Movie Recommendation using Metadata based Word2Vec Algorithm

Yeo Chan Yoon
SW/Content Research Laboratory
Electronics and Telecommunications Research Institute
Daejeon, Korea
ycyoon@etri.re.kr

Jun Woo Lee
SW/Content Research Laboratory
Electronics and Telecommunications Research Institute
Daejeon, Korea
leejw@etri.re.kr

Abstract— **Nowadays, recommending preferable item among huge number of item is essential on online market. Many content platforms, such as YouTube and Amazon, use recommendation techniques to recommend items. Therefore, various techniques have been studied to recommend desirable item for each users. In this paper, we propose a method for effectively recommending preferable movies for each users by using community user's movie rating information and movie metadata information with deep learning technology. The proposed method shows 0.165 performance improvement based on Rcall@100 as compared with the basline method.**

Keywords—recommendation of item, deep neural network, movie recommendation

I. Introduction

With the widespread use of the Internet, it has become very easy to use and purchase multimedia content such as music, movies, games, and offline products from online. However, the number of products provided from online is enormous; Finding a satisfactory item among them is very difficult. Therefore, in order to increase sales, the market operator should be able to appropriately recommend desirable products for users. As a result, global companies such as Google, Amazon, and MS have been developing and researching recommend system that automatically analyzes user preferences, and these attempts were successful.

Amazon [1] proposed a collaborative filtering method to recommend products by analyzing the community's common interests using usage history of users in the community. However, since the recommendation is based on the user's history, it is difficult to recommend a new item which does not have a usage history and recommend for a new user who does not have a purchase history. To solve this problem, Kula [2] and Sahebi [3] attempted to use metadata or cross domain information.

On the other hand, various deep learning based attempts have been studied to improve the performance of the recommendation. Covington [6], Koenigstein [7], and Van den Oord [8] have shown high performance by applying deep-learning-based recommendation techniques to YouTube, XBOX and Sportify platforms, respectively. Barkan et al [9] applied the word2vec algorithm, which is one of the widely known deep learning methods, to the recommendation service. They trained vector of each item by applying the algorithm with item usage history.

In this paper, we propose a method to recommend movie using metadata information based on Word2Vec algorithm. Using metadata information, the proposed method could effectively recommend recently released item which have been rarely used. This method is also excellent for finding similar item by comparing embedding of each item.

II. Related Works

A. Recommendation methods based on Collborative filtering

Linden [1] et al proposed a well-known collaborative filtering method that recommend items using user’s item usage pattern and purchase history among community users. This method showed very good performance and has been used in many online markets such as Amazon. In this method, only the user usage history is used; To improve performance, various methods using various resources such as metadata or other domain information has been studied.

Kula [2] et al used metadata to represent user and item as vector. The recommendation is created by using the user and item vector. To learn item vector, each property of item is embedded as vectors. The summation of each property vector is then used as item vector. User vectors are learned in a similar way using user-preferred metadata. However, in this case, since the metadata embedding values are simply added, the weight of metadata cannot be considered.

Sahebi [3] et al proposed a method of recommending item using cross domain information. He used information of common users from different community to exploit cross domain information.

Paterek[4] et al used Singular Value Decomposition (SVD) to reduce complexity of collaborative filtering method. As the number of items increases, the amount of computation of the collaborative filtering method becomes very large. By applying SVD, the dimension is reduced, and the collaborative filtering algorithm can be feasible. The performance is also improved

Wang [5] et al applied the collaborative filtering method to recommend scientific papers. The approach combines the advantage of traditional collaborative filtering and probabilistic

