Exploring Contextualized Tag-based Embeddings for Neural Collaborative Filtering

Tahar-Rafik Boudiba and Taoufiq Dkaki
IRIT, UMR 5505 CNRS, 118 Route de Narbonne, F-31062 Toulouse Cedex 9, France

Keywords: Folksonomies, Deep Learning, Tag-based Embedding, Social Tagging, Recommendation

Abstract: Neural collaborative filtering approaches are mainly based on learning user-item interactions. Since in collaborative systems, there are several contents surrounding users and items, essentially user reviews or user tags these personal contents are valuable information that can be leveraged with collaborative filtering approaches. In this context, we address the problem of integrating such content into a neural collaborative filtering model for rating prediction. Such content often represented using the bag of words paradigm is subject to ambiguity. Recent approaches suggest the use of deep neuronal architectures as they attempt to learn semantic and contextual word representations. In this paper, we extended several neural collaborative filtering models for rating prediction that were initially intended to learn user-item interaction by adding textual content. We describe an empirical study that evaluates the impact of using static or contextualized word embeddings with a neural collaborative filtering strategy. The presented models use dense tag-based user and item representations extracted from pre-trained static Word2vec and contextual BERT. The Models were adapted using MLP and Autoencoder architecture and evaluated on several MovieLens datasets. The results showed good improvements when integrating contextual tag embeddings into such neural collaborative filtering architectures.

1 INTRODUCTION

Effective recommendation systems require the ability to predict how a user would rate a given item. To this end, several classes of recommendation approaches are commonly used, Content-Based Filtering (CBF), Collaborative Filtering (CF), and hybrid approaches (Zhang et al., 2019). Classical CF approaches are based either on Matrix Factorization (MF) techniques or on simple user-item vector similarity methods. However, these models have the common property of being essentially linear models, i.e they linearly combine user and item latent factors. Since deep learning approaches have shown potential in representation learning, they improved the performance of traditional recommender systems (RS) by enabling neural models to include content information from various sources. New neural collaborative filtering approaches capture more complex user-item interactions and achieve a high abstract level of content description. The use of content information is a way to advance understanding of items and users in order to provide a better recommendation (Zhang et al., 2019). Such content is often presented as textual data such as *user's reviews* or *user's tags* and it is commonly used to describe items and users' profiles using the bag of words representation. Although such representations commonly known as one-hot vectors are efficient for computing user-item similarity, many problems such as ambiguity and vocabulary mismatch have been raised (Hassan et al., 2018). To overcome such issues, word embedding representations (Rücklé et al., 2018) have been proposed to better capture semantic. Some approaches make use of semantic vectors extracted from pre-trained Word2vec neural language model instead (Nguyen et al., 2020). Such models achieve quality recommendations with their ability to extract high-level representations from raw data. Precisely, much attention was paid in order to establish which kind of user and item dense representations to use at the top layer of the neural network. Works have discussed text-based aggregation techniques (Wu et al., 2019), some others suggest the concatenation of mean Word Embedding (Rücklé et al., 2018) since they compute word average embedding representations.

In this paper, we have considered tag embeddings as semantically rich tag-based representations and we exploit them into effective CF models for rating prediction. We have built several hybrid models mixing

Boudiba, T. and Dkaki, T.
Exploring Contextualized Tag-based Embeddings for Neural Collaborative Filtering.
DOI: 10.5220/0010793300003116
In *Proceedings of the 14th International Conference on Agents and Artificial Intelligence (ICAART 2022) - Volume 3*, pages 158-166
ISBN: 978-989-758-547-0; ISSN: 2184-433X

neural CF with tagging information integrated into a training process. For this purpose, we handled word vector representation to include more valuable tag' semantic and so to enhance existing neural CF models. We compared different tag embedding representations from pre-trained static (Word2vec) and contextual BERT models. Furthermore, We evaluated the impact of using tag embeddings through two neuronal model's architecture (MLP and autoencoder). We provided empirical results from MovieLens Dataset (*10 M, 20M et 25M*).

The main contributions of this paper are summarized as follows:

- Integrate efficiently tag-embedding representations into neural CF models.
- Evaluate the impact of static and contextual embedding representations and comparing model architecture.
- Extensive series of experiments on real data from several MovieLens data sets.

The remaining of the paper is organized as follows. The next section reviews recent research works that are related to tag-based personalized recommendation using neuronal networks and word vector representation. We gathered works that describe neural approaches from a collaborative filtering point of view, specifying the most used neural architectures. Section 2 presents some background. Section 3 highlights the basis of our proposed models. Section 4 details datasets, evaluation metrics, and experimental settings. Section 5 gives the evaluation results and discusses performance comparison with baselines. Section 6 is the final section of this paper.

