Story Embedding: Learning Distributed Representations of Stories based on Character Networks (Extended Abstract)*

O-Joun Lee[1] and **Jason J. Jung**[2,†]
[1]Future IT Innovation Laboratory, Pohang University of Science and Technology, Republic of Korea
[2]Department of Computer Engineering, Chung-Ang University, Seoul, Republic of Korea
ojlee112358@postech.ac.kr, j3ung@cau.ac.kr

Abstract

This study aims to represent stories in narrative works (i.e., creative works that contain stories) with a fixed-length vector. We apply subgraph-based graph embedding models to dynamic social networks of characters that appeared in stories (character networks). We suppose that interactions between characters reflect the content of stories. We discretize the interactions by discovering the subgraphs and learn representations of stories by predicting occurrences of the subgraphs in corresponding character networks. We find subgraphs rooted in each character on each scene in multiple scales, using the WL (Weisfeiler-Lehman) relabeling process. To predict occurrences of subgraphs, we apply two approaches: (i) considering changes in subgraphs according to scenes and (ii) focusing on subgraphs on the last scene. We evaluated the proposed models by measuring the similarity between real movies with vector representations that were generated by the models.

1 Introduction

Various studies have been conducted to represent stories with a computational model. Most of the studies [Callaway and Lester, 2002; Mani, 2012; Jung *et al.*, 2017] concentrated on what happened in stories (i.e., events). To convey the meanings of events, they have proposed representations that cover vast and detail information, such as actors in the events (characters), behaviors of the characters, purposes of the characters, consequences of the behaviors, causality between the events, and so on. This information was annotated using graphs [Purdy and Riedl, 2016] or markup languages [Mani, 2016].

Another approach is character networks, which are dynamic social networks of characters that appeared in the stories [Lee and Jung, 2019a; Labatut and Bost, 2019]. The character network is not comparable with the event-based models in terms of semantic-richness. Since the character network represents only the existence and frequency of interactions between the characters, this model cannot handle the exact meanings of the interactions or events. However, previous studies have shown that extracting character networks from narrative multimedia is much easier than composing the event-based models [Lee and Jung, 2019a]. Also, they exhibited that information in the character network is enough for a few practical applications: classifying characters [Weng *et al.*, 2009; Park *et al.*, 2012; Tran and Jung, 2015], summarizing [Tran *et al.*, 2017; Bost *et al.*, 2019], and recommending [Lee and Jung, 2019b] narrative multimedia. Since a story is a series of events and interactions between characters describe the events [McKee, 1997], we can assume the content of the events from tendencies of the interactions.

However, the above two models have a common limitation. They are graphical data representations, which aim to represent a single narrative work. Thus, these models have difficulties in comparing a story with another story. Although our previous studies [Lee and Jung, 2018; Lee and Jung, 2019b] attempted to estimate the similarity between stories, they commonly used hand-crafted and heuristic features. One of them [Lee and Jung, 2018] used cohesion of communities in character networks. Another one [Lee and Jung, 2019b] analyzed locations of major events in stories.

Since these hand-crafted features are designed for particular media or tasks, they are not much reusable in other applications or studies. For example, the number of characters and major events is different from kinds of media. TV series or epic novels will contain far more characters and major events than movies. This difference will affect both the community-based and event-based similarity estimation.

This study aims to learn task-agnostic and media-independent representations of stories by embedding character networks. The existing studies have already shown that structural features of character networks are effective for analyzing stories [Labatut and Bost, 2019]. Therefore, we apply unsupervised graph embedding techniques to embed character networks.

To focus on the structural features, we discover substructures from character networks using the WL (Weisfeiler-Lehman) relabeling process [Shervashidze *et al.*, 2011]. Then, with Doc2Vec [Le and Mikolov, 2014], we represent a story by predicting which substructures appear in its char-

[*] This is an extended abstract of an article *"Story Embedding: Learning Distributed Representations of Stories based on Character Networks"* published in [Lee and Jung, 2020].

[†] Corresponding Author.

acter network. We propose two story embedding models that consider time-sequential features of stories and not.

