I.J. Information Technology and Computer Science, **2019, 12, 1-18**
Published Online December 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijitcs.2019.12.01

ComPer: A Comprehensive Performance Evaluation Method for Recommender Systems

Alaa Alslaity
University of Ottawa, Ottawa, K1N 6N5, Canada
E-mail: aalsl005@uottawa.ca

Thomas Tran
University of Ottawa, Ottawa, K1N 6N5, Canada
E-mail: ttran@eecs.uottawa.ca

Received: 10 June 2019; Accepted: 25 October 2019; Published: 08 December 2019

Abstract—Recommender Systems are receiving substantial attention in several application areas (such as healthcare systems and e-commerce), where each area has different requirements. These systems are multifaceted by nature. So, many metrics, which are sometimes contradictious, are introduced to assess different aspects. The existence of several alternatives and dimensions to recommendation approaches complicate the evaluation of recommender systems. In such a situation, it is desirable to evaluate and compare recommenders in a united way that assesses the multifaceted aspects of these systems fairly and uniformly. Despite the abundance of evaluation dimensions, the literature still lacks an evaluation method that evaluates the multiple properties of these systems, all at once. As a potential solution, this paper proposes an evaluation methodology that provides a multidimensional assessment of recommender systems. The proposed method, which we call *ComPer*, combines the most common evaluation dimensions into a single, yet, general evaluation metric. *ComPer* is inspired by the idea that a recommender system mimics human beings; hence, it can be seen as a human and its outputs can be assessed as human's outputs. Up to our knowledge, this is the first evaluation approach that deals with recommenders as humans. *ComPer* aims to be thorough (by combining multiple dimensions), simple (by presenting the final result as a single value), and independent (by providing setting-independent results). The applicability of the proposed methodology is evaluated empirically using three different datasets. The initial results are promising in the sense that *ComPer* is able to give comparable results regardless of the experimental settings.

Index Terms—Recommender Systems, Recommendation Evaluation, Experiments Replication, performance, unified evaluation.

I. Introduction

Evaluating Recommender Systems (RSs) is a nontrivial task due to several reasons. First, different types of RSs use different algorithms, and hence, have different inputs and outputs. Second, recommendation results are of different types (such as rating prediction and a recommendation list); based on the nature of the results, the same metric can be evaluated differently. Finally, due to the multifaceted nature of RSs, a variety of evaluation metrics (or dimensions) have been introduced to the field.

In the beginning, it was common to evaluate RSs using only one evaluation dimension (the most commonly evaluated dimension was the recommendation *Correctness* or *Accuracy*). Later, there became an increasing consensus that a single metric is insufficient to evaluate the actual performance/effectiveness of recommenders [1]. For instance, recommending an item with the highest ratings is not necessarily always a useful recommendation because the user might already see this item. Hence, there were several calls to consider other dimensions. To respond to this requirement, researchers kept introducing new dimensions actively, which results in a wide variety of dimensions being introduced in the literature.

Having multiple evaluation dimensions is a double-edged saw. On the one hand, it enriches the evaluation process of RSs; but on the other hand, the vast variety of dimensions makes it challenging to select an appropriate dimension. The existence of multiple evaluation options requires significant time and effort to design a proper experiment [2]. Besides, the abundant of evaluation options makes it easy to find an evaluation design that suits an algorithm but ineligible for others, which represents an increasing barrier to fairly compare different studies [3]; For example, good *Scalability* means that if the system scales up to the point where there exist a huge number of items to be recommended, it can make recommendations within a reasonable time; to increase the system's scalability, some algorithms recommend items based on only a part of the item set instead of considering the whole dataset. Although this solution increases Scalability, it decreases Coverage and *Correctness* [2]. For such cases, an algorithm author may only report her algorithm's *Scalability*. Nevertheless, the

diversity of existing approaches is a very positive aspect. However, "it should be critically analyzed to ensure improvement in the field" [4].

Having said that, we cannot deny that there is a lack of uniformity and fairness in the current evaluation methods; as a result, each study deploys its own way to evaluate its proposed recommender [31]. Hence, a unified evaluation methodology becomes essential to compare recommenders [5]. According to Avaspour et al. [2], a better framework or standard for understanding the relationship between dimensions is required. Also, it is essential to evaluate RSs from multiple aspects because this makes systems more comprehensive [31]. As a potential solution, this paper proposes an evaluation approach called ***Com****prehensive* ***Per****formance* evaluation (***ComPer***), which is inspired by the idea that an RS can be viewed as a human being with cognitive skills, and so, the recommendation process can be analogized to a human learning process. Consequently, the outcomes of the recommendation process can be evaluated in the same manner as we evaluate the outcomes of the humans' learning process.

For this, *ComPer* innovates a mapping between Bloom's taxonomy (a well-known classification of educational learning objectives [6]) and the main phases of RS (i.e., information collection, learning, and recommending). Based on this mapping, a correlation between the most common evaluation dimensions and Bloom's cognitive dimension is inferred. The inferred correlation is used to calculate the final value of *ComPer*.

ComPer aims at achieving three goals: thoroughness, simplicity, and independency. To achieve the former goal, *ComPer* considers multiple evaluation dimensions instead of focusing on a single dimension. The second goal, simplicity, is satisfied by presenting the final result of the evaluation as a single value, which implicitly reflects multiple dimensions. Independency means that *ComPer* can provide results that are independent of data and evaluation settings. Independency leads to provide a fair comparison between different recommenders.

We assessed the applicability of the proposed methodology through empirical experiments using three different datasets. The experiments show promising results such that *ComPer* achieves its goals. However, further in-depth experiments are still required to assure the applicability of *ComPer*; so, we invite researchers and businesses to test the proposed approach through user studies on their real systems.

The contribution of this article is twofold; first, an analogy between RS and human. This analogy opens up prospects for advancing the research in the field of RSs. Second, it proposes a new evaluation approach (called *ComPer*) for RSs. Up to our knowledge, *ComPer* is the first evaluation approach that considers RSs as humans. In addition, *ComPer* mitigates the high dimensionality issue mentioned above; such that, it does not introduce a new evaluation metric; instead, *ComPer* combines multiple conventional evaluation metrics. The advantage of relying on conventional metrics is that these metrics have already been discussed and utilized in the literature.

The rest of this paper is organized as follows: Section II reviews the literature and provides a discussion about related methodologies and their limitations. Section III briefly describes the main concepts of the paper. Section IV introduces the analogy between recommenders and humans, and it describes the proposed evaluation approach. The experimental evaluation is discussed in section V, followed by a discussion about the validation of the proposed approach in section VI. Then the paper is concluded in section VII.

II. Related Work

The literature has recently witnessed increasing attention to the evaluation methodologies of RSs. Particularly, there is almost a consensus on the necessity of a common methodology to evaluate the multifaceted nature of RSs. Hence, several researchers highlighted the need for such an evaluation methodology. This section presents some of the evaluation models that have been proposed recently, along with a discussion about their limitations compared to the one that is proposed in this paper.

