Dynamic convolutional neural network for eliminating item sparse data on recommender system

Hanafi [a,b,1,*], Nanna Suryana [b,2], Abdul Samad Hasan Basari [b,3]

[a] Universitas Amikom Yogyakarta, Indonesia
[b] Univeristi Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia
[1] hanafi@amikom.ac.id; [2] nsuryana@utem.edu.my; [3] abdsamad@utem.edu.my
* corresponding author

ARTICLE INFO

Article history
Received October 17, 2018
Revised November 22, 2018
Accepted November 27, 2018
Available online November 28, 2018

Keywords
Recommender system
E-commerce
Convolutional
Deep learning
Collaborative filtering

ABSTRACT

Several efforts have been conducted to handle sparse product rating in e-commerce recommender system. One of them is the inclusion of texts such as product review, abstract, product description, and synopsis. Later, it converted to become rating value. The previous works have tried to extract these texts based on bag of word and word order. However, this approach was given misunderstanding of text description of products. This research proposes a novel Dynamic Convolutional Neural Network (DCNN) to improve meaning accuracy of product review on a collaborative filtering recommender system. DCNN was used to eliminate item sparse data on text product review while the accuracy level was measured by Root Mean Squared Error (RMSE). The result shows that DCNN has outperformed the other previous methods.

1. Introduction

E-commerce has changed the methods applied by many companies in business transactions. It has become not only an alternative but an imperative. These business organizations are faced with the challenges of finding the best way to conduct and develop their businesses in the digital economy. As a result of this, most of them have moved their total operations to web-based applications while some are settling subsection, then spinning them off as independent online business entities [1]. The need to provide buyers with relevant information about products has brought about the development of recommender system which is used in delivering this information to them through websites portal or mobile phones [2]. This system is an essential part of the e-commerce industry used in promoting sales and services on various online websites and mobile applications. For instance, Movies on Netflix watched 80 percent of movies watched on Netflix came from recommendations [3], and 60 percent of video clicks came from YouTube recommendation system [4]. In addition, according to Schafer [5], sales agents recommended by NetPerceptions system achieved higher average cross-sell value and higher success rate (60 percent and 50 percent respectively) than the ones based on experimental traditional techniques.

Recommender system can be divided into four types based on technical approach [6], [7]. The first, content-based, is the mechanism that involves product classification. It uses information retrieval to generate product recommendation. The second is the knowledge-based which is developed for specific recommendation. The particular character in this approach provides information rarely needed by individuals, for example, house, loan, insurance, car etc. The third is the demographic-based which provides recommendation based on demographic information. The fourth is the collaborative filtering which is founded on past behavior of the user. This method is the most successfully applied approach in larger e-commerce company because of its ability to provide recommendation characters that are product fit, provide relevant information, and accurate [8]. It adopts rating through the use of rating matrix to assess the similarities in the feedback of users in order to generate recommendation for products. However, the method is limited by the population of people, which is about 1%, that gives ratings on products. This problem is popularly called sparse data and when it is in extreme condition it is known as cold start. When this happens, the system cannot generate any recommendation.

One of the several efforts towards solving the problem is by extracting information from items. Many authors have come up with different approaches of extracting content feature, for instance, den Oord et. al. [9] developed a music recommendation using convolutional neural network (CNN) to classify music based on acoustic type for the purpose of tackling unpopular music (long tail). Another approach introduced by Jaradat [10] is about extracting deep color feature to recommend fashion using color classifier.

Collaborative filtering is the most popular recommendation engine applied in the e-commerce business because of its ability to provide relevant, accurate information and give surprise product information to users [11]. However, it is limited by sparse data caused by the inability of users to give ratings [12]. Several attempts have been made to reduce sparse data for it to be more accurate. The use of auxiliary information has proven to be effective in improving this accuracy. Most of them have been successful in increasing robustness in sparse data such as audio features in music [9], [13], color features on online fashion shop [10], and documents in online news [14]. According to Chen et. al. [15], valuable information can be extracted from reviews for the purpose of enhancing product recommendations. This can be done through several methods despite the fact that the raw review information is in unstructured textual form that cannot be easily understood by the system. Qu et. al.[16] proposed the use of bag of word method to analyze customer review on movies so as to increase its rating value. Using this method can provide optimal results [17], [18], despite the fact that it has difficulty in capturing contextual meaning of texts. For example, it cannot detect surrounding words like "not good", "not bad", "not very well" and this the reason its prediction of rating results is sometimes inaccurate. Therefore, there is a challenge of the method to use in increasing the level of accuracy when convolutional neural network such as sentiment analysis is applied [19].

