INTERNATIONAL JOURNAL
ON INFORMATICS VISUALIZATION

journal homepage : www.joiv.org/index.php/joiv

Enhance Document Contextual Using Attention-LSTM to Eliminate Sparse Data Matrix for E-Commerce Recommender System

Hanafi [a,*], Anik Sri Widowati [b], Jaeni [a], Jack Febrian Rusdi [c]

[a] Department of computer science, University of Amikom Yogyakarta, Jl. Ringroad Utara Condongcatur Depok, Sleman, Indonesia
[a] Faculty of economic and social, Universitas Amikom Yogyakarta, Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Indonesia
[c] Department of Informatics Engineering, Sekolah Tinggi Teknologi Bandung, Bandung, Indonesia

Corresponding author: [*]hanafi@amikom.ac.id.ac.id

***Abstract*—E-commerce has been the most important service in the last two decades. E-commerce services influence the growth of the economic impact worldwide. A recommender system is an essential mechanism for calculating product information for e-commerce users. The successfulness of recommender system adoption influences the target revenue of an e-commerce company. Collaborative filtering (CF) is the most popular algorithm for creating a recommender system. CF applied a matrix factorization mechanism to calculate the relationship between user and product using rating variable as intersection value between user and product. However, the number of ratings is very sparse, where the number of ratings is less than 4%. Product Document is the product side information representation. The document aims to advance the effectiveness of matrix factorization performance. This research considers to the enhancement of document context using LSTM with an attention mechanism to capture a contextual understanding of product review and incorporate matrix factorization based on probabilistic matrix factorization (PMF) to produce rating prediction. This study employs a real dataset using MovieLens dataset ML.1M and Amazon information video (AIV) to observe our ATT-PMF model. Movielens dataset represents of number sparse rating that only contains below 4% (ML.1M). Our experiment report shows that ATT-PMF outperforms more than 2% on average than previous work. Moreover, our model is also suitable to implement on huge datasets. For further research, enhancement of product document context will be a good factor in eliminating sparse data problems in big data problems.**

***Keywords*— Sparsity data; recommender system; matrix factorization; e-commerce; attention mechanism; PMF.**

Manuscript received 20 Jan. 2022; revised 11 Mar. 2022; accepted 17 Apr. 2022. Date of publication 30 Sep. 2022.

I. Introduction

E-commerce is the most popular application to provide online transactions on the internet, and E-commerce influences a high impact on the growth of global economic enhancement. In everyday life, we cannot escape from online transactions such as newspapers to read, videos to watch, food to deliver, games to play, and many friends to confirm. That means, in every life, we need online transactions [1].

E-commerce services require a mechanism to serve essential product information to customer or buyer candidates. This engine is responsible for famous computing information, called the recommender system. A recommender system is an automatic engine to calculate product fit information for a customer. The successful adoption of the recommender system influences target marketing value. A recommender system is a very important tool to achieve business revenue for an e-commerce company.

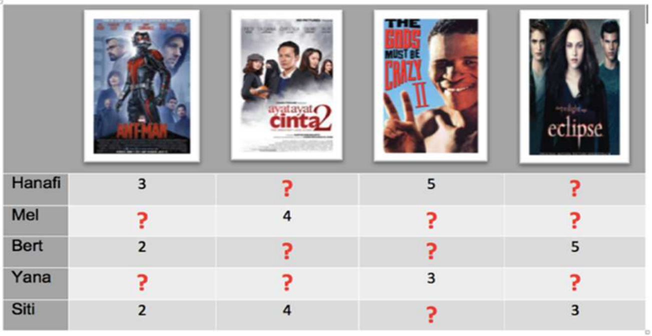
There are four recommender system algorithm model classifications to build e-commerce: collaborative filtering, content-based, knowledge-based, and demographic-based filtering. According to the literature [2]–[5], collaborative filtering is the most effective algorithm for product recommendation. Collaborative filtering uses user behavior activity records in the past, especially rating information, and rating is an expression of satisfied representation about a service or product. Therefore, collaborative filtering is more useful than another algorithm, such as content-based, that considers characteristic product calculation.

The early collaborative filtering model depended on a memory-based algorithm and a popular nearest neighbor. The memory-based algorithm is a very simple model to calculate product recommendations, and they no need data training and no need to develop a complex algorithm. Memory-based is suitable for a simple model, small dataset, and no greater number of transactions. However, memory-based have shortcomings when applying new additional data dan massive datasets. Another essential drawback of memory-based is that they cannot be integrated with other information such as user information or item information representation. An example of a product recommendation presentation can be seen in Fig. 1 below.

Fig. 1 E-commerce for online movie market from Netflix

It is a unique competition that belongs to the Netflix corporation. They will give the winner 6 million dollars for those who successfully achieve more than 10% over existing Netflix engine performance. Most academicians, experts, and researchers consider applying another collaborative filtering model based on a latent factor famously called matrix factorization, even though latent factors were developed in early 2000 by Sarwar [6]. The latent factor model uses singular value decomposition (SVD) that employs low rank dimensional. The advantage of the matrix factorization model is that it can be integrated with other information, including the product's or user's side information. The application of latent factor of matrix factorization has become popular in the Netflix competition aims to reach effectiveness level in producing rating prediction.

The fundamental problem of applying matrix factorization in sparse rating datasets has been raised. The sparse level rises when the number of ratings is minimal in the level below 4% or an extreme sparse level below 1%. The effectiveness of rating prediction using latent factors degrades significantly when applied to sparse rating. Figure 1 below illustrates unrating products causes sparse data. It impacts product information for customers inaccurately and has become a major problem in collaborative filtering issues. Most researchers involve product side information, specifically product document information such as product description, product information, and product testimony [7]–[9].


