

# ExMrec2vec: Explainable Movie Recommender System based on Word2vec

Amina SAMIH, Abderrahim GHADI, Abdelhadi FENNAN  
Faculty of Sciences and TechniquesList Laboratory  
University Abdelmalek EssaadiTangier-Morocco

**Abstract**—According to the user profile, a recommender system intends to offer items to the user that may interest him. The recommendations have been applied successfully in various fields. Recommended items include movies, books, travel and tourism services, friends, research articles, research queries, and much more. Hence the presence of recommender systems in many areas, in particular, movies recommendations. Most current Machine Learning recommender systems serve as black boxes that do not provide the user with any insight into or justification for the system's logic. What puts users at risk of losing their confidence. Recommender systems suffer from an overload of information, which poses numerous problems, including high cost, slow data processing, and low time complexity. That is why researchers in have been using graph embeddings algorithms in the recommendation field to reduce the quantity of data, as these algorithms have been successful in the last few years. This work aims to improve the quality of recommendation and the simplicity of recommendation explanation based on the word2vec graph embeddings model.

**Keywords**—Recommender system; explainable artificial intelligence machine learning; Word2vec

## I. INTRODUCTION

With the advent of the internet today, we are witnessing an enormous information overload. This exponential growth in data results in difficulty organizing and analyzing this basic information but opens up new avenues on the paths of knowledge. The question is no longer to have the information but to find the relevant information simultaneously; from there, recommendation systems were born [1]. To manage the situation of information overload, graphs embeddings [2] appears as a solution that aims to present the properties of graphs in a vector or a set of vectors in a low-dimensional space while preserving the graph topology much as possible that offers more accurate recommendations. Graph embeddings can be classified into three main categories [3]: factorization graph embedding, random-walk graph embedding and deep graph embedding. Recommendation systems play an essential role in improving the user experience of online services to reduce the efforts of humans in the search for objects of interest. Examples of these systems include recommending a set of items such as Amazon products, advertisements from products presented by Google based on search history, movie recommendations from Netflix, Yahoo, Movies, and Movielens [4] [32].

It often happens that some recommended articles are not expected for users and thus cause confusion. Explainability, therefore, becomes a criterion closely linked to the quality of

the recommendation system. Indeed, supporting the recommendations with explanations justifying the proposals helps to improve efficiency, transparency, and user satisfaction, which allows for better user loyalty. The literature offers some methods to generate explanations of recommendations; however, these methods have several limitations.

Explanations are not easy to obtain regardless of their importance. Indeed, with the recent success of the machine learning neural networks, the Recommender system relies increasingly on these algorithms because of the black-box nature of machine learning algorithms to improve their performance and predictions for transparency [4].

An emerging field called Explainable Artificial Intelligence [5] recently discussed the problem of deep learning explicitness and other transparent machine learning algorithms by providing methods that generate high-quality predictions as well as intuitive explanations of the results.

Therefore, being aware of the importance of addressing information overload and explainability issues in recommender systems and under the background of the word2vec model and explainable artificial intelligence, we propose in this paper a word2vec-based approach for the explainable recommendation. Precisely, we extract relationships between movies from user history to build the vocabulary of the word2vec technique to make recommendations. Then we match these recommendations with a human-friendly explanation style. The rest of the paper is organized as follows:

- 1) We introduce the basics of this work, Recommender systems, Word2vec model, and Explainable Artificial Intelligence.
- 2) We analyze related works in this area.
- 3) We present our proposed approach, and we discuss how it helps present a simple explanation for a normal user through an illustrative example. We discuss experiments made to evaluate our proposition.
- 4) We present our conclusions and directions for the future.

## II. RECOMMENDER SYSTEMS

Currently, there are many aspects of modern life, and we are being overwhelmed by this information large quantity, or this reason, recommender systems have been helpful since their birth in the mid1990s to address this wealth of data by suggesting to an active user in the field of his interests a list of

objects or a unique object [6]. There are three main recommendation approaches (see Fig. 1) content-based recommendation, collaborative filtering-based recommendation, and hybrid recommendation [7].

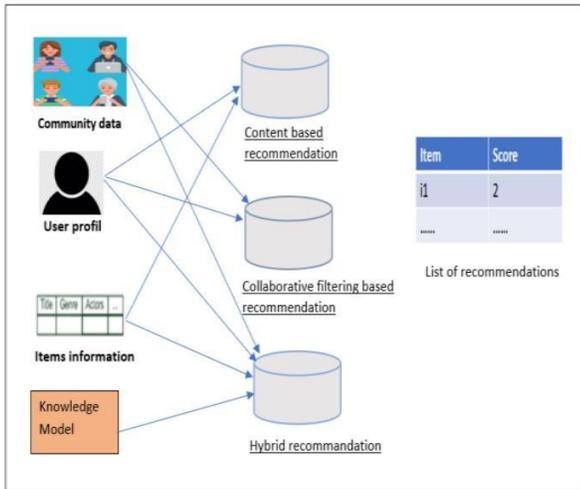


Fig. 1. Major Recommendation Approaches.

