IJCNN 2019. International Joint Conference on Neural Networks. Budapest, Hungary. 14-19 July 2019

A Deep Neural Network Model for Predicting User Behavior on Facebook

Hanen Ameur*, Salma Jamoussi† and Abdelmajid Ben Hamadou‡
Multimedia InfoRmation system and Advanced Computing Laboratory
University of Sfax
Sfax, Tunisia
*ameurhanen@gmail.com, †salma.jamoussi@isimsf.rnu.tn, ‡abdelmajid.benhamadou@gmail.com

***Abstract*—On Facebook, the social media site, liking, commenting and sharing the posts or status of other users are usually considered to be the key mechanisms for exchanging opinions about different topics. Due to the non-availability of data and security constraints, only few research studies have analyzed such behavior. In this paper, we introduced a novel deep neural network model for user behavior prediction (like and comment). We presented an embedding representation method for the textual content of comments and posts based on the contextual recursive auto-encoders model. The users were represented using a deep joint auto-encoders model to fuse the users' like and comment information, and train the users' combined embedding representation. Then, the user behaviors towards a given post were embedded into the same feature space of users and posts, using the joint auto-encoders model. Thereafter, we used a fully connected layer for behavior prediction. To train and evaluate the effectiveness of the proposed method, we also constructed a large dataset collected from Facebook. The experimental results show that the proposed method could achieve better results than the previous alternative methods.**

***Index Terms*—Behavior prediction, Deep learning, Embedding representation, Contextual Recursive Auto-Encoders, Joint Auto-Encoders, Facebook.**

I. Introduction

THE rising popularity of online social media networks (e.g. Facebook, twitter, Digg) have driven the studies on social behavior to a new higher level. Social behavior prediction systems have received a vast amount of interest the past few years. Nowadays, social media networks provide researchers with a valuable and rich social data, which allows them to get insights of human behaviors and analyze them. The user behavior is considered as a fundamental element in social media, covering various social activities that the users can do online. Each social media network is characterized by different activities such as, like, comment, share, tweet, retweet, rating, reply,etc. Understanding and predicting a user's behavior is an important and valuable task in several recommendations, and personalization applications, such as e-commerce activities, advertising policies, detecting online protest participation, cybersecurity and detecting manifestations. Thus, it is considerably important to analyze the users' behavior.

Previous methods studied this problem using various linguistic features, users' personal information and many other manually constructed features to achieve the prediction task. Feature engineering has always been a laborious task, required to obtain the external sources but these are difficult to get or not always available. Recently, the success that deep learning methods have achieved in such fields as computer vision and natural language processing has naturally motivated their application in a behavioral prediction task (e.g. retweet, rating). These methods aim at automatically learning optimal distributed feature representations from the data as an alternative to handcrafted feature engineering. There are two general ways of applying the deep learning methods to a behavior prediction system. These consist of either modeling the interaction among users and items (e.g. tweets, products, posts) or processing the raw features of users and items by exploiting textual content information. Among the most used deep learning methods in a behavior prediction system, we can cite the restricted boltzmann machines (RBM) [1], convolutional network (CNN) [2] and auto-encoders (AE) [3].

In this paper, we proposed a novel deep neural network method to predict how users behave towards given posts when they are connected to Facebook. We explored the capabilities of the auto-encoders architecture to embed the textual content and learn multi-view representations. To this end, the contextual recursive auto-encoders model was introduced to recursively combine the embedding representation of the words composing sentences by taking into account the word contextual information and respecting the word order. Thus, a joint auto-encoders model was proposed to learn multi-view representations from attributes which are automatically inferred from the users' likes and comments. These two users' views, on which our model is trained are initially represented in the same feature representation. This model was used to fuse users and posts information into the same feature space to embed user behavior and predict the behavior class.

The main contributions of this work can be summarized as follows: 1) we defined the problem of predicting how Facebook users behave towards a given post. We treated this problem as a multi-label classification (like and/or comment). 2) we proposed a novel deep neural network to solve this problem. The textual content of users' comments and posts were embedded with a new contextual recursive auto-encoders model in order to combine word embedding. A joint auto-encoders model was introduced to learn a fused representation of users from their like and comment views. This model is also used to combine the user and post representations to embed a

978-1-7281-2009-6/$31.00 ©2019 IEEE

Personal use is permitted, but republication/distribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

user behavior. 3) we constructed a large dataset from Facebook to train and evaluate the proposed model. Our experimental results proved that the proposed model achieved better results than baseline models.

II. Related Work

There are two research trends related to our work and to the proposed models but dealing with other kinds of social behavior (like rating and retweeting). The first is the work on the task of recommendation on social media and retweet prediction using deep learning techniques. The second is the work based on multi-view learning methods to learn shared representations. In this section, we briefly reviewed these two research areas and distinguished our work from the existing methods.