ANT-MAN
ayat-ayat cinta 2
THE GODS MUST BE CRAZY II
eclipse

Hanafi	3	?	5	?
Mel	?	4	?	?
Bert	2	?	?	5
Yana	?	?	3	?
Siti	2	4	?	3

Fig. 2 Unrating product cause sparse data; it has an impact on inaccurate product recommendation

The adoption of product document to increase the effectiveness of the latent factor model have been popular in the recent decade [10]. The majority of them consider reaching a contextual understanding of the document. Several strategies capture document understanding, such as [11]. This model implements TF-IDF and a statistical model to calculate and interpret document representation called bag of the word (BOW). This algorithm successfully increased the latent factor model based on probabilistic matrix factorization (PMF). PMF is the enhancement of the SVD model that considers probabilistic mechanisms based on gaussian normal distribution [12].

Adopting traditional natural language processing in latent factor (matrix factorization) has enhanced collaborative filtering. However, the traditional NLP faced the essential shortcoming in capturing understanding in context. An expert in NLP said that the contextual understanding of phrases could be reached by considering word order and subtle words; for example, "the cat chases the mouse" is a normal situation, while "the mouse chase the cat" is an abnormal condition. According to the bag of word mechanism, both phrases are similar in the meaning of text [13].

Some algorithms integrate product documents and user information into matrix factorization models such as PMF [14]–[17]. According to the experiment report, their deep learning model performs better than the traditional latent factor approach. A deep learning framework was adopted in the last five years. Deep learning enhances performance in several computer science applications such as image processing, voice recognizing, and natural language processing. Several researchers have implemented a deep learning model in recommender system—enhancement of matrix factorization based on PMF using auto encoder to calculate product document [18]. A similar model was proposed to advance SVD model [19].

The advance of the collaborative filtering model using CNN was proposed by Kim [20]. Their model, called ConvMF achieves better performance than previous work based on the autoencoder. The improvement of CNN on this model is due to the dimensional reduction mechanism to capture product document understanding. Unlike previous work, they consider word order and subtle words to interpret product documents. Like Kim, another word order mechanism proposed by Hanafi [21] by sequential aspect mechanism where adoption of sequential mechanism achieves more useful over-dimensional reduction based on CNN. Table 1 represents the state-of-the-art collaborative filtering algorithm by enhancing document of product understanding.

TABLE I
COLLABORATIVE FILTERING ALGORITHM BY INTEGRATING PRODUCT DOCUMENT AND LATENT FACTOR MODEL

Ref.	Collaborative filtering algorithm model
[11]	**BOW**: Collaborative filtering model using PMF and TF-IDF to capture document understanding of product review document.
[14]	**LDA**: Collaborative filtering using PMF and integrated with a statistical approach to interpreted product review document.
[15]	**CTR**: Collaborative filtering using PMF integrated with topic modeling to interpret product review documents.

[18]	**CDL**: Collaborative filtering using PMF and integrated with auto encoder to interpret document understanding of product review document.
[15]	**HCDR**: Collaborative filtering model using SVD and auto encoder to capture document understanding of product review document by considering dimensional reduction aspect of the phrase.
[20]	**ConvMF**: Collaborative filtering model using PMF and CNN to capture document understanding of product review document by considering dimensional reduction aspect of phrase.
[21]	**LSTM-PMF**: Collaborative filtering model using PMF, word embedding, and LSTM to capture document understanding of product review document by considering the sequential aspect of the phrase.
[22]	**SRCMF**: Collaborative filtering model using PMF, and CNN aims interpreted social review document of the product.
[23]	**Att-ConvMF**: Collaborative filtering model using latent factor and optimizing product document understanding using CNN and attention mechanism.

Capturing document contextual understanding of product review becomes an important aspect. In recent five years, several algorithm models have obtained some research, such as word embedding based on word2vec, Fastest, glove and so on. In recent years, several models have been using bidirectional word vector representation.

According to the above explanation, in the early decade, the traditional NLP model became famous for integrating document representation into latent factor models such as BOW, LDA, and CTR. However, in recent five years, some researchers have involved deep learning models in enhancing document contextual understanding, such as CDL, CNN, LSTM, and Attention model. All of them employ a latent factor model based on PMF that aims to improve rating prediction effectiveness. Different from previous work where Attention mechanisms have no adoption in sequential aspect in LSTM. Therefore, this study is expected to contribute to generating a contextual understanding of product documents using sequential to the sequential aspect, which integrated Attention and LSTM. Thus, this study combines Attention-LSTM into traditional latent factors based on PMF to enhance the effectiveness of rating prediction in large datasets.

II. Materials and Method

A. Probabilistic Matric Factorization (PMF)

PMF is a very popular latent factor model for producing rating predictions. Many researchers consider to adoption PMF in many research and application for a recommendation. PMF is an advanced version of SVD. The essential work mechanism can be explained as follows: *M* represents a movie, and *N* represents the user. While a rating value represents integer from 1 to k, Rij represents user *i* with movie *j*. While $U \in R^{D\times N}, V \in R^{D\times M}$. Then, *U* and *V* represent the user and movie latent factors. Aim to calculate rating prediction can be computed by $R_{ij} = U_i^T, V_j$. PMF is SVD version with Gausian normal distribution approach. The vector representation of a user and movie resulted from the distribution with rating correspondent, where the distribution mechanism can be computed with equation 1 as follows:

$$p(R_{ij}|U_i, V_j, \sigma^2) = N(R_{ij}|U_i^T V_j, \sigma^2) \quad (1)$$

A mechanism to transform the latent vector of the item, the PMF model considers applying a zero-mean spherical gaussian prior that can be calculated with equation 2 as follows:

$$p(V|\sigma_V^2) = \prod_{j=1}^{M} N\,(V_i|0, \sigma_V^2 I) \quad (2)$$

While a mechanism to transform the latent vector of the item, the PMF model considers applying a zero-mean spherical gaussian prior that can be calculated with equation 3 as follows:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} N\,(U_i|0, \sigma_U^2 I) \quad (3)$$