#### A. Content-based Recommendation

This approach works by analyzing the content of an item to compare it to a profile that describes the user's interests. For example, if the item description describes a certain kind of music, the system will compare that description to the user's profile to predict its usefulness [4]. A content-based recommendation system is a filtering system that uses elements description and user data to identify to users items that are most likely to appeal to them.

That is to say. It is an approach that uses the content of documents to compare them to a profile's themes. The system then learns about the user's interests and uses these details to predict the usefulness of a given document. For example, if the user likes or dislikes specific themes or attributes, the system will suggest a document with these characteristics [8]. Content-based recommendations have several advantages and disadvantages. The strengths of this approach are:

- User autonomy: Content-based recommendation techniques treat users independently. Thus, only the evaluations of each user are taken into account to build their user profile and make the recommendation [9].
- Immediate consideration of a new item: the recommendation of items based on content-based filtering is not necessarily linked to the evaluation by a user; the item can be recommended without being evaluated [10].

However, this approach tackles many challenges:

- Content analysis limited: this requires detailed descriptions of the items and a highly organized user profile to produce recommendations. Unfortunately, that does not always happen.

- Subspecialization of content: Users shall only receive recommendations similar to items already defined in their profiles.
- No immediate embarkation of a new user: before the system can interpret its preferences, a user must evaluate several items and make the appropriate recommendations.

Several content-based filtering solutions are available [11] we cited:

- News Dude: a personal data system using the synthesized language to read articles of users.
- LIBRA: is a book recommendation system based on the content that uses web-based book information. The system can explain any recommendation made to users that will enable them to trust the system's recommendations fully.
- Pandora: is a web-based radio (available in the United States only) which incorporates a Music Genome Project automated music recommendation service. Where the user first specifies an artist's name or a music title. The system then handles the rest of the music list and selects the titles closer to the first choice musically.

#### B. Collaborative Filtering based Recommendation

This method uses the history of ratings to predict future interests. It assumes that various people have common interests in multiple products, then those interests are likely to extend to other products [12]. Collaborative filtering is a recommendation algorithm that predicts articles (whatever they are, books, films, press articles) that users will appreciate in the future. This approach aims to predict what a person will like in the future. The algorithm uses the history of that user and all the existing information in the system about other users. These algorithms can detect users who have similar tastes to exploit this information for recommendations [1].

For example, suppose it turns out that Amina, Mouna, and Imad have liked similar songs in the past and that Amina listens or gives a favorable opinion on the last song of Salma Rachid. In that case, Mouna and Imad likely like it too, and it is relevant to recommend it to them. We can use the data in a collaborative filtering system in several ways. These methodologies are categorized into two prominent families: algorithms based on memory and models [4]. There are several advantages to the collaborative filtering-based recommendation approach; the most important are [10]:

- Surprise effect: The user may receive an appropriate surprise recommendation that is often desirable for himself. For instance, if user a is close to user b because he is only watching comedies and if b loves a different genre, the system will recommend this film because of its proximity.
- Domain knowledge not required: collaborative recommendation is based only on item scores. Therefore, it does not require any knowledge of items.

However, using this recommendation approach can lead to several issues:

- Cold start problem: There is no sufficient user or item information to make relevant predictions in the recommendation system. The cold start problem is one of the most significant limitations, which lowers system performance. This new user's profile or article is empty because it is not rated; its taste is thus unknown.
- Sparsity: a situation where users evaluate a little of the total number of articles available in a database. This leads to a sparse matrix with a rate of missing values that can reach 95% of the articles' inability to locate successful neighbors and the generation of weak recommendations. In addition, the scarcity of data.
- The gray sheep problem: Users with different tastes out of the ordinary will not have many similar users. So it will not be easy to make relevant recommendations for this kind of user.

Among the most popular collaborative systems we cited [11]:

- Ringo: is a system based on collaborative filtering that recommends music albums and artists. This system gives a list of 125 artists to the user when this user first enters the system to rate them according to his preference.
- Amazon.com: is an example of an e-commerce recommendation engine that utilizes scalable
- collaborative peer-to-article filters to recommend online products for various product types. The algorithm of calculation evolves regardless of the user number and the object number.
- GroupLens: is a collaborative filtering system based on the client/server architecture; the system recommends News Usenet, a high-volume Internet discussion list service.

### C. Hybrid Recommendation

A hybrid recommendation is another class of recommendation that seeks to overcome the limitations of the previously discussed other approaches. It combines two or more different techniques of recommendation. Content-based and collaborative filtering combination is the most popular hybrid technique. That is, the content of items and the ratings of all users are used [13].