978-1-5386-4710-3/18/$31.00 ©2018 IEEE

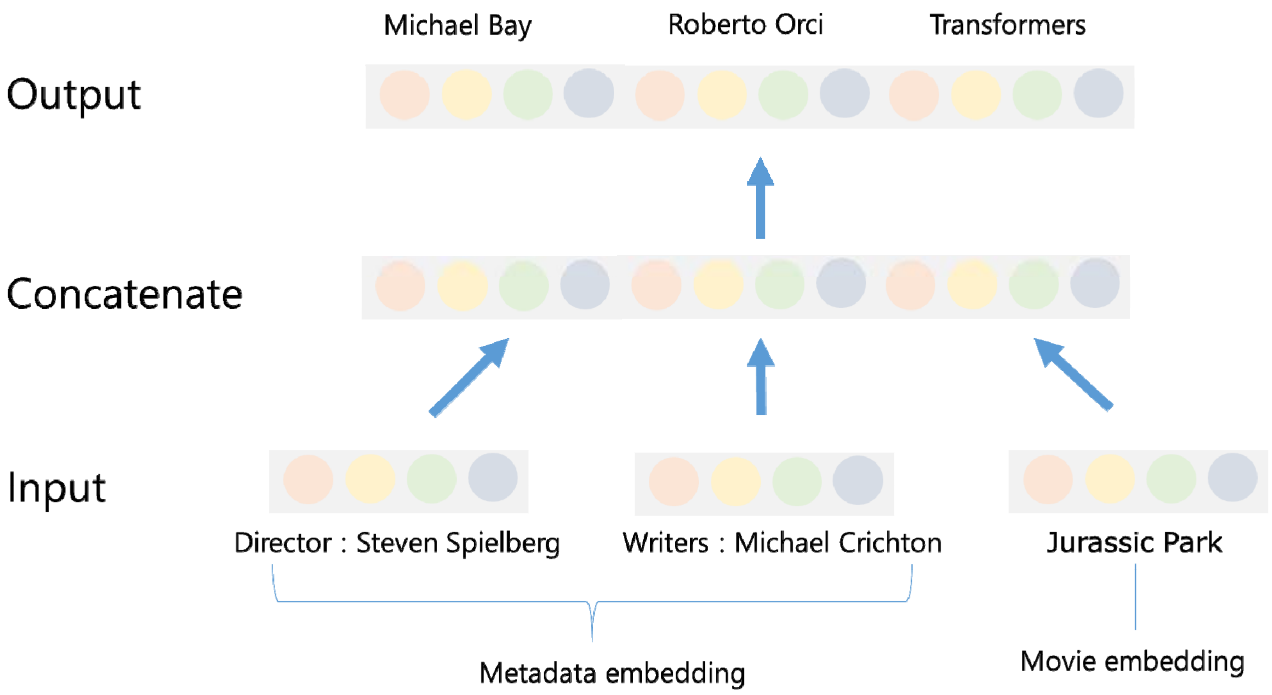

Fig. 1. Structure of proposed model

topic modeling. It provides an interpretable latent structure for users and items, and can form recommendations about both existing and newly published articles.

B. *Content based methods for recommendation*

Collaborative filtering method is very effective for recommending item. However, it is difficult to recommend new items to users or recommend item for new user; because there is no usage history for them.

Chow [6] et al proposed a method for recommending mobile game using content based approach. They built similarity graph by analyzing common meta data between games. To build graph, they brought concept from well-known Page Rank algorithm.

Debnath [7] et al uses social network graph to consider the importance of each attribute of item. They also proposed a hybrid approach of collaborative filtering and content based recommendation to improve performance.

Van den Oord [8] et al proposed a content based recommendation method which represent music as vector to find similar song with deep learning technique. With this method, recently released music which have no usage history could be recommended.

The content-based recommendation method is suitable for recommending a new item, but it is not effective for recommending an item to a new user. Elkahky[13] et al solved this problem by using web browsing and searching history. They learned preferable topic of user by analyzing search query and clicked URL, and it is used to recommend item to user.

C. *Deep learning methods for recommendation*

Covington [6] et al proposed deep learning based method of recommending videos suitable for each user for YouTube service. They used two deep learning network to generate candidates and rank the candidates. In the first network, candidates of video are generated among tens of millions of videos. Then the second deep-learning network is exploited to rank the videos to recommend preferable videos for each user.

Barkan [9] et al also applied Word2Vec, one of the widely known deep-learning techniques, to recommend music and games. In order to recommend games, they used the games included in the same purchase list as input and output. For the music recommendation, they used the set of singers played by the same user as input and output to learn the vector of the singers and then used them for recommendation.

