JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 1, Maret 2024, Pp. 231-241

# MOVIE RECOMMENDATION SYSTEM USING HYBRID FILTERING WITH WORD2VEC AND RESTRICTED BOLTZMANN MACHINES

**Muhammad Aryuska Pradana[1)], Agung Toto Wibowo[2)]**
1. Telkom University, Indonesia
2. Telkom University, Indonesia

**Article Info**
**Keywords:** Collaborative filtering; Content-based filtering; Hybrid filtering; Normalized Discounted Cumulative Gain; Precision; RBM; Restricted Boltzmann Machine; Word2Vec.

**Article history:**
Received 21 November 2023
Revised 5 December 2023
Accepted 19 December 2023
Available online 1 March 2024

DOI : https://doi.org/10.29100/jipi.v9i1.4306

* Corresponding author.
Muhammad Aryuska Pradana
E-mail address:
aryuskap@gmail.com

**ABSTRACT**
Recommender systems are designed to provide interesting information to users and assist them in making choices. With the help of a recommender system, users can feel more comfortable using an application. In this final project, we will implement a hybrid filtering method using two techniques: Word2Vec as the algorithm for content-based filtering and Restricted Boltzmann Machine for collaborative filtering. The Word2Vec algorithm will utilize a pre-trained model provided by Google, while the Restricted Boltzmann Machine algorithm will utilize the TensorFlow library. The dataset used for this project will be Movie Lens. The goal of this final project is to evaluate the accuracy and performance of the recommender system using various metrics such as Precision and Normalized Discounted Cumulative Gain.

## I. INTRODUCTION

DIGITAL media users expect personalized experiences and relevant recommendations [1]. Recommendation systems serve as tools to fulfill these needs. Artificial intelligence in recommendation systems and search engines greatly assist in manifesting information from unlimited data sources [2]. Popular recommendation systems like Netflix, Apple TV, and Disney+ have initiatives to understand user preferences and provide relevant product recommendations [3].

The use of recommendation systems can address the problem of information overload for users by providing only items that align with their interests and behaviour [4]. Various methods have been developed for recommendation systems, leveraging collaborative filtering, content-based filtering, or hybrid filtering [4].

Collaborative filtering is a commonly used method in recommendation systems, utilizing ratings from other users with similar preferences [5]. The user-item matrix used in collaborative filtering can become large and scattered, leading to the sparsity problem, where there is limited data on users, and the cold start problem.

Content-Based Filtering is a method that utilizes item descriptions to recommend similar items based on user preferences [7]. Since content-based filtering is based on item data, it does not require data from other users, making it easier to scale to a large number of users [8] and can address the cold start problem for users.

To address the limitations of these two methods, they can be combined in a hybrid filtering approach [6]. Hybrid filtering optimizes recommendation systems by leveraging the strengths and compensating for the weaknesses of both methods. There have been several studies conducted on hybrid filtering-based recommendation systems in the past five years.
In reference [15], a study was conducted on a mobile-based recommendation system for tour packages using CBR (Case-Based Reasoning). The hybrid method used in this research involved the utilization of Naive Bayes, Bayes

Theorem, and Dempster-Shafer. Naive Bayes was employed to calculate probabilities such as age and visit frequency, while Bayes Theorem was used to calculate probabilities related to country, gender, and tourist destinations to be visited. Dempster-Shafer was utilized to determine their combination. The research reported an accuracy rate of 95% and an error rate of 5% based on the conducted accuracy evaluation.

In reference [16], a study was conducted on a movie recommendation system using a hybrid approach with K-Nearest Neighbours (KNN) and Restricted Boltzmann Machine (RBM). The author employed the Weighted Average Rating hybrid approach on the MovieLens dataset. In the Content-Based method using KNN, the researcher sought similarities between movies based on attributes such as release year, genre, box office hit rate, and film descriptions. The research reported an RMSE (Root Mean Square Error) value of 1.1155 for the ML100k dataset and 1.0652 for the ML1M dataset.

In reference [17], a study was conducted on a hybrid recommendation system based on association rules using the MovieLens dataset. The researchers utilized content-based association rule mining and employed the WEKA software to generate association rules and perform the necessary data mining for implementation and testing. The study utilized a hybrid approach called Feature Augmentation, where association rules served as the augmented profile, and Content-Based TF-IDF was used as the primary recommendation method. The research reported an MAE (Mean Absolute Error) value of 0.871 and an RMSE (Root Mean Square Error) value of 1.104.

