \mathbf{X}_{v_1}^{\mathcal{G}} + \alpha_{v_2} \cdot \mathbf{X}_{v_2}^{\mathcal{G}},$$

Thus we get the subgraph-specific node embedding, denoted as $\mathbf{X}^K, \mathbf{X}^M, \mathbf{X}^U$.

We then conduct subgraph-level attention to facilitate the flow of information between the three information sources. Generally, nodes in different subgraphs only contain information from one subgraph. To learn a more comprehensive representation and facilitate the flow of information between subgraphs, we enable the information exchange across various subgraphs using the movie nodes and the review nodes, both appearing in two subgraphs, as the information exchange ports. Specifically, we propose a novel subgraph-level attention to automatically learn the weight of each subgraph and fuse the information learned for different subgraphs. To be specific, the learned weight of each subgraph $(\boldsymbol{\beta}_K, \boldsymbol{\beta}_M, \boldsymbol{\beta}_U)$ can be computed as:

$$(\boldsymbol{\beta}_K, \boldsymbol{\beta}_M, \boldsymbol{\beta}_U) = \mathrm{attn}_g(\mathbf{X}^K, \mathbf{X}^M, \mathbf{X}^U),$$

where attn_g denotes the subgraph-level attention layer. To learn the importance of each subgraph, we transform subgraph-specific embedding through a feedforward layer and then calculate the similarity between transformed embedding and a subgraph-level attention vector $\mathbf{q}$. Furthermore, we take the average importance of all the subgraph-specific node embedding as the importance of each subgraph. Taking $\mathcal{G}^K$ and $\mathcal{G}^M$ as an example, the shared nodes of these two subgraphs are movie nodes. The importance of each subgraph, denoted as w^K, w^M, can be formulated as:

$$w^V = \frac{1}{|\mathcal{V}_{mv}^V|} \sum_{j \in \mathcal{V}_{mv}^V} \mathbf{q}^{\mathrm{T}} \cdot \tanh(\mathbf{W} \cdot \mathbf{x}_j^V + \mathbf{b})$$

where $V \in \{K, M\}$, $\mathbf{q}$ is the subgraph-level attention vector for each subgraph. Then the weight of each subgraph can be shown as:

$$\beta^K = \frac{\exp(w^K)}{\exp(w^K)+\exp(w^M)},\ \beta^M = \frac{\exp(w^M)}{\exp(w^K)+\exp(w^M)}$$

After obtaining the weight, the subgraph-specific embedding can be fused, formulated as:

$$\mathbf{X}_{mv} = \beta^K \cdot \mathbf{X}_{mv}^K + \beta^M \cdot \mathbf{X}_{mv}^M$$

Similarly, for review nodes, we can get the fused representation $\mathbf{X}_{rv}$. Our proposed subgraph-level attention enables the information to flow across different views and subgraphs.

3.4 Overall Interaction

One layer of our proposed MVSD layer, however, cannot enable the information interaction between all information sources (e.g. the user-review subgraph and the knowledge subgraph). In order to further facilitate the interaction of the information provided by each view in each subgraph, we employ ℓ MVSD layers for node representation learning. The representation of movie nodes and review nodes is updated after each layer, incorporating information provided by different views and neighboring subgraphs. This process can be formulated as follows:

$$\mathbf{X}^{(i)} = \mathrm{MVSD}(\mathbf{X}^{(i-1)}),$$

where

$$\mathbf{X}^{(i)} = [\mathbf{X}_k^{\mathcal{G}^\mathcal{K}(i)}, \mathbf{X}_s^{\mathcal{G}^\mathcal{K}(i)}, \mathbf{X}_m^{\mathcal{G}^\mathcal{M}(i)}, \mathbf{X}_s^{\mathcal{G}^\mathcal{M}(i)}, \mathbf{X}_m^{\mathcal{G}^\mathcal{U}(i)}, \mathbf{X}_s^{\mathcal{G}^\mathcal{U}(i)}]$$

We use $\mathbf{h}^{(i)}$ to denote the representation of reviews after adopting the *i*-th MVSD layer.

3.5 Learning and Optimization

After a total of ℓ MVSD layers, we obtain the final movie review node representation denoted as $\mathbf{h}^{(\ell)}$. Given a document label $a \in \{\text{SPOILER}, \text{NOT SPOILER}\}$, the predicted probabilities arer calculated as $p(a|\boldsymbol{d}) \propto \exp\big(\mathrm{MLP}_a(\mathbf{h}^{(\ell)})\big)$. We then optimize MVSD with the cross entropy loss function. At inference time, the predicted label is $\mathrm{argmax}_a p(a|\boldsymbol{d})$.

4 Experiment

4.1 Experiment Settings

Datasets. We evaluate MVSD and baselines on two spoiler detection datasets:

Table 3: Accuracy, AUC, and binary F1-score of MVSD and three types of baseline methods on two spoiler detection datasets. We run all experiments **five times** to ensure a consistent evaluation and report the average performance as well as standard deviation. MVSD consistently outperforms the three types of methods on both benchmarks. * denotes that the results are significantly better than the second-best under the student t-test.