Enhancement of text has been applied by many researchers in building recommendations, for example, Wang [20] use classical method of bag of word from users to predict rating value but this method still has its own limitations such as misunderstanding in detecting the meaning of text or sentences. This has led to inaccuracy in the extraction of sentences from product reviews. This can be observed from IMDB movie (www.imdb.com) review portal which include several text resources such as product description, user opinion, user comment, and testimony with the aim of improving accuracy. Feedback provided by customers can be said to be the best form of product rating. According to marketing theory, majority of customers will not buy a new product without experience. They will consider an opinion or testimony from others that have had deep experience about the product. Other previous approaches, such as categorical Bag of Word (BOW), Collaborative Topic Regression [21], and Latent Dirichlet Allocation (LDA) [17], [18] were succeed to generate item latent factor from text sentence document. However, they failed to catch the suitable view of contextual meaning point.

The aim of this study was enhancing DCNN as subclass of Convolutional Neural Network (CNN) to capture contextual meaning of the consumer product review to produce item latent factor which was combining with probabilistic matrix factorization. The evaluation results by Root Mean Squared Error (RMSE) showed that the proposed approach was outperforming the previous works.

2. Method

In handling the problem of minimum contextual meaning of text sentences convolutional neural network (CNN) is considered. This is a state of the art machine learning platform that can outperform others applications, such as information retrieval [22], [23], computer vision [24], and natural language processing (NLP) [25]–[27]. On sentiment analysis, Kim [27] proposed that the approach is effective in sentence classification. Table 1 shows various approaches that have been significantly relevant in different researches in recent years.

Table 1. Comparison of proposed Dynamic-CNN with product rating prediction approaches

No	Method	Type of auxiliary Information	Domain Application	Disadvantage issue	Ref.
1	Convolutional Neural Network	Music	Music	Not compatible with text sentences content	[9], [28]
2	Convolutional Neural Network	Colour	Fashion	Not compatible with text sentences content	[10]
3	Convolutional Neural Network	Sentiment analysis	Sentiment analysis	Not compatible to predict a rating	[27]
4	LDA	Text sentences	Movie, book	Ignore contextually	[17], [18]
5	CTR	Text sentences	Movie	Ignore contextually	[21]
6	Dynamic-CNN	Text sentences	Movie, video		Our approach

After LDA model approach [17], [18], collaborative topic regression [21] which was categorized into bag of word and word order was also applied to have deep contextual meaning but this method has been observed to the most shallow of all the sentence models introduced. Kim et. al. [29] proposed a model adopted from [27] to predict rating based on convolutional neural network aimed at capturing all contextual meaning in text sentences document. This model combines both convolutional neural network and probabilistic matrix factorization and it has the ability of outperforming others based on latent item factor. However, its limitation includes high computation perspective and longer time of computation. On the other hand, it has less scalability point of view.

A variant of convolutional neural network has been proposed [26], and it was dubbed Dynamic Convolutional Neural Network (DCNN) for sentences classification. In a language processing, the system ability to represent accurately meaning is important. Therefore, Dynamic K-Max Pooling was implemented in the semantic modeling to handle varying length input sentences, and to generate a graph feature over sentence. It adequate to capture long and short-range relations. The network does not depend on a parse tree and is easily implementing to any language. In solving the problem of sparse data identified, an hybrid of DCNN and probabilistic technique is proposed to improve the result of previous research that adopted convolutional neural network approach to extract content feature based on text sentence document [29].

2.1. Text Sentences Description and Probabilistic Matrix Factorization

The proposed Dynamic Convolutional Neural Network and Probabilistic Matrix Factorization (DCNN-MF) (Fig. 1) involves three steps: 1) Exploitation of the probabilistic approach, and implementation of critical factor in bridging Probabilistic Matrix Factorization and DCNN to calculate both rating and item text sentences description; 2) Definition of a structure that produces text sentences latent model by calculating items text sentences description documents; and 3) Optimization of the latent variables of DCNN.

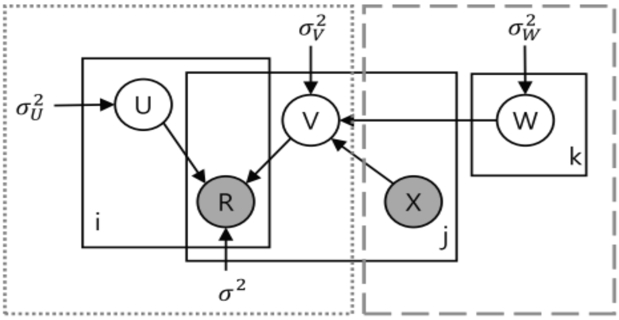

Fig. 1. Left side with smooth dash is DCNN, and right side with a Coarse dash is the Probabilistic approach [29]

2.2. Dynamic Convolutional Neural Network Adopted Sentiment Analysis

The objective of the DCNN architecture is to obtain text sentences documents latent vectors from documents items, then it is utilized to create the latent factors items among epsilon variables. Fig. 2 reveals that the DCNN architecture contains four layers: 1) embedding layer, 2) convolutional layer, 3) pooling layer, 4) output layer.