In reference [18], a study was conducted on a recommendation system using hybrid deep collaborative filtering with Singular Value Decomposition (SVD) and Restricted Boltzmann Machine (RBM) on the MovieLens dataset. In this research, SVD was applied to the user-item matrix to decompose it into the best low-rank approximation of the original matrix. Then, the user-item matrix was embedded using a deep neural network called Restricted Boltzmann Machine (RBM). Finally, the two aforementioned methods were separately performed, and a weighted hybridization process was employed to combine the results and obtain the final recommendations in the recommendation system. The research reported MAE (Mean Absolute Error) and RMSE (Root Mean Square Error) values for the MovieLens 100k dataset, with an MAE of 0.6699 and an RMSE of 0.9577. For the MovieLens 1M dataset, the obtained values were an MAE of 0.6187 and an RMSE of 0.8931.

In this paper, the Restricted Boltzmann Machine (RBM) algorithm is used because it is effective in encoding distributions and computationally efficient. Additionally, the activations in the hidden layers can be used as features to improve model performance [9]. RBM is utilized to build a model by leveraging the values of each film in the Movie Lens dataset. For content-based filtering, Word2Vec is used as it can capture the semantic meaning of words in film descriptions. Word2Vec can generate high-quality word embeddings from large corpora quickly and can be employed in neural networks to understand natural language [10].

## II. RESEARCH METHODOLOGY

The objective of this final project is to implement a hybrid method in the recommendation system to generate Top-N items and evaluate the performance of the developed recommendation system. After loading the dataset, the next step involves pre-processing the data, which will then be split into two sets: the training set and the test set. The training set will be utilized to train the model.

The test set, on the other hand, will be used to measure the accuracy and performance of the built model. This project will utilize a hybrid method by combining content-based filtering and collaborative filtering. For content-based filtering, the Word2Vec algorithm will be used, specifically the Skip-Gram model. As for collaborative filtering, the Restricted Boltzmann Machine (RBM), a deep learning algorithm, will be employed. The hybrid process will utilize the Cascade method, where the RBM model will be implemented first, and the results provided by the RBM model will be passed on to the Word2Vec model, as depicted in Fig 1.

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 1, Maret 2024, Pp. 231-241

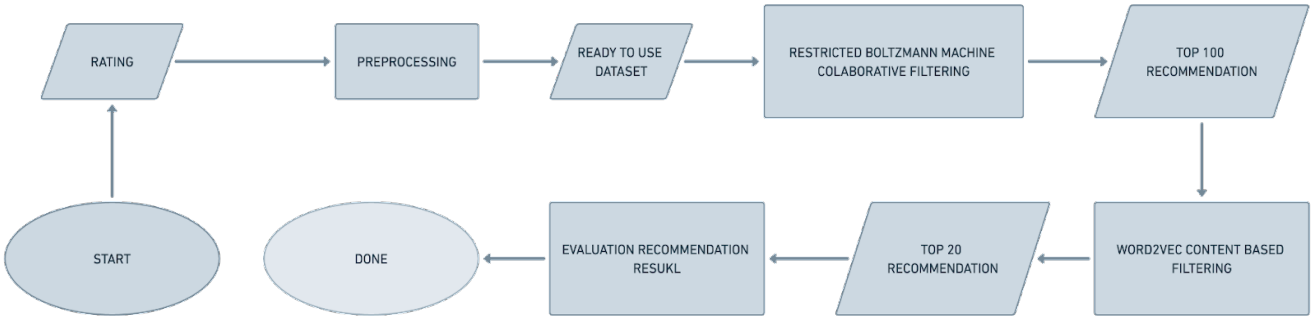


Fig. 1. Recommendation design system

## A. *Dataset*

The dataset utilized in this final project comprises movies obtained from the Movie Lens dataset, which is accessible on the website https://grouplens.org/datasets/movielens/1m/. Since this dataset does not include film descriptions, the film descriptions will be scraped from the website https://www.imdb.com/ using the IMDb ID attribute from the Movie Lens dataset. The process of scraping can be visualized in Fig 2.

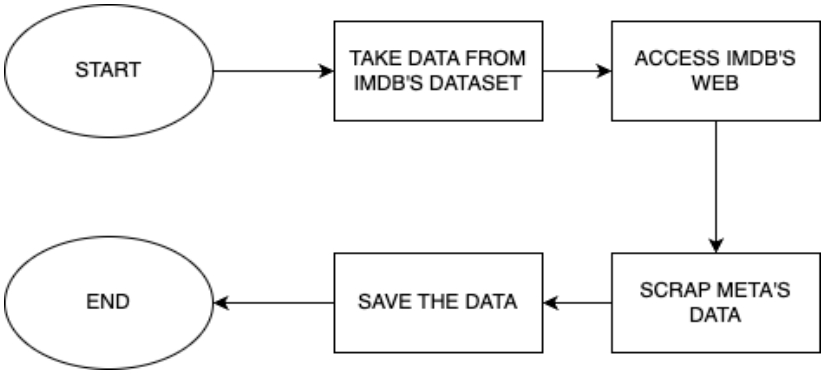


Fig. 2. Scrapping meta data from imdb's website

The first step in the scraping process is to retrieve data from the Movie Lens dataset, specifically the column that contains the IMDb ID. Then, you would access the IMDb website using the IMDb ID as a parameter. Next, you would extract the metadata by parsing the HTML tags. Finally, you would save the scraped data into a CSV file.