Cheng [10] et al suggested a method of combining linear model and deep learning model for the mobile application recommendation service. Features such as age of application, user demographics are embedded for deep learning model and are also exploited for linear model. This method showed higher performance than using linear model or deep learning model separately.

III. PROPOSED METHOD

In this paper, we propose a simple and effective item recommendation model based on Word2Vec algorithm with metadata information.

The Word2vec algorithm is one of the deep learning methods widely used in various tasks related to document analysis such as document clustering [11] and recommendation [9]. In this paper, we propose new Word2Vec algorithm using metadata information and show high performance. Figure 1

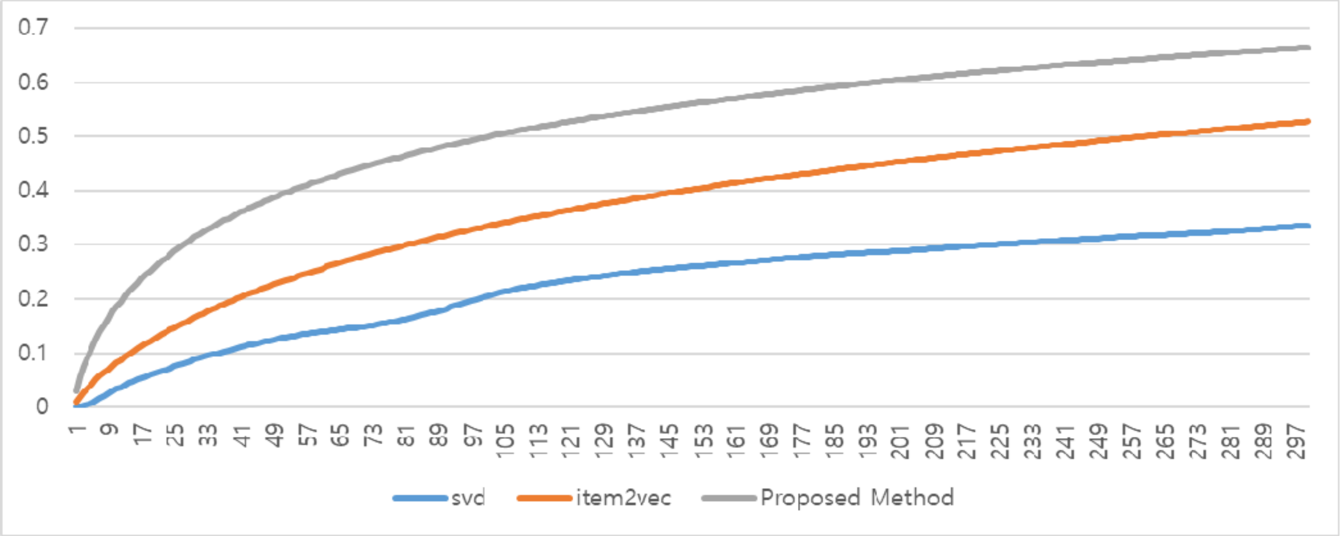

Fig. 2. Structure of proposed model

shows the structure of proposed model. The proposed method exploits various metadata of movie, such as movie director, actor, production year, production cost, movie tag and so on. Values of these metadata are embedded as vector and are used as input and output of proposed Word2Vec network. Movie embedding is also used as input and output with meta data embedding.

A. *Training embedding of each movie*

The proposed Word2Vec algorithm uses movie embedding and metadata embedding as input. The input and output pairs can be obtained from the user's movie purchase and viewing history data. Movies which is rated as highest score from a same user are set to a positive movie set, and randomly selected movie pairs from the set are used as input and output of the Word2Vec Network. For example, if a user watches four movies A, B, C, D and assigns five, two, five and five rating to each movie, the pairs of movie (A and C), (A and D) and (C and D) are set to positive examples which are used to train the Word2Vec network.

