Hybrid filtering and semantic sentiment analysis by deep learning for recommendation systems

Badiâa Dellal-Hedjazi
Computer Science Faculty
USTHB
Algiers, Algeria
bhedjazi@usthb.dz

Zaia Alimazighi
Computer Science Faculty
USTHB
Algiers, Algeria
zalimazighi@usthb.dz

***Abstract*— Faced with the ever-increasing complexity, volume and dynamism of online information, recommendation systems are among the solutions that anticipate the needs of users and offer them items (articles, products, web pages, etc.) that they are likely to appreciate. Unlike traditional recommendation models, deep learning offers a better understanding of user requests, objects features, historical interactions between them and can process massive amounts of data. In this work, we realize two recommendation systems. The first one is based on MLP deep neural network adapted to data already defined by their features. In addition to the use of deep learning, we offer a new hybrid recommendation solution between the demographic approach and the content-based approach in order to eliminate the limits of each and to combine their strengths, through a deep neural network adapted for massive data. Experimentation with our approach has produced good results in terms of accuracy and speed, whether through the use of deep learning or the hybridization of content-based and demographic filtering, which is a particular type of collaborative filtering. The second system is based on semantic sentiment analysis through the use of LSTM deep neural network for sentiment analysis and topic modeling with LDA for semantic analysis. Semantic analysis is used to study the meaning of a text, whereas sentiment analysis represents its emotional value. The precision of the obtained results allows relevant recommendations and the integration of the two systems allows multi-faceted exploitation of the data and consequently offers a new, complete and promising approach for recommendation systems.**

Keywords—recommendation system, deep learning, MLP, LSTM, content based filtering, demographic filtering, sentiment analysis, semantic analysis, LDA

I. INTRODUCTION

The amount of information available to everyone is constantly growing. It is highly heterogeneous and almost infinite in certain areas. It then becomes imperative to assist the user and to facilitate his access to resources likely to interest him and adapted to his needs and therefore help him through suggestions. Recommendation systems allow targetting and offering information that is relevant to the user. These systems consist of filtering information in a personalized way in order to automatically offer the user recommendations (films, products, etc.) that adapt to their profile (preferences, interests). These propositions are made on the basis of its interactions with the system based on implicit or explicit feedback [1], but also on other criteria, such as demographic data such as age, occupation, gender and the geographic area. The recommendation systems make it possible to retain the user by learning more about their needs [2]. Different approaches exist for recommendation where the most used are content-based filtering and collaborative filtering. The first technique consists in recommending to the user products similar to the products appreciated in the past, on the other hand the second has the principle of recommending one or more items to a given user based on the opinions of other users sharing tastes and similar preferences. Both approaches have their strengths and weaknesses, hence the adoption of hybridization to overcome their limits.

Traditional approaches to recommendation systems have limitations such as cold start (new user, new item) and are not efficient enough to exploit the value of big data. The volume of data is too large to analyze the correlations between these data, test all the hypotheses and derive a value. Hence the orientation towards the construction of models by machine learning.

Learning by deep neural networks (deeplearning) has shown its effectiveness (precision, speed) in solving various problems compared to other techniques. It can process a large set of diverse and changing data. The more data a deeplearning system receives, the more precisely it learns. The use of deeplearning in recommendation systems has shown that they can improve recommendations quality in relevance, novelty and diversity [3].

In addition to using a number of item and user-related characteristics to produce recommendations, it is possible to exploit the content of comments, analyze and understand them semantically using deep learning to generate recommendations. based on this understanding.

For this, we present, in this work, two recommendation systems. The first recommender system is a hybrid filtering system based on learning from features using deep learning through a MLP (Multi-Layer Perceptron) type deep network, and a second recommender system based on content analysis by sentiment analysis using an LSTM type deep network and semantic analysis more specifically topic modeling using the LDA technique.

The objectives of this work are: (1) To use big data for the analysis and the construction of a recommendation model based on deeplearning. (2) Resolve the limits of recommendation approaches precisely cold start by the hybridization of recommendation based on content and demographic. (3) Understand user contents (comments, reviews,…) through semantic sentiment analysis using deep learning and topic modeling techniques.

In this work, we present recommendation systems in section 2, deep learning in section 3, the works on recommendation systems by deep learning in section 4 and our contribution in sections 5 and 6 for our first system (system 1) and the sections 7 and 8 for system2. We end with a conclusion and some perspectives.

II. RECOMMENDATION SYSTEMS

Recommendation systems are software tools and techniques that provide suggested items to a user [7] [8]. An item can designate any element that can be offered to a user such as a product, a service, media items (movies, music etc.) or a collection of item information (web page, portal, etc.) [9] . A recommendation system helps users make their choice in an area where they have little information to sort and evaluate possible alternatives. The importance of such systems arises in environments where gigantic amounts of online information exceed the research capabilities of a human being [6]. They offer users a filtered list of items according to their given preferences either explicitly where the user evaluates the items or implicitly by the interpretation of their actions. These recommendations are usually personalized or non-personalized like recommendation of top n-best selling books.

For recommendations that are relevant and tailored to user preferences, various approaches are adopted [4] [5] [6], the three main ones being: collaborative filtering, content-based filtering and hybrid filtering. There are also demographic recommendation systems (particular case of collaborative filtering), knowledge-based SRs and utility-based recommendation systems [4]. .

A. Content-based filtering

Recommends items to the user with similar descriptions to items they have previously enjoyed [10] [11]. It consists in comparing the items not assessed by the user with his profile [1], represented by all the items he has assessed, by calculating the similarity between them [12] [13]. The recommendation is generated according to the user profile and the item profile. The item profile is represented by the characteristics expressing its semantic content [5].

This type of filtering is (1) Adaptable because it becomes more precise by increasing the evaluations. (2) Independent from other users. It only depends on the target user's ratings and builds their profile and (3) recommends new items even before they are rated [11]. But this type of filtering presents (1) the Problem of the new user with undefined preferences (cold start for users: User Cold Start) [11]. (2) limit of the analysis because it is difficult in certain types of items (multimedia, etc.). Unlike collaborative filtering which deals with all types of items. (3) Overspecialization [11]: Since the system can only recommend items similar to the user profile. The diversity of recommendations is a quality criterion. The user should receive diverse and non-homogeneous relevant recommendations.

B. Collaborative filtering

Collaborative filtering (CF) [6] [12] predicts the usefulness of items for a particular user basing on items previously rated by other users [5].

Unlike content-based filtering, collaborative filtering does not need item descriptions for recommendations and is better suited for multimedia items.

A CF system is made up of three main processes: (1) evaluation of recommendations: Users evaluate the items that they consult explicitly or implicitly [17], (2) the formation of communities for each user by its proximity to the evaluations of other users [17], and (3) the recommendation: The system predicts user item interest based on the evaluations that community members have done on this item [17].

There are two classes of CF [18], memory-based algorithms also called heuristic-based [5] or neighborhood-based algorithms, and those based on model.

1) Memory based collaborative filtering : considers all the user evaluations available when calculating the recommendation. To predict the ratings that a user may give to items they have not rated, similarity is calculated by grouping the closest neighbors into communities and then estimating the user's rating. Similarity is measured between users or between items [11] [19] [20].

2) Model based collaborative filtering: learns models from data. Once the models found, the prediction, for a user, is automatically generated from his profile. Typically, machine learning and statistical techniques are used [6].

There is a particular case of collaborative filtering which is the demographic recommendation [17] based on the categorization of users according to their demographic information (age, occupation, city of origin, etc.) [4]. Collaborative filtering has the strong points (1) independence of content, (2) cross-genre niches [6], (3) adaptable but has the limits (1) cold start, (2) sparcity (3) shilling [6].

For more efficiency the two types of filtering (content, collaborative) can be hybridized (combined) [4] [21] [22].

In addition to classical information retrieval techniques, several machine learning techniques are used such as the naive Bayes classifier, the nearest K's, and recently deep learning. The recommendation becomes a classification task. From the description of the items, the system infers a model making it possible to classify an unobserved item for each user. The classification is in two classes (interesting, not interesting) or several classes [11].

III. DEAP LEARNING

The need for big data analysis in all fields and especially recommendation systems has led to the use of machine learning and especially deep learning (deeplearning) based on artificial neural networks.