As the name implies, the hybrid-based recommendation combines one or more of the preceding methods [14]. In addition, it solves challenges like cold start and sparsity by combining components from other recommendation systems or relying on their reasoning [15].

This type of recommendation tries to overcome the shortcomings of the previous two approaches. It is a synthesis of two or more different recommendation techniques [1]. There are several ways to do hybridization, and no consensus has been defined by the research community [16] to

hybridize. However, Burke [17] identified seven different ways of hybridizing.

Among the most popular hybrid recommendation systems, we cited [11]:

- Netflix: an online movie service that allows users to rent movies for a monthly fee based on an inventory of priority movies they want to see. Films will be sent to users. If the user has finished watching the film, the system automatically returns the DVD by mail and mail the next DVD, free of post. The length of subscription for subscribers is related to the number of films they see and enjoy. If they do not find films that interest them, subscribers tend to drop out of service. Therefore, company needs to provide subscribers with movies they will love. Thus, the company promotes.
- Cinmatch: an automated system weekly analysis of cumulative films by using Pearson's correlation coefficient to all other movies to determine the list of "like" films that might appeal to users. It uses scores then calculates a multivariate regression, based on these correlations, based on the real-time online section of the system, to determine a unique, personalized prediction for each recommended film. If there is no personalized recommendation, the average score is used from all scores given to the film. These forecasts can be seen on the website through red stars.

### D. Other Sub Approaches of Recommendation

1) *Demographic recommendation*: this approach uses demographic records such as age, gender, education, etc., to identify classes of users. This approach does not suffer from the new person problem because he does not use reviews to furnish recommendations. [1][18].

2) *Recommendation based on knowledge [1]*: This strategy takes the knowledge on products, such as features and preferences explicitly requested by consumers.

3) *Context-aware recommendation [19]*: This approach generates more relevant recommendations through contextual information like time, location, and social data.

4) *Recommendation based on psychological knowledge [1]*: Recommendations based on psychological conditions like emotion, convincing, carefulness, presence, etc., can describe it.

## III. WORD2VEC

A well-known word embedding algorithm is Word2Vec [20]. It builds on two-layer neural networks to learn vector representations of the words composing the input text to represent close digital vectors for words that share similar contexts.

Word2Vec has two neural, CBOW and Skip-Gram architectures. The first receives a word context, i.e. the terms it entails in a sentence, as an input and tries to predict the word. The second word is used to indicate his context and to provide an input (see Fig. 2).

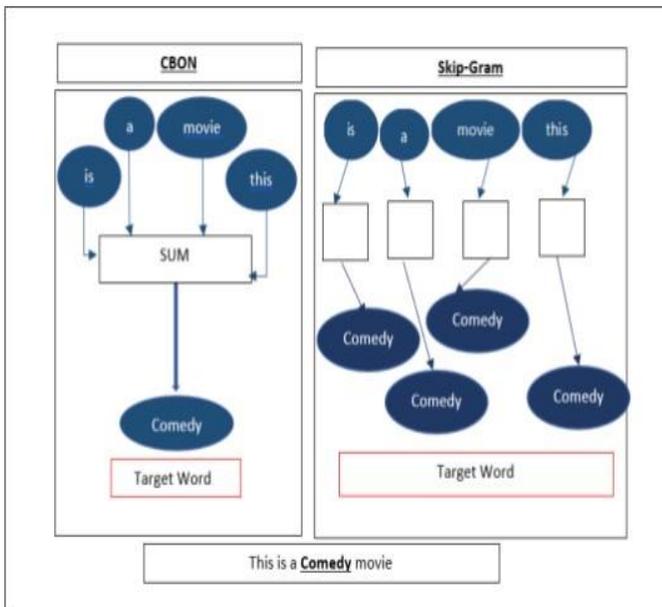


Fig. 2. CBOW and Skip-Gram Architectures of Word2vec Model.

#### IV. EXPLAINABLE ARTIFICIAL INTELLIGENCE

According to Lamy and al. [21], explainable artificial intelligence is an area that focuses on designing intelligent systems capable of explaining their outputs to humans, such as robots, artificial agents, planners, and machine learning algorithms. Several explanation methods and strategies, particularly for machine learning algorithms, have been proposed in the quest to make artificial intelligence systems explainable in a relatively short period. Methods of explaining are classified.[22] (see Fig. 3) in three categories:

1) *The complexity of interpretability*: The more complex the model is, the harder it is to interpret it. There is often a compromise between interpretability and accuracy. Therefore, we can classify explainable techniques in two ways according to the complexities of machine learning algorithms:

a) An inherently and intrinsically interpretable algorithm (but less accurate), such as a decision tree.

b) Post hoc techniques consisting of a very highly complex and uninterpretable black- book model and use a separate set of strategies to complete a reverse technique to explain it without affecting or even knowing the inner part of the system.