A methodological description framework and a formalization of the offline evaluation procedure of Time-Aware RSs (TARS) are provided in [7]. Through their survey and analysis of the TARS literature, the authors found clear divergences in the evaluation protocols and methodologies. Consequently, they identified a set of key conditions; based on these conditions, they introduced a categorization of evaluation protocols for TARS. Finally, the authors suggested methodological guidelines that may help researchers to select the proper combination of evaluation conditions. Comparing to our work, this study focuses on only one type of RSs, namely the Time-Aware RS. Also, it does not provide a comprehensive enough evaluation framework; instead, it provides some evaluation guidelines that can be followed to evaluate TARS.

Another evaluation approach is proposed by Shahab et al. [32]. The goal of the proposed approach is to avoid biased and fake ratings in the dataset. For this, the paper introduced a user's sincerity measure that concerns of eliminating feedback of insincere users. This measure is calculated based on different factors, including user visits to the review page and the time spent on that page. Based on this information, the authors define the Product Importance Score (PIS), and the Product Preference Score (PPS). Then, the authors obtained the Comprehensive Veracity Measure (CVM), which is defined as the average of different evaluation measures. Although this evaluation measure considers multiple measures, it is considered limited because it only considers accuracy-related measures (such as Mean Average Precision (MAP), Mean Reciprocal Rank (MRR), false-positive rate (FPR) and false-negative rate (FNR)); As mentioned above, *Accuracy* does not reflect the actual performance of RSs.

Meyer et al. [1] provide a methodology to analyze the efficiency of RSs in an industrial context. They

categorized RSs into four structuring functions: *help to decide, help to compare, help to discover, and help to explore*. Based on these functions, the authors proposed an offline evaluation protocol; the main idea of this protocol is to map each one of these four functions to the appropriate evaluation measures. Although the proposed protocol aims at considering different recommendation functions for RSs in general, it fails to consider different evaluation dimensions; That is, the protocol focuses mainly on evaluating the accuracy of the recommendations using different measures based on the aforementioned functions.

Mojisola et al. [11] aim at providing an evaluation approach that can alleviate the challenges of both online and offline evaluation protocols. The goal of the proposed approach is to be a compromise solution that gets benefits from the accuracy of online evaluation and the simplicity of offline experiments. Hence, Mojisola et al. investigated the use of crowdsourcing as a supporting source of willing users to evaluate Technology Enhanced Learning (TEL) RSs. Their results show that crowdsourcing can be a potential alternative for evaluating TEL. Although this research shows promising results, it is still in its early stages, and it has been investigated in a single domain of RSs, which is the TEL recommender systems.

Some researchers focus on introducing protocols that serve as supportive solutions to enhance traditional evaluation settings rather than introducing an evaluation methodology. In [12], for instance, the authors developed a group testing framework that aims at providing means to enhance evaluating group recommendation. The main goal of the proposed framework is to generate synthetic groups that are modelling actual group preferences automatically. The generated groups are parameterized to test different group contexts. The proposed framework contains two main components: First, group modelling, which defines specific group characteristics that will be used to form the synthesized groups. Second, group formation, which is a mechanism that identifies compatible groups from a dataset of single users by applying the defined model (i.e., the group model that is identified in the first component). This framework provides a useful tool for evaluating group recommenders. However, it has nothing to do with other types of recommenders, which is a limitation comparing to our proposed approach.

Another supportive solution has been proposed by Said et al. [19]. The goal of this model is to propose a method for fair data splitting that can give more accurate results for recommenders' correctness. It is designed specifically for the recommendation task (i.e., the task of recommending an item or a set of items). The protocol has been developed with a "*find good item*" scenario in mind. The authors suggested criteria to choose candidate items (or users) for the training and testing sets. The goal of this splitting method is to mitigate the issue that is presented by the bias of the accuracy values to the users with many items in the dataset. Also, it aims to make sure that all available data for each user is used for training the algorithm. This protocol is limited in terms of applicability and dimensionality; that is, it is limited to the top-k recommendation algorithms (i.e., algorithms that provide top-k recommended items), and it considers the accuracy dimension only.

Other researchers focus on evaluating the usability aspects of RSs; Pu et al. [9], for instance, proposed a user-centric model, called *ResQue*. The model has been built based on usability-oriented research in the RS field, as well as principles from popular usability evaluation models. *ResQue* is a user study-based model that consists of thirteen constructs and a total of sixty questions. It categorizes the questions into four higher-level constructs, which are: Perceived system qualities, users' beliefs, users' subjective attitudes, and their behavioural intentions. The model aims to assess the perceived qualities of recommenders, such as usability, usefulness, and interface quality. This framework presents a unified approach to user-driven evaluation. However, it is time-intensive for participants, and it may be overly costly for focused hypothesis testing. Also, it focuses only on the usability aspects of RS.

AppFunnel [10] is another example of a usage-centric evaluation protocol; it adopts the concept of conversion funnels to the mobile app RSs. The main goal of *AppFunnel* is to extend the evaluation of mobile app RSs from only considering the instant use of the app to considering the long-term use of it. To do so, it allows the usage centric evaluation to consider application engagement stages along the applications' lifecycle beyond installation. *AppFunnel* suggests that users go through four stages in order to reach the final conversion. The final conversion represents the conversion of the application from a recommended app to one that is used in the long-term. The four stages are View, Installation, Direct Usage, and Long-term usage. Like many other models, *AppFunnel* is not applicable to all kinds of RSs; it can be applied for recommenders that recommend items that can be used for a long time (applications, in particular). Moreover, it considers only one dimension, which is user engagement, as a measure of RS quality.

Olmo and Gaudioso [33] propose an objective-based framework for the standardization of recommendation system evaluations. They view RSs as applications that consist of two main subsystems: interactive (called the **Guide**) and non-interactive (called the **Filters**). The **Guide** concerns *when* and *how* the recommended items should be presented to users. The **Filter** focuses on *what* to recommend (i.e., which item should be shown to the user). Accordingly, they categorize RSs based on these two functions. Then, a performance metric (denoted as ***P***) has been introduced as the quantification of the final performance of a recommendation system over a set of sessions. ***P*** is defined as the number of selected relevant recommendations that have been followed by the user over a recommendation session. This work is comprehensive enough such that it covers a wide range of RSs. However, it concerns only one evaluation dimension. Also, an issue related to the introduced metric (***P***) is that it does not treat the usual problems that traditional

metrics do. For instance, if, in one session, the user followed ten items (i.e., ten items were indeed useful) out of a million recommendations, this is considered as good as if the system displayed ten recommendations all of which the user followed (i.e., all of which were useful).

These are the evaluation methods (or models) that are most relevant to our work. We observed that the literature suffers from three main limitations comparing to our proposed approach. These limitations can be summarized as follows:

1. *Limited domain*: the proposed model is application-based (i.e., it is proposed to evaluate the RSs of a particular application area), such as [7, 10].
2. *Limited dimensionality*: protocols that consider only one or two evaluation dimensions, this issue exists almost in all the proposed protocols, such as [1, 19].
3. *Highly costly*: some protocols, such as [8, 11], focus on properties that cannot be evaluated offline; these protocols rely on the user study evaluation model. The problem with user-centric evaluations is that they are considered costly regarding time, effort, and money.

III. Preliminaries

This section gives a brief description of the main concepts that we rely on to introduce our approach. These components are Bloom's taxonomy, RS main phases, and the most common evaluation dimensions of RSs.