These algorithms allow computers to train on sample data in order to build a model that can perform the required task (prediction, recommendation, etc.) [24]. This learning can be supervised, unsupervised and by reinforcement [3]

The training data are gathered in a "dataset" (dataset) organized in a column table. This data can be numerical values, words, sentences, etc. To generate the model, the data is divided into (1) Learning data used to build the learning model, (2) Test data to test the model and correct errors and (3) Validation data: can be used to validate the model.

Neural networks (RNs) is one of the most used techniques for machine learning. An artificial neuron is a mathematical and computational representation of the biological neuron. Each artificial neuron is an elementary processor. It receives a varying number of inputs from upstream neurons or sensors. Each of its inputs is associated with a weight [26] representing the connection force. Each neuron has an output activation function that branches out to power a varying number of downstream neurons. An RN is made up of several layers of neurons. Neurons can be linked

by weighted connections with neurons in other layers or with those in the same layer [28]. We distinguish between the input layer, hidden layers and an output layer [24] [26].

Learning is a procedure by which the connections of neurons are adjusted by an information source [26]. The goal of this procedure is to make the network capable of reacting correctly to data which was not present in the learning base [29]. The learning process is based on the idea of propagating the error made at the output to the inner layers to modify the synaptic weights previously initialized with random values. For supervised learning, we have a set of examples (learning base) made up of pairs (input, desired output). During training, we present the examples one by one to the network which calculates the corresponding outputs. These calculations are done step by step from the entrance to the exit (forward propagation) [30].

The error between the actual output and the desired output is calculated. This error is then propagated back through the network by modifying synaptic weights [30]. This process is repeated for each example in the learning base. If, for all of the examples, the error is less than a chosen threshold, then the network has converged. The training consists in minimizing the quadratic error committed on the set of the examples, by adjustment of the weights by decreasing the gradient (ex. Descent of gradient) [30].

Deep learning consists of learning several levels of representation and abstraction to give meaning to data.

Deep Learning is based on neural networks and tailored to handle large amounts of data by adding layers to the network. A deep learning model has the ability to extract features from raw data through multiple layers of processing of linear and nonlinear transformations and learn about those features bit by bit through each layer with minimal human intervention [31] . The progress of deep learning has been possible in particular thanks to the increase in the power of computers and the development of large masses of data (big data) [32]. Several deep architectures exist like multilayer perceptrons (MLP) [27] [29], convolutional neural network (CNN) [33] [34], autoencoder (AE) [38] [39] [69], recurrent neural network (RNN) [35] [36] and their two variants Long Short-Term Memory (LSTM) [24] [35] and Gated Recurrent Unit (GRU) [38] [66].

IV. RELATED WORD

Deep neural networks have given immense success in speech recognition, computer vision and natural language processing. However, their exploration on recommender systems has been less studied [40]. In this section we present three main works based on deep learning. The first model [41] is developed to predict the ratings that a user would assign to a film based on the tastes of that user and other users who have viewed and classified the same film and other films. Using a 6-layer deep AE network with a non-linear activation function (SELU- Linear exponential linear units), dropout and iterative re-feeding [41] [42]. One of the key ideas of this work is dense re-feeding by re-feeding the output into the auto-encoder.

The second model [40] is a multilayer perceptron (MLP) for learning the item user interaction function, by working on implicit data. This model uses the flexibility, complexity and non-linearity of the neural network to create a matrix factorization recommendation system.

The three works cited above have provided a solution to the problem of subspecialization, but both have drawbacks, namely cold start, sparcity and user dependency.

Most work on recommendation systems by deeplearning opt for collaborative filtering. The major disadvantage of this approach is the cold start. To overcome the disadvantages of each and for more efficiency, we propose, in this work, two systems. The first system is a hybridization solution between the demographic and content approach. The second system is a semantic sentiment analysis solution for relevant recommendations.

V. SYSTEM 1 : HYBRID FILTERING

To overcome the cold start problem (new user, new item), we propose hybridization by injecting the features of content filtering with those of the demographic approach “Fig. 1.”. The features of the two approaches are the attributes of the items description and those of the users. The system entry is therefore a series of user then item features.

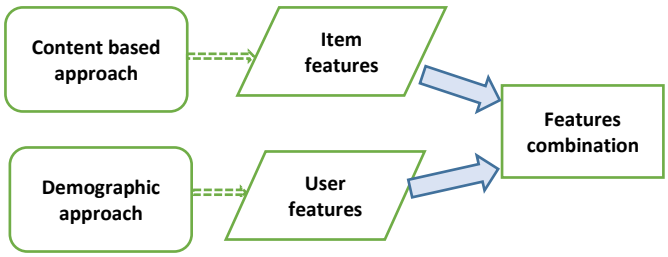

Fig. 1. Use case diagram of System 1

For relevant recommendations, we use a deep neural network to learn recommendation. Deeplearning allows us to generate a model capable of giving relevant predictions. Our learning data is a combination of user and item characteristics.

In order to offer relevant recommendation of various items to meet user needs and tastes, we have chosen movie recommendation as an application example. A movie has as attributes a title, a year of production, a genre and a director. A user is identified by its gender, age, occupation and zip code.

The user of our system or administrator, in our case, is able to perform the following tasks “Fig. 2.” :

1) Start learning: the learning process is started on the sampling data. This step requires data preprocessing (section V.C) at the start and it will generate a model at the end.

2) Stop learning: the administrator can end the learning process. The system will therefore record the weights learned during this process.

3) Recommendation: The administrator begins by choosing the possibilities according to which the system makes the recommendation: display the list of films to recommend for an existing user or for a new user, display the list of users for a new film, display the prediction for new user and new film, in the following the details of each:

• display movie list to recommend for an existing user who has already evaluated a few items and provided his

information. The user enters his identifier and obtains ratings predictions for the movie he has not seen. A list containing ten moviess with large prediction values is recommended.

• View the list of movies to recommend for a new user who has not done reviews in the past. This function requires you to enter information about this user, which are: sex, age and occupation. The recommendation is a list of ten movies with the highest prediction values.

• View the list of users for a new unrated movie. This requires entering the information for the new movie. The system displays all the users who may be interested in this film in order to recommend it to all these users.

• display the prediction for new user and new film which requires entering the information of the film as well as that of the user in order to find the prediction of the evaluation of this userutilisateur.

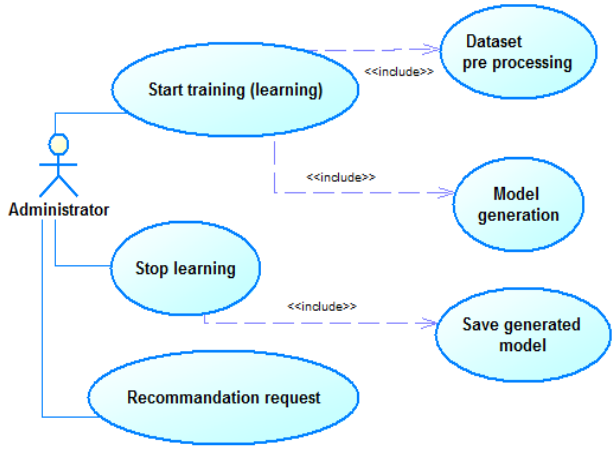

Fig. 2. Use case diagram of System 1

A. System 1 architecture

Our system consists of two main modules which are Learning module and Recommendation module. The first one generates a model based on the sampling data and the second allows to provide recommendations based on the generated model (Deep neural network MLP) "Fig. 3. ".