2) *The scoop of interpretability (local /global)*: these criteria help to define the perimeter of the interpretability of the model:

a) *Global explainability*: aims to make the decision-making process transparent for all data; this class of methods is valid when machine learning models are essential to inform decisions at the population level, such as consumption trends of drugs or climate change.

b) *Local explainability*: aims to provide explanations for a single decision in a restricted neighborhood of data, this class of interpretability methods is used to generate a particular explanation, in general, to justify the reason why the model made a decision specific, for instance. It should be noted that model-agnostic interpretations are generally post-hoc. Intrinsic methods are, by definition, techniques specific to a model.

3) *Dependence level on the machine learning model used*: Another meaningful way of classifying explanation model techniques is to determine whether they are:

a) *Agnostic models*: which means that they can be applied to any machine learning algorithm.

b) *Specific models*: which means that they can only be applied to a single type or algorithm class. The most popular methods of interpretation belong to the agnostic model.

Indeed, numerous model-agnostic methods, ranging from statistics, machine learning, and information science, have recently been developed. These are generally covered by four types [10]: Extraction of knowledge; Visualization; Methods of influence; Example explanations.

#### V. RELATED WORK

This section suggests an analysis of the few existing explainable recommender systems in literature to detect axes of improvement. The current explainable recommendation is mainly divided into four general classes: (i) Matrix Factorization, Topic Modeling, (iii) Deep learning, and (iv) Graph Embedding models.

We analyze the essential works proposed in each class below. The Explicit Model Factor (EFM) is proposed by Zhang and al [23] to produce explicit recommendations, which aim to ensure high predictive accuracy. First, they analyze the feelings expressed in user reviews sentences and their explicit product characteristics comments. The approach suggested can provide personalized explanations alongside the recommendations, using the explicit features, such as: "We recommend the product because you are interested in a particular feature and this product is good functionality performance." By telling the user, the model can even give recommendations.

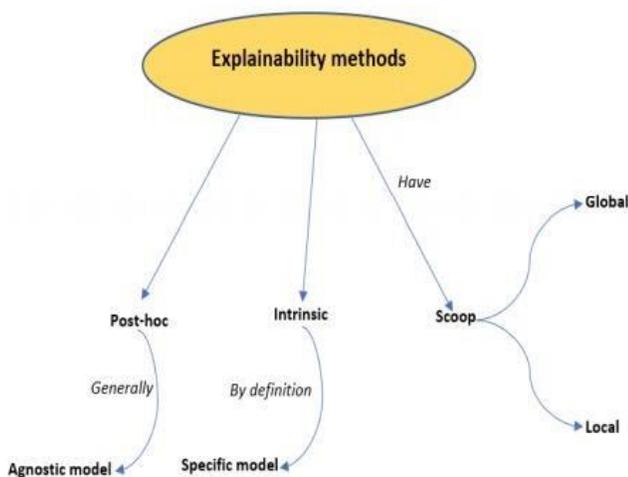


Fig. 3. Interpretable Machine Learning Methods.

Bauman and al. [24] proposed the Sentiment Utility Logistics Model (SULM), which extracts characteristics and user sentiment on those characteristics. Features and sentiments are integrated into a factorization matrix to predict unknown sentiments and ratings, which are ultimately used to generate recommendations. The proposed method provides the recommended items to users and provides the recommended characteristics for an element, and these characteristics serve as explanations for a recommendation. For example, the method may recommend cinemas and important aspects over which the user has control and optionally select, such as the time to go to the movies.

Based on topic modeling class, McAuley and Leskovec [25] proposed to understand the hidden factors in the models of latent factors based on the hidden subjects extracted from textual reviews (reviews). To achieve this goal, the authors proposed the Hidden Factor and Subject Model (HFT), which links the latent factor models and the latent Dirichlet allocation (LDA) by linking each dimension of the latent item (or user) vector to each dimension of the subject distribution in LDA using a softmax function. In considering the information from the reviews, the proposed method improves the accuracy of predictions. Following this idea, Tan and al. [26] proposed to model the recommendation of elements and user preferences in a unified semantic space based on textual reviews. An item is built in as a recommendable distribution by topic in the modeling process, and the topics covered in these higher rating reviews are repeated to improve relevance. Likewise, a user is integrated into the same space, determined by their historical rating behaviors. The recommendation and preference distributions are finally integrated into the latent factorization framework to match the truth. The explanations for the recommended items are derived from the latent learned subjects.