A. Bloom's Taxonomy

Bloom's Taxonomy is a model that was introduced to the education field in 1956 by Benjamin Bloom, an educational psychologist at the University of Chicago [6]. Bloom's taxonomy classifies people's ways of learning into three domains, namely: cognitive domain, affective domain, and sensory domain. It also defines assessment dimensions for each domain to evaluate different educational outcomes. The cognitive domain, which is the domain of particular interest of this paper, underlines six intellectual outcomes, which are:

1. *Remember*: the ability to retrieve knowledge from memory.
2. *Understand*: The ability to determine meanings from all kinds of messages.
3. *Apply*: The ability to implement a procedure in a given situation.
4. *Analyze*: The ability to break material into constituent parts and detect how the parts are related to each other.
5. *Evaluate*: the ability to make judgments.
6. *Create*: The ability to put elements together and recognize them into a new pattern.

B. Recommender Systems' Main Phases

According to Isinkaye et al. [13], a recommendation process is divided into three main phases; first, *Information collection*, this phase concerns about collecting users' information to build their profiles that will be used by the second phase. The performance of RSs is related strongly to the amount and type of information collected. This information can be collected explicitly (as inputs from the users), or implicitly (by inferring users' preferences indirectly such ass inferring preferences through user's behaviour). Second, the *Learning* phase in which the system concerns of applying learning algorithms to exploiting users' features from the information gathered in the previous phase. The last phase is the *Prediction/recommendation* phase. This phase focuses on predicting which items the user may (or may not) prefer. The result of this phase is affected by the previous two phases; the more the information collected is, and the better the learning algorithm used is, the more useful the recommendation is.

C. Recommender Systems' Evaluation Dimensions

More than sixteen different evaluation metrics or dimensions[1] have been introduced to the recommendation field [2]. Avazpour et al. [2] classified these dimensions into four categories: 1) Recommendation-centric, which involves dimensions that focus on the recommendations (i.e., the outcomes of RSs). 2) System-centric; it the dimensions that provide a way to assess the recommender itself. 3) User-centric, which assess the degree to which users' requirements (or needs) are fulfilled. 4) Delivery-centric, dimensions that assess the usefulness of the recommender (i.e., it focuses on the RS in the context of use). Those dimensions, along with their definitions, are listed in Table 1. A detailed discussion about the dimensions and how they can be evaluated can be found in [2, 20].

As it's described in the next subsection, *ComPer* deals with RSs as humans, so it concerns the cognitive aspect of recommenders (i.e., cognitive skills that are required to provide recommendations). That is, it focuses mainly on the recommendation process itself (represented by its main phases, as mentioned in the previous subsection) rather than on the other aspects of recommendation, such as usability or risk. Therefore, *ComPer* emphasizes the nine dimensions that belong to the first two categories; namely, the recommendation-centric and the system-centric categories, because these dimensions assess the recommender and its recommendations (analogously, human learning and the learning outcomes).

[1]In the literature, the terms "properties", "dimensions", and "metrics" are used interchangeably to represent the RSs' dimensions that we want to assess. To reduce ambiguity, we will use the term "dimension", unless mentioned otherwise.to assess. To reduce ambiguity, we will use the term "dimension", unless mentioned otherwise.

Table 1. Evaluation Dimensions and Their Categories [2]

Category	Dimension	Definition
Recommendation-centric	Correctness	The closeness of the recommendations or predictions to the actual user preferences
	Coverage	The proportion of items (or users) that the system can recommend (or the system can generate recommendations for)
	Diversity	The dissimilarity between recommended items
	Confidence	How confident is the recommender to its results (recommendations or predictions)
System-centric	Robustness	How stable is the RS in the presence of fake (or false) information
	Adaptivity, or Learning Rate	How adaptable (fast) is the system to new information
	Scalability	How scalable is the system under extreme conditions, such as a huge dataset?
	Stability	The consistency degree of the recommendations over time
	Privacy	Is there any potential risk users' privacy
User-centric	Trust	To what extent do users trust the system's recommendations
	Novelty	Recommending items that are new to the user.
	Serendipity	How surprising, yet successful are the recommendations?
	Utility	The value gained from the system for different actors, such as users and system owners
	Risk	How risky is the acceptance of the recommendation for the user?
Delivery-centric	Usability	How usable is the recommender (i.e., how easy is it to adopt it)?
	User preferences	How do users perceive the recommendation system?

It is worth to mention that the dimensions are not of the same popularity for RSs. For example, evaluating the "*Accuracy*" of a programming code recommender is more important than evaluating its "*Diversity*." Unfortunately, there is no consensus in the literature about the most important evaluation dimensions. Thus, we surveyed the literature to discover which are the most common dimensions out of the nine considered ones.

The study considered six proceedings of the ACM RecSys [2] conference. Each paper published under the "Full-Length" track was selected as a relevant paper, the total number of articles considered after this step was 157 papers. Based on the abstract and the evaluation section of these articles, we excluded articles that do not include an evaluation section (i.e., they do not evaluate a recommender, or they do not describe the evaluation method). The final number of articles considered in this study was (135) articles.

Our results indicate that *Correctness* (also known as Accuracy), is the most considered property; this popularity is not surprising as its well-known in the literature that the efficiency of RSs is widely assessed based on Correctness [28], and *Correctness* is one of the earliest dimensions introduced to the recommendation field [21]. *Coverage* and *Diversity*, are the second popular dimensions, followed by *Robustness*, *Scalability, and privacy*. The results also show that *Confidence*, *Adaptability*, and *Stability* have not been evaluated by any of the articles. So, we omitted them from our framework. In addition, we omitted *Privacy* because introducing a measure to evaluate it is a very difficult task, and measuring its effect is still not fully explored [2]. Therefore, our framework considers the following five common evaluation dimensions: *Accuracy*, *Coverage*, *Diversity*, *Robustness*, and *Scalability*.

IV. Proposed Evaluation Approach

This section discusses our proposed evaluation approach, which We call the ***Com****prehensive* ***Per****formance* Evaluation (***ComPer***). The proposed approach is inspired by our vision that an RS, with its recommendation tasks, could be analogous to human beings with their cognitive skills. The essence of the proposed approach is to obtain a correlation between RSs, presented by their main phases and common dimensions, at one side, and humans' cognitive skills, presented by Bloom's taxonomy and its cognitive dimension, on the other side.

The rest of this section is organized as follows: First, it introduces the analogy between recommenders and humans. Second, it shows our suggested mapping between RS's main phases and Bloom's learning objectives. Third, it describes how we deployed this mapping to infer a numerical correlation matrix between learning objectives and Evaluation dimensions. Finally, it describes an algorithm that shows the steps of getting the final value of *ComPer*.

A. Recommender and Human: An Analogy

This section illustrates our proposed mappings between the cognitive skills of human beings and the main phases of RSs by providing an analogy between a salesperson, Alice, and an arbitrary RS, as follows:

To give recommendations, RS collects information about users to build their profiles, and it exploits users' features to predict their preferences. Analogously, a salesperson Alice tries to recognize her customers to become aware of their preferences in order to suggest items that may be of their interest. Both, recommender

[2] https://recsys.acm.org/

and Alice, collect information about their users (or customers) to build enough knowledge (i.e., to learn) about users' (or Customers') preferences, then they use this knowledge to suggest or recommend items that match their users' (or customers') preferences.