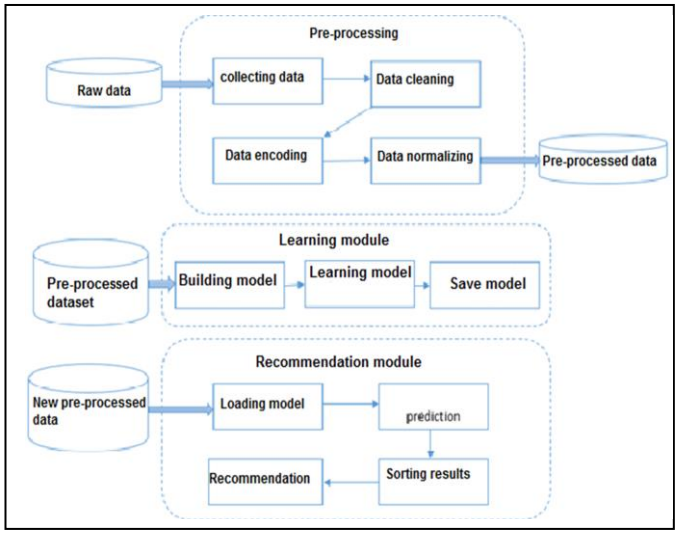

Fig. 3. *Global System 1 architecture*

B. Pre-processing step :

Consists of transforming raw data into preprocessed data to be injected into the network and consists of:

- Data collection: We have gathered the attributes extracted from different sources in the same set containing: sex, age and user occupation, title, year of production and genre of the film.
- Data cleaning: fill in missing values, identify or remove outliers, like zip-code because it does not have a fixed structure in the whole dataset.
- Data encoding: allows converting attributes to digital format since neural models are based on mathematical calculations.
- Data normalization: Min-Max normalization consists of reducing the data range between 0 and 1 (or -1 to 1 with negative values) because it is easier for the network to learn on this scale.

According to our case study, the data preprocessing algorithm is as follows:

Algorithm 1 : Pre-processing algorithm

Input :
U: user informations file
M: movie informations file
R: ratings (évaluations) file
Output :
M : Nework input matrixMatrice
Var id_U, id_F, Eval , id_User, id_Film :integer
L_User: User informations list
L_Movie : Movie informations list
L_Attribute: informations list of L_User, L_Movie, Eval
D : file containing L_Attribute list
A : one codified instance of file D
B : one normalized instance of file A
Begin
Read_file (U) ; Read_file (F) ; Read_file (R) ;
for (i=1 to size(R)) **do**
id_U :=retrieve user id of instance i
id_M := retrieve movie id of instance i
Eval := retrieve rating id of instance i
j=1
while (j <= taille(U)) **do**
if (id_User == id_U) **then**
L_User := all user informations of instance j
end if
j :=j+1
end while
k=1
while (k <= taille(M)) **do**
if (id_Movie == id_M) **then**
L_Movie := all informations of instance k; **End if**
k :=k+1
end while
L_Attribut := L_User+ L_Movie + Eval
write(D, L_Attribute)
end for
Read_file(D)
for (k=1to size(D)) **do**
A := encoding of D file instance k
B:= normalizer (A)
for (j=1 to size(L_Attribute)) **do** M [k][j]:=B[j]; **End for**
end for
return M

C. Learning module :

Our deep architecture is of MLP type. It is the architecture most suited to classification problems with characteristics of data already defined as input. Our MLP is made up of multiple layers with varying number of neurons and functionally grouped and fully connected. It consists of an input layer, 12 hidden layers and an output layer "Fig. 4. " . This architecture is obtained after several variations of the network parameters (number of hidden layers, the number of neurons per layer, the activation function, etc.).

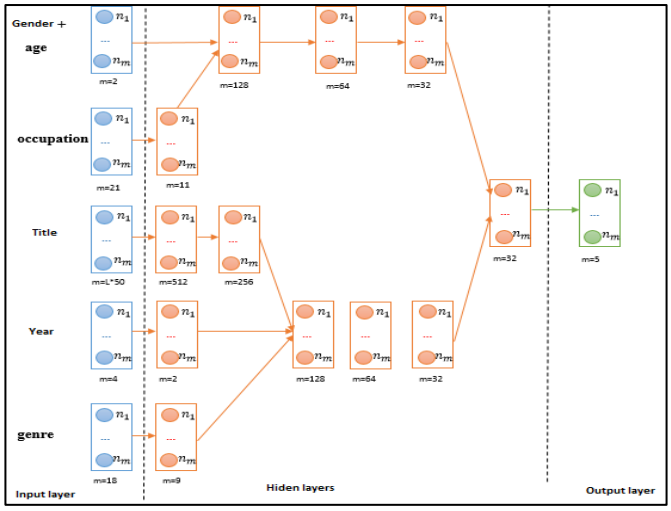

Fig. 4. *Our deep neural network model (MLP) structure*

The neurons in the layers are grouped into sets of information as the network learns the hidden characteristics of the user and the item before combining them for prediction. The input layer is made up of subsets of neurons. Each of them represents an attribute of the preprocessed dataset. The coding of each attribute is at least in two digits which requires more than two neurons. The character m represents the number of neurons per layer. The parameter m is set as follows:

- User gender and age group: m = 2, one neuron for gender and the other for age..
- User occupation: m = 21 neurons for a maximum of 21 occupations where the neuron corresponding to the occupation is activated.
- Movie title: m = L * 50 L is the maximum number of words in the title, 50 is the size of the weight vector of each word in the title.
- Year of the movie: m = 4 one neuron per digit.
- Genre of movie: m = 18 each neuron refers to a genre (Action, Comedy, etc.).
- Output layer: m = 5 for 5 stars (ratings), each neuron takes the probability of belonging of the input data to each star. The neuron with the greatest probability is the prediction of the network.

D. Learning the model :

We first choose a dataset, preprocess it and divide it into 80% for learning, 10% for validation and 10% for testing. Learning is done by passing data through the deep neural network followed by the error calculation between the results found by the network and the expected results. This error is minimized on each iteration by applying back propagation.

E. Save the model: At the end of each iteration, the new weights for each parameter matrix are saved in files.

The algorithm of the learning process is given below:

Algorithm 2 :Learning algorithm

input :
M : Network input matrix. Each row is a couple <x, y> where x=M [1..n-1] is the network input vector and y= M[n] the class of this input
output :
W : weight matrix
Var :
nb_epoch : integer
y' : network output
begin :
Random initializing of network weights w_j
while (nb_epoch<= 50) **do**
for each <x, y> **do**
enter x to network and calcule the output y'
for each y' faire
calculate error
end for
update W after errors of each y'
end while
return W

F. Recommendation module: The recommendation is made via the learned model. The processing in this module goes through several stages that are*:*

- Loading the learned model with its weights.
- Calculation of the prediction which gives an estimate of the preference values using the model learned with new preprocessed data. These new data are the films not seen by the target user.
- Results sort: recommended items are sorted in descending order of their predicted values.
- Items Recommendation with highest predicted values.

VI. SYSTEM 1: IMPLEMENTATION AND EXPERIMENTS

Our system is made in two large modules. The movie recommendation prediction model learning module. This model is an MLP-like deep neural network and an exploitation module of this model for recommendation. Our system first performs data preprocessing because neural networks only accept numeric values as input, preferably between 0 and 1. 80% of the data in our dataset is used for training the model. deep, 10% for validation and 10% for testing.

After initial network configuration, supervised learning is performed on the learning data. The network parameters are set after several tests. To make a recommendation we use the saved model for new data that it has not practiced. These are preprocessed and the results predicted by the model are ranked to recommend the best. We use the MovieLens dataset from Minnesota University. This dataset is a reference of real data for testing collaborative filtering. It contains the files users.csv, movies.csv and ratings.csv. It

contains 1000209 ratings for 3900 movies made by 6040 MovieLens users since the year 2000. The value of a vote is between 1 and 5.

The file containing user information users.csv "Fig. 5." is structured into five columns containing user id, gender, age range, occupation and zip code.code.

```
3500  3500::M::35::7::20650
3501  3501::F::35::7::30309
3502  3502::F::18::7::80503
3503  3503::M::18::4::59715
3504  3504::M::18::7::02215
3505  3505::F::25::15::55455
3506  3506::M::18::12::80503
```

Fig. 5. A part of users.csv file

TABLE I. ATTRIBUTES OF USERS.CSV FILE

Field	Description
UserID	User identifier
Gender	User gender (male, female)
Age	The user's age range
Occupation	User occupation (artist, doctor, …)
Zip-code	Geographic area

movies.csv file "Fig. 6." is structured in three columns (Table II).

```
158  159::Clockers (1995)::Drama
159  160::Congo (1995)::Action|Adventure|Mystery|Sci-Fi
160  161::Crimson Tide (1995)::Drama|Thriller|War
161  162::Crumb (1994)::Documentary
162  163::Desperado (1995)::Action|Romance|Thriller
163  164::Devil in a Blue Dress (1995)::Crime|Film-Noir|Mystery|Thriller
164  165::Die Hard: With a Vengeance (1995)::Action|Thriller
165  166::Doom Generation, The (1995)::Comedy|Drama
```

Fig. 6. A part of movies.csv file

TABLE II. ATTRIBUTES OF MOVIES.CSV FILE

Champs	Description
MovieID	Movie identifier
Title	Movie title and year of release
Genre	Movie genre (action, Drama, …)

Ratings are taken from the ratings.csv file:

```
256371  1564::112::3::974740192
256372  1564::1752::2::974740741
256373  1564::2701::1::974741203
256374  1564::153::2::974741038
256375  1564::1912::4::974739950
```

Fig. 7. A part of ratings.csv file

ratings.csv file is structured in four columns (Table III).