Deep learning has gained much attention in the recommendation research community, and it has also been widely used for explainable recommendations [27]. Seo and al [28] proposed to model user preferences and element properties using convolutional neural networks (CNNs) during textual revisions with local and global dual focus when predicting notes from a user; the model selectively chooses the critical words with different attention weights, and with the learned attention weights the model can indicate which part of a review is more critical to the current prediction, in addition, the model can also highlight relevant words in comments as an explanation to help users understand the recommendations. Likewise, Wu and al. [31] merged user-article interaction and review information into a unified framework. User reviews are summarized carefully as content characteristics built into user/item embedding to predict final grades.

Based on graph embedding class, Ma and al. [29] proposed a new common learning framework to integrate the induction of explainable rules from a knowledge graph into the construction of a rule-driven neural recommendation model. The framework runs two modules to complement each other to generate effective and explainable recommendations to have a better capacity for generalization to tackle the cold start problem because induced rules can supplement the recommendation module, especially as the circumstances of

the same user can change over time, so this approach can achieve better results if authors integrate the axis of psychological recommendations. To create explanations, Ai and al. [30] suggested adopting knowledge graph embedding and designed a user-item knowledge graph; knowledge base embeddings are learned in the graph. The shortest way from the user to the recommended item can be explained via the knowledge graph.

By analyzing the approaches mentioned above, two glaring limitations emerge, existing explainable recommender systems are mainly based on comments written by users, which is not systematically available; indeed, users prefer to be served with minimum interaction with the system (users do not like comments) thus the lack of users comments and reviews impacts the quality of explainability of existing approaches. Second, the explanations provided are complex, unintuitive, and hard to understand by an average user (he needs the technical background to understand explanations).

## VI. PROPOSED APPROACH

Inspired by existing explainable recommendation models, this approach aims to overcome explainable recommender system limitations.

- 1) Complexity of interpretations.
- 2) The necessity for textual user reviews.

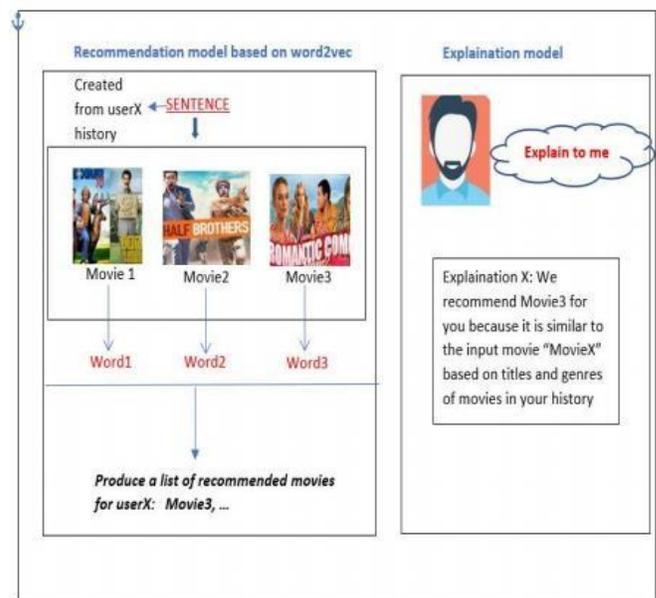


Fig. 4. An Overview of ExMrec2vec Approach.

We offer an embedded method of recommendation based on the word2vec algorithm. It is an ad-hoc model-agnostic; the explanation process is triggered after the recommendation process independently. Fig. 4 shows, through a simplified example (userX enters as input movieX), an overview of our explainable recommendation approach. The basic idea is to construct the vocabulary of the word2vec algorithm from the user history and generate recommendations for this user after training the word2vec model with the created vocabulary (see recommendation model based on word2vec in Fig. 4); then

generate a simple explication for our average user. The word2vec model requires word and sentence equivalents, so we consider each movie as a word, and movies with similar user ratings are placed in the same sentence. To learn the movie embeddings to find relationships between movies to produce recommendations. We present a simple explanation for our lambda user; we will not show the details of embeddings learned from the word2vec model to an average user. We interpret recommended results generally; as we said, we try to overcome the limitation of complexity in the explanations in existing works in this field.

### VII. EXPERIMENT AND EVALUATION

We used two available datasets to evaluate our proposition, MovieLens 20M Dataset, maintained by the MovieLens research team, will be used to evaluate our proposition. It has 20 million ratings and 465,000 tag applications from 138,000 users who have tagged 27,000 movies, and the latest movielens dataset contains 100,000 ratings applied to 9,000 distinct movies by 600 users.

We used the "movies.csv" and "ratings.csv" files from the downloaded datasets. "movies.csv" is a lookup table for the movie's id and its name. "ratings.csv" contains the ratings of all the users for all the movies. We first separated the data into training and test sets. The test set is for the evaluation of the model. We used precision, recall, and F-1 score at K to evaluate our model performance.