2 BACKGROUND

Using multi-layer perceptron (MLP) for feature representation is highly efficient, even though it might not be as expressive as autoencoders (Zhang et al., 2019). MLP-based RS considers user and item representations in the context of CF as input features or as sparse vectors using one-hot encoding (He et al., 2017). The embedding layer is a fully connected layer that projects a sparse representation to a dense vector. The obtained user (item) embedding is the latent vector for a user (item). User and item embeddings are then jointly fed into neural CF layers to map the latent vectors with prediction scores. Autoencoder-based recommender systems are on the other hand used either to learn lower-dimensional feature representations at the bottleneck layer (Ouyang et al., 2014) or to fill the blanks of an interaction matrix in the reconstruction layer (Sedhain et al., 2015). Autoencoder-Based CF is a meaningful illustration of this method. It consists of interpreting the CF paradigm from the autoencoder perspective. Since deep learning networks have made breakthroughs in data representation learning from various data sources, effective content-aware hybrid neural recommendation models have been able to handle learning representations of user preferences, item features and textual interactions (Liu et al., 2020; Hassan et al., 2018). Yet, neural recommendation models attempt to introduce in addition, tag semantics-aware representations based on distributional semantic used as features (Liu et al., 2020). Musto et al., (Musto et al., 2015) exploit Word2vec approach to learn a low dimensional vector space word representation and exploited it to represent both items and user profiles in a recommendation scenario. Zhang et al., (Zhang et al., 2016) proposed to integrate traditional matrix factorization with Word2vec for user profiling and rating prediction. Liang et al., (Liang et al., 2018) exploited pre-trained word embeddings from *Word2vec* to represent user tags and construct item and user profiles based on the items' tags set and users' tagging behaviors. They use deep neural networks (DNNs) and recurrent neural networks (RNNs) to extract the latent features of items and users to predict ratings. Moreover, TagEmbedSVD (Vijaikumar et al., 2019) uses pre-trained word embedding from *Word2vec* for tags to enhance personalized recommendations that are integrated to an SVD model in the context of cross-domain CF. Other works (Hassan et al., 2018) take advantage of network embedding techniques to propose embedding-based recommendation models that exploit CF approaches. Exploiting rating patterns often require the use of a neural network-based embedding model that is first pre-trained. Features are extracted and integrated into a CF model by fusing those features with latent factors thanks to non-linear transformations that better leverage abstract content representations and so perform higher quality recommendations. Since pre-training word embedding from large-scale corpus became widely used in different information retrieval tasks, it was also exploited to generate recommendations by ranking user-item matrix from users' similar tags vocabulary. Models such as Word2vec (Le and Mikolov, 2014) or GloVe (Pennington et al., 2014) for instance learned meaningful user tag representations by modeling tag co-occurrences. However, these methods don't consider the deep contextual information that some single word tags may suffer. Moreover, they do not handle unknown tags. In contrast, contextualized word rep-

resentations such as BERT (Devlin et al., 2018), have been proposed to overcome the lack of static word embeddings, since it was shown that such contextual neural language model improves the performance of many downstream tasks.

2.1 Related Work

In the following, we introduce some recommendation models of the literature that have handled neural CF approaches (Ouyang et al., 2014; Sedhain et al., 2015; Chen et al., 2019). Those models resolved user rating prediction. Some of them have been adapted to include tagging content (Zheng et al., 2017), they are mostly composite through which multiple neural building modules compose a single distinguishable function that is trained end-to-end. Here, we introduce some summary definitions related to tagging that will allow us to address later most common architectures and topologies giving recommendation strategies for each of them. A folksonomy F can be defined as a 4-tuple $F = (U,T,I,A)$, where U is the set of users annotating the set of items I, $U = \{u_1,u_2,...u_M\}$ where each u_i is a user. T is the set of tags that includes the vocabulary expressed by the folksonomy. I is the set of tagged items by user $I = \{i_1,i_2...i_N\}$. $A = \{u_m,t_k,i_j\} \in U \times T \times I$ is the set of annotations of each tag t_k to an item i_j by user u_m. We have also considered R as the set of user ratings $r_{u,i}$.