TABLE III. RATINGS.CSV FILE ATTRIBUTES

Champs	Description
UserID	User identifier
MovieID	Movie identifier
Rating	User (*UserID*) evaluation for movie (MovieID)
Timestamp	Date (in seconds) of evaluation.

The supervised learning is based on Rating field of the dataset corresponding to the label or class of a row of features in the dataset.

A. Dataset pre-processing

In order to achieve our goal, we started collecting the necessary attributes (features) from the three dataset files and cleaning them by removing outliers, and saved them in a single file. This one is structured in eight columns separated by '; ', which are : sex, Age, occupation, title, year, gender and rating "Fig. 8.".

```
16285  0;18;12;princess bride the;1987;Action|Adventure|Comedy|Romance;5
16286  0;25;17;powder;1995;Drama|Sci-Fi;2
16287  0;35;20;great escape the;1963;Adventure|War;4
16288  0;25;7;hurricane the;1999;Drama;4
16289  0;45;7;all quiet on the western front;1930;War;5
16290  0;25;19;breaking the waves;1996;Drama;5
16291  1;35;11;year of living dangerously;1982;Drama|Romance;4
16292  0;25;12;fisher king the;1991;Comedy|Drama|Romance;2
16293  1;18;1;boys;1996;Drama;1
16294  0;35;19;rear window;1954;Mystery|Thriller;4
```

Fig. 8. A part of the bbtained file (dataset)

Before entering data in our model we encode it (Table IV).

TABLE IV. OUR DATASET ATTRIBUTES

Attribute	Codification
Gender	Female : 0 and male : 1
Age	The 7 age ranges are encoded between 0 and 6, The obtained value is divided by 6to be between 0 and 1.
Occupation	M = 21 occupations in vector of 0 with 1 in occupancy cell
year	Each digit of the year on 9 positions to have digits between 0 and 1.
Genres	There are 18 genres. Coded in a vector of 0 except the boxes representing the genres of the film are at 1
Rating	Vector of size M = 5, the evaluation box is set to 1 the others are set to 0

1) Title processing : consists of **extracting the year** of production, **deleting special characters**, **converting to lowercase** and then **encoding**.

2) Extraction of production year : which becomes an attribute of the dataset, as well as the elimination of translation of movie title in mother language in some titles.

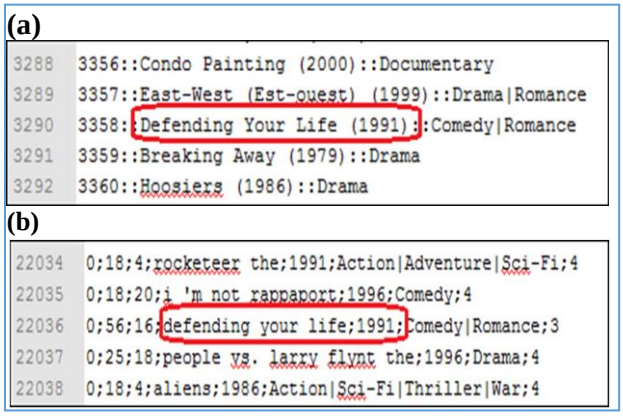

Fig. 9. Extraction of year, (a) before, (b) after

3) Removal of special characters: is to remove all special characters from itle. These characters can be punctuation marks, special characters (bars, slashes, etc.) "Fig. 10. ".

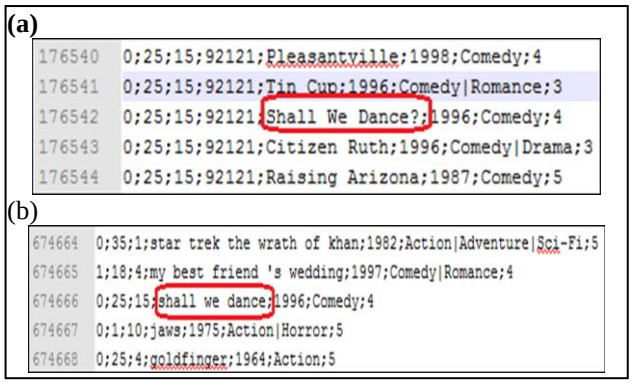

Fig. 10. Removal of special characters, (a) before, (b) after

4) Converting to lowercase: we apply text normalization by converting all words to lowercase..

5) Encoding title : Each word in the title is represented by a weight vector. The latter is obtained with Glove (Global Vectors for Word Representation). It is an unsupervised learning algorithm for obtaining vector representations of words developed by Stanford[1]. The result of Glove is a file containing the word with its weight. It is structured in fifty-one columns separated by spaces. The first column represents the word and the others represent the weights of the word "Fig. 11. ".

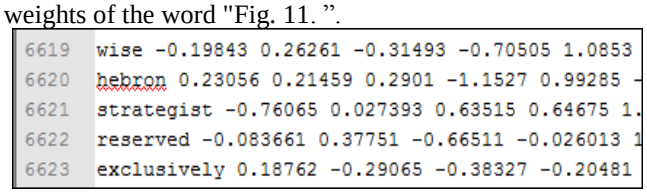

Fig. 11. Output of GLOVE model

B. The model evaluation metrics

For the evaluation of our model we use some metrics that are:

1) Cost function : allows the error calculation between the values generated by the network and the actual values [78]. In this work, we have chosen the root-mean-square-error RMSE as in (1). It calculates the root of the mean value of all the squared differences between the actual and predicted scores.

$$RMSE = \sqrt{\sum \frac{(y_{pred} - y_{real})^2}{N}} \quad (1)$$

y_{pred} is the predicted rating, y_{real} is the actual rating and N is the number of examples.

2) Accuracy : denotes the proportion of correct predictions made by the model. Taking into account that the use of the scale of ratings of films can be different from one user to another. Take the example of two users who evaluate an item that they both consider it very good but that on a scale of 1 to 5, one gives it the value 5 while the other gives it the value 4, even if the two users have the same judgment on the item but the second user is strict in the evaluation. For this we have defined our own function, described by (2) *:*

$$Acc = \frac{\sum_{i=1}^{N}(argmax(P_i) = argmax(Y_i)) + \sum_{i}^{N}(|argmax(P_i) - argmax(Y_i)| = 1)}{N} \quad (2)$$

Where N is the number of example, P_i is the prediction of the model, Y_i is the real value.

C. MLP Network settings

There is no general rule for setting the values of network hyper-parameters. In order to have the best performance, it is necessary to carry out several trainings with the different combinations of values of these parameters.

1) Activation function: The hyperbolic tangent (equation 3) in hidden layers, is preferable because centered on zero and the softmax function in output in order to normalize the results in probabilities in the interval [0-1] useful for the non-binary classification.

$$f(x) = tanh(x) = (2/1 + e^{-2x}) - 1 \quad (3)$$

where *x* is the weighted sum

2) The batch size : In our case, we cannot transfer the entire dataset to our model at once, because we have massive data and the power of our computer does not allow us to process it at the same time. For this, we have divided our dataset into batch. The size of each batch is 128.

3) The optimizer : based on the gradient descent which consists in changing the model weights in the opposite direction to the signs of the partial derivatives by a factor called the learning rate [44]. We used Adam optimizer because the results show that it works well and better than other stochastic optimization methods [45].

4) Epoch : An epoch is when all the samples in the dataset are passed through the network once. We trained our model on 50 epochs.

[1] http://nlp.stanford.edu/data/glove.6B.zip

5) Iteration : Is the number of batches required to complete an epoch. In our work and for a batch of size 128 instances, we need 6252 iterations, the latter is calculated by formula (4)*:*

$$\frac{Total\ number\ of\ examples}{Batch\ size} \qquad \text{(VI4)}$$

6) Learning rate: Is a positive scalar which determines the step with which the optimization function updates the weights generally between 10^{-3} and 1) [43]. We used the value 0.001.