The metadata obtained from the scraping process typically includes information such as the film's genre, producer, language, and actors/actresses involved in the film.

TABLE I
MOVIES DATA

| Movie ID | Title | Genres |
|---|---|---|
| 1 | Toy Story (1995) | Adventure\|Animation\|Children\|Comedy\|Fantasy |
| 2 | Jumanji (1995) | Adventure\|Children\|Fantasy |
| 3 | Grumpier Old Man (1995) | Comedy\|Romance |

TABLE II
RATINGS DATA

| User ID | Imbd ID | Tmbd ID |
|---|---|---|
| 1 | 114709 | 121 |
| 2 | 113497 | 8844 |
| 3 | 113228 | 15602 |

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 1, Maret 2024, Pp. 231-241

TABLE III
LINKS DATA

| MOVIE ID | IMBD ID | TMBD ID |
|---|---|---|
| 1 | 114709 | 121 |
| 2 | 113497 | 8844 |
| 3 | 113228 | 15602 |

TABLE IV
DESCRIPTION DATA

| Movie ID | Metadata |
|---|---|
| 1 | Action English Jean-ClaudeVanDamme PowersBoothe RaymondJ.Barry WhittniWright RossMalinger DorianHarewood KateMcNeil MichaelGaston AudraLindley BrianDelate SteveAronson MichaelR.Aubele KarenEliseBaldwin JenniferD.Bowser PatBrisson GlendaMorganBrown JopheryC.Brown WilliamCameron |

## B. *Preprocessing*

In the content-based filtering method, pre-processing will be performed on the film descriptions. The data will be processed through several stages, including punctuation removal, case folding, stop words removal, and tokenizing.

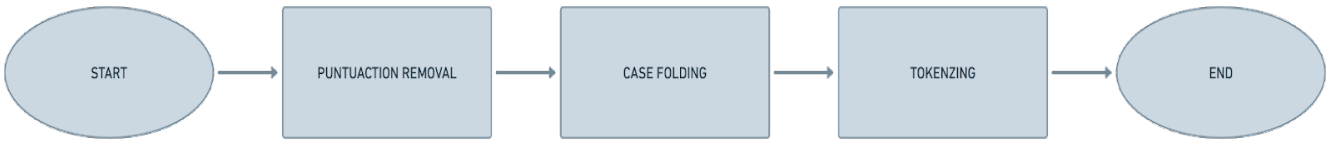


Fig. 3. Pre-processing workflow

### 1) *Punctuation Removal*

In this stage, special characters and punctuation marks will be removed.

TABLE V
PUNCTUATION REMOVAL EXAMPLE

| Input | Output |
|---|---|
| Action English Jean-ClaudeVanDamme PowersBoothe RaymondJ.Barry WhittniWright RossMalinger DorianHarewood KateMcNeil MichaelGaston AudraLindley BrianDelate SteveAronson MichaelR.Aubele KarenEliseBaldwin JenniferD.Bowser PatBrisson GlendaMorganBrown JopheryC.Brown WilliamCameron BernardCanepari JayCaufield | Action English Jean ClaudeVanDamme PowersBoothe RaymondJ Barry WhittniWright RossMalinger DorianHarewood KateMcNeil MichaelGaston AudraLindley BrianDelate SteveAronson MichaelR Aubele KarenEliseBaldwin JenniferD Bowser PatBrisson GlendaMorganBrown JopheryC Brown WilliamCameron BernardCanepari JayCaufield' |

### 2) *Case Folding*

The next step is case folding, where the letters will be converted to lowercase. In this task, lowercase conversion or lowercase transformation will be applied.

TABLE VI
CASE FOLDING EXAMPLE

| Input | Output |
|---|---|
| Action English JeanClaudeVanDamme PowersBoothe RaymondJBarry WhittniWright RossMalinger DorianHarewood KateMcNeil MichaelGaston AudraLindley BrianDelate SteveAronson MichaelR.Aubele KarenEliseBaldwin JenniferDBowser PatBrisson GlendaMorganBrown JopheryC Brown WilliamCameron BernardCanepari JayCaufield | action english jeanclaudevandamme powersboothe raymondjbarry whittniwright rossmalinger dorianharewood katemcneil michaelgaston audralindley briandelate stevearonson michaelr.aubele karenelisebaldwin jenniferdbowser patbrisson glendamorganbrown jopheryc brown williamcameron bernardcanepari jaycaufield |

### 3) *Tokenizing*

In the pre-processing stage of content-based filtering, word tokenizing refers to the process of splitting a text into individual words or tokens[13].