Model	Kaggle			LCS		
	F1	AUC	Acc	F1	AUC	Acc
BERT (Devlin et al., 2019)	44.02 (±1.09)	63.46 (±0.46)	77.78 (±0.09)	46.14 (±2.84)	64.82 (±1.36)	79.96 (±0.38)
RoBERTa (Liu et al., 2019)	50.93 (±0.76)	66.94 (±0.40)	79.12 (±0.10)	47.72 (±0.44)	65.55 (±0.22)	80.16 (±0.03)
BART (Lewis et al., 2020)	46.89 (±1.55)	64.88 (±0.71)	78.47 (±0.06)	48.18 (±1.22)	65.79 (±0.62)	80.14 (±0.07)
DeBERETa (He et al., 2021a)	49.94 (±1.13)	66.42 (±0.59)	79.08 (±0.09)	47.38 (±2.22)	65.42 (±1.08)	80.13 (±0.08)
GCN (Kipf and Welling, 2016)	59.22 (±1.18)	71.61 (±0.74)	82.08 (±0.26)	62.12 (±1.18)	73.72 (±0.89)	83.92 (±0.23)
R-GCN (Schlichtkrull et al., 2018)	63.07 (±0.81)	74.09 (±0.60)	82.96 (±0.09)	66.00 (±0.99)	76.18 (±0.72)	85.19 (±0.21)
SimpleHGN (Lv et al., 2021)	60.12 (±1.04)	71.61 (±0.74)	82.08 (±0.26)	63.79 (±0.88)	74.64 (±0.64)	84.66 (±1.61)
DNSD (Chang et al., 2018)	46.33 (±2.37)	64.50 (±1.11)	78.44 (±0.12)	44.69 (±1.63)	64.10 (±0.74)	79.76 (±0.08)
SpoilerNet (Wan et al., 2019)	57.19 (±0.66)	70.64 (±0.44)	79.85 (±0.12)	62.86 (±0.38)	74.62 (±0.09)	83.23 (±1.63)
MVSD (Ours)	**65.08*** (±0.69)	**75.42*** (±0.56)	**83.59*** (±0.11)	**69.22*** (±0.61)	**78.26*** (±0.63)	**86.37*** (±0.08)

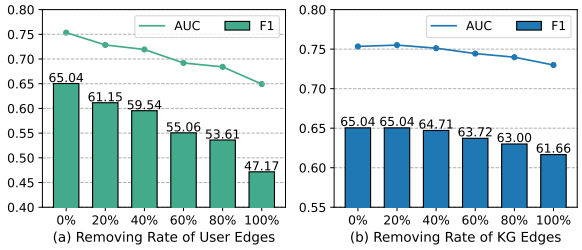


Figure 3: MVSD performance when randomly removing the edges in the user interaction network and external knowledge subgraph. Performance declines with the gradual edge ablations, indicating the contribution of external knowledge and user networks.

- **LCS** is our proposed large-scale automatic spoiler detection dataset. We randomly create a 7:2:1 split for training, validation, and test sets.

- **Kaggle** is a publicly available movie review dataset presented in a Kaggle challenge (Misra, 2019). We present more details about this dataset in Appendix D.

Baselines. We compare MVSD against 9 baseline methods in three categories: pretrained language models, GNN-based models, and task-specific baselines. For pretrained language models, we evaluate BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), BART (Lewis et al., 2020), and DeBERETa (He et al., 2021a). For GNN-based models, we evaluate GCN (Kipf and Welling, 2016), R-GCN (Schlichtkrull et al., 2018), and SimpleHGN (Lv et al., 2021). For task-specific baselines, we evaluate DNSD (Chang et al., 2018) and SpoilerNet (Wan et al., 2019).

4.2 Overall Performance

Table 3 presents the performance of MVSD baseline methods on the two datasets. **Bold** and underline indicate the best and second best performance. Table 3 demonstrates that:

- MVSD achieves state-of-the-art on both datasets, outperforming all baselines by at least 2.01 in F1-score. This demonstrates that our various technical contributions, such as incorporating external knowledge and user networks, multi-view feature extraction, and the cross-context information exchange mechanism, resulted in a more accurate and robust spoiler detection system.

- Graph-based models are generally more effective than other types of baselines. This suggests that in addition to the textual content of reviews, graph-based modeling could bring in additional information sources, such as external knowledge and user interactions, to enable better grounding for spoiler detection.

- Among the two task-specific baselines, SpoilerNet (Wan et al., 2019) outperforms DNSD (Chang et al., 2018), in part attributable to the introduction of the user bias. Our method further incorporates external knowledge and user networks while achieving better performance, suggesting that robust spoiler detection requires models and systems to go beyond the mere textual content of movie reviews.

4.3 External Knowledge and User Networks

We hypothesize that external movie knowledge and user interactions on movie review websites are essential in spoiler detection, providing more context

Table 4: Ablation study concerning multi-view data and the graph structure on Kaggle Dataset. The semantic view, knowledge view, and meta view are denoted as S, K, and M respectively. The knowledge subgraph, movie-review subgraph, and user-review subgraph are denoted as $\mathcal{G}^K$, $\mathcal{G}^M$ and $\mathcal{G}^U$.

