"Killing Me" Is Not a Spoiler: Spoiler Detection Model using Graph Neural Networks with Dependency Relation-Aware Attention Mechanism

Buru Chang[†] **Inggeol Lee**[‡] **Hyunjae Kim**[‡] **Jaewoo Kang**[‡*]
Hyperconnect[†] Korea University[‡]
buru@hpcnt.com
{ingulbull,hyunjae-kim,kangj}@korea.ac.kr

Abstract

Several machine learning-based spoiler detection models have been proposed recently to protect users from spoilers on review websites. Although dependency relations between context words are important for detecting spoilers, current attention-based spoiler detection models are insufficient for utilizing dependency relations. To address this problem, we propose a new spoiler detection model called SDGNN that is based on syntax-aware graph neural networks. In the experiments on two real-world benchmark datasets, we show that our SDGNN outperforms the existing spoiler detection models.

1 Introduction

Spoilers on review websites, which reveal critical details of the original works, can ruin an appreciation for the works. Review websites, such as Rotten Tomato, IMDb, and Metacritic, provide self-reporting systems that tag spoiler information to warn users of spoilers. However, since self-reporting systems depend solely on the active participation of users, they cannot handle the fast-growing volume of newly generated reviews. During the past decade, several machine learning-based spoiler detection (SD) models have been proposed to solve the inefficiency of self-reporting systems. Guo and Ramakrishnan (2010) proposed an automatic SD model that measures the similarity between reviews and synopses of movies. Support vector machine (SVM)-based SD models using handcrafted features have been proposed (Boyd-Graber et al., 2013; Jeon et al., 2016). Recently, attention-based SD models that utilize metadata of review documents achieve state-of-the-art performance on the SD task (Chang et al., 2018; Wan et al., 2019).

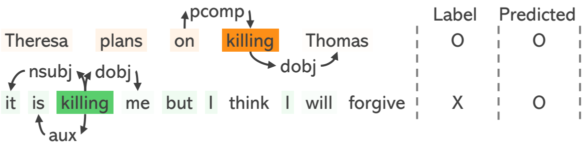

Figure 1: Attention-based models focused on the word "killing" because the word is frequently used in spoiler sentences, which results in incorrect predictions.

However, the attention-based SD models have a lack of using dependency relations between context words. Dependency relations are useful for capturing the semantics of given sentences and detecting spoilers. As shown in Figure 1, although the phrase "killing me" is not a spoiler because the phrase is a metaphor, the attention-based SD models often focus on the word "killing" and classify sentences that contain the phrase "killing me" as spoilers. By providing the information that the word "me" is used as the direct object of the verb "killing," SD models can understand that the phrase is a metaphor.

In this paper, we propose SDGNN, which is a new **S**poiler **D**etection model based on syntax-aware **G**raph **N**eural **N**etworks (GNNs) (Marcheggiani and Titov, 2017) for leveraging dependency relations between context words in sentences to fully capture the semantics. With the success of GNNs We also propose a dependency relation-aware attention mechanism, which is a modification of the gating mechanism used by syntax-aware GNNs, to be suitable for the spoiler detection task. In SD, considering the relative importance of dependency relations. However, existing syntax-aware GNN-based models compute the importance of each dependency relation individually in sentences without considering the context of the given

[*] Corresponding author.

[†] This work was done while the author was affiliated with Korea University.

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 3613–3617
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

sentence. Our proposed dependency relation-aware attention mechanism considers the relative importance of dependency relations. Also, we adopt a previously proposed genre-aware pooling method (Chang et al., 2018) to utilize the genre of works efficiently. In the experiments, we demonstrate the effectiveness of SDGNN on two real-world benchmark datasets in both quantitative and qualitative ways.

2 Our Approach

SDGNN classifies whether a given sentence $x = (w_1, w_2, \cdots, w_n)$ is a spoiler sentence. SDGNN consists of three stages: contextualized word representation, dependency relation-aware attention mechanism, and genre-aware pooling.

Contextualized Word Representation Each word w in the given sentence x is represented with the pretrained word embedding vector (Pennington et al., 2014). We then utilize bi-directional LSTMs (Hochreiter and Schmidhuber, 1997) to encode contextualized word representations $\mathbf{h}^{(0)} \in \mathbb{R}^d$.