7) Dropout rate: The dropout makes it possible to solve the problem of over-fitting. It consists in deleting a few neurons randomly and temporarily during the learning phase according to a defined percentage. In our approach we have chosen a percentage of 10%.

D. Used technologies

For the development of our application we used, among others, Python for programming, Spyder, Tensorflow and Keras for high-level neural networks, Numpy for scientific computing and Nltk for natural language processing.

E. Expérimentations and results

We performed several executions by varying the network parameters (number of hidden layers, number of neurons per layer). Each execution takes several hours. The results are shown in the following figures “Fig. 12.”. Blue curves for evaluation of learning performance and orange curves for validation data.

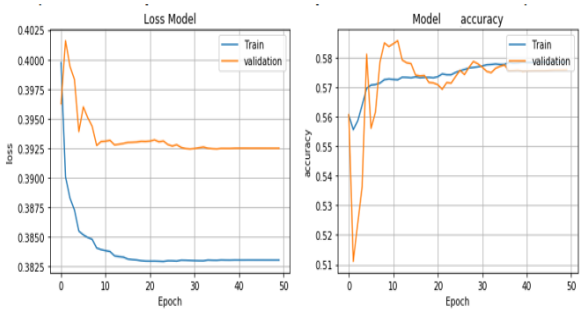

Fig. 12. Our MLP performance with 5 hidden layers

Accuracy value reaches 0.58 and does not improve after iteration 38 “Fig. 13.”.

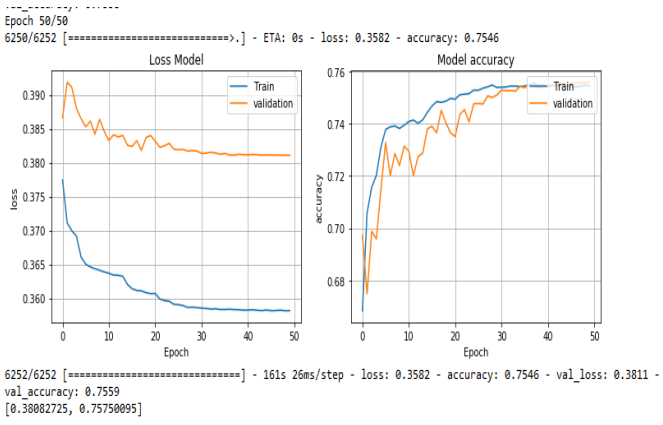

Fig. 13. Our final MLP performance

We notice that for the cost function its value decreases until stagnation. The accuracy increases for the learning and validation data until it stagnates at 0.75 over 50 epochs. The performance of the end network is better.

F. Our application interfaces

The interface “Fig. 14 ", allows launching learning process with viewing of loss and accuracy indicators evolution.

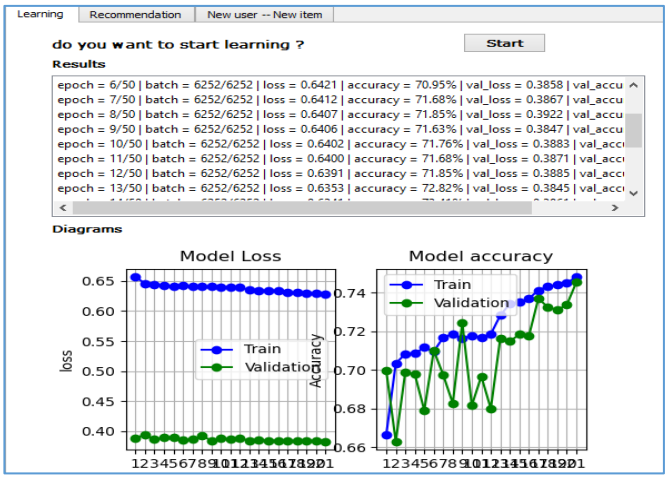

Fig. 14. Launching learning

The following interface "Fig. 15.” displays recommendations to an existing user.

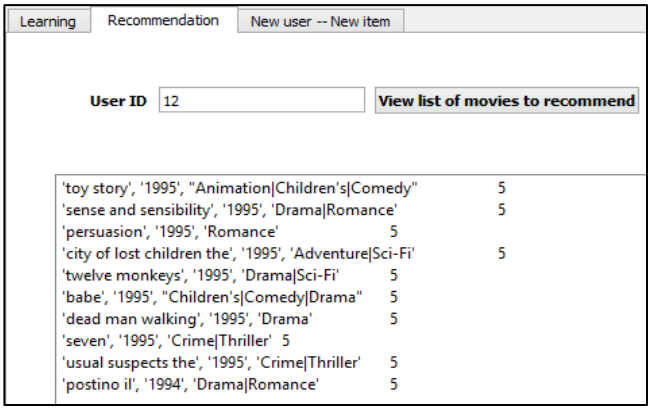

Fig. 15. Recommendation interface

The following interfaces “Fig. 16.” and “Fig. 17.” allows viewing the cases of the new user and / or new movie. If it is the case of a new movie, the list of users for that movie and the list of existing users that this movie might be interested in appears.

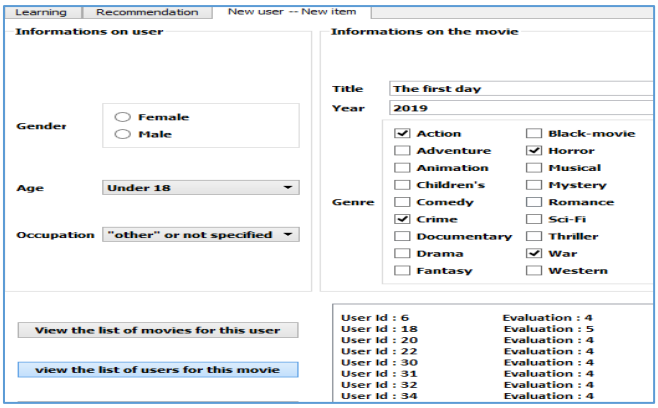

Fig. 16. Recommendation. : New movie case

If it is the case of a new user, a list of recommended movies appears “Fig. 17.”.

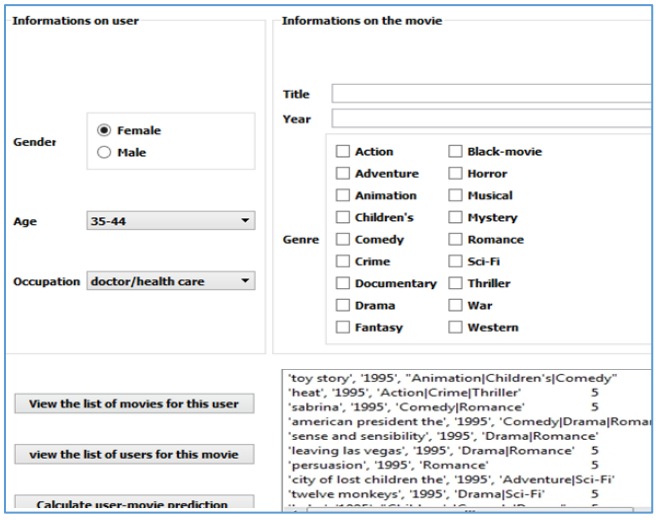

Fig. 17. Recommendation : New user case

If a new user matches a new movie, the system predicts the user's rating for that movie “Fig. 18.”.

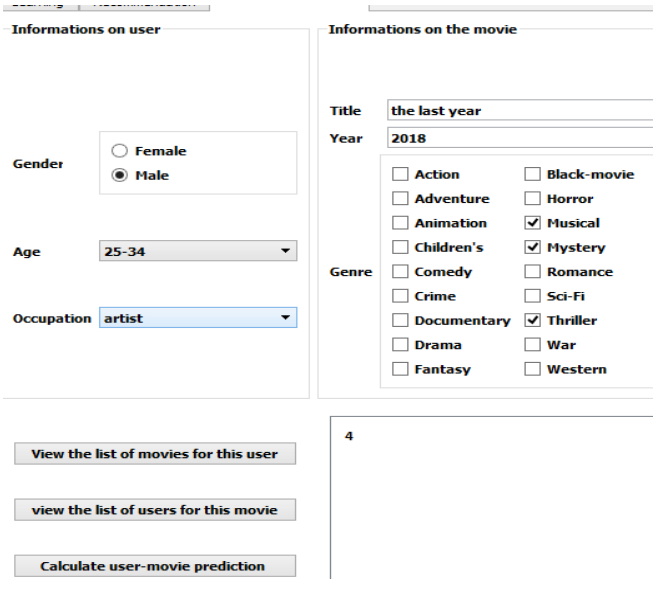

Fig. 18. New user and new movie prediction

The values of the final parameters were fixed after several variation tests. From the results obtained we conclude that our recommendation system by deep neural networks allows to solve most of the limitations of collaborative and content-based approaches which are: new user, new item, cold start, sparsity and subspecialization.