Category	Setting	F1	AUC	Acc
multi-view	-w/o S	38.47	61.37	78.15
	-w/o K	62.13	73.46	82.73
	-w/o M	52.99	68.07	79.46
	-w/o O, K	40.05	61.97	78.25
	-w/o O, M	56.44	70.05	80.66
graph structure	-w/o $\mathcal{G}^K$	61.66	72.99	83.12
	-w/o $\mathcal{G}^U$	47.17	64.93	78.00
	-w/o $\mathcal{G}^M$, $\mathcal{G}^K$	56.54	69.98	81.71
	-w/o $\mathcal{G}^M$, $\mathcal{G}^K$	46.65	64.89	78.03
ours	MVSD	**65.08**	**75.42**	**83.59**

and grounding in addition to the textual content of movie reviews. To further examine their contributions in MVSD, we randomly remove 20%, 40%, 60%, 80%, or 100% edges of the knowledge subgraph and user-review subgraph, creating settings with reduced knowledge and user information. We evaluate MVSD with these ablated graphs on the Kaggle dataset and present the results in Figure 3 (a). It is illustrated that the performance drops significantly (about 10% in F1-score when removing 60% of the edges) when we increase the number of removed edges in the user-review subgraph, suggesting that the user interaction network plays an important role in the spoiler detection task. As for the knowledge subgraph, the F1-score drops by 3.38% if we remove the whole knowledge subgraph, indicating that external knowledge is helpful in identifying spoilers. Moreover, it can be observed in Figure3 (b) that the F1-score and AUC only dropouts slightly when removing part of the edges in the knowledge subgraph. This illustrates the robustness of MVSD, as it can achieve relatively high performance while utilizing a subset of movie knowledge.

4.4 Ablation Study

In order to study the effect of different views of data, we remove them individually and evaluate variants of our proposed model on the Kaggle Dataset. We further remove some parts of the graph structure to investigate, Finally, we replace our attention mechanism with simple fusion methods to evaluate the effectiveness of our fusion method.

Table 5: Model performance on Kaggle when our attention mechanism is replaced with simple fusion methods.

View-level	Subgraph-level	F1	AUC	Acc
Ours	Max-pooling	53.73	68.50	79.29
Ours	Mean-pooling	62.27	73.40	83.23
Ours	Concat	61.07	72.63	82.97
Max-pooling	Ours	63.19	74.21	82.86
Mean-pooling	Ours	63.60	74.36	83.30
Concat	Ours	62.90	74.00	82.83
Ours	Ours	**65.08**	**75.42**	**83.59**

Multi-View Study We report the binary F1-Score, AUC, and Acc of the ablation study in Table 4. Among the multi-view data, semantic view data is of great significance as AUC and F1-score drop dramatically when it is discarded. We can see that discarding the external knowledge view or removing the knowledge subgraph reduces the F1-score by about 3%, indicating that the external knowledge of movies is helpful to the spoiler detection task. However, external knowledge doesn't show the same importance as the directly related semantic view or meta view. We believe this is because the external knowledge is not directly related to review documents, so it can only provide auxiliary help to the spoiler detection task.

Graph Structure Study As illustrated in Table 4, after removing the use-review subgraph, the reduced model performs poorly, with a drop of 18% in F1. This demonstrates that the user interaction network is necessary for spoiler detection.

Aggregation and Interaction Study In order to study the effectiveness of the hierarchical mechanism that enables the interaction between views and sub-graphs, we replace the two components of our hierarchical attention with other operations and evaluate them on the Kaggle Dataset. Specifically, we compare our attention module with concatenation, max-pooling, and average-pooling.

In Table 5 we report the binary F1-score, AUC, and Acc. We can see that our approach beats the eight variants in all metrics. It is evident that our approach can aggregate and fuse multi-view data more efficiently than simple fusion methods.

4.5 Qualitative Analysis

We conduct qualitative analysis to investigate the role of external movie knowledge and social networks for spoiler detection. As shown in Table 6, with the guide of external knowledge and user

Table 6: Examples of the performance of three baselines and MVSD. Underlined parts indicate the plots.

Review Text	Label	DeBERTa	R-GCN	SpoilerNet	MVSD
Kristen Wiig is the only reason I wanted to see this movie, and she is insanely hilarious! (...) Wiig plays Annie, (...) becomes jealous of Lillian's new rich friend, Helen. Annie slowly goes crazy and constantly competes against Helen (...)	True	False ✗	False ✗	False ✗	True ✓
The new director was horrible. Not even comparable to Chris Columbus. He changed the entire format of the school (...) why was there a deer next to harry across the lake, he didn't mention that and yet he still put the deer in the movie (...)	False	True ✗	True ✗	True ✗	False ✓
(...) This scene involves Harry getting bombarded by ugly, little squid like creatures and is awe inspiring. And more happens. Harry is having a certain dream over and over again. Lord Voldemort wants to return and he does.	False	False ✗	False ✗	False ✗	True ✓
(...) I remember that for four years in high school, I was a high school nerd/loner, and I liked it. I was shy, I was socially awkward, and I was one of those guys who happened to have a thing for one of the popular girls (...)	False	True ✗	True ✗	True ✗	False ✓

networks, MVSD successfully makes the correct prediction while baseline models fail. Specifically, in the first case, the user is a fan of Kristen Wiig. Guided by the information from the social network, MVSD finds that the user often posted spoilers related to the film star, and finally predicts that the review is a spoiler. In the second case, the user mentioned something done by the director of the movie. With the help of movie knowledge, it can be easily distinguished that what the director has done reveals nothing of the plot.