TABLE VII
TOKENIZING EXAMPLE

| Input | Output |
|---|---|
| action english jeanclaudevandamme powersboothe raymondjbarry whittniwright rossmalinger dorianharewood katemcneil michaelgaston audralindley briandelate stevearonson michaelr.aubele karenelisebaldwin jenniferdbowser patbrisson glendamorganbrown jopheryc brown williamcameron bernardcanepari jaycaufield | [‘action’, ‘english’, ‘jeanclaudevandamme’, ‘powersboothe’, ‘raymondjbarry’, ‘whittniwright’, ‘ rossmalinger’, ‘dorianharewood’, ‘katemcneil’, ‘michaelgaston’, ‘audralindley’, ‘briandelate’, ‘stevearonson’, ‘michaelraubele’, ‘karenelisebaldwin’, ‘jenniferdbowser’, ‘patbrisson’, ‘glendamorganbrown’, ‘jopheryc’, ‘brown’, ‘williamcameron’, ‘bernardcanepari’, ‘jaycaufield’] |

### *C. Data Splitting*

After the preprocessing process, the next step is data splitting, where the data is divided into two parts: the training data and the testing data. The training data is used to train the model, while the testing data is used to evaluate the performance of the trained model. The data splitting is typically done with a ratio of 80% for training data and 20% for testing data.

### *D. Model Training*

After the data splitting process, the subsequent step involves training the hybrid model using the training data. The hybrid model combines the Word2Vec algorithm and the Restricted Boltzmann Machine (RBM). Word2Vec will be used in the content-based filtering method, which utilizes the data from the film descriptions. On the other hand, the Restricted Boltzmann Machine will be used in the collaborative filtering method, which relies on user ratings data.

The training process will start with training the Word2Vec model. Once the output from the Word2Vec model is obtained, the training will continue with the Restricted Boltzmann Machine. The output from the Word2Vec model will be used as input for the training of the RBM model.

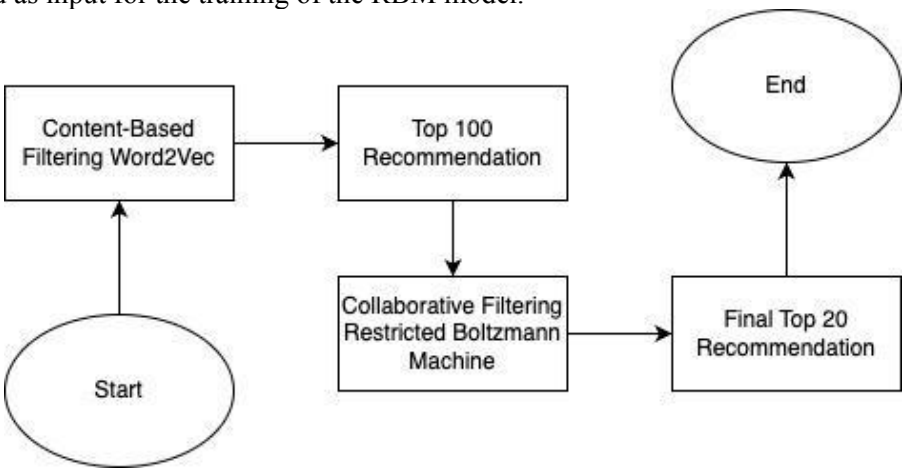


Fig. 4. The Working principle of model training

#### *1) Word2Vec Model*

The initial step is to create user profiles by extracting the film descriptions from movies that have received a rating of >= 3. Once the user profiles are acquired, the following step involves comparing the similarity between the user profiles and the recommendations generated by the RBM model. The recommendations will be sorted based on the similarity scores between the user profile and each film provided by the RBM model. In the process of creating the Word2Vec model, default parameters from the Gensim library will be used.

#### *2) Restricted Boltzmann Machine*

In this process, the training data is used in the collaborative filtering method, leveraging user ratings data and the Restricted Boltzmann Machine (RBM) algorithm. RBM works by creating two models known as the hidden layer and the visible layer. The hidden layer is responsible for learning features from the information inputted to the visible layer. In this case, the visible layer will contain x neurons, where x is the number of films in the dataset. Each neuron will hold a normalized value between 0 and 1, where a value of 0 means the user has not watched that film, and a value close to 1 indicates the user's preference for the film represented by the neuron.

Once the input is provided, the RBM model will undergo training, utilizing the hidden layer to learn features. These features will be used to reconstruct the input data, specifically to predict ratings for films that the user has not watched, which will be recommended to the user.

### *E. Model Testing*

After successfully training the model, the next step is to perform testing on the trained model. The purpose of model testing is to measure how well the model performs and evaluate the quality of its output. Once testing is completed, the results of the testing process will be obtained. These results will then be further evaluated to assess the performance and effectiveness of the model.