The movie and metadata vectors given as inputs are initialized with pretrained embedding using the Word2vec algorithm. Two methods are used to obtain the pretrained metadata vector. First, we use the same method as describe above. That is, for the movies rated highest score by the same user, pairs of metadata included in the movies are used as input and output of the Word2Vec network. Second, if there are different metadata values included in a same movie, the pair of the metadata values are used as input and output data. For example, if a movie contains two tags (#Smooth, #Romantic), it is used as training example. When the Word2Vec algorithm is applied to the training set, vector for each metadata value can be learned. The pretrained item vector is also learned using a similar method.

The Input embedding of Figure 1 is generated by concatenating those of pretrained metadata embedding and movie embedding. With the input embedding, we train the Word2Vec Network and finally could learn the embedding of movies that reflect the characteristics of metadata.

B. *Prediction of User Vector*

The user vector could be obtained by summing up the movie embedding watched by the user. The movie embedding is multiplied by weight that reflects the user's preference. The weight of each movie is obtained using the watching time of the movie with the following equation.

$$V_u^{user} = \sum_i watch\_time(i) \times V_i^{item} \qquad (1)$$

In the above formula, V_u^{user} indicates the user u's latent vector, V_i^{item} indicates the vector of the i-th movie and watch_time (i) indicates the viewing time of the user u of the i-th movie.

C. *Generation of recommend list*

In this paper, we use the inner product of the item vector and the user vector in order to rank the movies for each user [10].

$$\hat{r}_{ui} = \mu + b_u + b_i + V_u^{user^T} V_i^{item} \qquad (2)$$

In the above formula, $\hat{r}_{ui}$ indicates the predicted rating for the movie i of user u, μ is the average rating of the entire movie, b_u indicates the average rating of the movie u, b_i is the average rating of the movie i, $V_u^{user^T} V_i^{item}$ indicates the inner product of user u and item i. The formula can be used to sort movies based on their predictive ratings and recommend movies that have high scores for a specific user.

IV. EVALUATION

We evaluate our method on movie data. The experimental results were measured using the Recall @ M measure. The Item2Vec method proposed by Barkan [7] and the SVD-based

TABLE I. EXAMPLE OF RECOMMENDATION OF MOVIE

Liked	Recommend
La Luna (2011) Mary Poppins (1964) Spirited Away (2001) 3 Idiots (2009) Man from Earth, The (2007) Moulin Rouge (2001) Midnight in Paris (2011) Zootopia (2016) Inside Out (2015) Nobody Knows (2004) Whiplash (2014) Inception (2010) Dogville (2003) Last Holiday (2006) Lee Daniels' The Butler (2013)	**Her (2013)** Gone Girl (2014) **12 Years a Slave (2013)** **The Martian (2015)** **Paperman (2012)** **Birdman: Or (The Unexpected Virtue of Ignorance) (2014)** **The Theory of Everything (2014)** Dallas Buyers Club (2013) Mad Max: Fury Road (2015) **Interstellar (2014)** **Intouchables (2011)** Kingsman: The Secret Service (2015) **About Time (2013)** **Spotlight (2015)**

method [4] which is frequently used for collaborative filtering are used as comparable methods.

A. Dataset

A well-known MovieLens[13] 10M data set was used for the experiment set. The MovieLens dataset consists of 10681 movies, and about 10 million movie ratings by 71567 users.

B. Experimental Result

Figure 2 shows the performance of the proposed method and the comparison method. The vertical axis shows the Recall@N and the horizontal axis shows the rank N. The proposed method is superior to that of SVD method and Item2Vec method. Based on Recall@100, the performance of the proposed method is 0.498, which is 0.165 superior to the item2vec performance of 0.333. This shows that learning Word2Vec using metadata is effective.

Table 1 shows an example of recommending a movie based on a user's favorite movie. A bolded movie represents a movie in which the actual user is satisfied with the recommendation result. As you can see in the example, we can confirm that the user is satisfied with the most of the recommendation result.