B. Capturing Document Contextual with LSTM and Attention Mechanism

1) LSTM

RNNs, or recurrent neural networks, are artificial neural networks in which the data from the previous step is used as input for the following step. The main problem with RNNs, on the other hand, is the occurrence of gradient vanishing and exploding problems during back-propagation. For this reason, Hochreiter and Schmid Huber [24] developed the Long Short-Term Memory (LSTM) in 1997 as a solution to the problem. A modified version of recurrent neural networks, Long Short-Term Memory (LSTM) networks, can learn information from earlier time steps and apply it to subsequent time steps. Cell states, as opposed to the conventional feed-forward neural networks, are used to channel data as it flows through the LSTM network. LSTMs are capable of selectively remembering and forgetting information in this manner. As a result, its gating mechanism allowed RNNs to overcome the "short-term memory" problem. LSTM units are composed of a cell, an input gate, an output gate, and a forget gate, the last gate in the chain. The cell retains data over a random period, and the three gates continuously regulate the flow of information into and out of the cell.

LSTM is a sub-class of neural network and categorical feed-forward neural network. One advantage of the LSTM approach is considering connecting the past information section and the current section process. According to the context of NLP, this is an essential aspect of capturing a contextual understanding of phrases in a document. The LSTM consists of several hidden stages related to the input layer, output layer, the hidden state, and prior process. The advantage of LSTM is to initiate sequential aspects due to some critical calculation stages in the hidden stage. The detailed algorithm of the LSTM model can be seen on Fig 3 as follow.

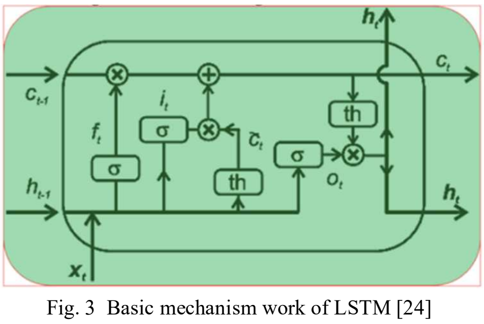


Fig. 3 Basic mechanism work of LSTM [24]

$$
\begin{aligned}
i &= \sigma(x_i U^i + S_{t-1} W^i)\\
f &= \sigma(x_i U^f + S_{t-1} W^f)\\
o &= \sigma(x_t U^o + S_{t-1} W^o)\\
g &= tanh(x_i U^g + S_{t-1} W^g)\\
c_t &= c_t - Of + gOi\\
s_t &= tanh(c_t) - O_o gOi
\end{aligned} \tag{4}
$$

i, f, o It represents the input gate (i), (f) forget gate, and (o) output gate symbols. They all have equations that are similar to one another; the only difference is that their parameter matrices are different. It is referred to as a gate because the sigmoid function determines whether the value is *0* or *1*.

g It is the symbol for the hidden state, computed based on the current input and the previous hidden state in which it appears.

c_t It represents the internal memory of the hidden state and serves as a symbol for it. A combination of the previous memory c_{t-1}, multiplied by a forget gate, and the new hidden state g, multiplied by the input gate, has resulted in the final result.

s_t It represents the memory of the hidden state and serves as a symbol for it. The output gate multiplies the computed output of the hidden state on s_t, which results from the hidden state computation.

2) *Attention Mechanism*

In the recent five years, the attention mechanism has been an essential finding in deep learning technology. Furthermore, it is a method that mimics the focus of the human mind. Deep learning enhances artificial intelligence models widely used in computer science applications, including natural language processing [25] and image processing [26]. Sequence to sequence (seq2seq) models from the domain of neural machine translation was used to create it initially. When using the seq2seq approach, the encoder analyzes the input data and compresses it into a context vector of a fixed length (sentence embedding), and the decoder uses the context vector in its computations to produce an output that has been transformed. Seq2Seq challenges have shown the enormous strength of this architecture, but a critical flaw hampers it. Sentence embedding is generated in a single vector that becomes more difficult for a machine to process as the length of the input data increases. As a result, it cannot store longer input data because it tends to forget portions of it.

To help neural machine translation remember long source sentences, Bahdanau [27] introduced an attention mechanism. This mechanism creates shortcuts between a single context vector and the entire source input rather than building a single context vector from the start. The weights of these shortcut connections can be adjusted for each output feature. Data that is not critical is emphasized while the rest fades into the background. The illustration figure of attention can be seen in Fig. 4.

Multiple attention weights are calculated for each of the inputs because not all can be used in generating the corresponding output. Hence, the attention mechanism calculates multiple attention weights. The weighted sum of the annotations is used to create the context vector C_i for the output result y_i.

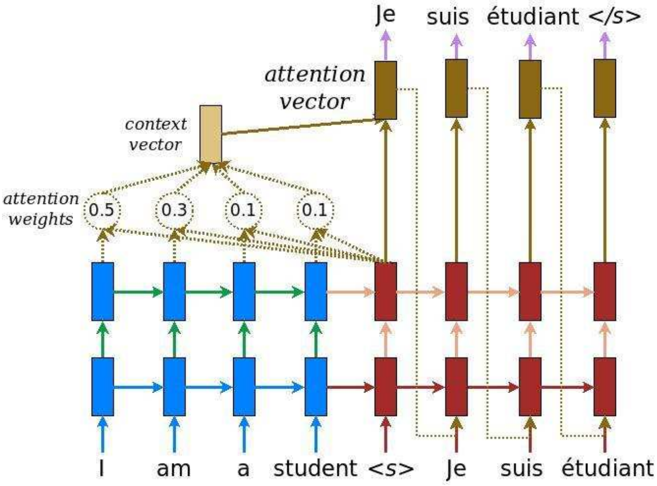


Fig. 4 Basic work of Attention mechanism [27]

While not every input would be applied in producing the correspondent result, the attention algorithm computes multiple attention weights marked by $\alpha(t,1)$, $\alpha(t,2)$, .., $\alpha(t, t)$. The context vector C_i for the output result y_i is generated by implementing the weighted sum of the formulation:

$$C_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{7}$$

The attention weights are calculated by normalizing the result of a feed-forward neural network explained by the function that catch aligning among input at j with result at i. In the following equation, a softmax function is used to compute the weights α_{ij}.