### *F. Evaluation*

#### *1) Precision*

Precision is a measure used in model testing to determine how accurately the model can predict data classified as the positive class [5]. Here is the formula for precision. In this thesis, relevant movies are assumed to be movies that have been rated > 3 by users or movies of the same genre as what the user has rated > 3.

$$Precision = \frac{TP}{TP + FP}$$

- TP (True Positive) = The value of the rating successfully predicted and classified as a suitable recommendation for the user's preference.
- FP (False Positive)= The value of the rating that was not successfully predicted and classified as an unsuitable recommendation for the user's preference.

#### *2) NDCG*

Cumulative Gain (CG) is an evaluation metric used in recommendation systems to measure the quality of recommendations based on the number of recommended items that are relevant to the user. This metric calculates the relevance of each recommended item and accumulates them to determine the overall quality of the recommendation list [15].

$$CG = \sum_{i=1}^{n} relevance_i$$

- relevance = the relevance item given by recommendation system
- n = the total of item given by recommendation system

Discounted Cumulative Gain (DCG) is a quantitative measure of ranking quality in recommendation systems. DCG measures the cumulative relative value of a set of recommendation results based on their rankings [14].

$$DCG = \sum_{i=1}^{n} \frac{2^{relevance_i}}{log_2(i+1)}$$

- relevance = the relevance item given by recommendation system
- n = the total of item given by recommendation system

NDCG (Normalized Discounted Cumulative Gain) is an evaluation metric for recommendation systems that measures the quality of item recommendations based on their ranking order. NDCG takes into account the position of the item in the recommendation list and the relevance of the item to the user's preferences [14]. In this thesis, the ranking is calculated based on the average ratings of each film.

$$NDCG = \frac{DCG}{relevance(DCG)}$$

## III. EXPERIMENT SETTINGS

To optimize the performance of our hybrid model, which combines Word2Vec and Restricted Boltzmann Machine (RBM), we will conduct a series of tests with different parameter variations. The purpose of these tests is to assess the impact of these parameters on the quality of the generated recommendations. The parameters to be tested include:

### A. *Minimal Count*

This parameter determines the minimum number of occurrences a word must have in the dataset to be considered when building the word vectors. Words that appear less than the specified “minimal count” value will be ignored during the vector construction process. Lower values for “minimal count” will include more words in the vector building process, while higher values will filter out less common words [11]. In this test, the “minimal count” value will be varied between 1 and 3.

### B. *Vector Size*

This parameter determines the dimensionality of the vectors generated for each word in the Word2Vec model. A larger "vector size" value allows more information to be represented by each word vector [11]. In this test, the "vector size" value will be varied between 500 and 1200 to assess the impact of different vector dimensions on the quality of the recommendations

### C. *Epoch*

Epoch refers to the number of iterations or cycles performed when training the Word2Vec model. In each epoch, the model updates its internal parameters based on the training data. More epochs lead to a deeper understanding of the text used by the model [11]. In this test, the number of epochs will be varied between 30 and 50 to examine the effect of different training iterations on the quality of the recommendations

### D. *Hidden Units*

Hidden units in an RBM are intermediate nodes that capture higher-level features or latent representations of the input data. The number of hidden units is a hyperparameter that influences the model's capacity and ability to capture complex patterns [12].In this test, the number of hidden units will be varied between 128 and 500 to explore the impact of different hidden unit configurations on the quality of the recommendations .

### E. *Learning Rate*

The learning rate in an RBM determines the magnitude of weight updates during training. It is a hyperparameter that influences the convergence speed and stability of the training process, and its selection should be carefully considered to achieve optimal performance[12]. In this test, the learning rate will be varied between 0.8 and 1.0 to examine the effect of different learning rates on the quality of the recommendations.

### F. *N First Recommendation*

This parameter refers to the number of recommendations that will be generated by the first model, which will determine the input for the second model's recommendations. In this test, the number of recommendations from the first model will be varied between 750 and 1000. The choice of this parameter will impact the amount of information provided to the second model and can influence the overall quality and diversity of the final recommendations.

We will conduct testing using the aforementioned parameters in 5 different scenarios, with the respective parameter values listed in Table VIII.