5 Related Work

Automatic spoiler detection aims to identify spoiler reviews in domains such as television (Boyd-Graber et al., 2013), books (Wan et al., 2019), and movies (Misra, 2019; Boyd-Graber et al., 2013). Existing spoiler detection models could be mainly categorized into two types: keyword matching and machine learning models. Keyword matching methods utilize predefined keywords to detect spoilers, for instance, the name of sports teams or sports events (Nakamura and Tanaka, 2007), or the name of actors (Golbeck, 2012). This type of method requires keywords defined by humans, and cannot be generalized to various application scenarios. Early neural spoiler detection models mainly leverage topic models or support vector machines with handcrafted features. Guo and Ramakrishnan (2010) use bag-of-words representation and LDA-based model to detect spoilers, Jeon et al. (2013) utilize SVM classification with four extracted features, while Boyd-Graber et al. (2013) incorporate lexical features and meta-data of the review subjects (e.g., movies and books) in an SVM classifier. Later approaches are increasingly neural methods: Chang et al. (2018) focus on modeling external genre information based on GRU and CNN, while Wan et al. (2019) introduce item-specificity and bias and utilizes bidirectional recurrent neural networks (bi-RNN) with Gated Recurrent Units (GRU). A recent work (Chang et al., 2021) leverages dependency relations between context words in sentences to capture the semantics using graph neural networks.

While existing approaches have made considerable progress for automatic spoiler detection, it was previously underexplored whether review text itself is sufficient for robust spoiler detection, or whether more information sources are required for better task grounding. In this work, we make the case for incorporating external film knowledge and user activities on movie review websites in spoiler detection, advancing the field through both resource curation and method innovation, presenting a large-scale dataset LCS, an up-to-date movie knowledge base UKM, and a state-of-the-art spoiler detection approach MVSD.

6 Conclusion

We make the case for incorporating external knowledge and user networks on movie review websites for robust and well-grounded spoiler detection. Specifically, we curate LCS, the largest spoiler detection dataset to date; we construct UKM, an up-to-date knowledge base of the film industry; we propose MVSD, a state-of-the-art spoiler detection system that takes external knowledge and user interactions into account. Extensive experiments demonstrate that MVSD achieves state-of-the-art performance on two datasets while showcasing the benefits of incorporating movie knowledge and user behavior in spoiler detection.

Ethics Statement

We envision MVSD as a pre-screening tool and not as an ultimate decision-maker. Though achieving the state-of-the-art, MVSD is still imperfect and needs to be used with care, in collaboration with human moderators to monitor or suspend suspicious movie reviews. Moreover, MVSD may inherit the biases of its constituents, since it is a combination of datasets and models. For instance, pretrained language models could encode undesirable social biases and stereotypes (Li et al., 2022; Nadeem et al., 2021). We leave to future work on how to incorporate the bias detection and mitigation techniques developed in ML research in spoiler detection systems. Given the nature of the task, the dataset contains potentially offensive language which should be taken into consideration.

References

Miguel Arana-Catania, Elena Kochkina, Arkaitz Zubiaga, Maria Liakata, Robert Procter, and Yulan He. 2022. Natural language inference with self-attention for veracity assessment of pandemic claims. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1496–1511, Seattle, United States. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26.

Jordan Boyd-Graber, Kimberly Glasgow, and Jackie Sauter Zajac. 2013. Spoiler alert: Machine learning approaches to detect social media posts with revelatory information. In *Proceedings of the 76th ASIS&T Annual Meeting: Beyond the Cloud: Rethinking Information Boundaries*, ASIST '13, USA. American Society for Information Science.

Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The World Wide Web Conference*, WWW '19, page 151–161, New York, NY, USA. Association for Computing Machinery.

Buru Chang, Hyunjae Kim, Raehyun Kim, Deahan Kim, and Jaewoo Kang. 2018. A deep neural spoiler detection model using a genre-aware attention mechanism. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 183–195. Springer.

Buru Chang, Inggeol Lee, Hyunjae Kim, and Jaewoo Kang. 2021. "killing me" is not a spoiler: Spoiler detection model using graph neural networks with dependency relation-aware attention mechanism. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3613–3617, Online. Association for Computational Linguistics.

Hao Chen, Zepeng Zhai, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. 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)*, pages 2974–2985, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthias Fey and Jan Eric Lenssen. 2019. Fast graph representation learning with pytorch geometric. *arXiv preprint arXiv:1903.02428*.

Jennifer Golbeck. 2012. The twitter mute button: a web filtering challenge. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2755–2758.