Several works [4], [5] model users and/or items from the rating matrix using neural networks like auto-encoders or Restricted Boltzmann Machines (RBM). They are considered as collaborative-based techniques because they only use the rating matrix and ignore the text review. [6] used deep models of CNN and Deep Belief Network (DBN) to learn latent factors from music data for music recommendation. [7] applied a generalized Stacked AutoEncoder model for music recommendation. All these works have ignored the text review. Other studies [8], [9] have taken into account the review text to improve the recommendation. [8] proposed a model consisting of a matrix factorization technique and a Recurrent Neural Network (RNN). The matrix factorization is responsible for learning the latent factors of users and items, and the RNN models the likelihood of a review using the item latent factors. [9] proposed Deep Cooperative Neural Networks (Deep-CoNN) for rating prediction that consists of two parallel convolutional neural networks coupled in the last layers. One focuses on learning the users behaviors and the other learns item properties. [2] extends the DeepCoNN model by introducing an additional latent layer representing the target user-target item pair. [10] proposed a novel context-aware recommendation model, convolutional matrix factorization that integrates convolutional neural network into probabilistic matrix factorization. [11] proposed a deep knowledge-aware network that is a content-based on deep recommendation for news recommendation. In our work, we relied on the texts written by users and texts describing posts to predict the Facebook behavior.

In order to predict the retweet behavior, there are two categories of studies which investigate the problem from different view points. These are the matrix factorization based and the classification/regression based methods. [12] used the Factorization machine method to learn an interpretable user representation for retweet prediction by jointly modeling a user decision and interest. [13] proposed a flexible model that captures a number of behavior signals affecting a user retweet decision. [14] focused on learning users' retweeting behavior representations from a message content and author information. They proposed a hybrid co-factor matrix factorization to capture the interactive effect between a user message interaction and deep semantics of the message content using word2vec embedding. [8] proposed a deep neural network model for retweet prediction. They used the convolutional neural network to represent the embeddings of the user, the attention, the user interests, the author and the tweet. Afterwords, these embeddings would be combined in a fixed feature vector.

These previous studies worked on the tasks of the retweet and rating prediction. In our work, we investigated the problem of predicting the users like/comment behavior towards a post on a Facebook site. To the best of our knowledge, this is the first work that analyzes and predicts Facebook behavior.

Recently, multi-view unsupervised learning method variants have achieved very good results in various computer vision tasks. Some studies [15], [16] used the restricted Boltzmann machines to combine different modalities (video, audio, text). Other studies [3] proposed multi-view models based on the auto-encoders method to learn the shared representation, such as correlation neural network (deep-corrnet) [17], [18] and multimodal auto-encoders [19], [20]. In this paper, we used the deep auto-encoders to combine heterogeneous user information (like and comments), and to project the user's history and post onto a unified representation that fuses them together.

III. Proposed Model Architecture

This paper aims at presenting a new user behavior prediction model. We treated the user's behavior prediction task as a multi-label classification problem where each Facebook user can like or/and comment a given post. For a given user u_j ($j \in \{1, ..., n_U\}$, n_U is the total number of users in the corpus), we denote his reaction's history as a set of comments $\{c_1^j, c_2^j, ..., c_{N_j}^j\}$ and a set of liked post $\{p_1^j, p_2^j, ..., p_{M_j}^j\}$, where c_k^j $(k = 1, ..., N_j)$ is the k-th comment of a user u_j, p_k^j $(k = 1, ..., M_j)$ is the k-th post liked by the user u_j and N_j and M_j are the total number of a user i's comments and the total number of liked posts, respectively. Each post p_i ($i \in \{1, ..., n_P\}$, n_P is the total number of posts in the corpus) consists of a sequence of sentences, that represents the description and the message bearer in the post. Hence, given a user's history and a new post, we aim to predict whether the user will just click on "like" to show his agreement or/and will comment the post or will not react (be neutral).

An overview of the architecture of the proposed model is given in Fig. 1. The two main inputs of this model are the user and the candidate post[1], as shown in Fig. 1. We first introduce the Contextual Recursive Auto-Encoders "CoRAE" model for encoding the textual contents of comments and posts. Then, we discuss the process of joint auto-encoders network for modeling the user's history by generating the fused user embedding from two inputs (his comments and his liked posts). This network is used to combine the user embedding and post embedding to represent the user behavior towards a given post. Finally, we apply a fully-connected layer to obtain the final prediction.

[1]A Facebook post is composed of a sequence of sentences, that represents the description, the caption and the message bearer in the post.