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{5}$$

$$e_{ij} = a(s_{i-1}, h_j) \tag{6}$$

e_{ij} is the result value of a feed-forward neural network explained by the function a that tried to catch the aligning among input at *j* and the final result at *i*.

C. *Hybridization Attention-LSTM and PMF*

Following the attention-LSTM work, applying regression algorithms such as rating prediction in a recommender system based on a collaborative filtering model is unsuitable. The output of the hybrid model between attention and LSTM in a two-dimensional vector representation is not applicable to be implemented to predict the rating directly. The hybridization approach with matrix factorization as a latent factor model is needed to address the essential problem above to produce rating prediction tasks, such as PMF, non-negative matrix factorization (NMF), and SVD. The completion of our proposed model can be seen in Figure 5, including MovieLens and Amazon datasets, document pre-processing of product review, user pre-processing, mechanism of hybridization process, integrating of matrix factorization model, and evaluation result using RMSE.

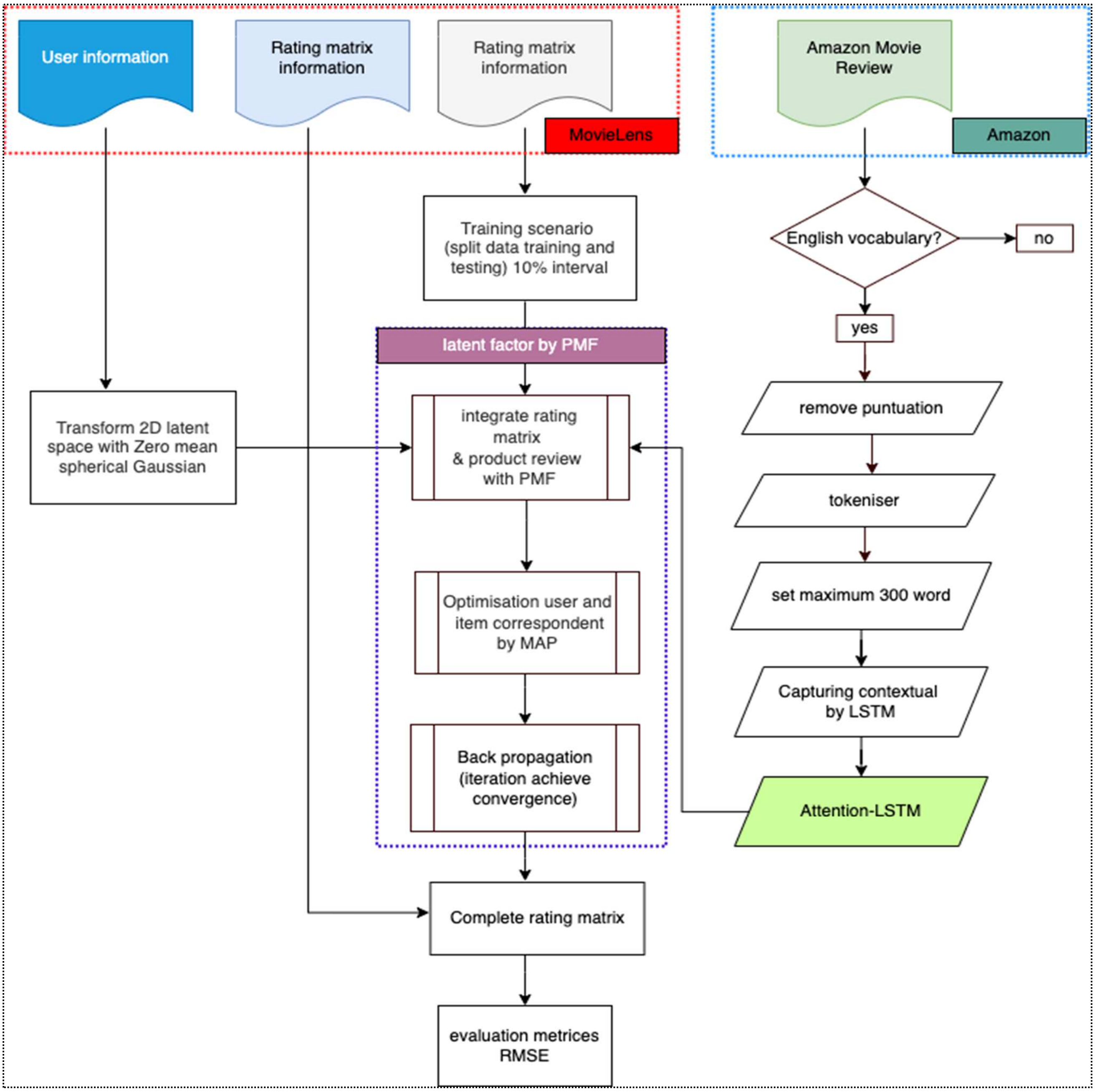

User information
Rating matrix information
Rating matrix information
MovieLens
Amazon Movie Review
Amazon
Training scenario (split data training and testing) 10% interval
English vocabulary?
no
yes
latent factor by PMF
integrate rating matrix & product review with PMF
Transform 2D latent space with Zero mean spherical Gaussian
remove puntuation
tokeniser
set maximum 300 word
Optimisation user and item correspondent by MAP
Capturing contextual by LSTM
Back propagation (iteration achieve convergence)
Attention-LSTM
Complete rating matrix
evaluation metrices RMSE