2 Preliminaries

This section presents the fundamental concepts and assumptions in this study. The existing studies [Lee and Jung, 2019a] represented the time-sequential features of stories by defining character networks as dynamic social networks. The nodes and edges of the character network are characters and interactions between the characters, respectively. Weights on the edges are interaction frequencies between characters. The character network is defined as follows;

Definition 1 (Character Network [Lee and Jung, 2019a]). *Suppose that n is the number of characters that appeared in a narrative work, $\mathbb{C}_\alpha$, and $\mathbb{C}_\alpha$ consists of L scenes from $s_{\alpha,1}$ to $s_{\alpha,L}$. When $N(s_{\alpha,l})$ indicates a character network on $s_{\alpha,l}$, $N(s_{\alpha,l})$ can be described as a matrix $\in \mathbb{R}^{n\times n}$. Each element of $N(s_{\alpha,l})$ denotes interaction frequency between two characters from $s_{\alpha,1}$ to $s_{\alpha,l}$. This can be formulated as:*

$$N(\mathbb{C}_\alpha) = N(s_{\alpha,L}) = \begin{bmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{n,1} & \cdots & a_{n,n} \end{bmatrix}, \tag{1}$$

where $a_{i,j}$ indicates the proximity of c_i to c_j when c_i is the i-th character in $\mathbb{C}_\alpha$.

In this study, we extract the character network from movie scripts. Characters were identified by their names, scenes were segmented using scene headings, and $a_{i,j}$ was measured by the number of dialogues that c_i spoke to c_j.

The character network represents interactions between characters. Since the interactions are designed to describe events in scenes, we can roughly estimate the meanings of events by analyzing the interactions. For example, in 'The Godfather' (1972), interaction frequencies of 'Michael Corleone' for his family members became larger according to running time. This change reflects that 'Michael Corleone' turned from normal people to a mafia boss. Therefore, we assume that vector representations of character network structures reflect the content of corresponding stories. We call the vector representations 'story vectors.' This is defined as follows;

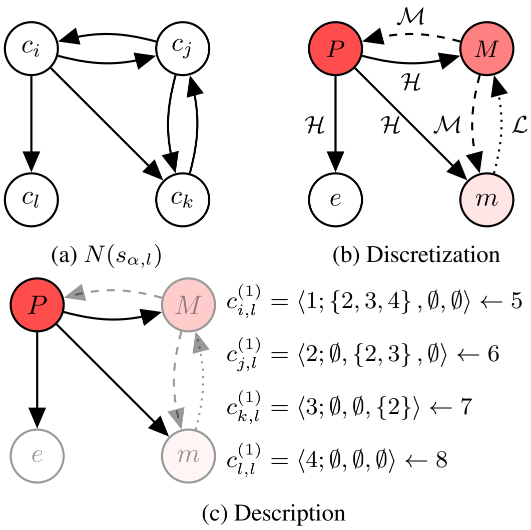
Definition 2 (Story Vector). *Let $\Phi(\mathbb{C}_\alpha)$ be a story vector of $\mathbb{C}_\alpha$. Closer locations of $\Phi(\mathbb{C}_\alpha)$ and $\Phi(\mathbb{C}_\beta)$ indicate that $N(\mathbb{C}_\alpha)$ and $N(\mathbb{C}_\beta)$ have more similar structures. We suppose that when $\mathbb{C}_\alpha$ and $\mathbb{C}_\beta$ contain similar stories, $N(\mathbb{C}_\alpha)$ and $N(\mathbb{C}_\beta)$ are structurally similar. This is formulated as:*

$$k(\mathbb{C}_\alpha, \mathbb{C}_\beta) \simeq k(N(\mathbb{C}_\alpha), N(\mathbb{C}_\beta)) = \langle \Phi(\mathbb{C}_\alpha), \Phi(\mathbb{C}_\beta)\rangle, \tag{2}$$

where $k(\cdot,\cdot)$ indicates a kernel function.