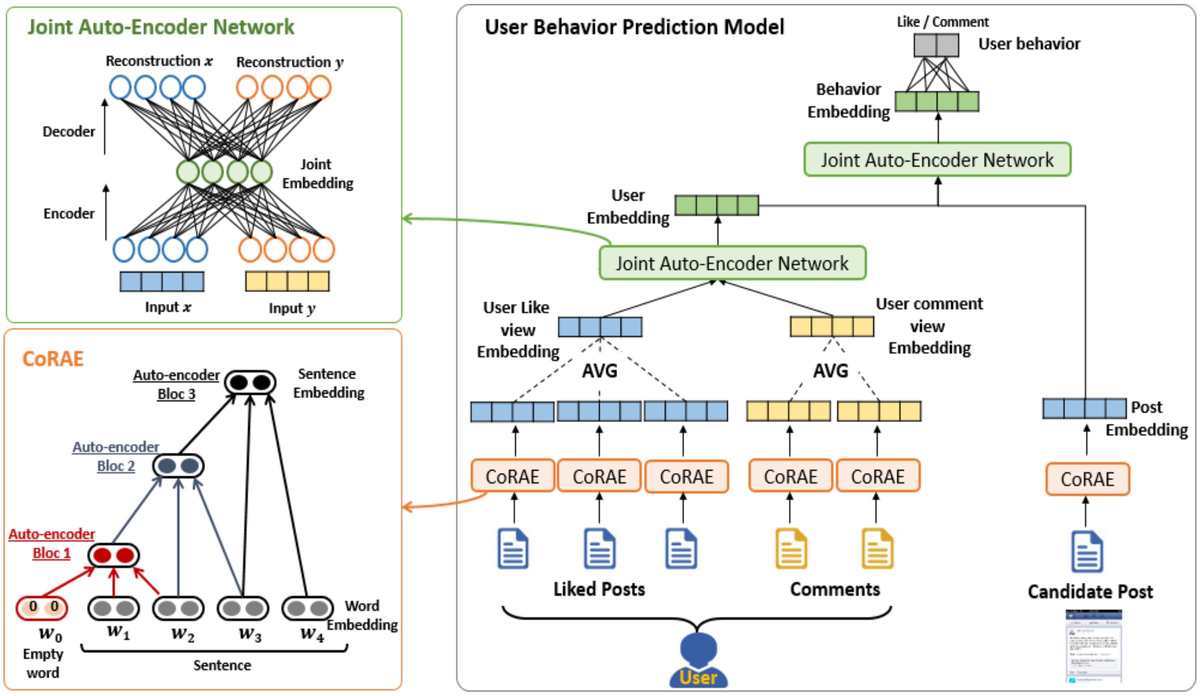

Fig. 1. Illustration of the proposed deep neural network for predicting user behavior.

A. Contextual Recursive Auto-Encoders

Before introducing the proposed deep neural network model, we presented our CoRAE model [21] which is used for embedding the comments and posts textual content. CoRAE aims at representing the sentence using the embedding vector of its words, based on a deep compositional technique, without resorting to the use of a given parse tree structure. In fact, the social media data (e.g. comments) are generally grammatically incorrect, which makes the parsing step very difficult.

The Facebook posts and comments can be generally written with a variety of texts of different length. For this reason, their textual content is represented by averaging the embedding vectors of its sentences using the equation $v_e = \sum_{i=0}^{n} e_{s_i}/n$, where, n is the total number of sentences making up a document (post or comment) and e_{s_i} is the embedding vector of sentence s_i.

We assume that a sentence s_i consists of a sequence of m words $s_i = (w_1, w_2, ..., w_m)$, where each word w_j is represented by a d-dimensional vector embedding $w_j = (x_1, x_2, ..., x_d)$. First, the word2vec embedding representation was used to convert each word to a word vector. Then, these word vectors were given as CoRAE's input. The proposed CoRAE model aims to map a sentence to a d-dimensional vector, based on its words and their corresponding embeddings. It recursively combines the word vectors constituting a sentence by considering the context and the word order. In fact, the meaning of a sentence is deduced iteratively by the appearance of its words. Besides, the meaning of a word is related to the context in which it appears. Indeed, the word is influenced by the meaning of its context words (the previous and the following words).

As shown in Fig. 1 (bottom left), the CoRAE model is considered as a concatenation of a sequence of auto-encoder blocs which are recursively trained. In every auto-encoder bloc, the parent node vector p_i (hidden layer) is computed using the formula 1, by merging three children (input layer), the word w_i and its neighbor context which are the w_i's previous content in s, p_{i-1} and the next word w_{i+1}.

$$p_i = f(W_1^i \begin{bmatrix} p_{i-1} \\ w_i \\ w_{i+1} \end{bmatrix} + b_1^i), \tag{1}$$

where (p_{i-1}, w_i, w_{i+1}) is simply the concatenation of the three children p_{i-1}, w_i and w_{i+1}, f is an element-wise activation function such as $\tanh$, $W_1^i \in \mathbf{R}^{d\times 3d}$ is the encoding weight matrix that we want to learn ($3d$ is the number of input units) and b_1^i is the encoding bias vector.

One way to obtain the best d-dimensional vector which represents its direct children is to decode the parent node vector p_i in a reconstruction layer using formula 2 and then to calculate the reconstruction error between the original input and its reconstructed vectors using equation 3. During training, the goal is to minimize the reconstruction error E_{rec} of all inputs.

$$\begin{bmatrix} p'_{i-1} \\ w'_i \\ w'_{i+1} \end{bmatrix} = f(W_2^i p_i + b_2^i) \tag{2}$$

$$E_{rec}(p_i) = \left\| \begin{bmatrix} p'_{i-1} \\ w'_i \\ w'_{i+1} \end{bmatrix} - \begin{bmatrix} p_{i-1} \\ w_i \\ w_{i+1} \end{bmatrix} \right\|^2 \tag{3}$$

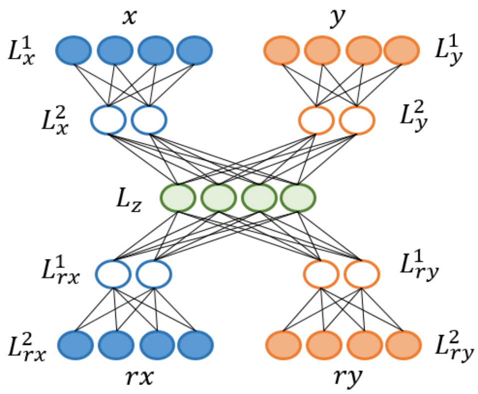

Fig. 2. The deep joint auto-encoders "DeepJAE" network.

where $(p'_{i-1}, w'_i, w'_{i+1})$ are reconstructed vector and W_2^i, b_2^i are the decoding weight matrix and bias vector respectively.