Fig. 5 Attention-LSTM and PMF framework (Our model)

PMF oversees computing the relationship between the factor of user's latent model and the factor of the product's latent space, which strengthens the user and related item. For instance, N as a symbol of the user and M as a symbol of the item, the strategy to compute the rating score is $R \in R^{NxM}$, while the formula to compute user representation is $U \in R^{KxN}$ and item representation can be computed by $V \in R^{KxN}$, then the table of products can be calculated by $U^T V$. Following to probabilistic perspective, the normal distribution can be computed by:

$$p(R|U,V,\sigma^2) = \prod_{i=1}^{N}\prod_{j-1}^{M}[N(R_{ij}|U_i^T V_j,\sigma^2)]^{I_{ij}} \qquad (8)$$

Where:

μ : mean of the total population
σ^2 : variance score
I_{ij} : an indicator mechanism to produce latent user model

1) User latent vector representation

MovieLens collects user information representations that only contain user information representation and rating of the product information. The user latent model territory employs a zero mean spherical Gaussian prior by incorporating the user data variance value σ^2 with the formula is as follows:

$$p(U|\sigma_U^2) = \prod_{i-1}^{N} N(U_i|0,\sigma_U^2 I) \qquad (9)$$

2) Item latent vector representation

Item document information representation is collected from AIV in the form of AIV item review. A product document 2D vector 50 is obtained after a series of processes based on the LSTM mechanism. From a probabilistic standpoint, the latent item model is given by:

$$p(V|W,X,\sigma_W^2 = \prod_{j}^{M} N(v_j|attention\_lstm(W,X_j)\sigma_W^2 I) \qquad (10)$$

While variable v_j as item representation that is produced by attention and LSTM framework. It can be obtained by:

$$v_j = attention\_lstm(W, X_j) + \varepsilon_j \tag{11}$$

3) Optimizing model and producing rating prediction mechanism

This is the last process consisting of 3 essential steps. When calculating an unknown quantity, the MAP statistic comes in handy. It is like the posterior distribution in terms of optimization and learning. To be more specific, it seeks to optimize the learning variable while considering the MAP application. This method employed log a posteriori using hyperparameters to analyze user and movie features. The complete computing scenario can be seen in the equation below:

$$\begin{aligned} &\max_{U,V,W} p(U, V, W|R, X, \sigma^2, \sigma_U^2, \sigma_v^2, \sigma_W^2) \\ =\ &\max_{U,V,W} [p(R|U, V, \sigma^2) p(U|\sigma_U^2) p(V|W, X, \sigma_V^2) p(W|\sigma_W^2) \end{aligned} \tag{12}$$

The training scenario to learn the relationship between user and item aims to minimize the loss function; the detailed formula can be seen in the equation below.

$$\begin{aligned} \mathcal{L}(U, V, W) = &\sum_{i}^{N} \sum_{j}^{M} \frac{I_{ij}}{2} (r_{ij} - u_i^T v_j)_2 + \frac{\lambda U}{2} u_{i2} \\ &+ \frac{\lambda V}{2} \sum_{j}^{M} v_j - attetnion\_lstm(W, X_j)_2 \\ &+ \frac{\lambda W}{2} \sum_{k}^{W_k} W_{k2} \end{aligned} \tag{13}$$

This step is critical because it involves optimizing W, which represents the weight variable and bias variable for each layer in the back-propagation algorithm, which is an important step in the process. The update mechanism is designed to optimize every layer, including *V*, *U*, and *W*, until convergence is required. It is optimized until convergence is required. The following is the formula that was used to predict the unknown rating:

$$\begin{aligned} r_{ij} \approx \mathbb{E}[r_{ij}|u_i^T v_j, \sigma^2] &= u_i^T v_j \\ &= u_i^T (attention\_lstm(W, X_j + \epsilon_j) \end{aligned} \tag{14}$$

D. Dataset

One of the most widely used datasets for e-commerce experiments is Movielens. In 1997, the University of Minnesota's School of Computing developed it. The MovieLens datasets [28], [29], were used in most recommender system experiments. We were looking for information to help with our own personal recommendations. These datasets have some categories that are dependent on how many ratings, how many users, how many products, and how dense the sparse ratings are. AIV's product review document was used in this experiment. This is a well-known Amazon dataset [30]. According to Table 3, here are some of the dataset's characteristics.

TABLE II
DATASET CHARACTERISTIC

Type of dataset	Total of users	Total of movie	Total of rating	Level of sparse
ml.1m	6040	3544	993482	4.64%
AIV	29757	15149	135188	0.03%

This experiment utilizes MovieLens dataset categorical ml-1m, which contains one million ratings at a sparse level of 4.64 percent, and Amazon information video (AIV) as product document representation because MovieLens dataset does not contain product document representation such as review, testimony, and product description. This is a critical factor in determining the performance of hybridization among Attention, LSTM, and PMF under certain sparsity level conditions.

E. Evaluation Metrics

It is necessary to evaluate the performance of Attention-LSTM-PMF. The RMSE evaluation matrix is the most frequently used technique for determining the effectiveness of rating prediction [17], [31], [32]. The experiment's scenario is divided into four parts, each of which divides the dataset by a 20% interval ratio, as follows: sparseness level 80% (20:80), sparseness level 60% (40:60), sparseness level 40% (60:40), sparseness level 20 (80:20).