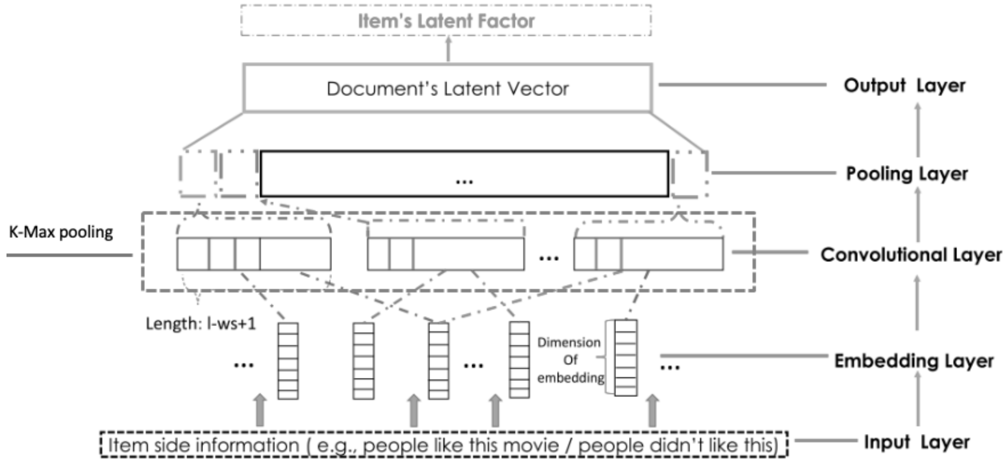

Fig. 2. Convolutional for a contextual recommendation (modified from [29])

Input Layer. The input layer is the primary information source that comes from the feature content, in this case, the product description. This is the representation of a product through the use of sentences. Based on different researches, it is different from music or fashion products recommendation where the pattern is clear. Products description undergoes many modifications at the preprocessing stage, such as the removal of punctuation marks and elimination of conjunctions.

Embedding Layer. The function of this layer is to transform a raw document into a numeric matrix according to the length of the words, which will be conducted as a convolution operation in the next segment. For instance, if number of words in a document is l, then a vector of each word can be concatenated into a matrix in accordance with the sequence. The vectors are initialized with a pre-trained word embedding model as in (1).

$$document = \begin{bmatrix} w_{11} & \cdots & w_{1i} & \cdots & w_{1l} \\ w_{21} & \cdots & w_{2i} & \cdots & w_{2l} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ w_{p1} & \cdots & w_{pi} & \cdots & w_{pl} \end{bmatrix} \quad (1)$$

where p stands for the dimension of word embedding and $w_{[1:p,i]}$ represents raw word i in the document.

Convolution Layer. This involves extracting contextual features from a sentence expression. The sentence conveyed by consumers on a movie was considered. Convolution architecture was applied to dispose the documents. The detail formulation is as shown in (2). A contextual feature $c_i^j \in R$ is extracted by jth that shared weight $w_c^j \in \mathbb{R}^{p \times ws}$, where ws as window size indicates the surrounding words number [29].

$$c_i^j = f(w_c^j * D\big(:, i: (i + ws - 1)\big) + b_c^j \quad (2)$$

where $b_c^j \in R$ is a bias for w_c^j, * is an operator of convolution, and $f(x)$ is a nonlinear activation function. Among these functions are sigmoid, tanh, and rectified linear unit (ReLU). ReLU was used to avoid the vanishing gradient issue that occurs minimum optimization convergence, and may evidence to minimize local value. Then, $c^j \in \mathbb{R}^{l-ws+1}$ as contextual feature vector of a document with $w_c^{\,j}$ can be constructed by:

$$c^j = \left[c_1^j, c_2^j, c_3^j, c_4^j, c_5^j, \ldots, c_{l-ws+1}^j\right] \tag{3}$$

On other sides, one type of contextual feature is captured by one shared weight captures while multiple types of contextual features were captured by multiple shared weights. This conditions occur possibility to produce contextual feature vectors as much as the number n_c of w_c (i.e., $w_c^{\,j}$ where $j = 1,2,\ldots,n_c$).

Pooling Layer. This layer did not only extract instance features from the convolution layer but also dealt with documents variable lengths (via pooling operation) that constructed a fixed-length feature vector. After the convolution layer was created, a document was represented by n_c contextual feature vectors. Each contextual feature vector has variable length (i.e., $l - ws + 1$ contextual feature). However, such representation causes two problems: 1) there are too many contextual c_i while most contextual features might not help to increase performance of the model; 2) varied length of the contextual feature vector caused problem to establish the following layers. Therefore, max-pooling was utilized to reduce the document representation into an n_c fixed-length vector by generating only the maximum contextual feature from each contextual feature vector as in (4).

$$df = [\max(c^1), \max(c^2), \ldots, \max(c^j), \ldots, \max(c^{n_c})] \tag{4}$$

where c^j is a contextual feature vector of length $l - ws + 1$ extracted by jth shared weight $w_c^{\,j}$.