In order to apply the auto-encoder recursively, the same steps were repeated. In fact, the above process was repeated until the last word w_m was reached and we had a reconstruction error at each parent node. Formally, the CoRAE model contains $m-1$ auto-encoder blocs, with m is the number of words in a given sentence. CoRAE begins the learning process with the generation of the parent node from the concatenation of (empty word w_0,w_1, w_2). It finishes the process by generating the last parent node vector from the three children (p_{m-2}, w_{m-1}, w_m).

B. Joint User Embedding Representation

Using CoRAE model, we encoded each post p and comment c with continuous vectors $v_p \in \mathbf{R}^d$ and $v_c \in \mathbf{R}^d$, respectively.

In order to perform a neural network framework for predicting the user's behavior toward a given post, we propose to represent the users in the same space where the posts are represented with continuous distributed vectors. Each user is represented by his history of reactions towards posts, i.e. liking and commenting some shared posts. In fact, the user expresses his approval and his adoption of the main idea published in the post by liking it, and he writes his opinion by commenting the post. For this reason, to build the best user embedding representation, it is crucial to highlight these two user views (liked posts and comments) that capture and characterize the user interest.

In the first step, we combine all the comments of each user by averaging their embedding vectors to represent the user j's comment view v_C^j. In the same way, we generate the representation of a user j's like view v_L^j by averaging the embedding vectors of all the liked posts. In the second step, our goal is to find the joint representation from these two user views. To do so, we performed the auto-encoder network with two disjoint inputs and outputs (one for each view), with separable hidden layers, as illustrated in Fig. 2. In other terms, the two views are available in the input and the both are reconstructed. Thus, this network includes a one fully connected hidden layer in common that interacts with both views in order to learn a joint representation. Indeed, the middle hidden layer activation is used as a bi-view embedding representation "fused representation".

Fig. 2 shows the deep joint auto-encoders "DeepJAE" topology that is equivalent to using two separate deep auto-encoders and tying them in one hidden layer. Each deep auto-encoder tries to reconstruct its input by following multiple encoding and decoding steps. As shown in Fig. 2 the encoder part in the joint auto-encoders consists of the layers (L_x^1, L_y^1, L_x^2, L_y^2 and L_z) while the decoder part consists of the layers (L_z, L_{rx}^1, L_{ry}^1, L_{rx} and L_{ry}).

Formally, let the data given by $x \in \mathbf{R}^d$ for the first view and $y \in \mathbf{R}^d$ for the second view (in our case, x is the user like view whereas y is the user comment view). In the hidden layers L_x^1 and L_y^1, the inputs x and y are encoded into lower dimension representations h_x^1 and h_y^1. $h_x^1 \in \mathbf{R}^d$ and $h_y^1 \in \mathbf{R}^d$ denote the activation of the hidden layers for x view and y view, respectively (see the equations 4 and 5). Then, a shared hidden layer L_z consists in merging h_x^1 and h_y^1 to produce the joint representation of both views. The fused representation $h_z \in \mathbf{R}^d$ is computed using the equation 6.

$$h_x^1 = f(W_{ex}^1 \ x \ + \ b_{ex}^1) \tag{4}$$

$$h_y^1 = f(W_{ey}^1 \ y \ + \ b_{ey}^1) \tag{5}$$

$$h_z = f(W_{ex}^2 \ h_x^1 \ + \ W_{ey}^2 \ h_y^1 \ + \ b_e^2) \tag{6}$$

where, $W_{ex}^{1,2}$, $W_{ey}^{1,2}$, b_{ex}^1, b_{ey}^1 and b_e^2 denote the weight matrix and the bias vectors.

Next, h_z will be decoded into two disjoint representations h_{rx}^1 and h_{ry}^1 with the same size of the h_x^1 and h_y^1 representations (see equation 7 and 8), in order to reconstruct h_x^1 and h_y^1. Finally, h_{rx}^1 and h_{ry}^1 representations are also decoded into rx and ry to reconstruct both view representations x and y by computing equations 9 and 10.