2.1.1 MLP-based Neural CF for Recommendation

Approaches of neural collaborative filtering (NCF) for rating prediction often involves dealing with binary property of implicit data. Some works (He et al., 2017; Chen et al., 2019) have in addition discussed the choice of the neural architecture to be implemented. A possible instance of the neural CF approach can be formulated using a multi-layer perceptron (MLP). As addressed in (He et al., 2017) the input layer (the embedding layer) is a fully connected layer that maps the sparse representations to dense feature vectors. It consists of two feature vectors $v^f_{(u)}$ and $v^f_{(i)}$ that describe user $v^U_{(u)}$ and item $v^I_{(i)}$ represented initially through one-hot encoding. The obtained user (item) embedding can be seen as the latent vector for user (item). The user embedding and item embedding are then fed into neural CF layers to map the latent vectors to prediction scores. Final output layer is the predicted score $\hat{r}_{u,i}$, and training is performed by minimizing the point wise loss between $\hat{r}_{u,i}$ and its target value $r_{u,i}$. *NCF* predictive model can be formulated as:

$$\hat{r}_{u,i} = \mathbf{MLP}(P_u^T . v^f_{(u)}, Q_i^T . v^f_{(i)} | P_u, Q_i, \Gamma_{\mathbf{MLP}}) \quad (1)$$

$P_u \in \mathbb{R}^{M \times K}$ and $Q_i \in \mathbb{R}^{N \times K}$ are latent factor matrix for users and items respectively. Γ_{MLP} denotes the model parameters of the interaction function that is defined as a multi-layer neural network.

2.1.2 Autoencoder-based CF for Recommendation

Another way to consider neural CF is to approach user-item rating as a matrix $X \in R^{m \times n}$ with partially observable row vectors that form a user $u \in$ the set of users $U = \{1...m\}$ given by the set of user ratings $r_{(u)} = \{X_{u1}...X_{um}\} \in R$ and column vectors from the set of items $i \in I = \{1...n\}$ also given by their corresponding ratings $r_{(i)} = \{X_{i1}...X_{in}\}$. An efficient neural method to encode each partially observed vector into law-dimensional latent space is to handle an autoencoder architecture as suggested in (Sedhain et al., 2015) that will reconstruct the output space to predict missing ratings for recommendation (Sedhain et al., 2015; Huang et al., 2020; Ouyang et al., 2014). Given a set of rating vectors $r_{(u)}$ and $r_{(i)} \in \mathbb{R}^d$, the autoencoder solves:

$$min_\theta \sum_{r \in R} = ||r - h(r;\theta)||^2 \quad (2)$$

Where $h(r;\theta)$ is the reconstruction of input $r \in \mathbb{R}^d$ that is defined as:

$$h(r;\theta) = f(W.g(Vr + \mu) + b) \quad (3)$$

$f(.)$ and $g(.)$ are activation functions associated to the encoder and decoder respectively and θ gather model parameters; $W \in \mathbb{R}^{d \times k}$ and $V \in \mathbb{R}^{k \times d}$ are weight matrices and $\mu \in \mathbb{R}^k$, $b \in \mathbb{R}^N$ biases. In an item-based recommendation perspective, the autoencoder applies $r_{(i)}$ as the set of input vectors. Weights associated to those vectors are updating during back-propagation.

3 OVERVIEW OF THE PROPOSED MODELS

In this section, we introduce our tag-aware neural models for recommendation. More explicitly, we integrate tag-based embeddings into two CF neural architectures, namely a Multilayer perceptron and an autoencoder. More explicitly, to integrate side information into predictive neural models a naive approach consists of appending additional user/item

bias to the rating prediction. We estimate that computing those biases can be handled either by handcrafted engineering or by implementing an appropriate CF strategy. A simple Neural collaborating filtering framework architecture implies considering the input layer(embedding layer) as a fully connected layer that projects sparse representation of users and items to dense vectors. To integrate explicitly tags vocabulary in a neural model for rating prediction, we have made use of feature vectors that we have considered as tag vector representations sharing a common embedding space using projection matrices. The obtained user (item) embedding can be seen as the latent vector for the user (item) in the tag latent space. Feature vectors $v^{f}_{(u)}$ and $v^{f}_{(i)}$ are reconsidered since we have projected tag representations into lower dimension using projected matrices **E** and **F**. Consequently, tag-based vector representation is expressed as a user feature vector $v^{\tilde{f}}_{(\tilde{u})}$:

$$v^{\tilde{f}}_{(\tilde{u})} = \frac{1}{|T_u|} \sum_{t_k \in T_u} E(t_k) \tag{4}$$

Such as $t_k \in \mathbb{R}^c$ is the embedding vector associated with tag k, and c denotes the embedding dimension. E denotes the projection matrix with $E \in \mathbb{R}^{d \times c}$.