Existing approach connected to the model sentences was modified by utilizing a convolutional engineering that exchanges wide convolutional layers with dynamic pooling layers given by dynamic k-max pooling. In the system, the width of a component maps a middle layer changes relying upon the length of the info sentence and this gave use the Dynamic Convolutional Neural Network as shown in Fig. 3 [26]. The system includes 1) Wide Convolution, 2) k-Max Pooling 3) Dynamic k-Max Pooling 4) Non-linear Feature Function. 5) Multiple Feature Maps.

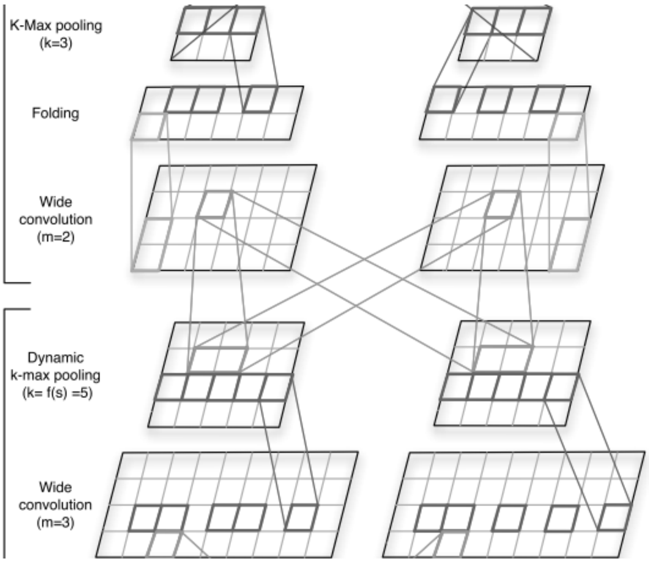

Fig. 3. Dynamic Convolutional Concept [26]

Output Layer. Generally, at this stage, high-level features obtained from the previous layer should be converted for a specific task. Thus, df was projected on a k-dimensional space of users and items' latent factors for the recommendation task, which completely produced a document's latent vector by using conventional nonlinear projection.

$$\mathrm{s} = \tanh(\mathrm{W}_{f2}\{\tanh(\mathrm{W}_{f1} d_f + \mathrm{b}_{f1})\} + \mathrm{b}_{f2}) \tag{5}$$

where $\mathrm{W}_{f1} \in \mathbb{R}^{f \times n_c}$, $\mathrm{W}_{f2} \in \mathbb{R}^{k \times f}$ are projection matrices, and $\mathrm{b}_{f1} \in \mathbb{R}^{f}$, $\mathrm{b}_{f2} \in \mathbb{R}^{k}$ is a bias vector for $\mathrm{W}_{f1}, \mathrm{W}_{f2}$ with $s \in \mathbb{R}^{k}$. Eventually, by the mentioned processes, the CNN architecture became a function that takes a raw document as input, and returns a latent vector of each document as output of (6).

$$s_j = dcnn(W, Y_j) \tag{6}$$

where W indicates all the weight and variables of bias to prevent clutter, Y_j denotes a raw document of item j, and s_j denotes a document's latent vector of item j.

2.3. Optimization Method

For the purpose of optimizing some variables such as item latent models, user latent models, maximum a posteriori (MAP) estimation, and DCNN weight and bias variables was applied as follows:

$$\max_{U,V,W}, p(U,V,W|R,X,\delta^2,\delta_U^2,\delta_V^2,\delta_W^2) = \max_{U,V,W}[p(R|U,V,\delta^2)p(U|,\delta_U^2)p(V|W,X,\delta_V^2)p(W|\delta_W^2)] \tag{7}$$

When a negative logarithm is applied on the equation, it can be reformulated as in (8).

$$\ell(U,V,W) = \prod_i^N \prod_j^M \frac{I_{ij}}{2}(r_{ij} - u_i^T v_j)2 + \frac{\lambda U}{2}\sum_i^N \|u_i\|_2 +$$

$$\frac{\lambda u}{2}\sum_i^N \|u_1\|_2 + \frac{\lambda v}{2}\sum_j^M \left\|v_j - dcnn(W,X_j)\right\|_2 + \frac{\lambda W}{2}\sum_k^{|w_k|}\|W_k\|_2 \tag{8}$$

where λU is σ^2/σ_U^2, λV is σ^2/σ_V^2, and λW is σ^2/σ_W^2.