TABLE VIII
PARAMETERS EXPERIMENT

| Scenario | Minimal Count | Vector Size | Epoch | Hidden Units | Learning Rate | N 1st Recommendation |
|---|---|---|---|---|---|---|
| 1 | 1 | 500 | 30 | 128 | 1 | 750 |
| 2 | 2 | 750 | 50 | 200 | 1 | 750 |
| 3 | 3 | 1000 | 50 | 300 | 1 | 750 |
| 4 | 3 | 1200 | 50 | 400 | 0.8 | 1000 |
| 5 | 1 | 900 | 30 | 500 | 1 | 1000 |

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 1, Maret 2024, Pp. 231-241

## IV. RESULT

TABLE IX
AVERAGE RESULT WITH RANDOM PARAMETERS

| Trials | Precision@20 | | NDCG@20 | |
|---|---|---|---|---|
| | Word2Vec -RBM | RBM-Word2Vec | Word2Vec -RBM | RBM-Word2Vec |
| 1 | 0.783 | 0.517 | 0.755 | 0.497 |
| 2 | 0.781 | 0.471 | 0.775 | 0.471 |
| 3 | 0.817 | 0.495 | 0.749 | 0.495 |
| 4 | 0.815 | 0.471 | 0.752 | 0.471 |
| 5 | 0.820 | 0.528 | 0.722 | 0.528 |

From the results shown in Table IX using random parameters, it can be seen that the average of both evaluation metrics, precision@20 and ndcg@20, has better values when using the hybrid cascade Word2Vec-RBM method compared to RBM-Word2Vec. The average precision@20 value is 0.6032, and the average ndcg@20 value is 0.7506.

The next experiment will involve finding the best combination of parameters from the three parameters tested in the previous experiment. The parameter combination to be tested includes the minimum count with options 1, 2, and 3, the vector size with options 500, 750, and 1000, and the hidden units with options 200, 300, and 400.

To minimize the experimental process, the first step is to find the best minimum count parameter using a vector size of 500 and hidden units of 200. The evaluation results for these metrics can be seen in Table 4.2.2.

In the table, it can be observed that the best minimum count value among the three tested values is 1, which yields the highest value among them. The precision@20 value is 0.7820914049299273, and the ndcg@20 value is 0.765315304318294.

TABLE X
AVERAGE RESULT TO GET BEST MINIMAL COUNT

| Minimal Count | Vector Size | Hidden Units | Result | |
|---|---|---|---|---|
| | | | Precission@20 | Ndcg@20 |
| 1 | 500 | 200 | 0,7820914049299273 | 0,765315304318294 |
| 2 | 500 | 200 | 0,7763730767710327 | 0,7599673847166963 |
| 3 | 500 | 200 | 0,7765110114993077 | 0,7668624814179662 |

The next step is to find the best vector size parameter among the three parameters tested, with a minimum count of 1 and hidden units of 200.

Looking at Table XI, the evaluation results for the metrics with a vector size of 500 have the highest values for precision@20 and ndcg@20. The precision@20 value is 0.7820914049299273, and the ndcg@20 value is 0.765315304318294.

TABLE IX
AVERAGE RESULT TO GET BEST MINIMAL COUNT

| Minimal Count | Vector Size | Hidden Units | Result | |
|---|---|---|---|---|
| | | | Precission@20 | Ndcg@20 |
| 1 | 500 | 200 | 0,7820914049299273 | 0,765315304318294 |
| 2 | 750 | 200 | 0,7813155695684347 | 0,7702083295975796 |
| 3 | 1000 | 200 | 0,7799884079196612 | 0,7624888986480376 |

JIPI (Jurnal Ilmiah Penelitian dan Pembelajaran Informatika)
Journal homepage: https://jurnal.stkippgritulungagung.ac.id/index.php/jipi
ISSN: 2540-8984
Vol. 9, No. 1, Maret 2024, Pp. 231-241

Then, the final step is to find the best hidden units parameter among the three values, considering the previously determined best minimum count and vector size values.

Looking at Table X, a hidden units value of 400 has the highest precision@20 value compared to the others, with a value of 0.8008083324364076. However, for the ndcg@20 metric, the highest value is obtained with a hidden units value of 200, which is 0.765315304318294.

TABLE X
AVERAGE RESULT TO GET BEST MINIMAL COUNT

| Minimal Count | Vector Size | Hidden Units | Result | |
|---|---|---|---|---|
| | | | Precission@20 | Ndcg@20 |
| 1 | 500 | 200 | 0,7820914049299273 | 0,765315304318294 |
| 2 | 750 | 300 | 0,7940056289941453 | 0,7577206401521747 |
| 3 | 1000 | 400 | 0,8008083324364076 | 0,7547546591460704 |

In [19] , the paper introduces an innovative non-IID framework for collaborative filtering using RBMs, incorporating user-user and item-item correlations. The use of real numbers in the visible layer and the integration of self-generated data contribute to improved recommendation accuracy. As shown at figure 5 it revealed that the use of multinomial visible layers outperformed real-valued visible layers. While the real-valued model initially performed better, it suffered from overfitting and fell behind the multinomial model as training progressed. Therefore, it is recommended to prefer the multinomial model due to its better performance and robustness. These findings highlight the importance of selecting appropriate visible layer representations in RBM-based recommendation systems for optimal results.