Our approach can be applied for several domains and not only for movies. It can be applied for job search or any type of items like books, products and so on.

VII. SYSTEM 2: SEMANTIC SENTIMENT ANALYSIS

Social networks produce massive data where are expressed users opinions and sentiments about people and companies. Recommendation can be produced by sentiment analysis and completed by semantic understanding of these sentiments. Our objective is to develop a recommendation system by analyzing the sentiment of the expressed opinions and then their semantic understanding in order to offer recommendations in the form of advices and guidance to companies (decision-makers).

The sentiment analysis is carried out with an LSTM deep neural network. The content is then analyzed semantically to find and understand the topics discussed in order to offer recommendations.

In the following sections we present sentiment analysis and deep learning. Then we present hird section we present the semantic analysis and in the fourth the recommender systems. The fifth and sixth sections are devoted to the development of our system. We end with a conclusion and perspectives.

A. Sentiment analysis

Sentiment analysis makes it possible to know the polarity (positive, negative or neutral) of users contents (comments, reviews) [47][48]. It was approached by the lexical approach but currently is essentially based on machine learning particularly deep learning which consists of training and testing a model to classify the text according to its polarity. Deep learning is a very effective technique for learning on large volumes of data in natural language processing [49]. A deep model is built on a multilayer perceptron model where the intermediate layers are more numerous and have automatic feature extraction capabilities. RNNs have repetitive connections to memorize past information. They are suitable for sequential data like text. An LSTM can learn a past long-term. In [50], for example, the authors performed sentiment analysis using SVM, CNN and LSTM where LSTM gave the best accuracy. Sentiment analysis is not enough to understand content. A semantic analysis is therefore necessary.

B. Semantic analysis

Semantic analysis allows understanding the meaning of users comments and opinions on a product or service. Different techniques are used to find the semantics and particularly the subject or theme addressed in a text such as Latent Semantic Analysis (LSA) [51] and Latent Dirichlet Allocation (LDA) [52]. LDA, which is more adapted for topic modeling, is a thematic probabilistic model for revealing the structure of hidden topics in documents. It is an unsupervised learning method. It maps the document to the subject list by assigning each word to a subject. The allocation is estimated on the basis of a conditional

probability. The set of words representing a given topic can be found by choosing the highest probability or by setting a threshold and selecting words with a probability greater than or equal to the threshold.

C. System 2 architecture

The idea in system 2 is the understanding of users contents through sentiment analysis completed and combined with semantic analysis. The purpose is to be able to recommend advices and guidance to decision-makers based on this understanding. This requires determining the sentiment of a content and its topics.
We propose then a system based on LSTM deep neural network for sentiment analysis and on topic modeling using LDA for semantic analysis.

Our system has six main functions:
1. Preprocessing: Data recovery and cleaning of punctuation, emails, URLs, etc.
2. Data embedding: Data encoding and vectorization with a tokenizer and Word2vec.
3. Learning: Training and testing the LSTM using a reviews dataset.
4. Prediction: predict the polarity of new tweets by the trained LSTM.
5. Semantic analysis: Determine topics of these new tweets by LDA.
6. Recommendation: Using the result of the sentiment and semantic analysis, offer advice by explaining the situation (Percentage positive/negative opinion, number of tweets, topics) in comparison with other companies of the same category and/or size.
The "Fig. 19" shows our System 2 architecture.

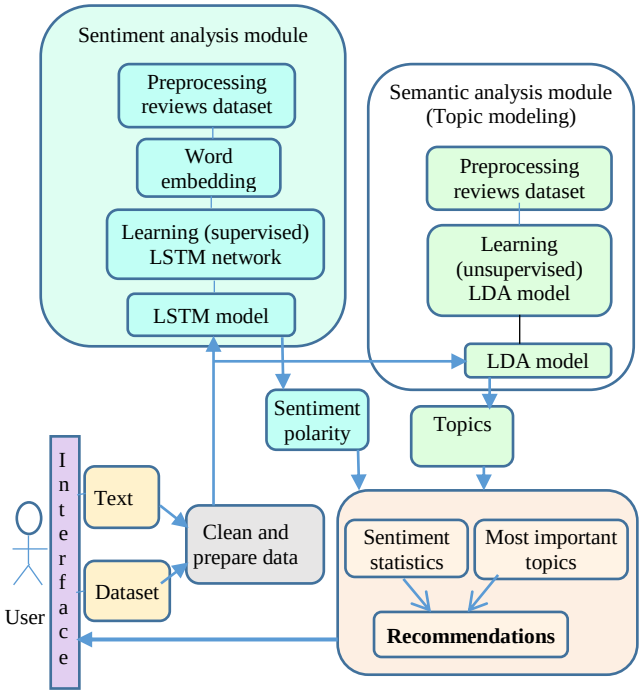

Fig. 19. System 2 architecture

D. LSTM model

For sentiment analysis, we build an LSTM by going through the following steps:

1) Data preparation:

- Use a reviews dataset and clean it as training data.
- Divide the data into two parts 80% for the train and 20% for the test.
- Encode each word with an integer representing it in an index (tokenizing).
  Example: ['Machine Learning Knowledge', 'Machine Intelligence']: [[2,1, 3], [2, 6]]
- Create a Word2Vec model (embedding) which finds for each word a vector of neighboring words.
  Example: The word good: Nice, awesome, wonderful, gorgeous….
- Link the result of Word2Vec model with the tokenizer encoding.

2) Learning:

- Passing each vector generated by the tokenizer and word2vec in the input layer of the LSTM.
- At each pass of a word of 100 values, the weight matrix of the LSTM network is updated and at the end of each sentence (sequence) the training accuracy and training loss are calculated.
- Test the model and saving it if the accuracy and loss are acceptable.

The "Fig. 20" represents the general diagram of our LSTM model.

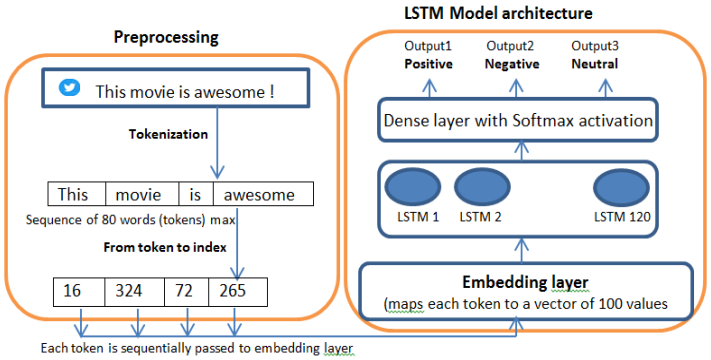

Fig. 20. LSTM model architecture

E. LDA model

LDA is a generative probabilistic model. Specifically it is a three-level ("Fig. 21") hierarchical Bayesian model, for a collection of discrete data (reviews dataset).

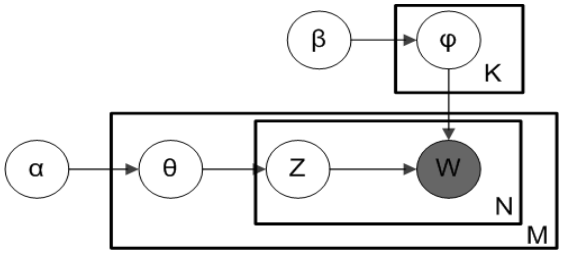

Fig. 21. LDA model

Where :
α : Probability on per document (review) topic distribution
β: Probability on per topic word distribution

θ_m: The topic distribution for document M
φ_k: The word distribution for topic K.
Z_{mn}: The topic for the n-th word in document M
W_{mn} : The specific word

LDA algorithm:

1. Assume there are *k* topics across all of the documents
2. Distribute these *k* topics across document *m* (Dirichlet distribution α) by assigning each word a topic.
3. For each word *w* in document *m*, assume its topic is wrong but every other word is assigned the correct topic.
4. Probabilistically assign word *w* a topic based on:

- what topics are in document *m*
- how many times word *w* has been assigned a particular topic across all of the documents (distribution *θ*)

5. Repeat this process a number of times for each document.

VIII. System 2: Implementation and Experiments

For training, we use a dataset of 3.1 Million Amazon reviews divided into two parts (Train = 80% and Test = 20%) and converted to CSV (Pandas dataframe) format.