2.4. Probabilistic Matrix Factorization

The main idea of this approach is similar to Kim et. al. [29] but it was modified with the aim of increasing the understanding of contextual meaning and level of scalability. In doing this, integration of probabilistic matrix factorization (PMF) approach with the proposed dynamic convolutional neural network (DCNN) was considered. Assuming there are N users and M items, then learn rating was supposed to be represented by $R \in \mathbb{R}^{N \times M}$ matrix. Therefore, the objective was to find the user and item latent models ($U \in \mathbb{R}^{N \times M}$and $V \in \mathbb{R}^{N \times M}$) whose product ($U^T V$) reconfigure the rating R matrix. According to the probabilistic approach, the distribution of the situation through observe rating can be denoted by:

$$p(R|U,V,\sigma^2) = \prod_i^N \prod_j^M N\left(r_{ij}|u_i^T v_j, \sigma^2\right)^{I_{ij}} \tag{9}$$

where $N(x|\mu,\sigma^2)$ is the probability density function of the Gaussian normal distribution with mean μ and variance σ^2, and I_{ij} is an indicator function. To get the user latent model, conventional priori, a zero-mean spherical Gaussian prior was used on the model with variance σ_U^2.

$$p(U|\sigma_U^2) = \prod_i^N N(u_i|0,\sigma_U^2 I) \tag{10}$$

This is different from the tradition method such that there was consideration of three variables, 1) internal weight W in the approach; 2) X_j representing the document of item j; and 3) Epsilon variable as Gaussian noise. Considering the optimization of item latent model in predicting rating, then the equation (11) can be used to the generate final item latent model.

$$v_j = dcnn\left(W, X_j\right) + \in j \tag{11}$$

$$\in j \sim N(0, \sigma_v^2 I) \tag{12}$$

where weight w_k in W, zero-mean spherical Gaussian prior was assumed as in (13).

$$p(W|\sigma_w^2) = \prod_k N(w_k|0, \sigma_w^2) \tag{13}$$

where, the distribution item latent model is denoted by:

$$p(V|W, X, \sigma_v^2) = \prod_j^M N(v_j|\mathrm{dcnn}(\mathrm{W}, \mathrm{X}_j), \sigma_v^2 I) \tag{14}$$

where X represents a set of text sentences description document of items. A description of product latent vector obtained by the DCNN model was applied to the mean of Gaussian distribution noise of the product as employed as the variance of Gaussian distribution to bridge the gap between CNN and PMF in order to utilize both text sentences description and rating.

2.5. Measure Metric

Root Mean Squared Error (RMSE) was frequently used to measure the differences between sample and population values predicted by a model or an estimator and the values that were actually observed [30]. It represents the standard deviation sample of the differences between predicted values and observed values. It is the most popular metric used in evaluating accuracy of rating prediction. The RMSE between the predicted and actual rating is given by:

$$RMSE = \sqrt{\frac{\sum_{i,j}^{N,M}(r_{ij} - \hat{r}_{ij})^2}{\#rating}} \tag{15}$$

2.6. Experiment Setting

This research was focused on developing a method to handle sparse data in items rating observed through the application of collaborative filtering recommender system. According to previous studies, the use of matrix factorization has not been able to effectively address this issue to the extent that the mistakes are most time seen in the results [31]. Therefore, this is a categorical lab scale research involving public datasets. MovieLens dataset was considered as the convince dataset in accordance with different researches carried out in the same field [32]. The second datasets involved include specific customer opinions on Amazon videos.

2.6.1. Device and library tools

In carrying out this experiment, several tools including software and hardware were used (Table 2). These include Python with several libraries such as TensorFlow for deep learning implementation and GeForce GTX 1001 for running convolutional neural networks supported by Quad Core Xeon 2.4 GHz processors.

Table 2. Devices and Library

No	Device/tools/library	Specification
1	Processor	Intel Xeon Quad core, 2.4 GHz
2	Memory	16 Gb
3	GPU	GeForce GTX 1001
4	Tensor Flow	Deep learning tools
5	Keras	Deep learning tools
6	Anaconda	Web interface
7	Python	Tool programming
8	Scikit-learn	Handle sparse module
9	Surface	Recsys data analytic module
10	NumPy	Learning library

2.6.2. Dataset

Two real-world datasets, MovieLens [32] and Amazon Information Video (AIV), were applied to show the robustness of the approach regarding rating prediction. These datasets contained consumer' explicit ratings for products on a scale of 1 to 5. Amazon dataset contained opinion for movie products as item description documents. Because MovieLens does not include item description documents, documents use corresponding items from IMDB server was generated. In accordance to Kim [27] and [21], a preprocessed description document was conducted for all the datasets as follows: 1) set the data with maximum length of raw documents to 300; 2) remove stop words; 3) calculate TF-IDF score for each word; 4) remove corpus specific stop words that have frequency higher than 0.5; 5) Choose top 8000 distinct words as a vocabulary; and 6) remove all non-vocabulary words from raw documents. Consequently, the average numbers of words per document were 97.09 on MovieLens-1m (ML-1m), 92.05 on MovieLens-10m (ML-10m) and 91.50 on Amazon Instant Video (AIV). Items that have no description documents in the dataset especially in Amazon dataset and user’s data that have just only less than 3 ratings were considered for removal. The resulting statistics of each data shows that three datasets have different characteristics on Table 3. Finally, even though some users have been removed during preprocessing, Amazon dataset still has more sparse data compared to the others.