3 Story Embedding

This section presents methods of how we apply the graph structure embedding methods to character networks in consideration of narrative characteristics.


c_i
c_j
c_l
c_k
P
M
e
m
$\mathcal{M}$
$\mathcal{H}$
$\mathcal{H}$
$\mathcal{H}$
$\mathcal{M}$
$\mathcal{L}$
P
M
e
m
$c_{i,l}^{(1)} = \langle 1; \{2,3,4\}, \emptyset, \emptyset\rangle \leftarrow 5$
$c_{j,l}^{(1)} = \langle 2; \emptyset, \{2,3\}, \emptyset\rangle \leftarrow 6$
$c_{k,l}^{(1)} = \langle 3; \emptyset, \emptyset, \{2\}\rangle \leftarrow 7$
$c_{l,l}^{(1)} = \langle 4; \emptyset, \emptyset, \emptyset\rangle \leftarrow 8$

(a) $N(s_{\alpha,l})$

(b) Discretization

(c) Description

Figure 1: Steps of the proximity-aware WL relabeling process from degree 0 to 1, where c_i, c_j, c_k, and c_l appear on $s_{\alpha,l}$. Darkness of nodes and kinds of lines indicate degree of centrality and proximity of characters, respectively.

3.1 Social Roles of Characters

Various substructures are used for embedding structural features of graphs, such as subgraphs, subtrees, meta-paths, and so on. This study employs subgraphs and uses the WL relabeling process to discover the subgraphs. The WL relabeling describes local structures around each node in multiple scales. Character networks are relatively smaller than ordinary social networks, and communities of characters are placed around the protagonists [McKee, 1997]. Therefore, the subgraphs, which represent structures around nodes with specific ranges, are appropriate for embedding character networks.

The WL relabeling considers only adjacency between nodes. However, interaction frequencies between characters are as important as the existence of interactions. For example, a character has interactions with both the protagonist and antagonist. Which one is closer to the character than the other is significant for determining the character's role. Thus, as shown in Fig. 1, we modified the WL relabeling process to cover the proximity information. We call this modification 'proximity-aware WL relabeling.'

In character networks, all nodes are intrinsic. To apply the WL relabeling, we should assign labels on the characters and interactions. We cluster the characters and interactions into three categories according to their centrality and weights, respectively [Weng *et al.*, 2009; Park *et al.*, 2012; Tran and Jung, 2015]. As shown in Fig. 1 (a) and (b), characters are clustered into main (M), minor (m), and extra (e) characters. Also, interactions are clustered into high ($\mathcal{H}$), meddle ($\mathcal{M}$), and low ($\mathcal{L}$) proximity. We use the k-means clustering method with three initial centroids: maximum, median, and minimum. Additionally, a character with the highest centrality is labeled as the protagonist (P).

Then, as shown in Fig. 1 (c), we describe subgraphs rooted in each character by using labels of adjacent characters and interactions. We assign the description on the character as a new label. In practice, we make a list by combining labels of adjacent characters and interactions (e.g., $M\mathcal{H}, m\mathcal{H}, e\mathcal{H}$ for c_i in Fig. 1). We sort the list, put a label of the character at first, and apply the hash function for the list. By iterating the description and relabeling, labels of characters represent broader structures. Initial labels (P, M, m, and e) are subgraphs on degree 0. Each iteration makes subgraphs on degree $d+1$ using subgraphs on degree d.

A subgraph represents the interactions of a character at a scale. Since we can conjecture the character's role from the interactions, we call the subgraph 'social role.' This is defined as follows;

Definition 3 (Social Role). *Let $c_{i,l}^{(d)}$ be a social role of c_i on $s_{\alpha,l}$ at a degree $d \in [0, D]$. $c_{i,l}^{(d)}$ is expressed by one-hop connectivity of c_i at degree $d-1$. This is formulated as:*

$$c_{i,l}^{(d)} = \left\langle c_{i,l}^{(d-1)}; \mathcal{H}_{i,l}^{(d-1)}, \mathcal{M}_{i,l}^{(d-1)}, \mathcal{L}_{i,l}^{(d-1)} \right\rangle, \qquad (3)$$

where $\mathcal{H}_{i,l}^{(d-1)}$, $\mathcal{M}_{i,l}^{(d-1)}$, and $\mathcal{L}_{i,l}^{(d-1)}$ indicate sets of social roles rooted in neighborhoods of c_i at degree $d-1$. These sets include neighborhoods that receive high, medium, and low proximity from c_i, respectively.