We notice that Alice, as a human being with cognitive skills, can perform the tasks of an RS and she go through the same phases (i.e., collect information, learn, and predict/recommend), as follows: during her work, Alice tries to *remember* her customers and understand their needs in order to **collect** correct **information** about their purchasing trends. After that, she analyzes the customer's attitudes based on the already collected information. Such an analysis leads Alice to **learn** her customers' preferences. By the end of this learning process, Alice reaches a level where she becomes able to link pieces of information together and creates patterns for each customer. Therefore, she is able to **predict** what may/may not attract a customer, and more than that, to **recommend** the best product that suits each customer.

The above analogy suggests that since Alice, as a human being, performed the same tasks as an RS, then RSs could be viewed as human beings and could have the comprehension skills of humans. Inspired by this idea, we can say that RSs can be analyzed and assessed, generally, in the same way that human's cognitive abilities are assessed. In particular, since the recommendation process is just like humans' learning process, we imply that an RS learning outcome (which are its predictions or recommendations) can be assessed in the same way that human learning outcomes are assessed. Thus, we argue that Bloom's cognitive dimension and its six learning objectives of humans could be used as a basis to evaluate the recognition abilities of an RS, as described in the next section.

B. Mapping Bloom's Learning Objectives and Recommender's main phases

Based on the definitions of RS phases, and Bloom's learning objectives, we inferred the following mappings[3]; to **collect information**, an RS should be able to retrieve information and construct meanings from different types of messages, which means that an RS should be able to *remember* and *understand*. To **learn**, an RS should be able to *apply* prior knowledge in different situations and *analyze* the interrelationships between different components of a system. Finally, an RS's **recommendation/prediction** depends on its ability to make a judgment, (i.e., *evaluate)*, and to recognize elements into new structures, (i.e., *create)*.

Fig.1 visualizes the aforementioned connections between each of the learning objectives and the main phases of RSs. At the top of the figure, we see the fundamental goal of the evaluation process (i.e., Quality of recommender). The middle level of the figure shows RS's main phases, while the bottom level shows human's learning objectives. The arrows depict the effect of one component on the quality of the other, as follows: the quality of a recommender is proportional to the recommendation process presented by its three main phases. This relation is depicted in the figure by an arrow heading toward the fundamental goal. In the middle, we can see two arrows facing the third phase (i.e., Recommend). These two arrows indicate that the output of this phase is proportional to the other two phases (as described in section III). At the lower level, each learning objective has an arrow toward a phase of the recommender; it shows that the quality of each phase of RS is affected by it is ability in two learning objectives (as described in the previous paragraph). That is, a better ability in objective (X) leads to better outputs of phase (Y). For instance, the recommender's **Learning** process gets better as the recommender's ability to *Analyze* or *Apply* is enhanced.

As a conclusion, Fig.1 shows that the overall quality of an RS is proportional to its recommendation process, which is proportional to the recommender's recognition skills (presented by the learning objectives). That is, the quality of recommender increases as its recognition skills increase.

The above mapping suggests that the evaluation dimensions of humans' cognitive skills can be adopted to evaluate the skills of RSs. These dimensions, however, are unfamiliar in the recommendation context. Thus, we established a numerical correlation between Bloom's learning objectives and the five most popular RS evaluation dimensions mentioned above; such that each learning objective is captured and assessed by these evaluation dimensions. The following subsection describes the process by which we inferred these correlations.

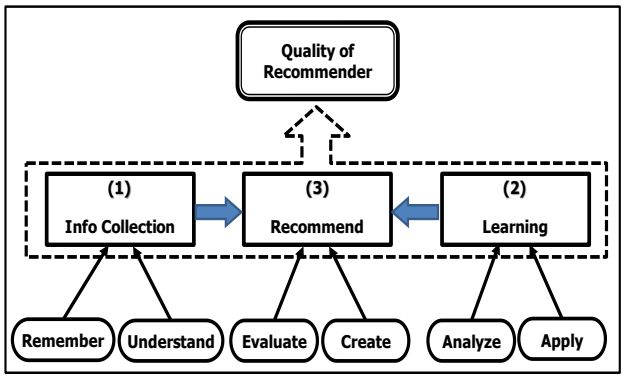

Fig.1. Mapping recommenders' main phases and Bloom's learning objectives

C. Correlating Bloom's Learning Objectives and Recommender's Evaluation Dimensions

As we have mentioned before, since the learning objectives of Bloom's cognitive dimension are unfamiliar in the recommendation literature, it is necessary to infer a correlation mapping between these dimensions and the evaluation dimensions of RSs. Thus, we inferred a correlation based on the definitions of both sides (i.e.,

[3] For clarity in this section, the words in bold represent the main phases of RS, and the words in italic represent the learning objectives of Bloom's cognitive dimension

learning objectives and evaluation dimensions) . This correlation is done through a three-step process, which involves domain experts, Natural Language Processing (NLP) and the aggregation of all the results, as depicted by Fig.2. The figure is divided into three horizontal parts, where each part represents the components of a single step, starting from step I at the top. Vertically, the figure shows the inputs, actors, and outputs of each step. At the right-hand side of the figure, we can see two matching operations (i.e., Matching 1 and Matching 2); these two operations were used to combine the results obtained from the three steps (as described in the subsequent discussion).

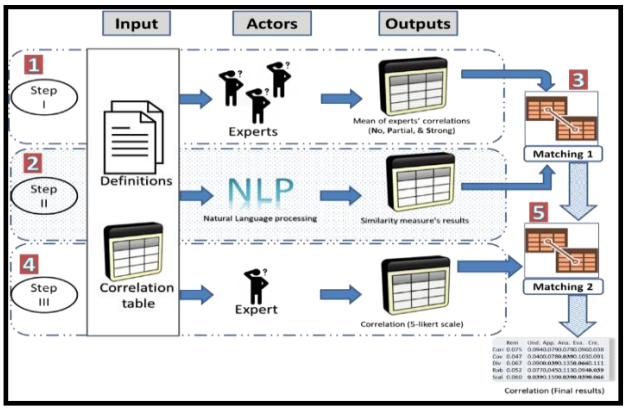

Fig.2. The process of correlating Bloom's learning objective and RSs evaluation dimensions

It can be noticed from Fig.2 that all the steps share the same inputs, which are: a document that contains the definitions and descriptions about each of the learning objectives and the evaluation dimensions, as well as an empty table (called correlation table) to be filled by the actors. For the results to be more accurate, we tried to find as many related descriptions as possible. These definitions have been collected from different resources, including research papers and web resources. As illustrated in Fig.2, the correlation process is ordered chronologically (from 1 to 5), and the order is presented as numbers in a square shape. The sequence is as follows: Step I, Step II, Matching 1, Step III, and finally Matching 2. The main difference between the three steps is in their output (i.e., the type of correlation values).