Table 3. Datasets characteristics

Datasets	#users	#movie	#rating	sparse	Additional information	Release
ML-100k	943	1.682	100,000	6.3%	demographic	4/1998
ML-1M	6.040	3.900	1.000.209	4.2%		2/2003
ML-10M	71.567	10.681	10.000.054	1.3%	95.580 #tags	1/2009
ML-20M	138.492	27.278	20.000.263	0.52%	465.564#tags	10/2007
AIV	29.757	15.149	135.188	0.030%		

3. Results and Discussion

This experiment considered the ability of DCCN model to capture contextual performance. It was applied in several sentences with similar words but with different contextual meaning. The result of the experiment is as shown in Table 4. The first sentence used the verb word "love" and it has the highest share weight score of 0.0431. In the last sentence, “love” was used as a noun and it has a share weight score of 0.0408. It can be concluded that DCNN has the ability of catching contextual meaning of text sentence in document and even distinguish smooth contextual diversity of similar words through their shared weight scores.

Table 4. Phrase test

Phrase test weigh	Score	Phrase test weigh	Score
Romeo love Juliet	0.0431	Romeo believe Juliet	0.0398
Romeo missed Juliet	0.0381	Juliet believe her love	0.0408

Based on the results shown in Fig. 4 and Table 5, the researchers demonstrated that at the beginning of the training process, the results of evaluation errors with RMSE were very high, but after the next iteration, the r value decreases significantly until it approached 30-50 levels where the value of the error reached a minimum of 0.8606681400. This means that the least error is reached at the 50th iteration. The training data process was automatically stopped when it reached the iteration level with the best performance in accuracy as shown in Fig. 4.

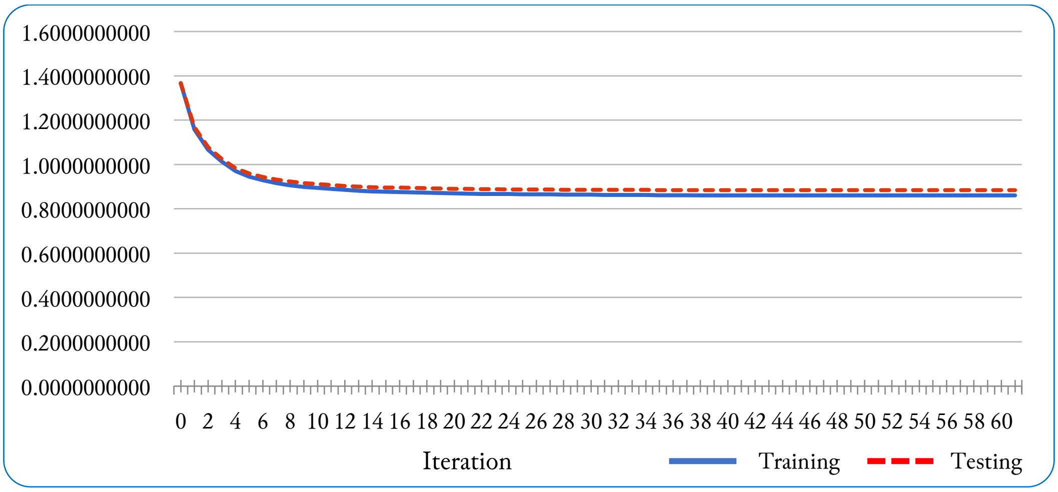

Fig. 4. Comparison of RMSE training and testing

The results of the iterations that produce a significant result on the level of accuracy are as presented in Table 5.