Similarly, if F denotes the projection matrix with $F \in \mathbb{R}^{d \times c}$, then the item feature vector $v^{\tilde{f}}_{(\tilde{i})}$ is expressed as:

$$v^{\tilde{f}}_{(\tilde{i})} = \frac{1}{|T_i|} \sum_{t_k \in T_i} F(t_k) \tag{5}$$

We denoted T_u the set of tags of a user u and T_i as the set of related tags describing a particular item. Moreover, we have obtained embeddings for tags from *Word2vec* and *BERT* pre-trained neural model by handling projection matrices **E** and $\mathbf{F} \in \mathbb{R}^{d \times c}$.

3.1 CF-based MLP Model

Extended tag-based NCF predictive model can be reformulated relying on the previous NCF model that has been described in section 3.1 equation (1) as:

$$\hat{r}_{u,i} = \mathbf{MLP}(v^{\tilde{f}}_{(\tilde{u})}, v^{\tilde{f}}_{(\tilde{i})}, \theta_{\mathbf{MLP}}) \tag{6}$$

The user and item embeddings can be fed into a multi-layer neural model. Where, $\hat{r}(u,i)$ is the rating score for a user on an item. **Figure** 1 details an instance of the model. Prediction Pipeline exploits user and item vectors extracted from dense space representation (***Figure*** $1_{(\mathcal{A})}$), hidden layers are added to learn interactions between user and item latent features, a regressor at the last hidden layer is set to produce the final rating. (***Figure*** $1_{(\mathcal{C})}$) is a static module in which dense representations are computed through inner product of user and items embedding' representations. Tag embedding representations are extracted from neural pre-trained language model (***Figure*** $1_{(\mathcal{E})}$).

3.2 CF-based Autoencoder Model

Following the autoencoder paradigm, instead of encoding user vectors containing user ratings to be predicted like in Autorec (Sedhain et al., 2015), we have extended a multilayered autoencoder architecture to integrate element wise product of pre-trained tag-based embeddings. Such embeddings are concatenated with the user rating representations and are projected on a dimensional latent (hidden) space. As such, user' rating $r(u_m, i_l)$ of a particular user is reconstructed using an objective function θ that minimizes:

$$\sum ||r(u_m,i_l) \oplus (v^{\tilde{f}}_{(\tilde{u})} \otimes v^{\tilde{f}}_{(\tilde{i})}) - h(r(u_m,i_l) \oplus (v^{\tilde{f}}_{(\tilde{u})} \otimes v^{\tilde{f}}_{(\tilde{i})}); \theta)||^2 \tag{7}$$

Where $(r(u_m,i_l),\theta)$ is the reconstruction of the input $r(u_m,i_l) \in \mathbb{R}^d$. The operator $\otimes$ denotes element-wise multiplication between user and item feature vectors. The operator $\oplus$ denotes a concatenation operator. *tanh* is the selected activation function. **Figure** $1_{(\mathcal{B})}$ presents a detailed instance of the model. Prediction Pipeline exploits user and item vectors extracted from dense space representation. Such representations are concatenated with user rating and fed as input of the autoencoder model. Layers are added to learn interactions between user and item latent features to be compressed in a dense space. User's ratings reconstruction from the dense space produce the final rating.

4 EXPERIMENTS

In this section, we have conducted experiments intending to answer the following research questions:
RQ1: Are tag-based contextual embeddings efficient representations to be used in a neural CF model compared to static tag-based embedding representations?
RQ2: Which extended neural collaborative architecture perform significant improvement and ranking quality for a rating prediction task?

4.1 Experimental Settings

1. **Datasets:** The data sets describe 5-stars ratings and free-text tagging from *MovieLens*, a movie