$$h_{rx}^1 = f(W_{dx}^1 \ h_z \ + \ b_{dx}^1) \tag{7}$$

$$h_{ry}^1 = f(W_{dy}^1 \ h_z \ + \ b_{dy}^1) \tag{8}$$

$$rx = f(W_{dx}^2 \ h_{rx}^1 \ + \ b_{dx}^2) \tag{9}$$

$$ry = f(W_{dy}^2 \ h_{ry}^1 \ + \ b_{dy}^2) \tag{10}$$

where, $W_{dx}^{1,2}$, $W_{dy}^{1,2}$, $b_{dx}^{1,2}$ and $b_{dy}^{1,2}$ denote the weight matrix and the bias vectors. We assume that each layer admits f as a non-linear activation function (tanh, softmax or relu). Training the joint auto-encoders is achieved by reducing the distance between the original data (input vectors x and y) and its reconstruction (output vectors rx and ry). In fact, it consists in minimizing the error of reconstructing x^i from rx^i and y^i from ry^i using equation 11 based on the mean squared error "mse" distance.

$$Err = \frac{1}{n}\sum_{i=1}^{n}(\|rx^i - x^i\|^2 + \|ry^i - y^i\|^2) \tag{11}$$

C. User Behavior Prediction

At this stage, all Facebook posts are represented by embedding vectors using the CoRAE model. Additionally, all users (their history) are also represented in the same feature space by applying the deep joint auto-encoders "DeepJAE" model which allows fusing the information about a user likes and comments. Hence, given a user embedding v_u and a post embedding v_p, our objective was to predict the user behavior towards this post. In this paper, user behavior prediction task was dealt as a multi-label classification problem where each Facebook user can like or/and comment or be neutral to a post.

1) User Behavior Embedding Representation: To achieve the classification step, we proposed to represent the user behavior toward a candidate post giving the pair (v_u, v_p). For this reason, we applied the joint auto-encoders "JAE" model in its simplest form (without any separated hidden layers) to learn the behavior embedding representation, as shown in Fig. 1 (top right). Indeed, the JAE model aims at learning to reconstruct the inputs v_u and v_p. It is trained to encode these inputs into a better joint representation between user and post. This joint representation (shared representation) is considered as the user behavior embedding representation which is calculated as follows.

$$v_b = f(W\ v_u\ +\ V\ v_p\ +\ b) \tag{12}$$

Where W and V are the weight matrix and b denotes the bias vector. Then, the behavior representation v_b is decoded into rv_u and rv_u to reconstruct v_u and v_p using the following equations:

$$rv_u = f(W'\ v_b +\ b') \tag{13}$$

$$rv_p = f(V'\ v_b +\ b') \tag{14}$$

where W' and V' denote the weight matrix, b' is the bias vector and f is the non-linear activation function.

2) Behavior Prediction as Multi-Label Classification: The multi-label classification was performed using a fully-connected layer with a logistic (Sigmoid) activation function σ (see equation 15). This layer contains k output neurons that correspond to the studied classes, namely: "like" and "comment". The multi-label classification task was treated as K different binary and independent classifications, where each output neuron decides whether the behavior example belongs to a class or not. In other words, the probabilities of each class are independent of the probabilities of the other classes.

$$\sigma(x)\ =\ \frac{1}{1+e^{-x}} \tag{15}$$

where x denotes the input of the final layer which is the activation of the middle hidden layer "shared layer" for user v_u and post v_p. The output value of σ function in each neuron is a probability between $(0,1)$; if this value is above or equal a cut-off threshold T_σ (which is tuned by grid search on the validation dataset), then it is worth 1 and 0 otherwise. We used a binary cross-entropy loss function L (see equation 16). In this study, the stochastic gradient descent (SGD) was applied to optimize our prediction model by setting the learning rate, decay rate and momentum.

$$L\ =\ -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{K}[y_j^{(i)}log(\widehat{y}_j^{(i)})+(1-y_j^{(i)})log(1-\widehat{y}_j^{(i)})] \tag{16}$$

where n is the number of training examples. K is the number of classes corresponding to the number of output neurons ($K = 2$). $y_j^{(i)}\ \in \{0,1\}$ and $\widehat{y}_j^{(i)}$ are the j-th real label and the j-th predicted label of the i-th example, respectively.

IV. Experiments

In this section, the performed experiments were presented together with their corresponding results. Before showing our method performance, we first focused on the dataset construction. We also described the used baseline models and compared them to our own user behavior prediction model.

A. Data Construction

We explored the Facebook corpus that was created and collected by [22]. Indeed, this corpus is extracted from the political Tunisian pages during the period [1-Jan-2011, 1-Aug-2012]. It contains 22 pages among the most popular ones in Tunisia during the revolution period. Each page is a set of posts where each is characterized by a description, message, users' likes list, comments list, publication date, etc.

To analyze and predict the user behavior, we collected the dataset from these Facebook pages in the following ways. First, we selected 1192 anonymous users among the most active ones. In this step, we crawled 4703 Facebook posts. In these posts, we find that there are 39081 users' behaviors (reactions) with 30329 likes and 8752 comments. Second, we prepared the dataset as a set of multilabel behaviors (1|0: like, 0|1: comment and 1|1: like and comment). In the case where a user has commented a post several times, we concatenate them in the same comment by separating them by "<ssss>". So, we obtained 33789 users behaviors. Then, we performed the pre-processing step applied in [22] to prepare the textual contents and avoid noise. Finally, we randomly split the total dataset into three subsets (70% as training data, 10% as validation data, 20% as test data). We repeated this decomposition step to form two examples of corpora (corpus 1 and corpus2) with different training and test data-sets. This allowed us to examine the performance of the proposed models facing two different corpora. In the following, we displayed the results relative to each corpus as well as the summary of the results by averaging those obtained with each corpus.