Table 5. The sample result of RMSE with loss, time, and convergence

Iteration	loss	Time (s)	Convergence	Training	Testing
25	-425,150.7780200000	15,317.5830000000	0.0102520000	0.8657584780	0.8874200000
26	-421,253.7979800000	15,377.0910000000	0.0091660000	0.8652984400	0.8870500000
27	-417,590.9481300000	15,431.1590000000	0.0086950000	0.8648384040	0.8867300000
28	-414,121.9125500000	15,362.2370000000	0.0083070000	0.8642983660	0.8863800000
29	-410,903.1628100000	15,419.9840000000	0.0077720000	0.8637583310	0.8860700000
30	-408,008.0179300000	15,461.0030000000	0.0070460000	0.8633483100	0.8858600000
31	-405,164.8094100000	15,381.8280000000	0.0069690000	0.8628982790	0.8855800000
32	-402,464.3062600000	15,884.8680000000	0.0066650000	0.8624482480	0.8853000000
33	-400,048.5160300000	15,445.6540000000	0.0060020000	0.8620982240	0.8850800000
34	-397,844.5245400000	15,389.2030000000	0.0055090000	0.8619082150	0.8850100000
35	-395,551.9407500000	15,515.5340000000	0.0057630000	0.8615481910	0.8847900000
36	-393,582.3736500000	15,478.9330000000	0.0049790000	0.8612881770	0.8846600000
37	-391,618.9421900000	15,402.3890000000	0.0049890000	0.8610081620	0.8845300000
38	-389,571.3054600000	15,453.6220000000	0.0047180000	0.8606681400	0.8843300000
39	-389,441.3044600000	15,451.6160000000	0.0047680000	0.8606682100	0.8842100000
40	-389,351.3064600000	15,433.6210000000	0.0042180000	0.8606683200	0.8841300000

Table 6 shows the average RMSE of Probabilistic Matrix Factorization (PMF), Collaborative Topic Regression (CTR), Collaborative Deep Learning (CDL) and the approach with various ratios for testing and training data on four datasets. First, it can be observed that ConMF approach model has the overall best performance on item side information based on product description documents. This implies that incorporating side information is effective. The proposed approach, DCNN, outperforms other previous ones founded on traditional text sentiment analysis using bag of word and word order (CTR). It shows that increasing deeper understanding of text sentences description would increase the level of accuracy. It was discovered that composition ratio of training datasets has impact on accuracy such that it constantly increases with ratio training and testing composition ration. The application of DCNN is more effective and efficient than the use of traditional methods such as CDL and CTR.

Table 6. Comparison RMSE on ML-10M with existing approach

Technical Approach	Ratio of Training Set (ML-10M)						
	20%	*30%*	*40%*	*50%*	*60%*	*70%*	*80%*
PMF	1.0168	0.9711	0.9497	0.9354	0.9197	0.9083	0.8971
CTR	1.0124	0.9685	0.9481	0.9337	0.9194	0.9089	0.8969
CDL	1.0044	0.9639	0.9377	0.9211	0.9068	0.8970	0.8879
DCNN	**0.9885**	**0.9490**	**0.9223**	**0.9052**	**0.8891**	**0.8825**	**0.8698**

4. Conclusion

The research which was conducted using a dynamic convolutional neural network (DCNN) and a standard probabilistic matrix factorization revealed that the inclusion of side information in product description is effective in movie e-commerce business. It also found that the application of this method can increase the level of accuracy compared to previous models. Its involvement has also been identified to increase deeper understanding of texts which can be used in predicting rating of a product. There is possible integration of this method with another standard matrix factorization such as SVD, NMF, SVD++. This hybridization approach has been proven to have the capability of improving performance in rating prediction and applied text sentence document to increase the effectiveness of item latent factor representation which is very essential in upgrading the performance of recommendation machines.

Acknowledgment

Our appreciation goes to everyone that supports this research starting from University Teknikal Malaysia Melaka (UTEM), Malaysia; Universitas Amikom Yogyakarta, Indonesia; PT. Time Excelindo, Yogyakarta, Indonesia which is the Holding Company in Indonesia; and anonymous reviewers who gave valuable critiques and suggestions for the improvement of this paper.

References

[1] R. Gulati and J. Garino, "Get the right mix of bricks & clicks.," *Harv. Bus. Rev.*, vol. 78, no. 3, pp. 107–114, 2000, available at : https://europepmc.org/abstract/med/11183973.

[2] Hanafi, N. Suryana, and A. Sammad, "Evaluation of e-Service Quality, Perceived Value on Customer Satisfaction and Customer Loyalty: A Study in Indonesia," *Int. Bus. Manag.*, vol. 11, no. 11, pp. 1892–1900, 2017, doi: https://doi.org/10.3923/ibm.2017.1892.1900.

[3] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Trans. Manag. Inf. Syst.*, vol. 6, no. 4, p. 13, 2016, doi: https://doi.org/10.1145/2843948.

[4] J. Davidson *et al.*, "The YouTube video recommendation system," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 293–296, doi: https://doi.org/10.1145/1864708.1864770.

[5] J. Ben Schafer, J. A. Konstan, and J. Riedl, "E-Commerce Recommendation Applications," 2001, pp. 115–153, doi: https://doi.org/10.1007/978-1-4615-1627-9_6.

[6] Hanafi, N. Suryana, and A. Basari, "Deep Learning for Recommender System Based on Application Domain Classification Perspective: a Review," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 14, pp. 4513–4529, 2018, available at : http://www.jatit.org/volumes/Vol96No14/17Vol96No14.pdf.

[7] Hanafi, N. Suryana, and A. Basari, "An understanding and approach solution for cold start problem associated with recommender system : a literature review," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 9, 2018, available at : http://www.jatit.org/volumes/Vol96No9/26Vol96No9.pdf.