Sheng Guo and Naren Ramakrishnan. 2010. Finding the storyteller: Automatic spoiler tagging using linguistic cues. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 412–420, Beijing, China. Coling 2010 Organizing Committee.

Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. OpenKE: An open toolkit for knowledge embedding. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 139–144, Brussels, Belgium. Association for Computational Linguistics.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. 2020. Array programming with numpy. *Nature*, 585(7825):357–362.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021a. Debertav3: Improving deberta using electra-style pretraining with gradient-disentangled embedding sharing.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2021b. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Valentin Hofmann, Xiaowen Dong, Janet Pierrehumbert, and Hinrich Schuetze. 2022. Modeling ideological salience and framing in polarized online groups with graph neural networks and structured sparsity. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 536–550, Seattle, United States. Association for Computational Linguistics.

Linmei Hu, Tianchi Yang, Luhao Zhang, Wanjun Zhong, Duyu Tang, Chuan Shi, Nan Duan, and Ming Zhou. 2021. Compare to the knowledge: Graph neural fake news detection with external knowledge. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 754–763, Online. Association for Computational Linguistics.

Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous graph transformer. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 2704–2710. ACM / IW3C2.

Sungho Jeon, Sungchul Kim, and Hwanjo Yu. 2013. Don't be spoiled by your friends: spoiler detection in tv program tweets. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 7, pages 681–684.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Junzhuo Li and Deyi Xiong. 2022. KaFSP: Knowledge-aware fuzzy semantic parsing for conversational question answering over a large-scale knowledge base. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 461–473, Dublin, Ireland. Association for Computational Linguistics.

Yizhi Li, Ge Zhang, Bohao Yang, Chenghua Lin, Anton Ragni, Shi Wang, and Jie Fu. 2022. HERB: Measuring hierarchical regional bias in pre-trained language models. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 334–346, Online only. Association for Computational Linguistics.

Bin Liang, Qinglin Zhu, Xiang Li, Min Yang, Lin Gui, Yulan He, and Ruifeng Xu. 2022. JointCL: A joint contrastive learning framework for zero-shot stance detection. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 81–91, Dublin, Ireland. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

George Loewenstein. 1994. The psychology of curiosity: A review and reinterpretation. *Psychological bulletin*, 116(1):75.

Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, and Jie Tang. 2021. Are we really making much progress? revisiting, benchmarking and refining heterogeneous graph neural networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1150–1160.

Nikhil Mehta, Maria Leonor Pacheco, and Dan Goldwasser. 2022. Tackling fake news detection by continually improving social context representations using graph neural networks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1363–1380, Dublin, Ireland. Association for Computational Linguistics.

Rishabh Misra. 2019. Imdb spoiler dataset.

Moin Nadeem, Anna Bethke, and Siva Reddy. 2021. StereoSet: Measuring stereotypical bias in pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5356–5371, Online. Association for Computational Linguistics.

Satoshi Nakamura and Katsumi Tanaka. 2007. Temporal filtering system to reduce the risk of spoiling a user's enjoyment. In *Proceedings of the 12th international conference on Intelligent user interfaces*, pages 345–348.

Van-Hoang Nguyen, Kazunari Sugiyama, Preslav Nakov, and Min-Yen Kan. 2020. Fang: Leveraging social context for fake news detection using graph representation. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1165–1174.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep graph infomax. *ICLR (Poster)*, 2(3):4.

Prashanth Vijayaraghavan and Soroush Vosoughi. 2022. TWEETSPIN: Fine-grained propaganda detection in social media using multi-view representations. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3433–3448, Seattle, United States. Association for Computational Linguistics.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2605–2610, Florence, Italy. Association for Computational Linguistics.

Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. Ripplenet: Propagating user preferences on the knowledge graph for recommender systems. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 417–426.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, pages 1480–1489.

Huaxiu Yao, Ying-xin Wu, Maruan Al-Shedivat, and Eric Xing. 2021. Knowledge-aware meta-learning for low-resource text classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1814–1821, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Donghan Yu, Chenguang Zhu, Yuwei Fang, Wenhao Yu, Shuohang Wang, Yichong Xu, Xiang Ren, Yiming Yang, and Michael Zeng. 2022. KG-FiD: Infusing knowledge graph in fusion-in-decoder for open-domain question answering. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4961–4974, Dublin, Ireland. Association for Computational Linguistics.

Wenqian Zhang, Shangbin Feng, Zilong Chen, Zhenyu Lei, Jundong Li, and Minnan Luo. 2022. KCD: Knowledge walks and textual cues enhanced political perspective detection in news media. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4129–4140, Seattle, United States. Association for Computational Linguistics.

Zhi-Hua Zhou. 2018. A brief introduction to weakly supervised learning. *National science review*, 5(1):44–53.