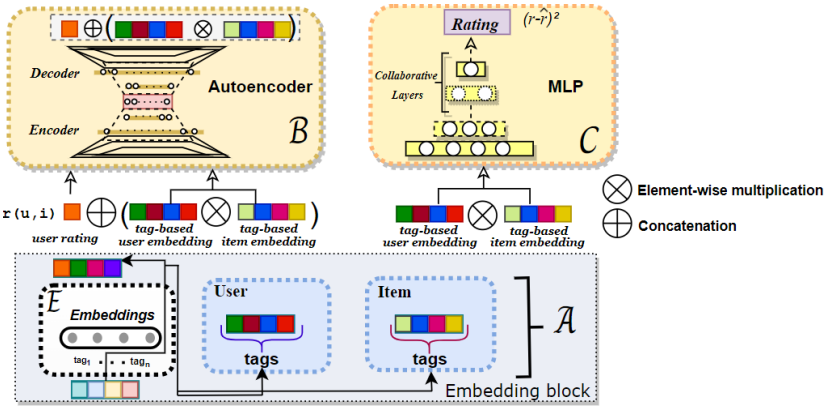

Figure 1: Extended NCF based on an MLP (on the right) and an Autoencoder (on the left).

recommendation service. We extracted user annotations from the ML-10M, ML-20M, and ML-25M data sets. Only users that have annotated and rated at least 20 movies were selected. We observed from **Table** 1 an unequal distribution of user rating classes, because of users trend scoring items with good rating values. This can lose models capacity to generalize. To overcome, we oversample minority classes (Chawla et al., 2002) by duplicating samples from the minority class and adding them to the training data.

2. **Hyper-parameters:**
After splitting the data in each dataset into random 90%, 10% training and testing sets, we hold 10% of the training set for hyper-parameters tuning. Then, we conducted 5 cross-fold validation strategy in each dataset and averaged RMSE measure. We have applied a grid search for hyper-parameters tuning such as the learning rate that we tuned among values $\in \{0.0001, 0.0005, 0.001, 0.005\}$, latent dimensions $\in \{100, 200, 300, 400, 500, 1000\}$ for both autoencoder and MLP architecture. We handled the Neural Collaborative Autoencoder with a default rating of 2.5 for testing set without training observations. The models were optimized thanks to the well known Adam optimizer.
3. **Evaluation Metrics:** We have evaluated rating prediction using two metrics: Mean Absolute Error (MAE) and Root Mean Square Error(RMSE). Both of them are widely used for rating prediction in recommended systems. Given a predicted rating $\hat{r}_{u,i}$ and a ground-truth rating $r_{u,i}$ from the user u for item i, the RMSE is computed as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(r_{u,i} - \hat{r}_{u,i})^2} \quad (8)$$

Where N indicates the number of ratings between users and items.

MAE is computed as follows:

$$MAE = \frac{1}{N}\sum_{u,i}|r_{u,i} - \hat{r}_{u,i}| \quad (9)$$

Indeed, we have also evaluated ranking accuracy using NDCG (Normalized Discounted Cumulative Gain (Järvelin and Kekäläinen, 2002)) at 10. For this purpose, we assumed rating values at 5 as being a good appreciation of a user regarding a movie. In contrast, rating values under 3 are considered as bad. Hence, the rating value of each movie is used as a gained value for its ranked position in the result. The gain is summed from the ranked position from 1 to n. To compute *NDCG*, relevance scores are set to six(5) points scale from 1 to 5 and denotes the relevance score from low to strong relevance. We set the Ideal DCG for user movies ranked in decreasing order of their ratings. NDCG values presented further are averaged over user testing set.

4.2 Tag-based Embedding Representations

We have considered tag-based embeddings thanks to word vector representations. We have extracted such tag-based embedding representations from pre-trained neural language models. Owing to the users' writing discrepancy, users' tags semantic meaning is often ambiguous. Tags can be composed of several words and may contain subjective expressions. They can also be unique words which can occasionally lead to a lack of context. That makes it difficult to integrate tags explicitly in an effective neural CF architecture. Our main objective is to map users, items and their tags' interaction in the same latent space. Rather to

exploit straightly dimensional latent space representations of users and items like in most neural collaborative approaches (He et al., 2017; He et al., 2018), we propose to project first both users' and items' representations into a dense tag space representation. Both previous neural approaches are somehow representative of our objective since they are from CF. We assume that users and items are represented by their corresponding tags. Particularly, they are represented from the aggregate average of their tag embedding representations.

4.2.1 Static *Word2vec* Tag-based Embeddings

We have handled static tag-based embedding vectors from Word2vec. We have exploited pre-trained vectors trained on part of Google News dataset (about 100 billion words) and have extracted user's tags embedding by associating them to a vector of a well known fixed size for each tag. However, we found that some tags were out of tag vocabulary, since those user tags represent respectively 8%, 5%, 5% of our Movielens Datasets 10M, 20M and 25 millions ratings. We fixed this issue by initiate those samples with random vector values. The inability to handle unknown or out-of-vocabulary words is one of limitation encountered when using such pre-trained model. Finally, each set of tags per user is represented through a multidimensional vector of $dim = 300$.