Generally, a Facebook user may not react to some posts that do not belong in his/her area of interest. To evaluate and validate our prediction model facing this kind of behaviors (called 0|0: neutral), we annotated 350 examples of a user's neutral behaviors towards some posts of the treated corpus. In fact, we added 100 neutral behavior examples to the validation set and the rest of the examples (250 neutral behaviors) to the test set.

User Embedding models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
AVG	77.79	84.41	80.97	79.61
Weighted AVG	78.27	84.15	81.10	79.84
Simple Auto-encoder	77.82	89.36	83.19	82.81
deep-cornet	79.49	87.56	83.33	82.86
JAE	79.82	87.65	83.55	82.99
DeepJAE	79.16	89.57	84.04	83.07

TABLE I

RESULTS OF DIFFERENT MODELS FOR THE JOINT USER EMBEDDING REPRESENTATION (CORPUS 1).

User Embedding models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
AVG	78.12	82.08	80.05	79.18
Weighted AVG	79.73	82.06	80.88	79.83
Simple Auto-encoder	79.5	85	82.15	81.53
deep-cornet	78.03	88.34	82.86	81.65
JAE	77.77	89.17	83.09	82.11
DeepJAE	79.73	88.05	83.68	82.49

TABLE II

RESULTS OF DIFFERENT MODELS FOR THE JOINT USER EMBEDDING REPRESENTATION (CORPUS 2).

User Embedding models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
AVG	77.95	83.24	80.51	79.39
Weighted AVG	79	83.10	80.99	79.83
Simple Auto-encoder	78.66	87.18	82.67	82.17
deep-cornet	78.76	87.95	83.09	82.25
JAE	78.79	88.41	83.32	82.50
DeepJAE	79.44	88.81	83.86	82.78

TABLE III

RESULTS OF DIFFERENT MODELS FOR THE JOINT USER EMBEDDING REPRESENTATION (AVERAGE RESULTS OBTAINED WITH CORPUS 1 AND CORPUS 2).

B. Experiment Configurations

We implemented the proposed models for generating embedding representation of users, posts and behaviors, based on R **Keras** package published by cran[2]. It facilitates the manipulation of neural network models (creation, training and evaluation). We empirically configured the values of hyper-parameters which control the models learning process, such as: epoch number, batch size, optimization function and activation function. In order to choose the best hyper-parameter values, extensive series of experiments were achieved by varying the parameters according to our dataset.

To initialize the word vectors, the publicly available word2vec[3] embedding vectors were used in this paper. They were trained using the skip-gram model on the input training data. In fact, the skip-gram model aims at finding word representations useful for predicting the surrounding words in a sentence. The number of vocabulary words is 25481. Among these, the most frequently used words were noted to appear at least 6 times, according to the words distribution. The best dimension of the word embedding vectors is 100. In the same dimension space, we represented the users, posts and behaviors. In the CoRAE model, We opted for “tanh” as an activation function, since the feature vector representing the words and sentences includes positive and negative values within -1 and 1.

To learn the joint user embedding representation, we relied on the DeepJAE model with separated hidden layers of 50 neurons. In fact, we applied the same dimension reduction (half) of the users views representation (like and comment) to construct a user representation in the same vector space of these views ($d = 100$). We compared the reconstruction error using the validation set for different batch sizes ranging from 10 to 200 examples. we found that using a 10-batch size, the reconstruction error using the validation set reaches 0.1483, then decreases to 0.1168 with 100-batch size and further increases with the other sizes. Therefore, we opted for 100-batch size. We used the Adam optimizer and we tested our model on the validation set with up to 150 epochs, we found that the compromise between time and model performance seems well respected by limiting the epoch number to 45.

In the proposed user behavior prediction model, the loss and accuracy were compared using the validation set by testing a list of batch sizes (50,100 and 200 examples). We found that with a batch size of 100 examples, we reached a loss of 0.4764 and an accuracy of 80.57 which was a satisfactory performance. Thus, we applied the early stopping method to ensure a sufficient number of epochs without falling on an over-fitting problem. So, relying on the experimental results, we decided to set the epochs number of 197.

To evaluate the performance of our model, we use the precision, recall, F-score and accuracy measures.

C. Experimental Results and Discussion

In the proposed behavior prediction model, “DeepJAE” model was used to learn the user embedding representation that combines its two views (like and comment) and the little deep “JAE” model was applied to compute the user behavior embedding from the user and post representations. In this section, we went through and evaluated the achieved results by comparing them with those obtained by other baseline models.

In order to evaluate the user embedding representation quality and the DeepJAE model effectiveness, we compared it to other models that combine the users’ like and comment views, namely:

- The weighted average was the baseline method that consists in averaging the like embedding vector (weighted by 0.4) and comment embedding vector (weighted by 0.6).