The training process's output was evaluated using RMSE evaluation matrix as the most popular evaluation model for rating prediction in recommender system research. The evaluation matrices have the following formula:

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

Finally, the Attention-LSTM-PMF prediction result was compared to the genuine rating collected from the most popular datasets for the recommender system (MovieLens). This is a mechanism to observe the effectiveness of the attention-LSTM model to support latent factors based on PMF.

III. Result and Discussion

Sparse data in recommender system territory has been a very important issue, and the growth of huge of data also increases the issue more popular. Combining Attention-LSTM and PMF is a solution to handle sparse data. This experiment attempts to observe the effectiveness of the algorithm. Figure 6 demonstrated that the training result involved an 80% sparseness level with a ratio of 20% data training and 80% data testing. The training scenario compared LSTM and PMF with Attention and without attention. The impact of attention shows that success in increasing effectiveness on rating prediction is even adopted at an extremely sparse level. Attention-LSTM and PMF are represented in red, and LSTM-PMF without PMF is represented in blue.

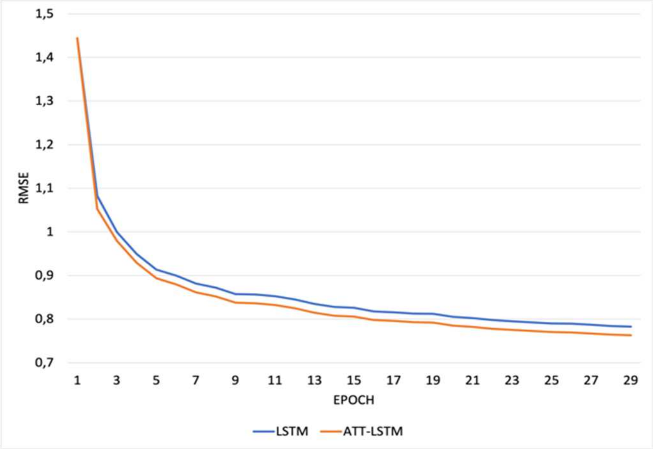


Fig. 6 RMSE result on sparseness level 80%

The next section scenario attempt to decrease the sparseness level to 40% (Fig. 7), 60% (Fig. 8), and finally to 80% (Fig. 8). In every section scenario, Attention-LTSM performs better than LSTM without Attention.

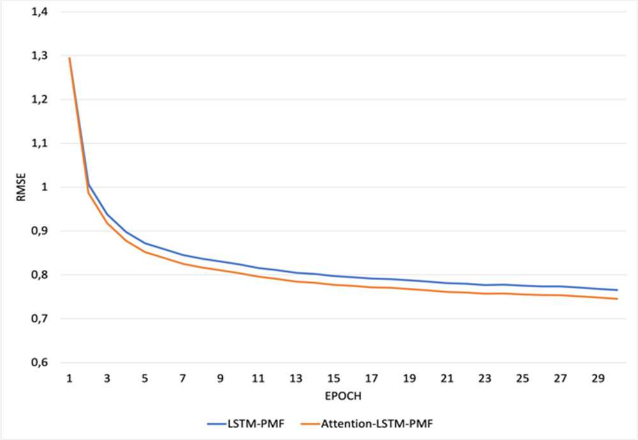


Fig. 7 RMSE result on sparseness level 60%

Refer to the RMSE evaluation report; every training scenario reaches below 0.80, where lower RMSE achievement is better. This result shows that Attention-LSTM proved better effective over previous work. The detailed experiment result can be seen in Fig. 10.

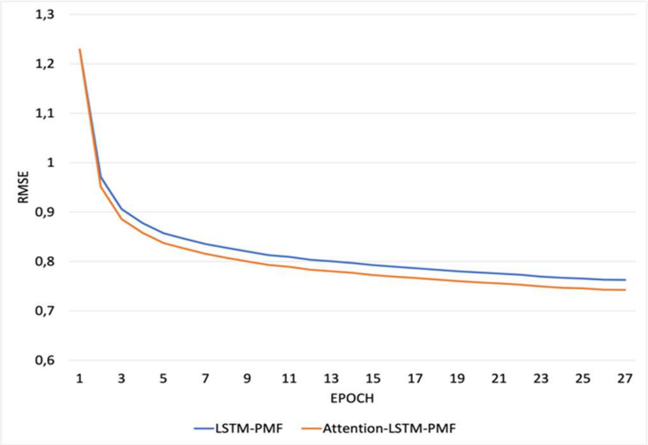


Fig. 8 RMSE result on sparseness level 40%

Our best achievement is demonstrated in Fig. 9, where the RMSE result reaches almost 0.731 and the Attention mechanism's success in supporting LSTM work to increase the effectiveness of PMF in generating rating prediction.

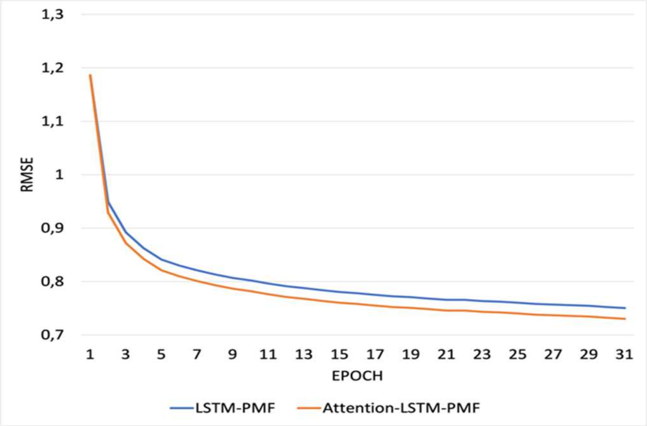


Fig. 9 RMSE result on sparseness level 20%

Fig. 10 shows the attention mechanism's effectiveness over LSTM in every training scenario. Orange color represents 80% sparseness level, green represents 40% sparseness level, grey represents 60%, and the last is blue color that represents 20% sparseness level. The key success factor of attention is increasing the share weight of the latent factor document of the product that is represented in *W* where this variable plays an essential role in increasing the value of variable *V*. In every case, specifically for natural language processing, seq2seq aspect success in increasing document understanding point of view.