A Graph-Based Social Text Analysis

Graphs and heterogeneous information networks are playing an important role in the analysis of texts and documents on news (Mehta et al., 2022) and social media (Hofmann et al., 2022). In these approaches, graphs and graph neural networks are adopted to represent and encode information in addition to textual content, such as social networks (Nguyen et al., 2020), external knowledge graphs (Zhang et al., 2022), social context (Mehta et al., 2022), and dependency relations between context words (Chang et al., 2021). With the help of additional information sources, these graph-based approaches enhance representation quality by capturing the rich social interactions (Nguyen et al., 2020), infusing knowledge reasoning into language representations (Zhang et al., 2022), and reinforcing nodes' representations interactively (Mehta et al., 2022). As a result, graph-based social text analysis approaches have advanced the state-of-the-art on various tasks such as misinformation detection (Zhang et al., 2022), stance detection (Liang et al., 2022), propaganda detection (Vijayaraghavan and Vosoughi, 2022), sentiment analysis (Chen et al., 2022), and fact verification (Arana-Catania et al., 2022). Motivated by the success of existing graph-based models, we propose MVSD to incorporate external knowledge bases and user networks on movie review platforms through graphs and graph neural networks.

B Limitations

We identify two key limitations:

- **MVSD** utilizes widely-adopted RGCN to model each subgraph, while there are more up-to-date heterogeneous graph algorithms like HGT (Hu et al., 2020), SimpleHGN (Lv et al., 2021). We plan to conduct experiments that replace RGCN with other heterogeneous graph algorithms. Besides, considering the subgraph structure of MVSD, we will test different heterogeneous graph algorithm settings in each subgraph to find out the most efficient algorithm for each subgraph.

- **LCS** is constructed based on IMDB, and the spoiler annotation is based on user self-report. Hence, it is likely that some label is false. In the next step of our work, we will check the labels with the help of experts and weak supervised learning strategy (Zhou, 2018).

C Heterogeneous Graph Construction Details

C.1 Movie-Review Subgraph

N1: *movie* The information about movies, especially the plot, is essential in spoiler detection. We use one node to represent each movie.
N2: *rating* Rating is an essential part of movie review. We use ten nodes to represent the numerical ratings ranging from 1 to 10.
N3: *review* We use one node to represent each movie review document.

We connect these nodes with three types of edges, denoted as $\mathcal{E}^M$:
R1: *review-movie* We connect a review node with a movie node if the review is about the movie.
R2: *movie-rating* We connect a movie node with a rating node according to the overall rating of the movie, rounded to the nearest integer.
R3: *rating-review* We connect a review node with a rating node based on its numeric score.

C.2 User-Review Subgraph

N4: *review* We use one node to represent each review document. Note that review nodes appear both in $\mathcal{V}^M$ (as N1) and $\mathcal{V}^U$ (as N4). Sharing nodes across subgraphs enables MVSD to model the interaction and exchange across different contexts.
N5: *user* We use one node to represent each user.
N6: *year* We use one node to represent each year, modeling the temporal distribution of spoilers.

We connect these nodes with three types of edges, denoted as $\mathcal{E}^U$:
R4: *review-user* We connect a review node with a user node if the user posted the review.
R5: *review-year* We connect a review node with a year node if the review was posted in that year.
R6: *user-year* We connect a user node with a year node if the user created the account in that year.

C.3 Knowledge Subgraph

N7: *movie* We use one node to represent each movie.
N8: *genre* We use one node to represent each movie genre.
N9: *cast* We use one node to represent each distinct director and cast member.
N10: *year* We use one node to represent each year.
N11: *rating* We use ten nodes to represent the numerical ratings ranging from 1 to 10.

We connect these nodes with four types of edges:

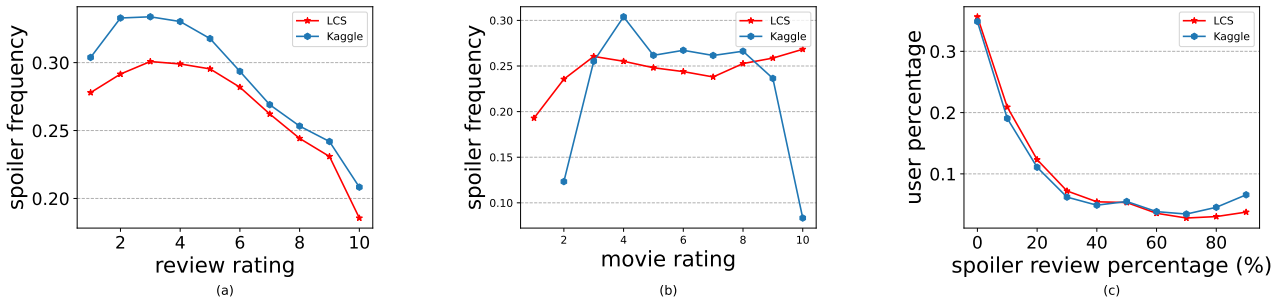


Figure 4: (a) The spoiler frequency of reviews with different ratings; (b) The spoiler frequency of reviews related to movies of different ratings; (c) The percentage of spoilers per user, spoiler review percentage intervals are divided every 10 percent.

Table 7: Statistics of our proposed LCS dataset.

Type	Number	Description
review	1,860,715	The posting time is from 1998 to 2022.
user	259,705	Users that posted these reviews.
movie	147,191	The released year is from 1874 to 2022.
cast	494,221	The cast related to the movies.
spoiler	457,500	24.59% of the reviews are spoilers.