**Precision@k:** The precision indicates the rate of relevant recommended items for all recommended items, the percentage of predictions we get right.

**Recall@k:** concerns the rate of relevant recommended items for all relevant items. The percentage of accurate labels captured when we recommend k labels per example.

**F-1@k:** is a combination of precision and recall. The F1 score takes values between 0 and 1. It turns 0 if one precision or recall is 0 and 1 if both precision and recall are 1.

To learn embeddings from our model, we need to have the equivalents of "word" and "sentence" from the downloaded datasets. Here we can consider that every "film" is a "word," Films with similar ratings are in the same sentence by a user. We considered all the movies "high rated" by a user to define the "meaning" for each movie. This process is applied to all the movies "low rated" by a user too.

Two models (for recommendation process) are trained with different parameters (see Table I and Table II):

TABLE I. PARAMETERS USED IN OUR IMPLEMENTATION

Parameters	Meaning
Number of epoch	Number of iterations
Min_count	More than min_count times a film must be maintained to be kept
Size	Size of the hidden layer
Workers	The number of threads for training
Sg	Sets the algorithm for training, we used skip-gram.
Hs	Negative sampling is used

TABLE II. PARAMETERS VALUES IN EACH MODEL

Parameters	Model1	Model2
Number of epoch	5	10
Min_count	10	5
Size	200	300
Workers	4	4
Sg	1	1
Hs	0	0

A set of recommendations is produced for each run on our proposition. The precision, recall, and F1-score of K can assess the performance of this task among the two defined models (K is the number of recommendations we made for each input).

In our implementation, precision indicated the rate of relevant recommended movies for all recommended movies, and recall indicated the rate of relevant movies for all relevant movies (movies high rated by user).

Note that the more movies a user has rated, the greater the accuracy and the lower the recall. On the other hand, the more recommendations made by our model (larger K), the lower the accuracy and the higher the recall, given the fixed number of "high rated" films. We constated these remarks when we interpreted precision and recall at K on a single model.

We extracted the "high rated" movies from both datasets and evaluated the precision, recall, and F1-score on both models we trained with different parameters are provided (see Table III and Table IV).

TABLE III. EVALUATION METRICS VALUES FROM MODEL 1 AND MODEL2 USING MOVIELENS 20M

Evaluation metrics	Model1	Model2
precesion@10	0.16	0.13
recall@10	0.07	0.059
F-1@10	0.095	0.08

TABLE IV. EVALUATION METRICS VALUES FROM MODEL 1 AND MODEL2 USING MOVIELENS LATEST

Evaluation metrics	Model1	Model2
precesion@10	0.15	0.13
recall@10	0.062	0.059
F-1@10	0.088	0.084

We constated from obtained results that model 1 is better than model 2 (with k =10, number of recommendation). So, our recommendation process will be based on module 1.

To demonstrate the effectiveness of our proposition, we compare our model ExMrec2vec with two traditional recommendation methods ("popular recommendation", "collaborative filtering"). based on the same evaluation metrics defined above (see Table V and Table VI) and using the two defined datasets.

TABLE V. COMPARISON RESULTS USING MOVIELENS LATEST

Evaluation metrics	ExMrec2vec	Popular model	Collaborative model
precesion@10	0.15	0.12	0.14
recall@10	0.062	0.053	0.05
F-1@10	0.088	0.077	0.081

TABLE VI. COMPARISON RESULTS USING MOVIELENS 20M

Evaluation metrics	ExMrec2vec	Popular model	Collaborative model
precesion@10	0.16	0.13	0.12
recall@10	0.07	0.066	0.069
F-1@10	0.095	0.072	0.073

Popular model: A common standard approach basically prescribes the foremost prevalent items that the user has not already devoured.

Collaborative model: This approach employs the memory of past users intuitive to compute users' similitudes based on items they've connecting (user-based approach) or compute items likenesses based on the clients that have connected with them (item-based approach). We used the time-based approach.

Obtained results (see Table V and Table VI) prove the effectiveness of our proposed model. (We plan to compare another advanced version of our model with other complicated algorithms in the future).

We implemented the user interaction scenarios with the recommendation explanation interfaces. Since we used the movielens datasets, we then positioned ourselves in the context of an online movie recommendation system called "ExMRec2vec". Typically, each user of the "ExMRec2vec" site will connect using his "login" and his password "password."

Take the example of the user "U17" who will connect (Fig. 5).

After logging in, the user enters a movie as input to search (see Fig. 6).