Dependency Relation-Aware Attention Mechanism While the gating mechanism in syntax-aware GNNs (Marcheggiani and Titov, 2017; Nguyen and Grishman, 2018) computes the scalar weight of each dependency relation, it does not consider the relative importance of dependency relations, which varies depending on the context of the given sentence. We present a dependency relation-aware attention mechanism that considers the relative importance of dependency relations in the given sentence. The relation-aware attention weights are computed as follows:

$$a^{(k)}_{L(u,v)} = g\left(\mathbf{h}^{(k)}_u \overline{\mathbf{W}}^{(k)} \mathbf{e}^{(k)}_{L(u,v)} + \bar{b}^{(k)}_{L(u,v)}\right), \tag{1}$$

$$\hat{a}^{(k)}_{L(u,v)} = \frac{\exp\left(a^{(k)}_{L(u,v)}\right)}{\sum_{v' \in \mathcal{N}(u)} \exp\left(a^{(k)}_{L(u,v')}\right)}, \tag{2}$$

where $\hat{a}^{k}_{L(u,v)}$ is a scalar attention weight of the dependency relation label $L(u,v)$ of the edge between word nodes u, v. g is the non-linear function and exp$(\cdot)$ is an exponential function. $\overline{\mathbf{W}}^{(k)} \in \mathbb{R}^{d \times d}$ is the attention weight matrix for the k-th layer. $\mathbf{e}^{(k)}_{L(u,v)} \in \mathbb{R}^d$ and $\bar{b}^{k}_{L(u,v)} \in \mathbb{R}$ are latent features of the dependency relation $L(u,v)$.

Finally, we aggregate the latent feature for each node u as follows:

$$\mathbf{h}^{(k)}_u = f\left(\sum_{v \in \mathcal{N}(u)} \hat{a}^{(k)}_{L(u,v)} \mathbf{W}^{(k)} \mathbf{h}^{(k-1)}_v + \mathbf{b}^{(k)}\right), \tag{3}$$

where $\mathbf{W}^{(k)} \in \mathbb{R}^{d \times d}$ and $b^{(k)} \in \mathbb{R}^d$ are the weight matrix and bias term, respectively, for the k-th layer. f is a non-linear function. $\mathbf{h}^{(0)}$ is the outputs of the bi-directional LSTMs in the previous stage.

There are two main differences between our proposed dependency-aware attention mechanism and the gating mechanism used by syntax-aware GNNS. First, the dependency-aware attention mechanism employs the softmax function to capture the relative importance of dependency relations, while the gating mechanism computes the scalar weights by the inner-product of latent features of words and dependency relations. Second, the gating mechanism utilizes only three dependency relations (*forward*, *backward*, and *self*) because of the over-parameterization issue. On the other hand, our proposed dependency relation-aware attention mechanism utilizes all the 82 types of dependency relations without suffering from the over-parameterization issue since the weight matrix in Equation 3 does not depend on the number of relations. The number of trainable parameters of SDGNN is proportionate to d^2 while that of syntax-aware GNNs is proportionate to $|L| \cdot d^2$, where $|L|$ is the number of relations.

Genre-Aware Pooling Genre information is useful for detecting spoilers. To leverage genre information, we employ a genre-aware pooling method following Chang et al. (2018). The genre-aware pooling computes the attention weights between the latent features of words and a genre feature captured from genre information of works. We then obtain a latent feature vector $\mathbf{x}$ for the given sentence x.

Optimization We compute the spoiler probability $\hat{y}$ of the given sentence x with the following the linear transformation:

$$\hat{y} = \sigma(\mathbf{w}\mathbf{x} + b), \tag{4}$$

where $\mathbf{w}$ and b are trainable parameters, and σ is a sigmoid function. We use the weighted binary cross entropy (Wan et al., 2019) as the loss function.

$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{x_i \in \mathcal{D}} (y_i \texttt{log}(\hat{y}_i) + \eta \cdot (1 - y_i)\texttt{log}(1 - \hat{y}_i)), \tag{5}$$

where y id the ground truth of spoiler information and $\mathcal{D}$ indicates the dataset. η is a hyperpameter used to balance the number of spoiler and non-spoiler labels in the training data. All the trainable parameters of SDGNN are updated by minimizing the loss function with gradient descent.

Statistics	Goodreads	TVTropes
# of Training Sentences	14,007,593	11,970
# of Validation Sentences	128,718	2,808
# of Test Sentences	3,536,341	1,477
# of Edge Types	82	82
# of Genre	542	30
Avg. # of Nodes per Sentence	17.7	21.03
Avg. # of Edges per Sentence	33.4	40.06
Avg. # of Genre per Sentence	4.95	2.40

Table 1: Statistics of the datasets.