In this study, we use Doc2Vec [Le and Mikolov, 2014] for embedding character networks. Doc2Vec trains vector representations by predicting occurrences of words in fixed-size windows. However, in graphs, the number of neighboring nodes is not constant. Narayanan et al. [2016] have proposed radial neighborhoods that define neighborhoods of a subgraph as subgraphs rooted in neighboring nodes on adjacent degrees. Since social roles also have temporal adjacency, we extend radial neighborhoods to temporal-radial neighborhoods as subgraphs rooted in neighboring nodes on adjacent degrees and scenes. This can be formulated as:

$$\begin{aligned}\mathcal{N}\left(c_{i,l}^{(d)}\right) = \Big\{ c_{j,m}^{(e)} \Big| & |i-j| \leq \Delta_L, \\ & |d-e| \leq \Delta_D, c_j \in \mathcal{N}(c_i) \Big\},\end{aligned} \qquad (4)$$

where $\mathcal{N}(\cdot)$ indicates neighborhood sets of nodes or subgraphs, and Δ_L and Δ_D are window sizes for scenes and degrees, which are commonly 1 in this study.

3.2 Learning Representations of Stories

We embed character networks using which social roles occur in the character networks. In other words, stories are represented by which types of characters appear in the stories. For learning the occurrences, we propose two models: (i) flow-oriented Story2Vec (Story2Vec-F) and (ii) denouement-oriented Story2Vec (Story2Vec-D).

Since stories are time-sequential, Story2Vec-F learns changes in social roles according to narrative time. Using the PV-DM (distributed memory model of paragraph vector) method in Doc2Vec, we predict co-occurrence probability between social roles in the temporal-radial neighborhoods. We estimate the co-occurrence probability of a social role given stories and neighboring social roles. As with the negative sampling [Mikolov *et al.*, 2013], the estimation is conducted by applying the sigmoid function to inner products between vector representations of the case and condition. Since Story2Vec-F has multiple conditions, we make a representative vector of the conditions by averaging vectors for the conditions. This is formulated as:

$$P\left(c_{i,l}^{(d)} \middle| \mathcal{N}\left(c_{i,l}^{(d)}\right), \Phi(\mathbb{C}_\alpha)\right) \simeq \sigma\left(\Phi\left(c_{i,l}^{(d)}\right)^{\intercal} \Phi_{\mathcal{N}}\left(c_{i,l}^{(d)}\right)\right), \quad (5)$$

where $\sigma(\cdot)$ indicates the sigmoid function, and $\Phi_{\mathcal{N}}(\cdot)$ denotes a representative vector of conditions.

Therefore, an objective function of Story2Vec-F maximizes the co-occurrence probability of occurred social roles given stories and neighborhoods and minimizes the probability of the other social roles. This can be formulated as:

$$\begin{aligned}\mathcal{L}_F\left(c_{i,l}^{(d)}\right) = & \log P\left(c_{i,l}^{(d)} \middle| \mathcal{N}\left(c_{i,l}^{(d)}\right), \Phi(\mathbb{C}_\alpha)\right) \\ & - \sum_{\forall \mathcal{S}_b \neq c_{i,l}^{(d)}} \log P\left(\mathcal{S}_b \middle| \mathcal{N}\left(c_{i,l}^{(d)}\right), \Phi(\mathbb{C}_\alpha)\right),\end{aligned} \quad (6)$$

where $\mathcal{S}_b$ is an arbitrary social role, and $\sum$ indicates the arithmetic mean operator. However, we cannot examine all social roles; 37,631 social roles were in 142 movies. Thus, we use the negative sampling to reduce computational loads for the negative samples. The second term of Eq. 6 is estimated as:

$$\sum_{j=1}^{k} \mathbb{E}_{\mathcal{S}_b \sim P_n(\mathcal{S})}\left[\log \sigma\left(-\Phi(\mathcal{S}_b)^{\intercal} \Phi_{\mathcal{N}}\left(c_{i,l}^{(d)}\right)\right)\right], \quad (7)$$

where $P_n(\mathcal{S}) \propto U(\mathcal{S})^{\frac{3}{4}}$ indicates a noise distribution of social roles, which is proportional to their unigram distribution. The first term is as with Eq. 5. The vector representations are updated by the gradient of the objective function for the representations.