4.2.2 Contextualized *BERT* Tag-based Embeddings

We have addressed extracting contextualised embeddings from *BERT* neural language model. For this purpose, we have assumed that the fist token which is '[CLS]' that captures the context is treated as sentence embeddings (Reimers and Gurevych, 2019). The word embedding sequence corresponding to each set of tags is entered into the pre-trained model. We have then handled the activation from the last layers of BERT model since the features associated with the activation in these layers are far more complex and include more contextual information. These contextual embeddings are used as input to our proposed models. Thus, each set of tags per user is represented through a multidimensional embedding vector of $dim = 768$. We have implemented the pre-trained bert-base model[1] (12 blocks of hidden dimension 768, 12 heads for attention) and defining the '[CLS]' which indicates the beginning of a sequence as well as the '[SEP]' that we used as a separation between two tags of a same sequence.

[1]BERT was pre-trained on a corpus composed of 11,038 unpublished books belonging to 16 different domains and 2,500 million words from English Wikipedia text passages

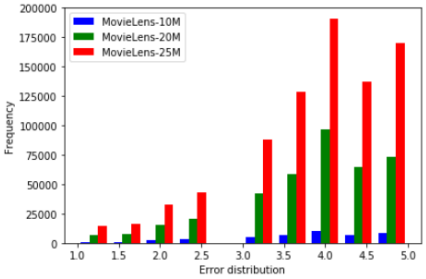

Figure 2: Frequency of user ratings per tagged item for each data set.

Table 1: Statistics of 10M, 20M and 25M extracted from MovieLens.

MovieLens	10-M	20-M	25-M
Users	71567	138000	162541
Movies	10681	27000	62423
Tags	95580	465000	1093360
Ratings	10000054	2000000	25000095

5 EVALUATION AND PERFORMANCE COMPARISON

To solve the **RQ1**, we extended neural models (He et al., 2017; Sedhain et al., 2015) by handling static and contextual tag-based embedding representations. We compared those models with recent neural models from CF that we set as baselines. We evaluated rating score accuracy using RMSE (Root Mean Square Error) and MAE (Mean Absolut Error). To address **RQ2**, we have implemented an MLP and an autoencoder-based CF architecture then, we compared the performance of each neural model according to tag-based embedding representations with which such models were integrated. Moreover, ranking accuracy metric was carried out among the different neural models using NDCG (Normalized Discount Cumulative Gain) at 10. We detailed models that have been compared hereafter:

- **Neural CF-MLP**(He et al., 2017)**:** Is a neural CF approach that exploits a multi-layer perceptron (MLP) to learn the user–item interaction function. The bottom input layer consists of two vectors that describe user u and item i in a binarized sparse vector (one-hot encoding), such model employ only the identity of a user and an item as input feature.

Table 2: A synthesis of RMSE and MAE values for each model including $ndcg@10$ scores.

Models	Evaluation measures								
	ML-10M			ML-20M			ML-25M		
	MAE	RMSE	ndcg@10	MAE	RMSE	ndcg@10	MAE	RMSE	ndcg@10
Neural CF-MLP$^{++}_{W2v}$	0.77	0.98	0.43	0.88	0.96	0.381	0.84	1.01	0.42
Neural CF-MLP$^{++}_{Bert}$	**0.72**	**0.93**	**0.46**	**0.791**	**0.86**	0.42	**0.791**	**0.83**	**0.46**
CF-Autoencoder$^{++}_{W2v}$	0.83	1.1	0.411	0.85	0.97	0.39	0.80	1.02	0.42
CF-Autoencoder$^{++}_{Bert}$	0.76	0.96	0.42	0.811	0.89	**0.44**	**0.798**	0.865	0.445
U-Autorec (Sedhain et al., 2015)	0.82	1.09	0.38	0.84	1.07	0.37	0.81	1.01	0.40
Neural CF-MLP(He et al., 2017)	0.73	0.98	0.44	0.89	1.025	0.39	0.87	0.92	0.43