[2]https://keras.rstudio.com/index.html

[3]http://deeplearning4j.org/word2vec.html

Behavior Prediction models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
DeepCoNN	79.78	86.03	82.78	81.75
Concatenate [U,P]+MLP	78.29	88.50	83.08	82.09
Our model				
with DeepJAE[U,P]	78.97	88.51	83.46	82.35
with JAE[U,P]	79.16	89.57	84.04	83.07

TABLE IV
THE PERFORMANCES OF DIFFERENT BEHAVIOR PREDICTION MODELS (CORPUS 1).

Behavior Prediction models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
DeepCoNN	79.45	84.77	82.02	81.37
Concatenate [U,P]+MLP	78.12	86.51	82.10	81.56
Our model				
with DeepJAE[U,P]	79.31	87.54	83.22	82.01
with JAE[U,P]	79.73	88.05	83.68	82.49

TABLE V
THE PERFORMANCES OF DIFFERENT BEHAVIOR PREDICTION MODELS (CORPUS 2).

Behavior Prediction models	Recall (%)	Precision (%)	F-score (%)	Accuracy (%)
DeepCoNN	79.61	85.4	82.4	81.56
Concatenate [U,P]+MLP	78.20	87.50	82.59	81.82
Our model				
with DeepJAE[U,P]	79.14	88.02	83.34	82.18
with JAE[U,P]	79.44	88.81	83.86	82.78

TABLE VI
THE PERFORMANCES OF DIFFERENT BEHAVIOR PREDICTION MODELS (AVERAGE RESULTS OBTAINED WITH CORPUS 1 AND CORPUS 2).

- The simple auto-encoder model contains three layers: the input layer represents the concatenation of the like and comment embedding vectors, the hidden layer corresponds to the user embedding and the output layer that reconstructs the input.

- The deep-cornet model[4] [17], [18] learns multi-view representation by minimizing the self reconstruction error and the cross reconstruction error and maximizing the correlation between the hidden representation of both views.

Tables I and II display the obtained results of our behavior prediction model with the corpus 1 and the corpus 2, respectively, using the baseline user embedding models and our joint auto-encoder model (JAE: with a single shared hidden layer and DeepJAE: deep version with separate hidden layers). Table III illustrates the average of the obtained results by these two corpora.

From these different Tables (I, II and III), the deep learning based models are noticed to clearly outperform the traditional weighted average. Thus, we remark that the weighted average slightly improves the results compared to the simple average. Indeed, we achieved an F-score of 80.99% with the weighted average and 80.51% with the simple average. These tables also show that the achieved results by the auto-encoder based models are close. The major difference between them is the reconstruction error way which was achieved either using a single input layer or two disjoint layers. In fact, the simple auto-encoder model aims to compute the reconstruction error as the difference between the concatenation of the two user views (given in one input layer) and its reconstructive representation. However, the Deep-cornet model consists in calculating the reconstruction error of each given user view in a separate input layer. Thus, it takes into account the reconstruction error of one view from the other view, which is very useful in the case of learning the image representation from a text representation and vice-versa, as presented in [17], [18]. But, in our case, it does not improve the results and requires additional computing time. Therefore, we may conclude that the best results are obtained by using the joint auto-encoders models (JAE and deepJAE) which separately calculate the reconstruction error of each user's view. Another important finding is that the DeepJAE model with separate hidden layers allows the user views to be encoded with multiple abstraction levels, which improves the results, allowing an F-score of 83.86%.

To the best of our knowledge, there are no studies that analyze and predict Facebook behavior (like and comment) as we have already stated. So, to evaluate the effectiveness of our deep model of user behavior prediction, we compared its results to those obtained using some other related behavior prediction models, namely:

- DeepCoNN model [9]: It was re-implemented in our study to make it flexible with our dataset. It consists of two parallel neural networks coupled in the last Sigmoid layer, one network for users (NetU) and another for posts (NetP). Users likes and comments, and posts descriptions are given to NetU and NetP respectively as inputs, and the corresponding behavior is produced as an output. In the first layer "look-up", users or posts are represented as matrices of word embeddings. The next layers are the common ones used in CNN based models, including the convolution, max pooling, and fully connected layers.

- Concatenate+MLP method consists in concatenating the user and post embedding vectors. The obtained feature vectors are passed into the full connection hidden layers to obtain higher-level representations. Then a Sigmoid layer is used to predict the labels.

In Tables IV, V and VI, we illustrate the experimental results, obtained using the DeepCoNN model and the behavior prediction models with different behavior representation ways starting from the user representation U (learned by DeepJAE model) and post representation P, namely: the concatenation in an MLP model (Concatenation[U, P] + MLP), the JAE model

[4]https://deeplearn.school.blog/2017/05/24/common-representation-learning-using-deep-cornet/

(JAE[U,P]) and the DeepJAE model (DeepJAE[U-P]). Tables IV and V present the obtained results by these models applied on corpus 1 and corpus 2, respectively. Table VI summarizes the overall performance of these models by averaging the results obtained by these two corpora.