Climaxes and denouements of stories have more influence on user impressions than the other parts [McKee, 1997]. Story2Vec-D concentrates on structures of character networks at last. Using the PV-DBOW (distributed bag-of-words version of the paragraph vector) method in Doc2Vec, Story2Vec-D predicts occurrence probability of social roles in only $N(\mathbb{C}_\alpha) = N(s_{\alpha,L})$. The method for predicting the occurrence probability is similar to Story2Vec-F. A difference is that Story2Vec-D has only one condition, stories. This can be formulated as:

$$P\left(c_{i,L}^{(d)} \middle| \Phi(\mathbb{C}_\alpha)\right) \simeq \sigma\left(\Phi\left(c_{i,L}^{(d)}\right)^{\intercal} \Phi(\mathbb{C}_\alpha)\right). \quad (8)$$

An objective function of Story2Vec-D maximizes the occurrence probability of social roles that appeared in $N(s_{\alpha,L})$ and minimizes the probability of the others. This can be formulated as:

$$\begin{aligned}\mathcal{L}_D\left(c_{i,L}^{(d)}\right) = & \log P\left(c_{i,L}^{(d)} \middle| \Phi(\mathbb{C}_\alpha)\right) \\ & - \sum_{\forall \mathcal{S}_b \notin \mathcal{S}(s_{\alpha,L})} \log P\left(\mathcal{S}_b \middle| \Phi(\mathbb{C}_\alpha)\right),\end{aligned} \quad (9)$$

where $\mathcal{S}(s_{\alpha,L})$ indicates a set of social roles discovered from $N(s_{\alpha,L})$. The second term of the objective function is estimated by the same method with Eq. 7, due to the negative sampling.

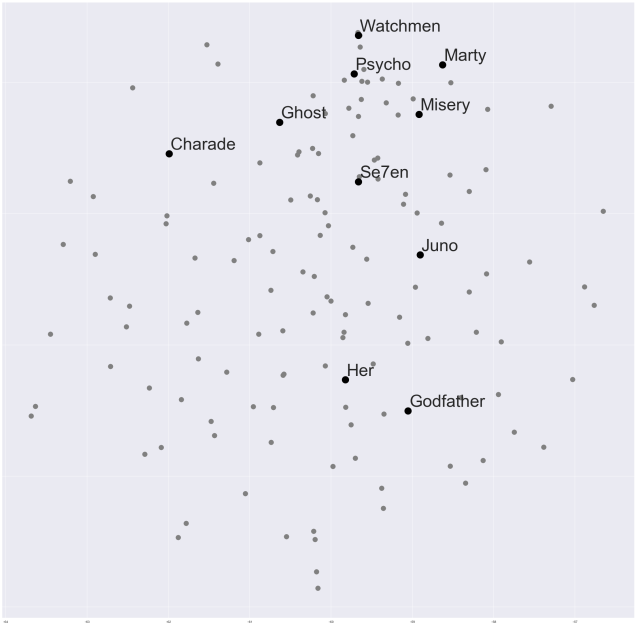


Figure 2: t-SNE projection results of story vectors composed for 142 real movies by the Story2Vec-U model.

We integrate the above two models by conducting the models alternately. First, Story2Vec-F predicts all social roles in a story and updates their representations. Then, Story2Vec-D also conducts the prediction and update for social roles in the last scene. We call the combined model 'unified Story2Vec (Story2Vec-U).' Fig. 2 presents the results of the Story2Vec-U model for real movies.

4 Evaluation

We evaluated the proposed model by comparing its accuracy for estimating similarity between stories with the genre similarity. Similarity between $\mathbb{C}_\alpha$ and $\mathbb{C}_\beta$ was measured by a multiplication of cosine similarity and inverse of Euclidean distance between $\Phi\left(\mathbb{C}_\alpha\right)$ and $\Phi\left(\mathbb{C}_\beta\right)$. The similarity estimation is conducted for 142 real movies[1]. The genre similarity is calculated by applying the Jaccard index on genre sets of movies, which were annotated in IMDB[2]. Character networks were composed by CharNet-Extractor[3] by analyzing movie scripts collected from IMSDb[4]. As ground truth, we asked 50 human evaluators to annotate similarity between the movies through a web application[5]. The accuracy of the estimated similarity was measured by absolute errors for the annotated one.