- **Neural CF-MLP^{++}:** Is an extension of Neural CF-MLPn, the model integrates in the bottom input layer two feature vectors that are described as tag embedding features of users and items. These features are extracted from word vector representation. User and item feature vectors are extracted from tag-based embeddings, with 300-dimensional word vectors from pre-trained Word2vec model Neural CF-MLP$^{++}{}_{Word2vec}$ and Neural CF-MLP$^{++}{}_{BERT}$ that exploits a 768-dimensional word vectors from pre-trained BERT model.
- **U-Autorec** (Sedhain et al., 2015)**:** U-AutoRec is a neural CF framework for rating prediction that exploits an autoencoder architecture. It takes user vectors as input and reconstructs them in the output layer. The values in the reconstructed vectors are the predicted value of the corresponding position.
- **CF-Autoencoder^{++}:** Our autoencoder-based neural collaborative approach that integrates as input tag embedding features by performing element-wise multiplication on their word vector representations and do concatenate such representations with user/item rating vectors to get the reconstructed ratings. We have termed the autoencoder-based model using static tag vector representations as CF-Autoencoder $^{++}{}_{Word2vec}$ meanwhile CF-Autoencoder$^{++}{}_{BERT}$ stands for autoencoder-based model using contextual tag vectors.

Results of our experiments are synthesized in **Table** 2. Firstly, as regards to ML-10M dataset, top RMSE and MAE scores are valued from Neural CF-MLP$^{++}_{Bert}$ model with $RMSE = 0.93$ and $MAE = 0.72$. Our proposed model has achieved good quality ranking to reach $NDCG@10 = 0.46$. We have also noticed that the static tag-based embedding extension of this model CF-MLP$^{++}_{W2v}$ has achieved well results outperforming the U-Autorec baseline (Sedhain et al., 2015) with $MAE = 0.77$, $RMSE = 0.98$ and has reached good quality ranking with $NDCG@10 = 0.43$. However, the baseline model Neural CF-MLP (He et al., 2017) has not been fully achieved. Secondly, in ML-20M dataset, the same model Neural CF-MLP$^{++}_{Bert}$ has shown top RMSE and MAE score with $MAE = 0.79$ and $RMSE = 0.86$. We observed that neural static tag-based embedding models perform acceptable results compared with baselines. However, top quality ranking has been achieved from the Autoencoder-based model extension CF-Autoencoder$++_{Bert}$ with $NDCG@10 = 0.44$ since this model performed well with $MAE = 0.811$ and $RMSE = 089$. Finally, in ML-25M dataset, impact of tag-based embeddings is established since RMSE score has shown significant improvement compared to baselines to reach $RMSE = 0.83$ for Neural CF-MLP$^{++}_{Bert}$ and quality ranking to $ndcg@10 = 0.46$. Moreover, we have also confirmed the effectiveness of such embeddings for CF-Autoencoder$++_{Bert}$ model that has performed $MAE = 0.79$ and $RMSE = 0.86$. We argue that these results can further be improved by increasing the volume of training data from tagging.

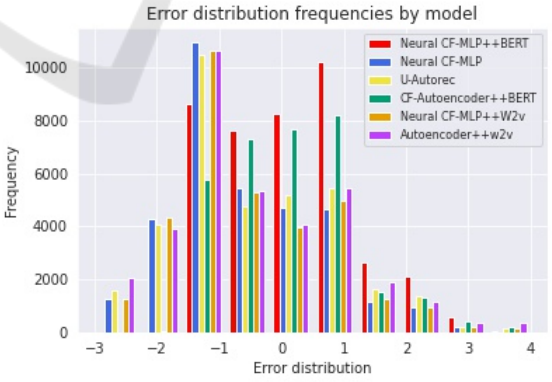

Figure 3: Overview of error distribution by frequency for each CF neural model.

The impact of exploiting contextualized tag-based embedding representations is also perceptible through studying error distribution when predicting user ratings. Such impact is summarized in **Figure** 3. Error distribution values have been gathered among test-

ing sets from the whole collection, including the data sets *ML-10M, ML-20M and ML-25M*. This is to propose an overall overview of the error distributions resulted from baselines compared with models that integrate tag-based static or contextualized embedding representations. We observe that global error distribution values from the models CF-MLP$^{++}_{Bert}$ and CF-Autoencoder$^{++}_{Bert}$ are most located in the interval $\in [-1,1]$ compared to the error distribution values of either the baselines or the models that integrate static-based embedding representations. As instance, the model CF-MLP$^{++}_{Bert}$ obtains a number of 8150 accurate predictions, which outperform Neural CF-MLP (He et al., 2017) with 4500 accurate predictions or U-Autorec (Sedhain et al., 2015) with less then 5000 accurate predictions. Moreover, CF-MLP$^{++}_{Bert}$ and CF-Autoencoder$^{++}_{Bert}$ exceed the performance of the models integrating static embedding representations with less then 4000 accurate predictions for both CF-MLP$^{++}_{W2V}$ and CF-Autoencoder$^{++}_{W2V}$ models. From these results, we confirm the effectiveness of tag-based embedding representations to be used with a neural CF approach for a rating prediction task.