A. Step 1: LSTM learning

We clean (preprocess) data.csv file, encode it with the tokenizer and then vectorize it with Word2Vec. We train and test our LSTM model with the following configuration.
LSTM Settings:
- The Word2Vec model was created with the following parameters:
• Vector size = 100 words, Number of epochs = 16, Minimum count = 50.
As results we obtained a Word2Vec model which contains:
• Vocabulary size = 51049 words,
Total word count = 877343 words.
• The embedding matrix of size 877343 x 100 (a vector for each word).
• The size of the sequence (phrase) = 80, - Number of epochs = 8, The batch-size = 64.
- Number of LSTM hidden layer nodes = 120 nodes.
- Softmax activation function.
- Model Inputs: 2 lists (review in vectors of max size 100 and corresponding sentiment).
- Model outputs: 3 values (% positive sentence, % negative sentence, % neutral sentence).
"Fig. 22" shows the graph representing the accuracy and loss of the train and the validation phases.

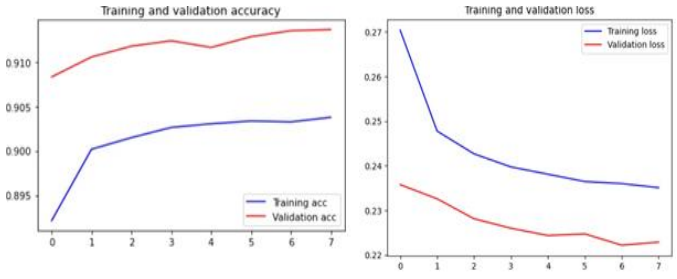

Fig. 22. Accuracy and loss for training and validation

The LSTM evaluation (test) gave:
accuracy: 0.9119, loss: 0.2173

B. Step 2: Using the pre-trained LDA Mallet

For topic modeling, we use the pre-trained LDA Mallet model. Instead of using simple LDA, we use LDA Mallet which allows us to improve the consistency coefficient of topics. This is because LDA Mallet is based on the optimized Gibbs sampling technique. LDA Mallet allowed us to optimize the consistency of topics (topics coherence) up to 65% and improve the result to find for example for 10,000 tweets per day up to 20 topics.

C. Sentiment analysis results

The system operation process is as follows:
• Tweet scrapping (new tweets for a specific brand).
• Import the trained LSTM model and predict the polarity of the cleaned tweets.
The evolution of the number of tweets according to their polarity for 30 days is given in "Fig. 23":

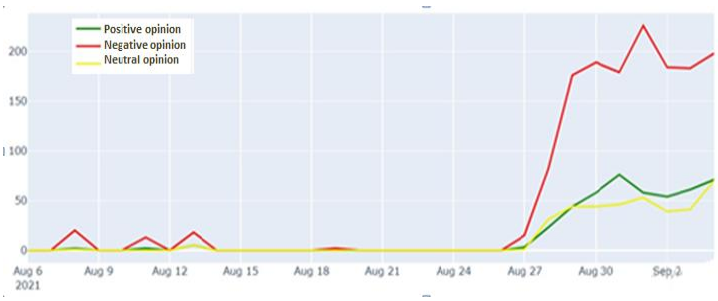

Fig. 23. One month sentiment evolution for a brand

"Fig. 24" represents a comparative table between the compared company (or brand) and the other companies of the same size and/or same category based on the polarity percentage.

	Last 30 days		
	My brand	Same category brand	Same size brand
Positive	19.7 %	29.9 %	31.4 %
Negative	64.1 %	40.7 %	38.9 %
Neutral	16.2 %	29.4 %	29.7 %

Fig. 24. 30 days sentiment evolution for a brand compared to other brands

D. Semantic analysis results

The semantic analysis is carried out using the pre-trained LDA Mallet model which allowed us to find the topics of new tweets whether they are positive or negative ("Fig. 25")

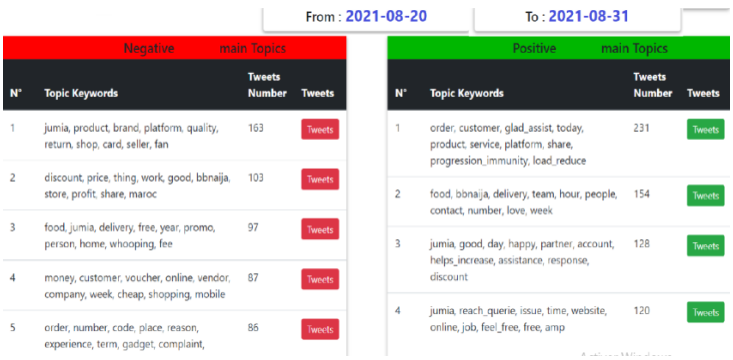

Fig. 25. Topic modeling results

E. Some recommendations

Based on the results of the sentiment and the semantic analysis, we offer recommendations to the user in the form of a summary of the results of the sentiment and semantic analysis and in the form of advices and guidance ("Fig. 26").

From : **2021-08-20** To : **2021-08-31**
Recommendations (advices and notes):
Companies in the same category : you have more reach than the average of the other companies. Keep expending and getting more customers on Twitter with different initiatives like Best comment…
Companies in the same size : You have more reach than the average companies of your level. Keep the good work to maintain the lead.
Companies with same presence : For brands that have the same presence on twitter as you 5660 tweet per month. You are doing a good job and you have a better positive percentage than the average of these companies. Keep monitoring your e-reputation and see what your customers are complaining about or being grateful for. This is the secret to better manage your e-reputation.

Fig. 26. Topic modeling results

IX. DISCUSSIONS

This work is a promising line of research for recommendation systems. Current work on recommendation systems deals with one approach and does not cover all the aspects that can be used to recommend.
Some recommendation systems exploit features related to items or users. Others have focused on content analysis by sentiment analysis or others (topics modeling as an example) but one cannot find operational systems or in research works that treat all these facets (features and content analyses) at the same time. Hence the interest of this work which is only a beginning and requires a finalization essentially for the following two points:
- Integration of the two systems.
- learning of recommendations after construction of a recommendation dataset with the help of experts and on the basis of semantic sentiment analysis.

X. CONCLUSION

In this work, we developed two complementary recommendation systems.

The first one is a hybrid filtering based recommendation system. It's a hybridization of content and demographic which is a particular case of collaborative filtering. Then we made a presentation on neural networks and particularly deep learning and their different architectures.

We built a MLP-type deep learning model where it's input is a set of content and demographic features. The MLP is learned on the basis of the Movielens dataset fragmented into two parts: the large part used for training the model, and the other for validation. The model is fed by training data and its performance is obtained after validation and reaching its improvement limit.

The second one is a semantic sentiment analysis based recommendation system.

For sentiment prediction, we built an LSTM network with 3.1 Million Amazon reviews splited into train and test data. The best learning result is used for efficient polarity prediction (classification).

For the semantic analysis we opted for the LDA model, more precisely LDA Mallet more efficient than simple LDA Model. It allows finding tweets topics whether they are positive, negative or neutral.

As perspectives:

• Add the dialect "Daridja" in our System, since we used the self-translation provided by Google encompassing several languages except "Daridja".

• Optimize topic modeling algorithms to improve topic consistency.

• for the recommendation, and with the help of experts, create a dataset of advices according to the results of the studies (sentiment and semantic analysis), train a model on this dataset for a more effective recommendation.

Our recommendation system can be integrated into an e-commerce site, adapted to other types of items (eg. books) and implemented under a big data or datalake platform.

REFERENCES

[1] Maatallah M. Une Technique Hybride pour les Systèmes de Recommandation[thèse de Doctorat]. Université Annaba , 2016.220.