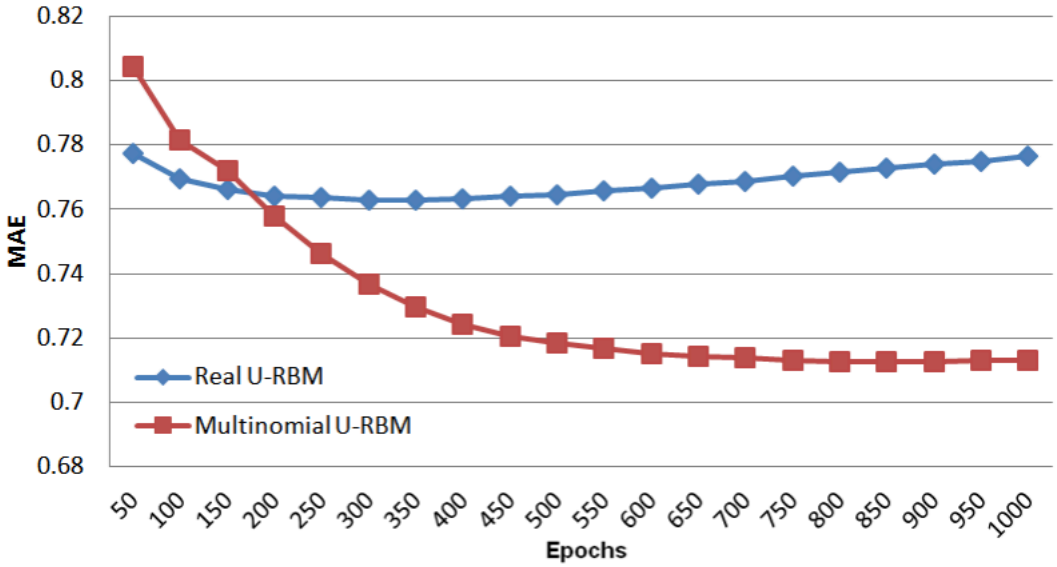


Fig. 5. MAE result of RBM in paper [x]

In [20] , paper evaluates the performance of Word2Vec, a popular word embedding technique, on the MovieLens dataset. The authors compare Word2Vec against User-to-User Collaborative Filtering (U2U) and Item-to-Item Collaborative Filtering (I2I) as baseline algorithms. The evaluation is conducted using a 5-fold cross-validation on the MovieLens dataset. Word2Vec is evaluated with two different vector sizes: 300 and 500. The performance is assessed using various metrics.

The results shown in table XI indicate that Word2Vec outperforms both U2U and I2I in terms of accuracy for most of the metrics considered. The size of the Word2Vec vectors does not significantly affect the overall

performance. These findings highlight the effectiveness of Word2Vec in capturing user-item correlations and making accurate recommendations in collaborative filtering scenarios.

TABLE XI
F1 SCORE RESULT CONTENT-BASED USING WORD2VEC IN PAPER [Y]

| Vector Size | Word2Vec | | RI | | LSI | |
|---|---|---|---|---|---|---|
| | 300 | 500 | 300 | 500 | 300 | 500 |
| F1@5 | 0.5056 | 0.5054 | 0.4921 | 0.4910 | 0.4645 | 0.4715 |
| F1@10 | 0.5757 | 0.5751 | 0.5622 | 0.5613 | 0.5393 | 0.5469 |
| F1@15 | 0.5672 | 0.5674 | 0.5349 | 0.5352 | 0.5187 | 0.5254 |

## V. CONCLUSION AND FUTURE WORK

Based on the conducted experiments, it can be concluded that the hybrid cascade approach, specifically the Word2Vec-RBM method, outperforms the RBM-Word2Vec method in terms of precision@20 and ndcg@20 metrics. The average precision@20 value for Word2Vec-RBM is 0.6032, compared to 0.4964 for RBM-Word2Vec. Similarly, the average ndcg@20 value for Word2Vec-RBM is 0.7506, while it is 0.4924 for RBM-Word2Vec. Therefore, the hybrid cascade approach, incorporating Word2Vec followed by RBM, is recommended for further research. Additionally, the study highlights the importance of parameter selection, indicating that a minimum count value of 1, a vector size of 500, and 400 hidden units yield optimal performance for precision@20 and ndcg@20. Further exploration of parameter combinations, evaluation with different datasets, refining the model architecture, and incorporating user feedback can contribute to enhancing the recommendation system's performance and user satisfaction.