6 CONCLUSION

Learning representations for recommendation represent a major challenge for the application of artificial intelligence along with its efficient role in recommendation applied to the massive amount of data extracted from users' folksonomies. Following the experiments, we come to the conclusion that neural networks models exploiting folksonomies using neural CF approaches, in particular, those using contextual tag-based embedding as user-item features provide more accurate models to rating prediction tasks. Moreover, we argued that exploiting contextual tag-based embeddings as features remains an effective way to include tags semantics into a recommendation process. We demonstrate that such embeddings can lead to quality models in the context of predicting user ratings. Tag-based embedding aggregation has been a promising way of improvement since, in the natural language embedding process, extracted representations can be described as semantic relationships. An interesting direction for forthcoming works is to investigate tag-based neighbor embedding representations into a neural collaborative filtering model for recommendation purposes.

REFERENCES

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority oversampling technique. *Journal of artificial intelligence research*, 16:321–357.

Chen, W., Cai, F., Chen, H., and Rijke, M. D. (2019). Joint neural collaborative filtering for recommender systems. *ACM Transactions on Information Systems (TOIS)*, 37(4):1–30.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Hassan, H. A. M., Sansonetti, G., Gasparetti, F., and Micarelli, A. (2018). Semantic-based tag recommendation in scientific bookmarking systems. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pages 465–469.

He, X., Du, X., Wang, X., Tian, F., Tang, J., and Chua, T.-S. (2018). Outer product-based neural collaborative filtering. *arXiv preprint arXiv:1808.03912*.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. (2017). Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Huang, R., Wang, N., Han, C., Yu, F., and Cui, L. (2020). Tnam: A tag-aware neural attention model for top-n recommendation. *Neurocomputing*, 385:1–12.

Järvelin, K. and Kekäläinen, J. (2002). Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196.

Liang, N., Zheng, H.-T., Chen, J.-Y., Sangaiah, A. K., and Zhao, C.-Z. (2018). Trsdl: Tag-aware recommender system based on deep learning–intelligent computing systems. *Applied Sciences*, 8(5):799.

Liu, H., Wang, Y., Peng, Q., Wu, F., Gan, L., Pan, L., and Jiao, P. (2020). Hybrid neural recommendation with joint deep representation learning of ratings and reviews. *Neurocomputing*, 374:77–85.

Musto, C., Semeraro, G., De Gemmis, M., and Lops, P. (2015). Word embedding techniques for content-based recommender systems: An empirical evaluation. In *Recsys posters*.

Nguyen, L. V., Nguyen, T.-H., and Jung, J. J. (2020). Content-based collaborative filtering using word embedding: a case study on movie recommendation. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pages 96–100.

Ouyang, Y., Liu, W., Rong, W., and Xiong, Z. (2014). Autoencoder-based collaborative filtering. In *International Conference on Neural Information Processing*, pages 284–291. Springer.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical*

methods in natural language processing (EMNLP), pages 1532–1543.

Reimers, N. and Gurevych, I. (2019). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Rücklé, A., Eger, S., Peyrard, M., and Gurevych, I. (2018). Concatenated power mean word embeddings as universal cross-lingual sentence representations. *arXiv preprint arXiv:1803.01400*.

Sedhain, S., Menon, A. K., Sanner, S., and Xie, L. (2015). Autorec: Autoencoders meet collaborative filtering. In *Proceedings of the 24th international conference on World Wide Web*, pages 111–112.

Vijaikumar, M., Shevade, S., and Murty, M. N. (2019). Tagembedsvd: Leveraging tag embeddings for cross-domain collaborative filtering. In *International Conference on Pattern Recognition and Machine Intelligence*, pages 240–248. Springer.

Wu, L., Quan, C., Li, C., Wang, Q., Zheng, B., and Luo, X. (2019). A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems (TOIS)*, 37(2):1–29.

Zhang, S., Yao, L., Sun, A., and Tay, Y. (2019). Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38.

Zhang, W., Yuan, Q., Han, J., and Wang, J. (2016). Collaborative multi-level embedding learning from reviews for rating prediction. In *IJCAI*, volume 16, pages 2986–2992.

Zheng, L., Noroozi, V., and Yu, P. S. (2017). 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*, pages 425–434.