[8] F. Ricci, L. Rokach, and B. Shapira, Eds., *Recommender Systems Handbook*, 2015, doi: https://doi.org/10.1007/978-1-4899-7637-6.

[9] A. den Oord, S. Dieleman, and B. Schrauwen, "Deep content-based music recommendation," in *Advances in neural information processing systems*, 2013, pp. 2643–2651, available at : http://papers.nips.cc/paper/5004-deep-content-based-.

[10] S. Jaradat, "Deep Cross-Domain Fashion Recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems - RecSys '17*, 2017, pp. 407–410, doi: https://doi.org/10.1145/3109859.3109861.

[11] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowledge-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013, doi: https://doi.org/10.1016/j.knosys.2013.03.012.

[12] E. Çano and M. Morisio, "Hybrid recommender systems: A systematic literature review," *Intell. Data Anal.*, vol. 21, no. 6, pp. 1487–1524, Nov. 2017, doi: https://doi.org/10.3233/IDA-163209.

[13] X. Wang and Y. Wang, "Improving Content-based and Hybrid Music Recommendation using Deep Learning," in *Proceedings of the ACM International Conference on Multimedia - MM '14*, 2014, pp. 627–636, doi: https://doi.org/10.1145/2647868.2654940.

[14] K. Park, J. Lee, and J. Choi, "Deep Neural Networks for News Recommendations," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17*, 2017, pp. 2255–2258, doi: https://doi.org/10.1145/3132847.3133154.

[15] L. Chen, G. Chen, and F. Wang, "Recommender systems based on user reviews: the state of the art," *User Model. User-adapt. Interact.*, vol. 25, no. 2, pp. 99–154, 2015, doi: https://doi.org/10.1007/s11257-015-9155-5.

[16] L. Qu, G. Ifrim, and G. Weikum, "The bag-of-opinions method for review rating prediction from sparse text patterns," in *Proceedings of the 23rd International Conference on Computational Linguistics*, 2010, pp. 913–921, available at : https://dl.acm.org/citation.cfm?id=1873884.

[17] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," in *Proceedings of the 8th ACM Conference on Recommender systems - RecSys '14*, 2014, pp. 105–112, doi: https://doi.org/10.1145/2645710.2645728.

[18] J. McAuley and J. Leskovec, "Hidden factors and hidden topics," in *Proceedings of the 7th ACM conference on Recommender systems - RecSys '13*, 2013, pp. 165–172, doi: https://doi.org/10.1145/2507157.2507163.

[19] A. Severyn and A. Moschitti, "Twitter Sentiment Analysis with Deep Convolutional Neural Networks," in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '15*, 2015, pp. 959–962, doi: https://doi.org/10.1145/2766462.2767830.

[20] J. Wang, "Predicting Yelp Star Ratings Based on Text Analysis of User Reviews," 2015, available at: https://pdfs.semanticscholar.org/1445/e46d8bf48f5739246c290340fbb15113902e.pdf.

[21] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 1235–1244, doi: https://doi.org/10.1145/2783258.2783273.

[22] J. Gao, P. Pantel, M. Gamon, X. He, and L. Deng, "Modeling Interestingness with Deep Neural Networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 2–13, doi: https://doi.org/10.3115/v1/D14-1002.

[23] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM '14*, 2014, pp. 101–110, doi: https://doi.org/10.1145/2661829.2661935.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: https://doi.org/10.1109/5.726791.

[25] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, no. Aug, pp. 2493–2537, 2011, available at : http://www.jmlr.org/papers/v12/collobert11a.html.

[26] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2014, pp. 655–665, doi: https://doi.org/10.3115/v1/P14-1062.

[27] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv Prepr. arXiv1408.5882*, 2014, available at : https://arxiv.org/abs/1408.5882.

[28] K. Iwahama, Y. Hijikata, and S. Nishida, "Content-based filtering system for music data," in *2004 International Symposium on Applications and the Internet Workshops. 2004 Workshops.*, 2004, pp. 480–487, doi: https://doi.org/10.1109/SAINTW.2004.1268677.

[29] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional Matrix Factorization for Document Context-Aware Recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, 2016, pp. 233–240, doi: https://doi.org/10.1145/2959100.2959165.

[30] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *J. Mach. Learn. Res.*, vol. 10, no. Dec, pp. 2935–2962, 2009, available at : http://www.jmlr.org/papers/v10/gunawardana09a.html.

[31] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer (Long. Beach. Calif).*, no. 8, pp. 30–37, 2009, doi: https://doi.org/10.1109/MC.2009.263.

[32] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm Trans. Interact. Intell. Syst.*, vol. 5, no. 4, p. 19, 2016, doi: https://doi.org/10.1145/2827872.