3 Experiments

3.1 Experimental Setup

Datasets We evaluated our proposed model on the two public spoiler datasets: **Goodreads** (Wan et al., 2019) and **TVTropes** (Boyd-Graber et al., 2013) [1]. The Goodreads dataset consists of spoiler sentences on book reviews, and only 3.22% of entire sentences are labeled as spoiler sentences. The TVTropes dataset consists of descriptions of 884 TV programs from the TVTropes site, and 52.7% of the descriptions are labeled as spoilers. The statistics of the datasets are summarized in Table 1.

Baseline Models We compared our proposed model with the following state-of-the-art SD models: **SVM** (Boyd-Graber et al., 2013; Jeon et al., 2016), **CNN** (Kim, 2014), **HAN** (Yang et al., 2016), **SpoilerNet** (Wan et al., 2019) and **DNSD** (Chang et al., 2018). Note that the implementation details about our experiments are described in the Appendix due to space limitations.

Metrics We use Area Under the Receiver Operating Characteristics curve (**AUROC**) used in Wan et al. (2019) as an evaluation metric. We also use an **F1** score following Chang et al. (2018).

Implemention Details We trained and evaluated the models on two TITAN X (Pascal) GPUs. We implemented SDGNN using PyTorch v1.1. We used Stanford CoreNLP (Manning et al., 2014) to generate dependency parse trees. We employed GloVe (Pennington et al., 2014) to represent word vectors in neural network-based models including SDGNN. Using the validation set and grid search, we searched optimal hyper-parameters for each SD model. All the neural network-based models were trained with the learning rate of 0.001 and the Adam optimizer (Kingma and Ba, 2014). A batch size of 1024 was used for training TextCNN, HAN, and DNSD, and a batch size of 512 was used for training SpoilerNet and SDGNN. To prevent overfitting, we applied L2-normalization with λ = 1e-5 and a dropout rate of 0.5. For TextCNN, we used 50 filters with kernel sizes of 3, 4, and 5. For efficient training on deep learning libraries, SDGNN set the maximum number of words to 50. For SDGNN, we used Leaky ReLU for the non-linear function g, and ReLU for f. We set $k = 2$ for SyntacticGCN, C-GCN, and SDGNN. We use $\eta = 0.05$ for the Goodreads dataset, which is unbalanced.

Models	Goodreads		TVTropes	
	AUROC	F1	AUROC	F1
SVM	0.880	0.162	0.735	0.698
TextCNN	0.904	0.188	0.779	0.738
HAN	0.915	0.190	0.785	0.750
SpoilerNet	0.924	0.194	0.808	0.768
DNSD	0.928	0.199	0.818	0.788
SDGNN	**0.938**	**0.210**	**0.828**	**0.801**

Table 2: Evaluation results on two benchmark datasets. The best results are highlighted in bold.

Models	Goodreads	
	AUROC	F1
SytacticGCN	0.933	0.204
C-GCN	0.923	0.193
SDGNN	**0.938**	**0.210**

Table 3: Evaluation results on the Goodreads dataset.

3.2 Results

The experimental results are summarized in Table 2. Evaluation results show that our proposed SDGNN outperforms all the baseline models including attention-based models. This result demonstrates that our proposed dependency relation-aware attention mechanism contributes to improving SD performance.

4 Analysis and Discussion

4.1 Analysis of Relative Importance

To further demonstrate the usefulness of the relative importance of dependency relations, we conducted quantitative and qualitative analysis.

Quantitative We compared SDGNN with the more syntax-aware GNN-based models, SyntacticGCN (Marcheggiani and Titov, 2017) and C-GCN (Zhang et al., 2018). We trained and evaluated the models on the Goodreads dataset. We

[1] We obtained the datasets from Wan et al. (2019) and Boyd-Graber et al. (2013), respectively.

Models	Genres	Sentences	Prediction	Label
DNSD	Romance	it 's killing me but i think i 'll forgive him no matter what he did or did n't do .	Positive	Negative
	Fantasy	the villains are decidedly vicious and in some cases insane .	Positive	Negative
SDGNN	Romance	it 's killing me but i think i 'll forgive him no matter what he did or did n't do .	Negative	Negative
	Fantasy	the villains are decidedly vicious and in some cases insane .	Negative	Negative

Table 4: Visualization of attention scores from DNSD, SpoilerNet, and SDGNN on test data.