Fig. 3 presents experimental results. The proposed models exhibited better performance than the genre similarity. The genre is the most widely-used taxonomy of narrative multimedia and mostly annotated by domain experts. This result underpins the effectiveness of the proposed models. Among the three proposed models, Story2Vec-D performed the highest accuracy. Climaxes and denouements of stories might have more influence on user impressions than the development of stories. Story2Vec-U outperformed only Story2Vec-F. The method for combining the dynamic and static approaches was useful, but not enough. Alternately conducting the two models might give too many learning opportunities to Story2Vec-F, comparing with Story2Vec-D.

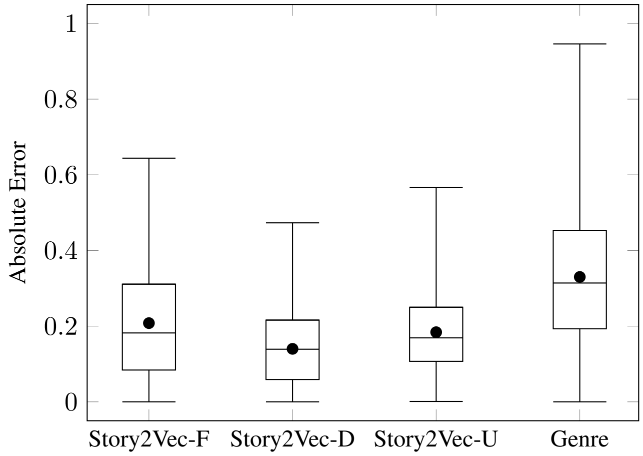


Figure 3: Accuracy of the proposed models for measuring the story similarity. Box-and-whisker plots present the first to third quartiles, maxima, and minima of absolute errors. Circular dots denote MAE.

5 Conclusion and Future Research

This study has proposed models for learning representations of stories by embedding character networks. Story2Vec-F/D reflect dynamic and static features, respectively, and Story2Vec-U combines the two approaches. Among the three, Story2Vec-D had the highest accuracy. Story2Vec-U improved the variance of the standalone cases but hindered the accuracy of Story2Vec-D. We should find a better method for integrating the dynamic and static approaches. This study also has the following research directions.

- Stories can be defined in various granularity levels, not only in a narrative work. Lee et al. [2020] partially solved this issue using the hierarchical representation learning, but still insufficient.
- Narrative multimedia have other physical features, excluding stories. We need a multi-modal representation learning method to cover both features.

Acknowledgments

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ICT Consilience Creative program (IITP-2019-2011-1-00783) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation). Also, this work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2020R1A2B5B01002207).

[1] https://github.com/O-JounLee/Story2Vec

[2] https://www.imdb.com/

[3] https://github.com/O-JounLee/CharNet-Extractor

[4] https://www.imsdb.com/

[5] http://recsys.cau.ac.kr:8084/movies/

References

[Bost *et al.*, 2019] Xavier Bost, Serigne Gueye, Vincent Labatut, Martha Larson, Georges Linarès, Damien Malinas, and Raphaël Roth. Remembering winter was coming. *Multimedia Tools and Applications*, 78(24):35373–35399, September 2019.

[Callaway and Lester, 2002] Charles B. Callaway and James C. Lester. Narrative prose generation. *Artificial Intelligence*, 139(2):213–252, August 2002.

[Jung *et al.*, 2017] Jai E. Jung, O-Joun Lee, Eun-Soon You, and Myoung-Hee Nam. A computational model of transmedia ecosystem for story-based contents. *Multimedia Tools and Applications*, 76(8):10371–10388, April 2017.

[Labatut and Bost, 2019] Vincent Labatut and Xavier Bost. Extraction and analysis of fictional character networks: A survey. *ACM Computing Surveys*, 52(5):1–40, September 2019.

[Le and Mikolov, 2014] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents. In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31th International Conference on Machine Learning (ICML 2014)*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 1188–1196, Beijing, China, June 2014. JMLR.org.