## REFERENCES

[1] M. Techlab, "Medium," *Medium*, Aug. 25, 2023. https://medium.com/geekculture/how-can-product-recommendation-system-benefit-your-business-65afd9cabfd8. (accessed Nov. 22, 2022).
[2] A. Singhal, P. Sinha, and R. Pant, "Use of Deep Learning in Modern Recommendation System: A Summary of Recent Works," *International Journal of Computer Applications*, vol. 180, no. 7, pp. 17–22, Dec. 2017, doi: https://doi.org/10.5120/ijca2017916055.
[3] A. Singhal, R. Kasturi, V. Sivakumar, and J. Srivastava, "Leveraging Web Intelligence for Finding Interesting Research Datasets," *Leveraging Web Intelligence for Finding Interesting Research Datasets*, vol. 1, Nov. 2013, doi: https://doi.org/10.1109/wi-iat.2013.46.
[4] F. O. Isinkaye, Y. O. Folajimi, and B. A. Ojokoh, "Recommendation systems: Principles, methods and evaluation," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 261–273, Nov. 2015, doi: https://doi.org/10.1016/j.eij.2015.06.005.
[5] Z. Xiang and U. Gretzel, "Role of Social Media in Online Travel Information Search," *Tourism Management*, vol. 31, no. 2, pp. 179–188, Apr. 2010, doi: https://doi.org/10.1016/j.tourman.2009.02.016.
[6] D. Kotkov, Jari Veijalainen, and S. Wang, "Challenges of Serendipity in Recommender Systems," *Challenges of Serendipity in Recommender Systems*, vol. 2, Jan. 2016, doi: https://doi.org/10.5220/0005879802510256.
[7] M. J. Pazzani and D. Billsus, "Content-Based Recommendation Systems," *The Adaptive Web*, pp. 325–341, 2007, doi: https://doi.org/10.1007/978-3-540-72079-9_10.
[8]"Content-based Filtering Advantages & Disadvantages," *Google Developers*, Jul. 18, 2022. https://developers.google.com/machine-learning/recommendation/content-based/summary
[9] S. Gupta, "A Literature Review on Recommendation Systems," *A Literature Review on Recommendation Systems*, vol. 07, no. 09, p. 3602, Sep. 2020, Accessed: Dec. 04, 2022. [Online]. Available: https://www.irjet.net/archives/V7/i9/IRJET-V7I9633.pdf
[10] "What Is Word2Vec?" ," *Pine Code*. www.pinecone.io/learn/roughly-explained/what-is-word2vec (accessed Dec. 06, 2022).
[11] R. Řehůřek, "gensim: topic modelling for humans," *radimrehurek.com*, Dec. 21, 2021. https://radimrehurek.com/gensim/models/word2vec.html (accessed Dec. 09, 2022).
[12] A. Fischer, "Training Restricted Boltzmann Machines," *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 441–444, May 2015, doi: https://doi.org/10.1007/s13218-015-0371-2.
[13] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based Recommender Systems: State of the Art and Trends," *Recommender Systems Handbook*, pp. 73–105, Oct. 2010, doi: https://doi.org/10.1007/978-0-387-85820-3_3.
[14] A. Dhinakaran, "Demystifying NDCG," *Medium*, Feb. 02, 2023. https://towardsdatascience.com/demystifying-ndcg-bee3be58cfe0 (accessed Jan. 09, 2023).
[15] X. WANG, F. LUO, C. SANG, J. ZENG, and S. HIROKAWA, "Personalized Movie Recommendation System Based on Support Vector Machine and Improved Particle Swarm Optimization," *IEICE Transactions on Information and Systems*, vol. E100.D, no. 2, pp. 285–293, 2017, doi: https://doi.org/10.1587/transinf.2016edp7054.
[16] D. K. Behera, M. Das, S. Swetanisha, and P. K. Sethy, "Hybrid model for movie recommendation system using content K-nearest neighbors and restricted Boltzmann machine," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, p. 445, Jul. 2021, doi: https://doi.org/10.11591/ijeecs.v23.i1.pp445-452.
[17] A. Alsalama, "A Hybrid Recommendation System Based on Association Rules," *TopSCHOLAR®*, 2013. https://digitalcommons.wku.edu/theses/1250 (accessed Jun. 27, 2023).
[18] K. R and M. J. A. S, "A Hybrid Deep Collaborative Filtering Approach for Recommender Systems," *A Hybrid Deep Collaborative Filtering Approach for Recommender Systems*, Jun. 2021, doi: https://doi.org/10.21203/rs.3.rs-651522/v1.
[19] K. B. Georgiev and Preslav Nakov, "A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines," *A non-IID Framework for Collaborative Filtering with Restricted Boltzmann Machines*, pp. 1148–1156, Jun. 2013.
[20] C. Musto, G. Semeraro, Marco de Gemmis, and P. Lops, "Word Embedding techniques for Content-based Recommender Systems: An Empirical Evaluation," *Word Embedding Techniques for Content-based Recommender Systems: An Empirical Evaluation*, vol. 1441, Jan. 2015.