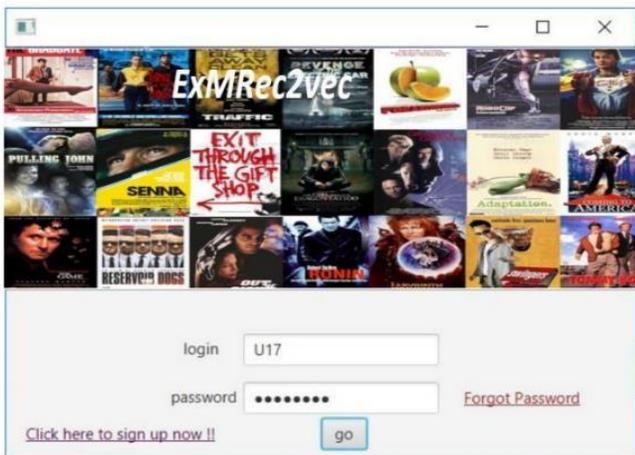


Fig. 5. Connection of User "U17".



Fig. 6. The Interface that Offers the Search for a Movie Entered by the Active user "U17".

Our U17 user has "Ratatouille (2007)" as a recommended movie to him. If he wants to know any explanations for this recommendation, once he clicks on "why" he will automatically have the reason available which explain this recommendation (see Fig. 7).

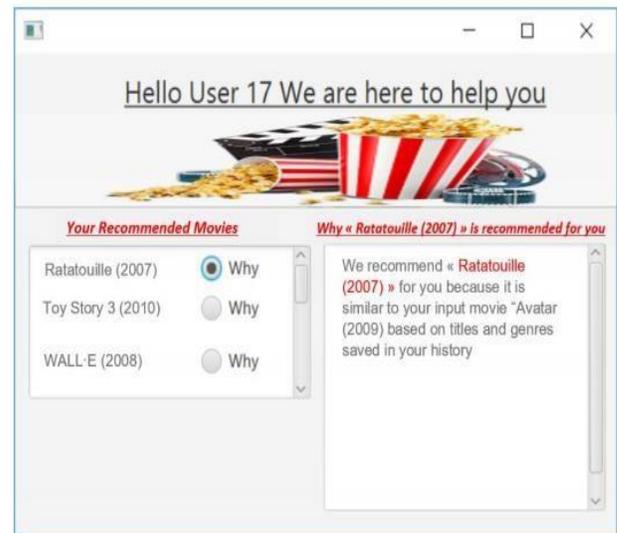


Fig. 7. The Interface of Recommendation and Explanation.

## VIII. CONCLUSION

In this article, we proposed a movie recommendation approach that allows improving the performance of movie recommender systems by accompanying recommended movies with simple explanations. Which increases recommender systems acceptance and makes them applicable to an even broader range of applications (not only for movies). The proposed approach is based on the word2vec model via using the history of users to produce recommendations, then presenting simple explanations through the input movie.

The conducted experiments on different parameters prove the quality of model 1, see Section VII, we demonstrate the effectiveness of our proposed model Exmrec2vec by conducting a comparison between our model and two traditional models.

We plan to enhance encoded word2vec vocabulary by introducing other social relations based on learned similarity using different machine learning algorithms and improving movie-movie relationships using the complementarity concept.