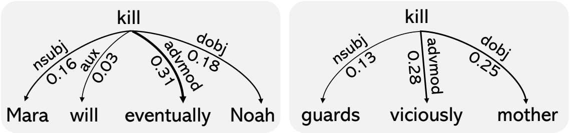

Figure 2: Partial graphs of dependency parse trees with dependency relation-aware attention weights.

utilize contextualized word representations and the genre-aware pooling method to SyntacticGCN and C-GCN. The evaluation results are summarized in Table 3. Our proposed SDGNN outperformed SyntacticGCN and C-GCN. This result demonstrates that our proposed attention mechanism is effective by considering the relative importance of dependency relations. Although SDGNN significantly reduced the number of parameters, SDGNN achieved better results compared to SyntacticGCN and C-GCN.

Qualitative In Figure 2, the attention weights of the adverbial modifier (advmod) linked to the words "eventually" and "viciously" are high, which indicates that adverbial modifiers frequently can be important hints for detecting spoilers. In the right partial graph, the attention weight of the (dobj) is relatively higher than that in the left partial graph. Since the word "mother" is not typically used as the object of the word "kill" in the original works, the phrase "kill mother" is a critical hint in detecting spoilers, and SDGNN effectively captures the phrase.

4.2 Case Study

We sample several sentences from the test set of the Goodreads dataset to explore how the models detect spoilers. Table 4 shows the visualization of attention scores in the pooling layer obtained by DNSD and SDGNN, respectively. The first sentence contains the verb "killing," but it is not a spoiler sentence because the phrase "killing me" is a metaphor. In this case, DNSD failed to correctly classify the sentence since DNSD cannot fully capture the semantics of the sentence. On the other hand, SDGNN focused on not only the word "killing" but also on the word "me" and classified the sentence correctly since SDGNN employs the dependency relation (dobj) between the word "me" and the word "killing".

The second sentence is a non-spoiler because it is obvious that villains are vicious in most original works. DNSD classified the sentence as a spoiler because the model solely focused on individual words such as "villains", "vicious", and "insane", rather than the understanding of the overall semantics of the sentence. On the other hand, SDGNN classified the sentence correctly as the word "are" is used to describe characters in many cases, and SDGNN understands the semantics of the sentence.

4.3 Discussion

Dependency Parsing on User-Generated Texts The spoiler datasets are user-generated texts, which are intrinsically noisy. To examine the influence of noises on dependency parsing results and the performance of SDGNN, we sampled 100 sentences from Goodreads. We manually classified whether the sentences are noise or not, and 28 of 100 sentences were classified as noisy sentences. Dependency parsing results on well-structured sentences seem good, but dependency parsing results on noisy sentences are poor. However, there is no significant gap in performance. SDGNN achieved 85.7% accuracy on noisy sentences and 87.5% accuracy on well-structured sentences. Since our proposed dependency relation-aware attention mechanism of SDGNN filters noisy information, SDGNN could detect spoilers even on noisy sentences.

Subjectivity in Judging Spoilers Since judging a sentence as a spoiler is a subjective task, label inconsistency occurs in spoiler datasets crawled from self-reporting systems. Guo and Ramakrishnan (2010) found that 23% of the labels of their manually labeled data is different from the original labels of IMDb reviews. One of the ways to mitigate label inconsistency is to solidify the definition of a spoiler. Although the TV Tropes site defines spoilers, efforts should be made for a more rigorous and linguistic definition in future studies. Another possible way is to employ reviewers' information in detecting spoilers. *Reviewer biases* of SpoilerNet can alleviate label inconsistency between users.

5 Conclusion

In this paper, we proposed a novel spoiler detection model called SDGNN which is based on syntax-aware GNNs that utilize dependency relations between context words. We also proposed a dependency relation-aware attention mechanism for considering the relative importance of dependency relations. In the experiments, our proposed SDGNN model achieved the state-of-the-art performance on two spoiler datasets. Our experimental results demonstrate the effectiveness of dependency relations in the spoiler detection task and our dependency relation-aware attention mechanism.

Acknowledgments

This research was supported by National Research Foundation of Korea (NRF-2020R1A2C3010638). This research was also supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Creative Consilience program (IITP-2020-0-01819) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

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*, page 45. American Society for Information Science.

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.

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*, pages 412–420. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Sungho Jeon, Sungchul Kim, and Hwanjo Yu. 2016. Spoiler detection in tv program tweets. *Information Sciences*, 329:220–235.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv preprint arXiv:1703.04826*.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Mengting Wan, Rishabh Misra, Ndapa Nakashole, and Julian McAuley. 2019. Fine-grained spoiler detection from large-scale review corpora. *arXiv preprint arXiv:1905.13416*.

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.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*.