Table 2 shows the results of finding a similar movie using a movie vector. Using a cosine similarity, a movie with a smaller distance from the target vector was judged as a similar movie. For the action black buster movie Fast & Furious 6, we can see the recommend results include movies in similar genres; and for the romantic movie Begin Again, we can see the lyrical movies were recommended as results.

V. CONCLUSION

In this paper, we proposed a method using metadata to embed movie and user for movie recommendation service. The proposed method showed higher performance than the baseline method using only user usage data. We also could represent user as vector by summing up the user-preferred-movie's vector. The example shows that the proposed method is very effective in recommending movie.

TABLE II. EXAMPLE OF FINDING SIMILAR MOVIE

Target movie	Similar movies
Fast & Furious 6 (2013)	White House Down (2013) Olympus Has Fallen (2013) Parker (2013) Now You See Me (2013) Good Day to Die Hard, A (2013)
Begin Again (2013)	Before We Go (2014) Walk of Shame (2014) Chef (2014) The Fault in Our Stars (2014) Love, Rosie (2014)

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) [No. R0118-16-1005, Digital Content In-House R&D].

REFERENCES

[1] Linden, Greg, Brent Smith, and Jeremy York. "Amazon. com recommendations: Item-to-item collaborative filtering." IEEE Internet computing 7.1 (2003): 76-80.

[2] Kula, Maciej. "Metadata embeddings for user and item cold-start recommendations." arXiv preprint arXiv:1507.08439 (2015).

[3] Sahebi, Shaghayegh, and Peter Brusilovsky. "Cross-domain collaborative recommendation in a cold-start context: The impact of user profile size on the quality of recommendation." International Conference on User Modeling, Adaptation, and Personalization. Springer, Berlin, Heidelberg, 2013.

[4] Paterek, Arkadiusz. "Improving regularized singular value decomposition for collaborative filtering." Proceedings of KDD cup and workshop. Vol. 2007. 2007.

[5] Wang, Chong, and David M. Blei. "Collaborative topic modeling for recommending scientific articles." Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2011.

[6] Chow, Anthony, Min-Hui Nicole Foo, and Giuseppe Manai. "HybridRank: A Hybrid Content-Based Approach To Mobile Game Recommendations." CBRecSys@ RecSys. 2014.

[7] Debnath, Souvik, Niloy Ganguly, and Pabitra Mitra. "Feature weighting in content based recommendation system using social network analysis." Proceedings of the 17th international conference on World Wide Web. ACM, 2008.

[8] Elkahky, Ali Mamdouh, Yang Song, and Xiaodong He. "A multi-view deep learning approach for cross domain user modeling in recommendation systems." Proceedings of the 24th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2015.

[9] Covington, Paul, Jay Adams, and Emre Sargin. "Deep neural networks for youtube recommendations." Proceedings of the 10th ACM Conference on Recommender Systems. ACM, 2016.

K. Elissa, "Title of paper if known," unpublished.

[10] Koenigstein, Noam, et al. "The Xbox recommender system." Proceedings of the sixth ACM conference on Recommender systems. ACM, 2012.

[11] Van den Oord, Aaron, Sander Dieleman, and Benjamin Schrauwen. "Deep content-based music recommendation." Advances in neural information processing systems. 2013.

[12] Barkan, Oren, and Noam Koenigstein. "Item2vec: neural item embedding for collaborative filtering." Machine Learning for Signal Processing (MLSP), 2016 IEEE 26th International Workshop on. IEEE, 2016.

[13] Cheng, Heng-Tze, et al. "Wide & deep learning for recommender systems." Proceedings of the 1st Workshop on Deep Learning for Recommender Systems. ACM, 2016.

[14] Yoon, Yeo□Chan, et al. "Fine□Grained Mobile Application Clustering Model Using Retrofitted Document Embedding." ETRI Journal 39.4 (2017): 443-454.

[15] Koren, Yehuda. "Factorization meets the neighborhood: a multifaceted collaborative filtering model." Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2008

[16] Harper, F. Maxwell, and Joseph A. Konstan. "The movielens datasets: History and context." ACM Transactions on Interactive Intelligent Systems (TiiS) 5.4 (2016): 19.