## REFERENCES

- [1] A. Samih, A. Ghadi, and A. Fennan, "Translational-Randomwalk embeddings based recommender system: a pragmatic survey", p. 9, *BDiot'21*, 2021.
- [2] H.Cai, V. W.Zheng and K.C.Chang, "A Comprehensive Survey of Graph Embedding: Problems, Techniques, and Applications"; in *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 9, pp. 1616-1637, 1D. Zhang, et al., "Common Mode Circulating Current Control of Interleaved Three-Phase Two-Level Voltage-Source Converters with Discontinuous Space-Vector Modulation," 2009 *IEEE Energy Conversion Congress and Exposition*, Vols 1-6, pp. 3906-3912, 2009.
- [3] P.Goyal and E.Ferrara; " Graph embedding techniques, applications, and performance: A survey ", *Knowledge-Based Systems*, vol. 151, p. 8-94, juill. 2018, doi:10.1016/j.knosys.2018.03.022.
- [4] A. Samih, A. Adadi, and M. Berrada, "Towards a knowledge based Explainable Recommender Systems "in *Proceedings of the 4th International Conference on Big Data and Internet of Things*, Rabat Morocco, oct. 2019, p.1-5, doi: 10.1145/3372938.3372959.
- [5] J. Lamy, B. Sekar, G. Guezennec, J. Bouaud, B. Séroussi, "Explainable artificial intelligence for breast cancer: a visual case-based reasoning approach", *Science Direct*, 94 :42-53, 2019.
- [6] P. Resnick and H. R. Varian," Recommender systems ", *Commun. ACM*, vol. 40, no 3, p. 56-58, mars 1997, doi:10.1145/245108.245121.
- [7] P. Melville and V. Sindhvani "Recommender Systems" In: Sammut C., Webb G.I. (eds) *Encyclopedia of Machine Learning and Data Mining*. Springer, Boston, 2017, MA. [https://doi.org/10.1007/978-1-4899-7687-1\\_964](https://doi.org/10.1007/978-1-4899-7687-1_964).
- [8] M. Seyednezhad, K. Cozart, J. Bowllan, and A. Smith , " A Review on Recommendation Systems : Context-aware to Social-based" , *IEEE Journal*, pp. 9-20, 2018.
- [9] I. Benouaret. « Un système de recommandation contextuel et composite pour la visite personnalisée de sites culturels », pp. 19-20 ,2018M. Young, *The Technical Writers Handbook*. Mill Valley, CA: University Science, 1989.
- [10] F. Isinkaye, Y. Foljimi and B.Ojokoh , "Recommendation systems : Principles, methods and evaluation", *Egyptian Information Journal*, pp.5-7, 2015.
- [11] E. Negre , *Les systèmes de recommandation : une catégorisation* . *Online :https://interstices.info/les-systemes-de-recommandation-categorisation/* ,2018 (Journal Online Sources style).
- [12] D. Sánchez-Moreno, A. B. Gil González, M. D. Muñoz Vicente, V. F. López Batista, and M. N. Moreno García; "A collaborative filtering method for music recommendation using playing coefficients for artists and users "; *Expert Systems with Applications*, vol. 66, p.234-244, déc. 2016, doi: 10.1016/j.eswa.2016.09.019.
- [13] R. Prasad and V. Kumari, "A Categorical review of recommender systems", *International Journal of Distributed and Parallel Systems (IJDPS)*, pp 70-79, 2012.
- [14] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System ", *IJCA*, vol. 110, no 4, p. 31-36, janv. 2015, doi: 10.5120/19308-0760.
- [15] P. B. Thorat, R. M. Goudar, and S. Barve, "Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System ", *IJCA*, vol. 110, no 4, p. 31-36, janv. 2015, doi: 10.5120/19308-0760.
- [16] S. BenTicha. *Recommandation personnalisée hybride*, pp 51-54, 2018.
- [17] R. Burke, *Hybrid recommender systems : Survey and experiments*. *User Modeling and User-Adapted Interaction*, pp.331-370, 2002.
- [18] P. B. Thorat, R. M. Goudar, et S. Barve, « Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. », *IJCA*, vol. 110, no 4, p. 31-36, janv. 2015.
- [19] Z. Champiri, S. Shahamari and S. Salim. A systematic review of scholar context-aware recommender systems. *Science Direct Journal*, pp. 1-3, 2015.
- [20] T. Mikolov, « Efficient Estimation of Word Representations in Vector Space », *Arxiv*, 2013.
- [21] J. Lamy, B. Sekar, G. Guezennec, J. Bouaud, B. Séroussi, "Explainable artificial intelligence for breast cancer: a visual case-based reasoning approach", *Science Direct*, 94 :42-53, 2019.
- [22] A. Adadi., B. Berrada, B.: *Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)*. *IEEE ACCESS*, Vol. 6, (2018) 52138–52160.
- [23] Y. Zhang, G. Lai, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 83–92. *ACM*, 2014.
- [24] K. Bauman, B. Liu, and A. Tuzhilin. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 717–725. *ACM*, 2017.
- [25] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. *ACM*, 2013.
- [26] Y. Tan, M. Zhang, Y. Liu, and S. Ma. Rating-boosted latent topics: Understanding users and items with ratings and reviews. In *IJCAI*, pp. 2640–2646, 2016.
- [27] X. Chen, Z. Yongfeng, X. Hongteng, Y. Cao, Z. Qin, and H. Zha. Visually explainable recommendation. *arXiv preprint arXiv:1801.10288*, 2018c.
- [28] S. Seo, J. Huang, H. Yang, and Y. Liu. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*, pages 297–305. *ACM*, 2017.
- [29] C. Ma., L. Ma, Y. Zhang, J. Sun, X. Liu, "M. (2020). Memory Augmented Graph Neural Networks for Sequential Recommendation". *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 5045-5052. <https://doi.org/10.1609/aaai.v34i04.5945>.
- [30] Q. Ai, V. Azizi, X. Chen, and Y. Zhang. Learning heterogeneous knowledge base embeddings for explainable recommendation. *arXiv preprint arXiv:1805.03352*, 2018.
- [31] L. Wu, C. Quan, C. Li, Q. Wang, and B. Zheng. "A context-aware user-item representation learning for item recommendation". *arXiv preprint arXiv:1712.02342*, 2017.
- [32] A. Samih, A. Ghadi, A. Fennan, *DEEP GRAPH EMBEDDINGS IN RECOMMENDER SYSTEMS: A SURVEY*, *Journal of Theoretical and Applied Information Technology* 15th August 2021. Vol. 99, No 15.