***Step I*:** Three domain experts were given the definitions of both learning objectives and evaluation dimensions along with the correlation table. They have been asked to infer the correlations between both concepts and to provide explanations of the rationale behind their inferences. For the inferred correlation, the experts have been asked to give one of three labels, which are N (No-Correlation), P (Partial Correlation), or S (Strong correlation). To explain how the experts, draw correlations, let's take as an example the learning objective "*Remember*" and the evaluation dimension "*Coverage*." As mentioned previously in section III, "*Remember*" represents the ability to retrieve and recall relevant knowledge from long-term memory, "*Coverage*" represents the proportion of available information for which recommendations can be made (i.e., to what extent does the RS covers the items or users' space). Based on these definitions, the experts had the following two observations: 1) the more the information available, the better the remembering skill will be, and 2) the more the item/user space covered (or retrieved), the higher the *Coverage* value will be. According to these observations, the correlation between "*Coverage*" and "*Remember*" was given the labels (S), (S) and (P) by the three experts. Based on these three labels, the final result (Table 2) indicates that *Coverage* is strongly (S) correlated to *Remember*.

Table 2. Correlating Evaluation Dimensions and Learning Objectives (Results of Step I)

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	P	P	P	S	S	S
Coverage	S	P	S	P	S	S
Diversity	S	S	P	S	P	S
Robustness	P	P	S	S	S	P
Scalability	S	N	S	P	P	N

Another example is the relation between "*Understand*" and "*Diversity*," where the objective *Understand* means demonstrating an understanding of facts by organizing and comparing the given description. On the other side, providing diverse recommendations indicates that by giving the recommender some information and descriptions about items, it can understand the idea that two items are dissimilar. Given these two definitions, experts have inferred a strong correlation between both concepts (illustrated with strong (S) label in Table 2).

Following the same rationale discussed above, experts have inferred the rest of the correlations. Each expert has sent back the output (i.e., the correlation table) filled with his/her suggested (and justified) correlation between each objective and the five evaluation dimensions. Then we merged the results obtained from these three experts' by considering the mean of the three labels, as follows: if the experts gave three different answers (i.e., N, P, and S), P is considered as the mean of these answers. If at least two experts gave the same answer, this mutual answer is considered as the mean value. Table 2 shows the results that have been obtained after this step.

***Step II*:** The correlations obtained in the first step indicates how strong the relation between each objective and the evaluation dimensions is. However, these values have two limitations; they are discrete by nature, and they are not measurable. To overcome these limitations, these

labels need to be mapped to more fine-grained numerical values. To do so, we adopted concepts from the NLP domain. We presented the problem as an NLP classification task, where each of the learning objectives is considered as a document category, and the evaluation dimensions are considered as documents that need to be classified under these categories.

In particular, we followed the approach presented by [26]; We presented each one of the learning objectives and the evaluation dimensions as a feature vector; that is, we created a bag-of-words for each of the objectives and the dimensions using the input document (i.e., the document that contains their definitions that has been provided as an input, as described at the beginning of this section). These words have been pre-processed by removing stop words and repeated words. Then, we did word stemming for the remaining words. Finally, we calculated the similarity between every two vectors using the *Dice measure* (or the Sorensen-Dice index [29]), as defined by (1).

$$Sim_{Dice} \frac{2\times\left|F_x \cap F_y\right|}{\left|F_x\right|+\left|F_y\right|} \quad (1)$$

Where F_x is the feature vector of objective x, and F_y is the feature vector of the evaluation dimension (y). The similarity Dice (Sim_{Dice}) value represents the correlation strength between an evaluation dimension and a learning objective, as illustrated in Table 3. For instance, *Remember* is more similar to *Correctness* (0.075) than to *Coverage* (0.047).

Table 3. Correlating Evaluation Dimensions and Learning Objectives (Results of StepII)

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	0.075	0.094	0.079	0.079	0.096	0.038
Coverage	0.047	0.040	0.078	0.093	0.103	0.091
Diversity	0.067	0.090	0.102	0.135	0.107	0.111
Robustness	0.052	0.077	0.045	0.113	0.094	0.086
Scalability	0.060	0.081	0.159	0.136	0.138	0.101

To assess the consistency between the results of this step and the results of the first step, we introduced a mapping between similarity values and the correlation labels (i.e., N, P, and S) obtained in Step I. To ensure that this mapping is representative, we calculated the difference (d) between the maximum and the minimum Sim_{Dice} values. Then the range (d) is divided into three intervals that correspond to N, P, and S, as depicted in Fig.3 and (2).

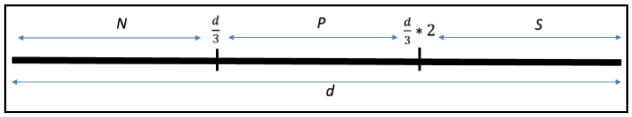

Fig.3. Mapping Dice similarity values to the correlation labels

$$SimToCorr = \begin{cases} N, & Sim_{Dice} \le \frac{d}{3} \\ P, & \frac{d}{3} \le Sim_{Dice} < \frac{d}{3}\times 2 \\ S, & Sim_{Dice} \ge \frac{d}{3}\times 2 \end{cases} \quad (2)$$

According to Table 3, the maximum Sim_{Dice} value is (0.159), and the minimum value is (0.038), the difference between these two values is ($d = 0.121$). accordingly, and based on (2), Table 4 shows the (N, P, and S) labels that correspond to each Sim_{Dice} value presented in Table 3.

Table 4. Mapping Similarity Values (Table 3) to the Correlation Labels (i.e., N, P, or S)

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	P	S	P	P	S	N
Coverage	P	N	P	S	S	S
Diversity	P	S	S	S	S	S
Robustness	P	P	P	S	S	S
Scalability	P	S	S	S	S	S

After comparing the results of step II and step I (i.e., Matching 1 in Fig.2), the matching shows around 50% consistency between the results of these two steps.Table 5 shows the results of Matching 1 operation, where the gray shaded cells with letter "O" represent the consistent values (i.e., where the two compared results are consistent), while the cells with "X" represent inconsistent results. For instance, the first cell in Table 5 (i.e., the correlation between Remember and Correctness) has an (O), which indicates that both the experts' step and the NLP step evaluate this correlation similarly. On the other hand, the next cell (the correlation between Remember and Coverage) has an (X), because the experts evaluated it as a strong correlation while the NLP evaluated it as a partial correlation.

The conflict between results can be attributed to the discrete nature of the correlations (obtained in step I) compared to the continuous range of similarity values

Table 5. Matching the Results of Step I and Step II

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	O	X	O	X	O	X
Coverage	X	X	X	X	O	O
Diversity	X	O	X	O	X	O
Robustness	O	O	X	O	O	X
Scalability	X	X	O	X	X	X

(obtained in step II). Since the experts have only three choices, they did not have the option to give a fuzzy rating; this, in turn, may cause some conflicts. To overcome this issue and to resolve these conflicts, step III takes place, as follows.

Step III*:* to resolve (or to mitigate) the conflicts between the previous two steps, another expert has been asked to infer the correlations for the conflicting results. At this step, unlike step I, the strength of the correlation has been evaluated on a 5-Likert scale from the weakest (1) to the strongest (5) correlation. To avoid bias, this expert does not know any information about the two previous steps. The expert has been given the autonomy to rate the correlation from 1 to 5 according to his own rational (for instance, if he sees that there's a strong correlation or not), without giving him any pre-defined label of the scores.

After getting the ratings from the last expert, the operation labelled "Matching 2" in Fig.2 is executed in order to examine if there is any improvement in the consistency with the conflict cases resulted after step II. To do so, we mapped the 5-Likert scale into the (N, P, and S) labels as follows; five (5) is considered as strong correlation (S), one (1) as no correlation (N), while 2, 3, and 4 as partial correlation (P).Table 6 shows these results. As we have mentioned, the expert has only considered the conflict cases of step II (i.e., "X" cells of Table 5); that is, other cells have not been considered, so these cells are filled with the "-" symbol.