[2] Jomaa I. Prise en compte des perceptions dans les systèmes de recommandations.[Thèse doctorat]. Ecole Centrale de Nantes, 2013.

[3] Stanislas L. Exploration des réseaux de neurones à base d'autoencodeur dans le cadre de la modélisation des données textuelles. [Thèse de doctorat] Université Quebec; 2016.112.

[4] Burke R. HybridRecommenderSystems: Survey and Experiments. User Modeling and User-Adapted Interaction.2002; 12(4):331–370.

[5] Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions.IEEE.2005; 17(6): 734-749.

[6] Naak A, Un système de gestion et de recommandation d'articles de recherche [mémoire Maîtrise Sciences]. Université Montréal ; 2009.

[7] Burke R. Hybrid web recommendersystems. In: Brusilovsky P. et al. The Adaptive Web. Springer Berlin Heidelberg; 2007.377–408.

[8] Mahmood T, Ricci F. Improving recommender systems with adaptive conversational strategies.ACM.2009 Jan.

[9] Pazos Arias JJ, Fernandez Vilas A, Diaz Redondo RP. Recommender systems for theSocial Web, 2012.

[10] Baeza-Yates R, Ribeiro-Neto B. Modern Information Retrieval. Addison-Wesley.1999.

[11] Ben Ticha S. Recommandation personnalisée hybride. [Thèse de doctorat].Metz : Université de Lorraine; 2015.140.

[12] Schafer JB, Konstan J, Reidl J. Recommender Systems in Ecommerce.ACM.1999.

[13] Montaner M, López B, De La Rosa JL. A Taxonomy of Recommender Agents on the Internet.Artif. Intell. Rev.2003.19: 285-330.

[14] Soualah-Alila F., CAMLearn*: Une architecture de système de recommandation sémantique sensible au contexte. [Thèse de Doctorat]. Université de Bourgogne ; 2015.166.

[15] Pazzani MJ, Billsus D. Content-based recommendation systems. In Brusilovsky P, Kobsa A, Nejdl W. The AdaptiveWeb. Springer-Verlag, Berlin, Heidelberg ;2007. 325–341.

[16] Taouli S, Benachenhou W. Utilisation de factorisation matricielle dans les Systèmes de recommandation sensible au contexte [mémoire de master], Université Tlemcen; 2017.59.

[17] Nguyen AT. COCoFil2 : Un nouveau système de filtrage collaboratif basé sur le modèle des espaces de communautés [thèse doctorat]. Grenoble: Université JF; 2006.162.

[18] Breese JS, Heckerman D, KadieC.Empirical analysis of predictive algorithms for collaborative filtering. UAI. 1998. 43–52.

[19] Ben Schafer J, Frankowski D, Herlocker J, Sen S. Collaborative filtering recommender systems. In: Brusilovsky P. et al. The Adaptive Web. Springer ; 2007.291–324.

[20] Sarwar B, Karypis G, Konstan J, Riedl J. Item based collaborative filtering recommendation algorithms.ACM.2001.285–295.

[21] Lemdani R. Système Hybride d'Adaptation dans les Systèmes de Recommandation [thèse doctorat]. Université Paris-Saclay ;2016.129.

[22] Pazzani MJ. A framework for collaborative, content-based and demographicfeltering. Artif. Intell. Rev. 1999;13 :393-408.

[23] Hadji pavlou E. Big data, surveillance et confiance: la question de la traçabilité dans le milieu aéroportuaire [Thèse doctorat] Université Côte d'Azur ; 2016.

[24] Bourehil M, Boutamine K. Génération de description en langage naturel à partir d'images utilisant de Deep Learning [mémoire de master]. Alger : Université USTHB ; 2017.

[25] Chapelle O, Schölkopf B, Zien A. Semi-supervised learning. In Adaptive computation and machine learning. MIT Press.2006.

[26] Ammar MY. Mise en Œuvre de Réseaux de Neurones pour la Modélisation de Cinétiques Réactionnelles en vue de la Transposition Batch/Continu, thèse doctorat,.I.N.P; 2007.

[27] Ben Rahmoune M. Diagnostic des défaillances d'une turbine à gaz à base des réseaux de neurones artificiels pour l'amélioration de leur système de détection des vibrations [thèse de doctorat]. Djelfa : Université Ziane Achour ; 2017.115.

[28] Asradj Z. Identification des systèmes non linéaires par les réseaux de Neurones[mémoire de magister]. Université Bejaia , 2009.90.

[29] Nouressadat M. Etude des performances des réseaux de neurones dynamiques à représenter des systèmes réels, [mémoire de magister]. Université de SETIF 1 ; 2009.

[30] Bellanger-Dujardin AS. Contribution à l'étude de structures neuronales pour la classification de signatures : Application au diagnostic de pannes des systèmes industriels et à l'aide au diagnostic médical [thèse de doctorat]. Paris : Université Paris XII. 2003.117.

[31] MoualekDjaloul Y. Deep Learning pour la classification des Images [mémoire de master]. Tlemcen : Université Abou Bakr Belkaid.2017.

[32] Boughaba M, Boukhris B.L`apprentissage profond pour la classification et la recherche d'images par le contenu [mémoire de master].: Université Ouargla;2017.

[33] Weibo L, Zidong W, Xiaohui L, Nianyin Z, Yurong LD, Fuad E. A survey of deep neural network architectures and their applications. Alsaadid Neurocomputing ; 234 (2017).

[34] Buyssens P, Elmoataz A. Réseaux de neurones convolutionnels multi-échelle pour la classification cellulaire. RFIA. Jun 2016.

[35] Randrianarivony MI. Détection de concepts et annotation automatique d'images médicales par apprentissage profond [mémoire master].Université d'Antananarivo; 2018.50.

[36] Zhanga Q, Yang LT, Chenc Z, Li P. A survey on deep learning for big data. Information Fusion.2018; 42 (2018 :146–157.

[37] Bouaziz M. Réseaux de neurones récurrents pour la classification de séquences dans des flux audiovisuels parallèles [Thèse de doctorat] Université d'Avignon ; 2017.130.

[38] Cho k, MerrienboerB, Gulcehre C, Bougares F, Schwenk H, Bahdanau D, Bengio Y. Learning phrase representations using rnnencoder-decoder for statistical machine translation Conference (EMNLP).2014;

[39] Jaumard-Hakoun A. Modélisation et synthèse de voix chantée à partir de descripteurs visuels extraits d'images échographiques et optiques des articulateurs. [Thèse de doctorat].: Université Paris 5 ; 2016.153.

[40] Xiangnan H, Lizi L, Hanwang Z, Liqiang N, Xia H, Tat-Seng C. Neural Collaborative Filtering.ACM. 2017.

[41] Kuchaiev O, Ginsburg B. Training Deep AutoEncoders for Collaborative Filtering. Stat-ML. 2017.

[42] Artem Oppermann, Deep Autoencoders For Collaborative Filtering. Predicting the Rating a User would give a Movie - A practical Tutorial, Towards Data Science, 2018.

[43] Ricci F, Rokach L, Shapira B. Introduction to Recommender Systems In Ricci F. et al. Recommender Systems Handbook. Springer; 2011.

[44] Abbad H. Estimation des distances des obstacles à partir des images à l'aide de l'apprentissage automatique auto-supervisé [mémoire de master].Université USTHB ; 2017.

[45] Brownlee J, Gentle Introduction to the Adam Optimization Algorithm for Deep Learning, 2017 in Deep Learning Performance.

[46] Soo-Min K et Eduard H , Determining the sentiment of opinions .

[47] Wiebe J et al., Annotating expressions of opinions and emotions in language, Kluwer Academic Publishers ; 2005.

[48] Liu B, Sentiment analysis and opinion mining, Morgan & Claypool Publishers ; 2012.

[49] Goodfellow I et al., Deep Learning, The MIT Press ; 2016.

[50] Guggilla C et al., CNN-and LSTM-based claim classificationin online user comments, In Proceedings of the International Conference on Computational Linguistics (COLING2016) ; 2016.

[51] Landauer, T. K., Foltz, P. W., & Laham, D. , Introduction to Latent Semantic Analysis. Discourse Processes, 1998 , 25, 259-284.

[52] Blei D.M., Ng A.Y. and Jordan, M.I., Latent Dirichlet Allocation. J Machine Learning Research Archive, 2003, 3, 993-1022.