Table 8: Statistics of the Kaggle Dataset.

Type	Number	description
review	573,913	The posting time is from 1998 to 2018.
user	263,407	Users that posted these reviews.
movie	1,572	The released year is from 1921 to 2018.
cast	7,865	The cast related to the movies.
spoiler	150,924	25.87% of the reviews are spoilers.

R7: *movie-genre* We connect a movie node with a genre node according to the genre of the movie.
R8: *movie-cast* We connect a movie node with a cast node if the cast is involved in the movie.
R9: *movie-year* We connect a movie node with a year node if the movie was released in that year.
R10: *movie-rating* We connect a user node with a rating node according to the rating of the movie.

D Dataset Details

We adopt two graph-based spoiler detection datasets, namely Kaggle (Misra, 2019) and our curated LCS. The two datasets are both in English. The publicly available Kaggle dataset only provides incomplete information. Hence, we retrieved cast information based on the movie ids and collected user metadata based on user ids. The statics details of Kaggle after retrieving are listed in table 8, and the statics details of our LCS are listed in table 7.

Table 9: Details of metadata contained in the dataset.

Entity Name	Metadata
Review	time, helpful vote count, total vote count, score
User	create at, badge count, review count
Movie	year, isAdult, runtime, rating, vote count
Cast	birth year, death year, involved movie count

D.1 Data Analysis

We compare **LCS** with another popular spoiler detection dataset Kaggle (Misra, 2019) and presents our findings in Figure 4. We investigate the correlation between spoilers and individual review scores, overall movie ratings, and the behavior of different users. Firstly, we investigate the correlation between spoilers and review scores. Figure 4(a) shows that whether a review containing spoilers has a strong connection with how well the user considers the movie. Additionally, we find that whether a review contains spoilers is also related to the public opinion of the movie, which is illustrated in Figure 4(b). These findings suggest the necessity of leveraging metadata and external knowledge of movies. In addition, we study the fraction of reviews containing spoilers per user. As illustrated in Figure 4(c), the 'spoiler tendency' varies greatly among users. This suggests that it is essential to utilize the user information and how they interact with different movies on review websites.

D.2 Metadata

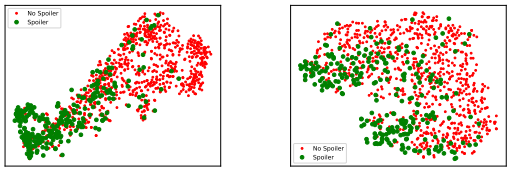
The metadata we collected for both datasets is listed in table 9.

E KG Details

The types of relations, triples, and the number of them are presented in table 10.

Table 10: Statistics of UKM.

Relation	Triple (head-rel.-tail)	Value
show_in	movie-show_in-year	147,191
rated	movie-rated-rating	147,191
genre_is	movie-genre_is-genre	147,191
is_director_of	person-is_director_of-movie	129,483
is_actor_of	person-is_actor_of-movie	379,696
is_actress_of	person-is_actress_of-movie	226,775
is_producer_of	person-is_producerr_of-movie	129,202
is_writer_of	person-is_writer_of	169,024
is_editor_of	person-is_editor_of-movie	49,817
is_composer_of	person-is_composer_of-movie	89,572
is_production_designer_of	person-is_production_designer_of-movie	11,838
is_archive_footage_of	person-is_archive_footage_of-movie	6,328
is_cinematographer_of	personcinematographer_of-movie	76,311
is_archive_sound_of	person-is_archive_sound_of	205
is_self_of	person-is_self_of-movie	129,483



(a) MVSD. (b) R-GCN.

Figure 5: T-SNE visualization of representations of reviews learned by MVSD and R-GCN.

F Experiment Details

Implementation. For pre-trained LMs, we utilize the pre-trained model to get the embeddings and transform them through MLPs. For DNSD and SpoilerNet, we follow the settings in their corresponding papers. For GNNs, we combined the three subgraphs into a whole graph and only utilize the semantic view embedding. We learn a representation for each review, and the representations are passed to an MLP for classification.

F.1 Baseline Details

We compare MVSD with pre-trained language models, GNN-based models, and task-specific baselines to ensure a holistic evaluation. We provide a brief description of each of the baseline methods, in the following.

- **BERT** (Devlin et al., 2019) is a language model pre-trained on a large volume of natural language corpus with the masked language model and next sentence prediction objectives.
- **RoBERTa** (Liu et al., 2019) improves upon BERT by removing the next sentence prediction task and improves the masking strategies.
- **BART** (Lewis et al., 2020) is a transformer encoder-decoder (seq2seq) language model with a bidirectional (BERT-like) encoder and an autoregressive (GPT-like) decoder.
- **DeBERTa** (He et al., 2021b) improves existing language models using disentangled attention and enhanced mask decoder.
- **GCN** (Kipf and Welling, 2016) is short for graph convolutional networks, which enables parameterized message passing between neighbors.
- **R-GCN** (Schlichtkrull et al., 2018) extends GCN to enable the processing of relational networks.
- **SimpleHGN** (Lv et al., 2021) is a simple yet effective GNN for heterogeneous graphs inspired by the GAT (Veličković et al., 2018).
- **DNSD** (Chang et al., 2018) is a spoiler detection framework using a CNN-based genre-aware attention mechanism.
- **SpoilerNet** (Wan et al., 2019) extends the hierarchical attention network (HAN) (Yang et al., 2016) with item-specificity information and item and user bias terms for spoiler detection.