Table 6. Correlating Evaluation Dimensions and Learning Objectives (Results of Step III)

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	-	S	-	P	-	N
Coverage	P	N	P	N	-	-
Diversity	P	-	N	-	P	-
Robustness	-	-	P	-	-	N
Scalability	P	N	-	N	N	P

After comparing the results of this step with the NLP results (Table 4), nine out of seventeen conflict cases are resolved.Table 7 represents the consistency table that is obtained after the Matching 2 operation.

Table 7. Matching the Results of Step II and Step III

	Rememb	Understa	Apply	Analyze	Evaluate	Create
Correctness	O	O	O	O	O	O
Coverage	O	O	O	X	O	O
Diversity	O	O	X	O	X	O
Robustness	O	O	O	O	O	X
Scalability	O	X	O	X	X	X

As a result of these three steps, we have confirmed (22) correlations out of (30) in total, as it is depicted in Table 7. To resolve the remaining (8) cases of conflicts, we compared their values that have been obtained as results of the three steps. The mean of these three values is considered the final correlation. For example, since the results of steps I, II, and III gave the correlation between *Understand* and *Scalability* N, S, and N labels, respectively, the value N is considered as a final result of this correlation.

We relied on the Dice values to give a numerical representation of the final correlation matrix, as follows: for results that show consistency with Dice results (i.e., consistent cells in Table 7), the corresponding Dice values are used. For the inconsistent cells (i.e., the eight "X" cells in Table 7, we used the average of all Dice results that belong to the corresponding correlation label. For example, for "N" labels, we took the average of the Dice values (as resulted from step II) that was categorized (according to equation (2)) under the "No-correlation" label; according to equation (2) and Table 3, two Dice values have been mapped to "N" label, these values are (0.038) and (0.04). The average of these two values is (0.039). Hence, since the correlation label between *Scalability* and *Understand* is "N" (as it is explained in the previous paragraph), it is given the value (0.039).

The final correlation results obtained after this process is presented in Table 8. The table shows the correlation strength between Bloom's learning objectives and each of the evaluation dimensions of RSs. Rows represent RS

Table 8. Correlation Matrix

	Remember	Understand	Apply	Analyze	Evaluate	Create
Correctness	0.075	0.094	0.079	0.079	0.096	0.038
Coverage	0.047	0.040	0.078	0.066	0.103	0.091
Diversity	0.067	0.090	0.066	0.135	0.066	0.111
Robustness	0.052	0.077	0.045	0.113	0.094	0.066
Scalability	0.060	0.039	0.159	0.066	0.066	0.066

dimensions and columns represent Bloom's learning objectives. Numbers represent the strength of the correlation; the higher the value, the stronger the correlation. For instance, Remember correlation with Correctness is higher than its correlation with Coverage

D. Visualization of the Correlation Between ComPer Components

The full visualization of *ComPer*'s components and their relationships is depicted in Fig.4. At the center of the figure, we can see the three main phases of a recommendation process, along with the learning objectives. The core of the model is divided into three parts based on RS's main phases; each part contains two corresponding learning objectives, as described in section B above. The evaluation dimensions are sorted based on their correlation strength with the corresponding objective; the closer the dimension to the center, the stronger the correlation with the objective. For instance, *Correctness* has the strongest correlation with *Remember*, and *Coverage* has the weakest correlation with it.

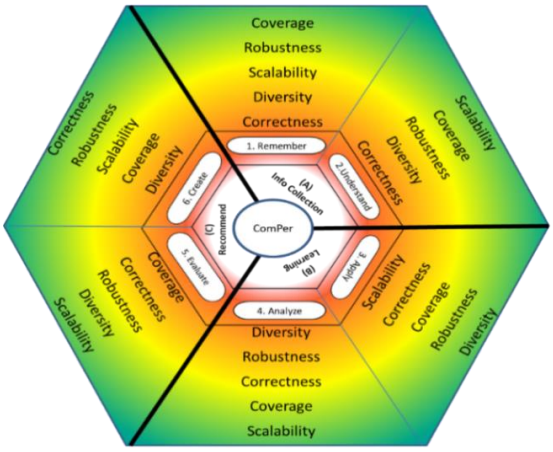

Fig.4. ComPer's main components

E. Formal Definitions

In order to present the applicability of the proposed framework, we formally define its components as follows: the set of all learning objectives is defined as O = {Remember, Understand, Apply, Analyze, Evaluate, Create}, and the set of evaluation dimensions of recommender as $D=\{V_{Correctness}, V_{Coverage}, V_{Diversity}, V_{Robustness}, V_{Scalability}\}$ where V means the values of the corresponding dimension. For instance, $V_{Correctness}$ is the value obtained when we evaluate correctness, $V_{Coverage}$ is the coverage value, and so on. $Corr_{ij}$ is the correlation between an objective ($o_i \in O$) and a dimension ($d_j \in D$), as presented in Table 8 Based on these definitions, we can define *ComPer* (the Comprehensive Performance evaluation) as the following

$$ComPer = \sum_{i=1}^{|O|}\sum_{j=1}^{|D|} d_j \times Corr_{ij} \quad (3)$$

Equation (3) represents the final value of the proposed method, which gives us the comprehensive performance of a recommender. The higher the *ComPer*, the better the recommender. It is noteworthy that *ComPer* depends mainly on dimensions that already exist in the literature. The rationale behind this is that these dimensions have already been validated, and they have been used in the literature. The aggregation of these dimensions, however, introduces two issues; firstly, not all dimensions have the same scale of results (i.e., not all of them get a value between 0 and 1, for instance). Secondly, the highest values do not necessarily mean better results. For example, when evaluating *Scalability* in terms of time complexity, the higher the time, the worse the recommender; Another example is regarding *Robustness*, which shows how tolerant the recommender is to biased or fake information. So, when calculating *Robustness* as the difference in accuracy before and after injecting false information, the highest value indicates the lowest *Robustness*. We refer to such dimensions as the "Negative Dimensions." Out of the five dimensions that *ComPer* considers, two of them are negative dimensions, which are *Scalability* and *Robustness*.

To overcome the first issue, we use a normalization technique using (4):

$$d' = 1 - \frac{1}{d+1} \quad (4)$$

Where d' is the normalized value of d. In this way, we normalize all the results to be in the range [0, 1]. It is noteworthy to mention that using other normalization methods (such as the min-max normalization) is possible. However, we used this ad hoc normalization because it is a general method, and it is applicable for any single value even if the range of values is unknown. To overcome the second issue (i.e., the negative dimensions), We find (d''), as follows ($d'' = 1 - d'$). Algorithm 1 shows the complete steps of the proposed evaluation approach.