From tables IV, V and VI, we can see that the performances of these models are consistent with our dataset. The combination between user and post embeddings using the joint auto-encoders models (JAE or DeepJAE) was observed to provide better results than the standard abstraction of the concatenation of user and post (Concatenate+MLP). Hence, the joint auto-encoders models obviously learn the best behavior embedding able to capture the relationship between user and post. We also notice that, in our prediction case, the joint auto-encoders model "JAE" with a single shared hidden layer is more effective than that with several separated hidden layers "DeepJAE". We obtain an F-score of 83.34% with DeepJAE and 83.86% with JAE. This can be explained by the fact that users and posts are already represented in a more compact and relevant way through several abstraction levels. The learning of the best behavior representation does not require several abstractions.

We remark, through these Tables, that the impact of CoRAE model in our method that captures the context and order of words in a text, compared to the CNN used in DeepCoNN model. Thus, the deepCoNN takes the users "like" and "comment" views in the same input, whereas our method applies the joint auto-encoders that learns the shared representation between these two user views.

V. Conclusion

In the context of this paper, we shed light on the problem of predicting the user behavior towards a Facebook post. We proposed a novel deep neural network that contains two main models; The first is the CoRAE model that allows representing the text by an embedding vector. The second is the joint auto-encoders model (JAE and DeepJAE) which aims at learning a shared distributed representation from two views. Using the DeepJAE model, the user embedding vector was learned from his like and comment views. Thus, the user behavior embedding was represented by fusing the user and post embedding with the JAE model. When tested on the evaluation dataset, the experimental results showed that the proposed model outperformed other previously introduced ones. In future work, we aim to predict whether users will write their comments to express their positive or negative opinions by integrating the notion of sentiment analysis on comments. We also aim to extend our model to be adapted to the newly added reactions such as: love, disgust and anger.

References

[1] H. B. Yedder, U. Zakia, A. Ahmed, and L. Trajkovi, "Modeling prediction in recommender systems using restricted boltzmann machine," in *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2017, pp. 2063–2068.

[2] R. Catherine and W. Cohen, "Transnets: Learning to transform for recommendation," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, New York, NY, USA, 2017, pp. 288–296.

[3] O. Kuchaiev and B. Ginsburg, "Training deep autoencoders for collaborative filtering," *CoRR*, 2017.

[4] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, New York, NY, USA, 2015, pp. 811–820.

[5] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *the 9 ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2016, pp. 153–162.

[6] X. Wang and Y. Wang, "Improving content-based and hybrid music recommendation using deep learning," in *Proceedings of the 22Nd ACM International Conference on Multimedia*, New York, NY, USA, 2014, pp. 627–636.

[7] H. Wang, N. Wang, and D.-Y. Yeung, "Collaborative deep learning for recommender systems," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2015, pp. 1235–1244.

[8] Y. Zhang, M. J. Er, R. Venkatesan, N. Wang, and M. Pratama, "Sentiment classification using comprehensive attention recurrent models," in *International Joint Conference on Neural Networks (IJCNN)*, July 2016, pp. 1562–1569.

[9] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2017, pp. 425–434.

[10] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016, pp. 233–240.

[11] H. Wang, F. Zhang, X. Xie, and M. Guo, "DKN: deep knowledge-aware network for news recommendation," *CoRR*, 2018.

[12] L. Hong, A. S. Doumith, and B. D. Davison, "Co-factorization machines: Modeling user interests and predicting individual decisions in twitter," in *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, New York, NY, USA, 2013, pp. 557–566.

[13] D.-A. Nguyen, S. Tan, R. Ramanathan, and X. Yan, "Analyzing information sharing strategies of users in online social networks," in *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Piscataway, NJ, USA, 2016, pp. 247–254.

[14] C. Wang, Q. Li, L. Wang, and D. D. Zeng, "Incorporating message embedding into co-factor matrix factorization for retweeting prediction," *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 1265–1272, 2017.

[15] N. Srivastava and R. Salakhutdinov, "Multimodal learning with deep boltzmann machines," *Journal of Machine Learning Research*, vol. 15, pp. 2949–2980, 2014.

[16] L. Pang and C.-W. Ngo, "Mutlimodal learning with deep boltzmann machine for emotion prediction in user generated videos," in *the 5th ACM on International Conference on Multimedia Retrieval*, New York, NY, USA, 2015, pp. 619–622.

[17] S. Chandar, M. M. Khapra, H. Larochelle, and B. Ravindran, "Correlational neural networks," *Neural Computation*, vol. 28, pp. 257–285, 2016.

[18] T. Ding, W. K. Bickel, and S. Pan, "Multi-view unsupervised user feature embedding for social media-based substance use prediction," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark, 2017, pp. 2275–2284.

[19] V. Vukotić, C. Raymond, and G. Gravier, "Bidirectional joint representation learning with symmetrical deep neural networks for multimodal and crossmodal applications," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, New York, NY, USA, 2016, pp. 343–346.

[20] C. Silberer and M. Lapata, "Learning grounded meaning representations with autoencoders," in *The Association for Computer Linguistics*, 2014, pp. 721–732.

[21] H. Ameur, S. Jamoussi, and A. B. Hamadou, "A new method for sentiment analysis using contextual auto-encoders," *Journal of Computer Science and Technology*, vol. 33, no. 6, pp. 1307–1319, 2018.

[22] ——, "A new emotional vector representation for sentiment analysis," in *Computational Linguistics and Intelligent Text Processing - 17th International Conference*, 2016, pp. 258–269.