Table 11: Hyperparameter settings of MVSD.

Hyperparameter	Value
GNN input size	768
GNN hidden size	128
GNN layer (in each MVSD layer)	1
MVSD layer L	2
# epoch	120
batch size	1,024
dropout	0.3
learning rate	1e-3
weight decay	1e-5
lr_scheduler_patience	5
lr_scheduler_step	0.1
Optimizer	AdamW

F.2 Hyperparameter Details

We present our hyperparameter settings in Table 11 to facilitate reproduction. The setting for both datasets is the same.

F.3 Computational Resources

Our proposed approach has a total of 0.9M learnable parameters. It takes about 10 GPU hours to train our approach on the Kaggle dataset. We train our model on a Tesla V100 GPU. We conduct all

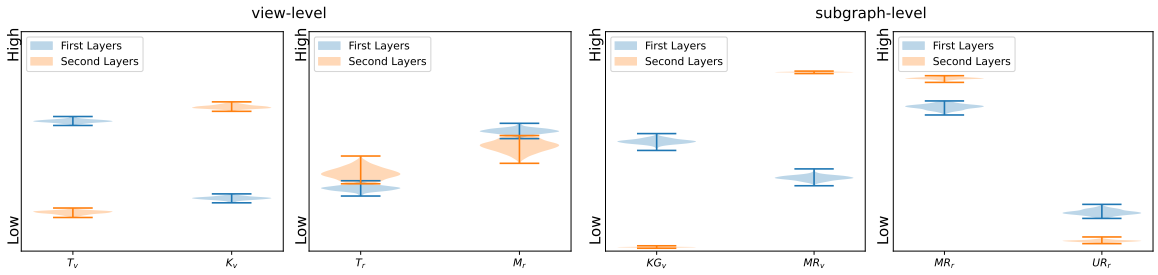


Figure 6: Attention weights learned by our hierarchical attention. Subscript v, r indicate the public nodes movie and review separately. T, M, and K refer to the textual view, the meta view, and the external knowledge view, respectively. This violin plot illustrates the different contributions of each view and subgraph and the process of interaction.

experiments on a cluster with 4 Tesla V100 GPUs with 32 GB memory, 16 CPU cores, and 377GB CPU memory.

F.4 Experiment Runs

For both datasets that have relatively large scales, we adopt the subsampling skill proposed in (Hamilton et al., 2017), which has been successfully used on large graphs (Velickovic et al., 2019). We conduct our approach and baselines five times on both datasets and report the average F1-score, AUC, and accuracy with standard deviation in Table 3. For the experiments in table 4, table 5, and figure 3, we only report the single-run result in the Kaggle dataset due to the lack of computational resources.

F.5 Visualization

To intuitively demonstrate the effectiveness of our representation method, we utilize T-SNE (Van der Maaten and Hinton, 2008) to visualize the representations of movie reviews learned by different models. Specifically, we choose our proposed MVSD and R-GCN (with the second highest performance) and evaluate them on the validation set of the small dataset. It can be observed in Figure 5b that the learned representations of different kinds are relatively mixed together. In contrast, representations learned by MVSD show moderate collocation for both groups of reviews. This illustrates that MVSD yields improved and more comprehensive representation with the effective use of multi-view data and user interaction networks.

F.6 Contribution of Views and Subgraphs

We introduce semantic, meta, and external knowledge views and utilize user-review, movie-review, and knowledge subgraph structures to represent multi-information. To further study the contribution of different views and sub-graphs. We extract the attention weight from the View-level attention layers and Subgraph-level attention layers and illustrate them in violin plots. We select representative features and present them in Figure 6. The four violin plots demonstrate that our proposed hierarchical attention can select the more important features from the variation of attention weight between the first and the second layer, indicating that the contributions of certain representations are varied as they capture features via the graph structure and attention mechanism.

G Significance Testing

To further evaluate MVSD's performance on both datasets, we apply one way repeated measures ANOVA test for the results in Table 3. The result demonstrates that the performance gain of our proposed model is significant on both datasets against the second-best R-GCN on all three metrics with a confidence level of 0.05.

H Scientific Artifact Usage

The MVSD model is implemented with the help of many widely-adopted scientific artifacts, including PyTorch (Paszke et al., 2019), NumPy (Harris et al., 2020), transformers (Wolf et al., 2020), sklearn (Pedregosa et al., 2011), OpenKE (Han et al., 2018), PyTorch Geometric (Fey and Lenssen, 2019). We utilize data from IMDB and following the requirement of IMDB, we acknowledge the source of the data by including the following statement: Information courtesy of IMDb (https://www.imdb.com). Used with permission. Our use of IMDb data is non-commercial, which is allowed by IMDB. We will make our code and data publicly available to facilitate reproduction and further research.