ALGORITHM 1: ComPer calculation process

Input: *O, D, correlation matrix (Corr)*
Output: *ComPer*
Begin:
$ComPer \leftarrow 0$
1) **for each** *d in D* **do**
$d' \leftarrow 1 - \frac{1}{d+1}$
$d \leftarrow d'$

$If d \in NegativeDimensions$
$d'' \leftarrow 1 - d'$
$d \leftarrow d''$

2) **for each** *o in O* **do**
$o \leftarrow \sum_{d \in D} (d * \mathrm{Corr}_{od})$

3) $ComPer \leftarrow \sum_{o \in O} o$
End

V. Experimental Evaluation

This section presents a practical implementation of the proposed approach. It serves as a proof of concept for *ComPer*. The experiments demonstrate the usability of *ComPer*, as well as its ability to provide results that are independent of evaluation settings. Also, it shows, by numerical examples, how easier it is to decide based on *ComPer* results comparing to the conventional evaluation.

All the experiments are carried on a laptop with 6 GB RAM, Intel Core i5-4200M CPU with four cores running at 2.5 GHz, and 500 GB hard disk. Windows 10 home edition was the operating system. All algorithms, evaluation measures, and other settings are implemented using Librec [14], an open-source Java library for RSs. Librec is one of the modern Java-based recommendation libraries that provides a large number of recommendation algorithms. Also, it is the only library, along with MyMediaLite [4], that can provide state-of-the-art algorithms besides those classical ones [22], and Librec runs faster according to a benchmark evaluation [14]. For this, many researchers found Librec a reliable library, so they have used it to implement and evaluate their algorithms [23, 24, and 25].

A. Configurations

The experiments compare two collaborative filtering algorithms[5], which are AspectModel [15] and PLSA [16]. Both are ranking algorithms (i.e., they provide a ranked list of top-k items). The performance of these algorithms has been assessed using three datasets; namely, Filmtrust, Movielens 100-k (or ML-100k, for simplicity), and CiaoDVD; we downloaded all the datasets from Librec official website [6]. Filmtrust is the smallest dataset comparing to the other two datasets. It was crawled by Librec team from the movie website, Filmtrust; it contains (35497) data records, each has three attributes: user-Id, movie-Id, and a rating. The ML-100k is a well-known dataset that is provided by the GroupLens research lab[7] for public use. It contains (100) thousands of ratings provided by around (1000) users on almost (1700) movies. It is also organized as triples (user Id, Item Id, and rating). The last and the largest dataset is the CiaoDVD dataset. The data was crawled by the Librec team in December 2013. In addition to users' ratings, CiaoDVD contains timestamps for ratings as well as trust information. Table 9 summarizes the properties of these three datasets.

Table 9. Datasets Properties

	Users	Items	Ratings	Sparsity
Filmtrust	1508	2071	35497	1.14%
ML-100k	943	1682	100000	6.3%
CiaoDVD	7375	99746	278483	0.038%

Each of the algorithms has been evaluated on the three datasets mentioned above. We split the data into training and testing sets randomly, based on ratings. A ratio-based data split has been followed, with a training set ratio varied from (0.2) up to (0.8), as steps of (0.2); that is, we have four different data splits for each dataset. The number of generated items for the Top-N recommendation has been fixed to be always N = 10.

B. Evaluation Dimensions

As we mentioned in section III, our framework considers five dimensions. It is worthwhile to mention that some of the evaluation dimensions can be evaluated using different measures or evaluation strategies. For this, it is necessary to clarify the measures used to assess each dimension, especially because *ComPer* depends heavily on the results of these measures. The following are brief explanations of the measures that we have used in the experiments for each of the evaluation dimensions.

- *Correctness*: correctness is the most common dimension in the literature. Different measures have been introduced to evaluate it, such as Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Precision, Recall, Area Under Curve (AUC), etc. Since our experiments consider ranking algorithms, we used the AUC to assess accuracy. The AUC measure is provided as a class within the Librec library. So, we instantiated an object of the "AUCEvaluator" class.
- *Coverage*: according to [2], coverage refers to either the item-space (catalog coverage) or user-space (prediction coverage). For the purpose of our experiment, we implemented the catalog coverage, which is implemented as the percentage of items that are recommended to all users within an experiment [17, 28].

[4] http://www.mymedialite.net/
[5] Both algorithms are implemented and included in Librec
[6] https://www.librec.net/datasets.html
[7] https://grouplens.org/about/what-is-grouplens/

- *Diversity*: Different measures have been introduced to evaluate diversity. For our experiments, we used the "diversityEvaluator" class that is already implemented in Librec. It calculates diversity as the average dissimilarity of all pairs of items in the recommended list at a specific cutoff position [27].
- *Robustness*: this dimension has been calculated as the Average Hit Ratio (Average-HR) shift after attacking the dataset [2]. To do so, we implemented a nuke attack (i.e., to inject fake profiles that try to nuke some of the items). Then we calculate the Average-HR for both, the original dataset (dataset without false information), and the attacked dataset (i.e., after injecting false information). The shift of the Average-HR is the difference between the two hit ratios. For instance, suppose that the Average-HR that we got before the attack is (0.23), and after the attack, it becomes (0.15), then the Robustness value, in this case, is the difference between these two values, which equals to (0.08).
- *Scalability*: according to [2], recommendation time is an important indication of system Scalability. Thus, we rely on the recommendation time (i.e., the time spent by a recommender to learn and recommend) as a measure of recommenders' Scalability.

C. Results

This section aims to show the difference between the conventional presentation of the results (i.e., presenting the results of each dimension separately)[8] comparing to the results of our proposed method. The section is divided into three subsections; each subsection demonstrates the applicability of *ComPer* in different situations, as follows; First, it discusses the results on a single dataset. That is, it shows how *ComPer* approach has the potential to provide consistent results for different data splits. Second, it demonstrates the consistency of *ComPer* over multiple datasets. Third, it investigates the benefit of using *ComPer*, even for averaged results. Finally, it shows whether *ComPer* results are indeed made difference, or it is just a combination of numbers like any regular combination.

1. Results on a Single Dataset

This subsection discusses the results obtained from the experiments on the Filmtrust dataset. Fig.5 shows the results as they are presented in the conventional way (a chart for each dimension); each figure shows the results of one dimension over the Filmtrust dataset, using four different splitting ratios (0.2, 0.4, 0.6, and 0.8). The charts show noticeable fluctuations in the superiority of one algorithm over the other; *Correctness* and *Robustness* charts show the superiority of the AspectModel algorithm, while *Coverage* and *Scalability* charts prove the opposite (i.e., the superiority of the PLSA algorithm). Also, fluctuations appear over the same dimension (as the *Diversity* chart shows). These fluctuations show that the plentiful of evaluation options make it easy to find an evaluation design that suits one algorithm but not the others, which in turn leads to, unfair (or biased) evaluation results [3].

ComPer, on the other side, combines all these results in a single, yet thorough value. After executing Algorithm 1, we got *ComPer* results, as depicted in Fig.6. The comparison between this figure and Fig.5 shows how *ComPer* mitigates the fluctuation issues that were apparent in Fig.5. Also, it exhibits the readability of *ComPer* results and the simplicity of analyzing these results to decide. That is, this section demonstrates the simplicity goal of *ComPer*, such that it assesses the effect of multiple dimensions in a single value. Also, it shows the consistency of *ComPer* over different experimental settings on the same dataset.

The next subsection demonstrates the independency of *ComPer* results. In particular, it examines the use of different datasets on the replicability of the results (i.e. whether the results over one dataset are comparable to the results over another dataset). Also, it investigates the effect of changing the data splitting ratio on the consistency of the results.