[Lee and Jung, 2018] O-Joun Lee and Jason J. Jung. Explainable movie recommendation systems by using story-based similarity. In Alan Said and Takanori Komatsu, editors, *Joint Proceedings of the ACM IUI 2018 Workshops co-located with the 23rd ACM Conference on Intelligent User Interfaces (ACM IUI 2018)*, volume 2068 of *CEUR Workshop Proceedings*, Tokyo, Japan, March 2018. CEUR-WS.org.

[Lee and Jung, 2019a] O-Joun Lee and Jason J. Jung. Integrating character networks for extracting narratives from multimodal data. *Information Processing and Management*, 56(5):1894–1923, September 2019.

[Lee and Jung, 2019b] O-Joun Lee and Jason J. Jung. Modeling affective character network for story analytics. *Future Generation Computer Systems*, 92:458–478, March 2019.

[Lee and Jung, 2020] O-Joun Lee and Jason J. Jung. Story embedding: Learning distributed representations of stories based on character networks. *Artificial Intelligence*, 281:103235, April 2020.

[Lee *et al.*, 2020] O-Joun Lee, Jason J. Jung, and Jin-Taek Kim. Learning hierarchical representations of stories by using multi-layered structures in narrative multimedia. *Sensors*, 20(7):1978, April 2020.

[Mani, 2012] Inderjeet Mani. *Computational Modeling of Narrative*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers, 2012.

[Mani, 2016] Inderjeet Mani. Animation motion in narrativeml. In Ben Miller, Antonio Lieto, Rémi Ronfard, Stephen G. Ware, and Mark A. Finlayson, editors, *Proceedings of the 7th Workshop on Computational Models of Narrative (CMN 2016)*, volume 53 of *OASICS*, pages 3:1–3:16, Kraków, Poland, July 2016. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

[McKee, 1997] Robert McKee. *Story: Substance, Structure, Style and the Principles of Screenwriting*. HarperCollins, New York, NY, USA, November 1997.

[Mikolov *et al.*, 2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26: Proceedings of 27th Annual Conference on Neural Information Processing Systems (NIPS 2013)*, pages 3111–3119, Lake Tahoe, Nevada, US, December 2013. Curran Associates, Inc.

[Narayanan *et al.*, 2016] Annamalai Narayanan, Mahinthan Chandramohan, Lihui Chen, Yang Liu, and Santhoshkumar Saminathan. subgraph2vec: Learning distributed representations of rooted sub-graphs from large graphs. arXiv preprint: 1606.08928, June 2016.

[Park *et al.*, 2012] Seung-Bo Park, Kyeong-Jin Oh, and GeunSik Jo. Social network analysis in a movie using character-net. *Multimedia Tools and Applications*, 59(2):601–627, January 2012.

[Purdy and Riedl, 2016] Christopher Purdy and Mark O. Riedl. Reading between the lines: Using plot graphs to draw inferences from stories. In Frank Nack and Andrew S. Gordon, editors, *Proceedings of the 9th International Conference on Interactive Digital Storytelling (ICIDS 2016)*, volume 10045 of *Lecture Notes in Computer Science*, pages 197–208, Los Angeles, CA, US, November 2016. Springer.

[Shervashidze *et al.*, 2011] Nino Shervashidze, Pascal Schweitzer, Erik Jan van Leeuwen, Kurt Mehlhorn, and Karsten M. Borgwardt. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12:2539–2561, September 2011.

[Tran and Jung, 2015] Quang Dieu Tran and Jai E. Jung. Cocharnet: Extracting social networks using character cooccurrence in movies. *Journal of Universal Computer Science*, 21(6):796–815, June 2015.

[Tran *et al.*, 2017] Quang Dieu Tran, Dosam Hwang, O-Joun Lee, and Jai E. Jung. Exploiting character networks for movie summarization. *Multimedia Tools and Applications*, 76(8):10357–10369, April 2017.

[Weng *et al.*, 2009] Chung-Yi Weng, Wei-Ta Chu, and Ja-Ling Wu. RoleNet: Movie analysis from the perspective of social networks. *IEEE Transactions on Multimedia*, 11(2):256–271, February 2009.