Orange color represents LSTM-PMF versus Attention-LSTM-PMF in 80% sparseness level. The green color represents LSTM-PMF versus Attention-LTM-PMF in 60% sparseness level, while the Grey color represents LSTM-PMF versus Attention-LSTM-PMF at 40% sparseness level, and the last experiment scenario can be seen in blue color, where this is represented on LSTM-PMF versus Attention-LSTM-PMF in 20% sparseness level. According to the experiment report in Fig 10, adopting an attention mechanism improved performance over LSTM-PMF.

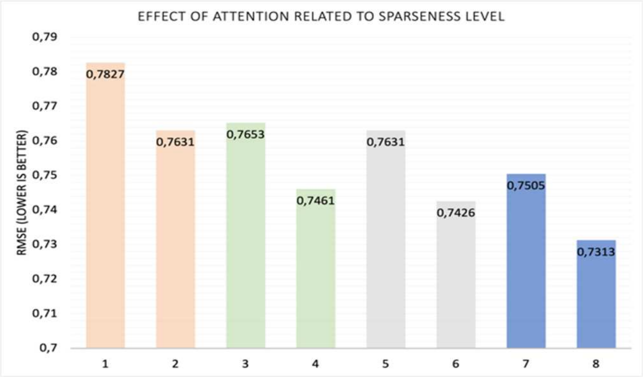


Fig. 10 RMSE comparison results on various sparseness level

The effectiveness of every PMF with the document understanding model is presented in Fig. 11. The schematic experiment comparison in this research only applied a 50% (50:50) sparseness level. The competitor in this experiment included auto encoder and PMF (blue color), CNN and PMF (red color), LSTM and PMF (green color), and our model Attention-LSTM and PMF (yellow color). Overall models have been state-of-the-art in recommender systems in the last five years. Our model achieves significant performance over another deep learning model.

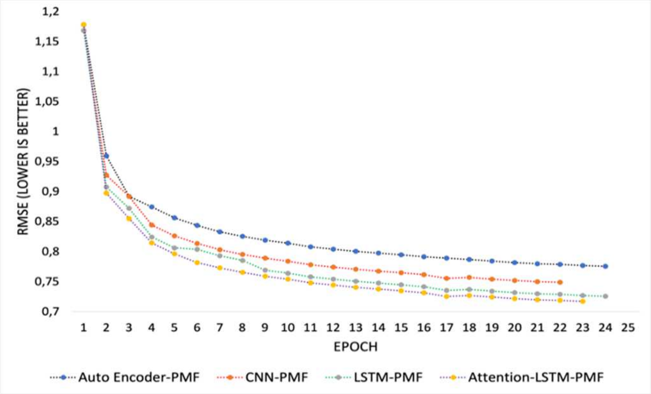


Fig. 11 RMSE comparison result over AE, CNN, LSTM, and Attention-LSTM

IV. CONCLUSION

Sparse data issues, which are caused by a minimum rating, continue to be a significant problem in the recommendation system. In this study, we proposed a latent factor model incorporating Attention, LSTM, and PMF while taking words sequential to calculate in word order to interpret document understanding and capture the contextual insight contained within the product review documents. According to the results of our experiment described above, our model outperformed previous work. Adoption Attention into LSTM-PMF performed well, according to the researchers, because of the impact of contextual insight representation of the document to support latent factors based on PMF in increasing the effectiveness level of generating ratings, which was believed to be the case. Furthermore, the involvement product documents developed with the help of Attention and LSTM improved the efficiency of the training process, allowing for greater convergence in an overall training scenario. Several methods are available to learn contextual insight interpretability, such as bidirectional word encoder representation. Considering the bidirectional model to improve contextual understanding of the document may improve the performance of matrix factorization in predicting the rating matrix. Future research work will be made more difficult because of this. PMF is a matrix factorization method that is a subset of the matrix factorization method. This can be further enhanced by incorporating other matrix factorization methods, such as SVD, SVD++, and Non-Negative Matrix Factorization (NNMF), which only take into rating information. With the help of some of the techniques previously mentioned, there is a possibility to improve the effectiveness of rating prediction in sparse data in large datasets.

ACKNOWLEDGMENT

The Universitas Amikom Yogyakarta Research Fund supported this work.

REFERENCES

[1] Hanafi, R. Widyawati, and A. S. Widowati, "Effect of service quality and online servicescape toward customer satisfaction and loyalty mediated by perceived value," *IOP Conf. Ser. Earth Environ. Sci.*, vol. 704, no. 1, 2021, doi: 10.1088/1755-1315/704/1/012011.

[2] Hanafi, N. Suryana, and A. Sammad, "An Understanding and Approach Solution for Cold Start Problem Associated with Recommender System : A Literature Review," *J. Theor. Appl. Inf. Technol.*, vol. 96, no. 09, pp. 2677–2695, 2018.

[3] Y. Koren, R. Bell, and C. Volinsky, "Matrix Factorization Techniques for Recommender Systems," *IEEE*, vol. 40, no. 8, pp. 42–49, 2009.

[4] Hanafi, N. Suryana, and A. Samad, "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.

[5] Hanafi, N. Suryana, and A. S. H. Basari, "Exploit Multi Layer Deep Learning and Latent Factor to Handle Sparse Data for E-commerce Recommender System," *SCITEPRESS*, no. Conrist 2019, pp. 343–351, 2020, doi: 10.5220/0009910603430351.

[6] B. M. Sarwar, G. Karypis, J. a Konstan, and J. T. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study," *Architecture*, vol. 1625, pp. 264–8, 2000, doi: 10.1.1.38.744.

[7] Hanafi, N. Suryana, and A. S. H. Basari, "Generate Contextual Insight of Product Review Using Deep LSTM and Word Embedding," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012006.

[8] Hanafi, N. Suryana, and A. S. B. H. Basari, "Convolutional-NN and word embedding for making an effective product recommendation based on enhanced contextual understanding of a product review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, 2019, doi: 10.18517/ijaseit.9.3.8843.

[9] Hanafi, N. Suryana, and A. S. H. Basari, "Deep Contextual of Document Using Deep LSTM Meet Matrix Factorization to Handle Sparse Data: Proposed Model," *J. Phys. Conf. Ser.*, vol. 1577, no. 1, 2020, doi: 10.1088/1742-6596/1577/1/012002.

[10] Hanafi, N. Suryana, and A. S. B. H. Basari, "Convolutional-NN and word embedding for making an effective product recommendation based on enhanced contextual understanding of a product review," *Int. J. Adv. Sci. Eng. Inf. Technol.*, vol. 9, no. 3, pp. 1063–1070, 2019, doi: 10.18517/ijaseit.9.3.8843.

[11] G. Ling, M. R. Lyu, and I. King, "Ratings meet reviews, a combined approach to recommend," *Proc. 8th ACM Conf. Recomm. Syst. - RecSys '14*, pp. 105–112, 2014, doi: 10.1145/2645710.2645728.

[12] R. Salakhutdinov and A. Mnih, "Probabilistic Matrix Factorization.," *Proc. Adv. Neural Inf. Process. Syst. 20 (NIPS 07)*, pp. 1257–1264, 2007, doi: 10.1145/1390156.1390267.

[13] C. C. Aggarwal, *Machine Learning for Text*. NEWYORK: Springer International Publishing AG, 2018.

[14] J. Liu and D. Wang, "PHD : A Probabilistic Model of Hybrid Deep Collaborative Filtering for Recommender Systems," in *ACML*, 2017, pp. 224–239.

[15] C. Wang and D. M. Blei, "Collaborative Topic Modeling for Recommending Scientific Articles," *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. - KDD '11*, p. 448, 2011, doi: 10.1145/2020408.2020480.

[16] Hanafi, E. Pujastuti, A. Laksito, A. Arfriandi, R. Hardi, and R. Perwira, "Handling Sparse Rating Matrix for E-commerce Recommender System Using Hybrid Deep Learning Based on LSTM , SDAE and Latent Factor," vol. 15, no. 2, pp. 379–393, 2022, doi: 10.22266/ijies2022.0430.35.

[17] Hanafi, N. Suryana, and A. S. B. H. BASARI, "Recommender System Based Tensor Candecomp Parafact Algorithm-ALS to Handle Sparse Data In Food Commerce Information Services," *IJSSST*, pp. 1–9, 2019, doi: 10.5013/IJSSST.a.19.06.60.

[18] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative Deep Learning for Recommender Systems," in *KDD conference*, 2015, pp. 1235–1244, doi: 10.1145/2783258.2783273.

[19] J. Wei, J. He, K. Chen, Y. Zhou, and Z. Tang, "Collaborative Filtering and Deep Learning Based Hybrid Recommendation for Cold Start Problem," *2016 IEEE 14th Intl Conf Dependable, Auton. Secur. Comput. 14th Intl Conf Pervasive Intell. Comput. 2nd Intl Conf Big Data Intell. Comput. Cyber Sci. Technol. Congr.*, pp. 874–877, 2016, doi: 10.1109/DASC-PICom-DataCom-CyberSciTec.2016.149.

[20] 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: 10.1145/2959100.2959165.

[21] Hanafi and B. M. Aboobaider, "Word Sequential Using Deep LSTM and Matrix Factorization to Handle Rating Sparse Data for E-Commerce Recommender System," *Comput. Intell. Neurosci.*, vol. 2021, no. 1, 2021, doi: https://doi.org/10.1155/2021/8751173 Research.

[22] X. Wang, X. Yang, L. Guo, Y. Han, F. Liu, and B. Gao, "Exploiting Social Review-Enhanced Convolutional Matrix Factorization for Social Recommendation," *IEEE Access*, vol. 7, pp. 82826–82837, 2019, doi: 10.1109/ACCESS.2019.2924443.

[23] B. Zhang, H. Zhang, X. Sun, G. Feng, and C. He, "Integrating an attention mechanism and convolution collaborative filtering for document context-aware rating prediction," *IEEE Access*, vol. 7, pp. 3826–3835, 2019, doi: 10.1109/ACCESS.2018.2887100.

[24] S. Hochreiter and J. Urgen Schmidhuber, "Lstm," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[25] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751, doi: 10.3115/v1/D14-1181.

[26] Hanafi, A. Pranolo, and Y. Mao, "Cae-covidx: Automatic covid-19 disease detection based on x-ray images using enhanced deep convolutional and autoencoder," *Int. J. Adv. Intell. Informatics*, vol. 7, no. 1, pp. 49–62, 2021, doi: 10.26555/ijain.v7i1.577.

[27] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 577–585, 2015.

[28] F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4: 19:1–19:19. https://doi.org/10.1145/2827872.

[29] F. M. Harper and J. A. Konstan, "The MovieLens Datasets: History and Context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1--19:19, 2015, doi: 10.1145/2827872.

[30] J. McAuley, "Amazon Product Data," 2021. .

[31] A. Gunawardana, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," vol. 10, pp. 2935–2962, 2009.

[32] Hanafi, N. Suryana, and A. S. B. H. Bashari, "Paper survey and example of collaborative filtering implementation in recommender system," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 16, 2017.