2. The Effect of Multiple Datasets

This subsection demonstrates the independency of *ComPer* by investigating the effect of changing experimental settings on the replicability of the results.

Fig.7 shows the results of comparing the two algorithms (AspectModel and PLSA) over three datasets. The figure is divided into (5) sections (a, b, c, d, and e), which represent the evaluation dimensions (Correctness, Coverage, Diversity, Robustness, and Scalability), respectively. Each section has three charts; starting from the left to the right-hand side, these charts represent the results obtained over Filmtrust, ML-100k, and CiaoDVD datasets, respectively.

Note that the results presented in Fig.7 are setting-dependent. That is, changing the experimental settings (dataset or splitting ratios) may generate conflicting results. The figure includes various examples of these conflicts. For instance, Fig.7-a shows that the AspectModel algorithm overcomes PLSA over Filmtrust and CiaoDVD datasets, but not over the ML-100k dataset. Also, Fig.7-c shows that the conflicting results can be generated because of changing the splitting ratios. For example, over the Filmtrust dataset, PLSA beats AspectModel when the algorithm uses 0.4 or 0.6 ratios for the training set, but not for the 0.2 and 0.8 ratios. These two examples and more demonstrate how the results can be affected by the evaluation settings. Fig.7 also assures the existence of the fluctuation issue that we noticed over a single dataset. For instance, over the CiaoDVD dataset, the AspectModel algorithm overcomes

[8] From now on, we will refer to this presentation as the "Conventional" way or presentation.

the PLSA algorithm in terms of *Correctness* and *Robustness*, while PLSA has superiority in terms of *Coverage* and *Scalability*.

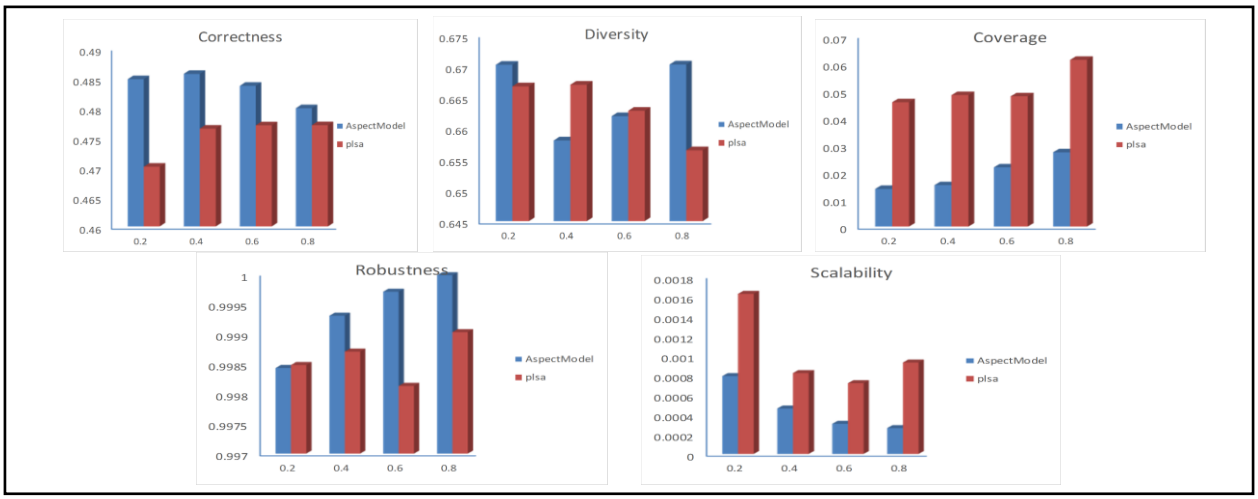

Fig.5. Conventional evaluation results over the Filmtrust dataset

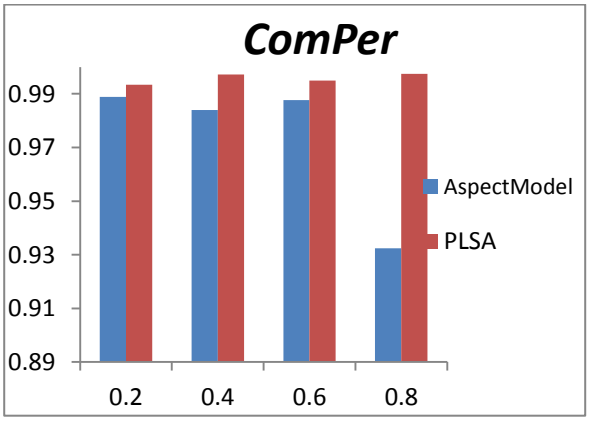

Fig.6. ComPer results over the Filmtrust dataset

These results confirm that the abundant of evaluation options makes it easy to find an evaluation design that suits a particular algorithm but ineligible for others, which represents an increasing impediment to fairly compare different studies [3]. C*omPer*, on the other hand, mitigates this issue by combining all these conflicting results into a consistent value that reflects all of them. Fig.8 depicts *ComPer* results over all datasets; it shows how the results have not been affected by changing the experimental settings. These observations show that the proposed approach opens the doors to compare recommendation algorithms fairly.

Another important observation here is that inferring conclusions through Fig.7 is very sophisticated, if not impossible; the figure doesn't show absolute superiority of one algorithm over the other on all datasets, neither it indicates the superiority of an algorithm on a single dataset. On the other side, we can easily draw conclusions through (*ComPer* results).

It is worthwhile to mention that the reason behind the fluctuated results in the conventional way is that the conventional metrics are sensitive to the evaluation settings. This sensitivity is one of the main issues for these metrics. For instance, *Correctness* is directly proportional to the dataset (i.e., the amount of available data). That is, an accurate recommendation is subject to the size and the density of the dataset. On the other hand, *ComPer* results show consistency over different experimental results because it considers different dimensions at one time; These dimensions influence each other. For instance, *Coverage* usually decreases as a function of *Correctness* [30]. That is, an increase in one dimension may cause a decrease in another one. *ComPer* reflects these contradictions by nature; For this, we can say that *ComPer* has the potential to evaluate recommenders fairly.

This subsection expounded how *ComPer* results are setting-independent. In addition, it assures the conclusions of the previous section, such that using *ComPer* results simplify the comparison between different algorithms and it mitigates the fluctuations appeared with the conventional results.

3. The Effect of Averaged Results

The previous subsections demonstrate the benefits of using our proposed method; they show, by practical examples, the simplicity and the independency of *ComPer* in comparison to the conventional presentation of the results. This section investigates the influence of averaging the results on alleviating the aforementioned issues.

To mitigate inconsistencies in the evaluation results, researchers usually consider averaging the results Table 10, Table 11, and Table 12 depict the averaged results of the experiments over the three datasets: Filmtrust, ML-100k, and CiaoDVD, respectively. Each cell of these tables is the average of the four values that we have obtained through different splits of the datasets. That is, for each dataset and each dimension, we averaged the results obtained from the four data splits (0.2, 0.4, 0.6, and 0.8).

It is worthwhile to mention that this section reports the results as tables instead of charts for readability purposes;

the values presented in the tables are of different scales, and hence, there is a big difference between small values (like *Correctness*) and big values (like *Scalability*). Thus, showing these values as a chart will be unclear or unreadable. Also, to enhance readability, better values are presented in a bolded style.

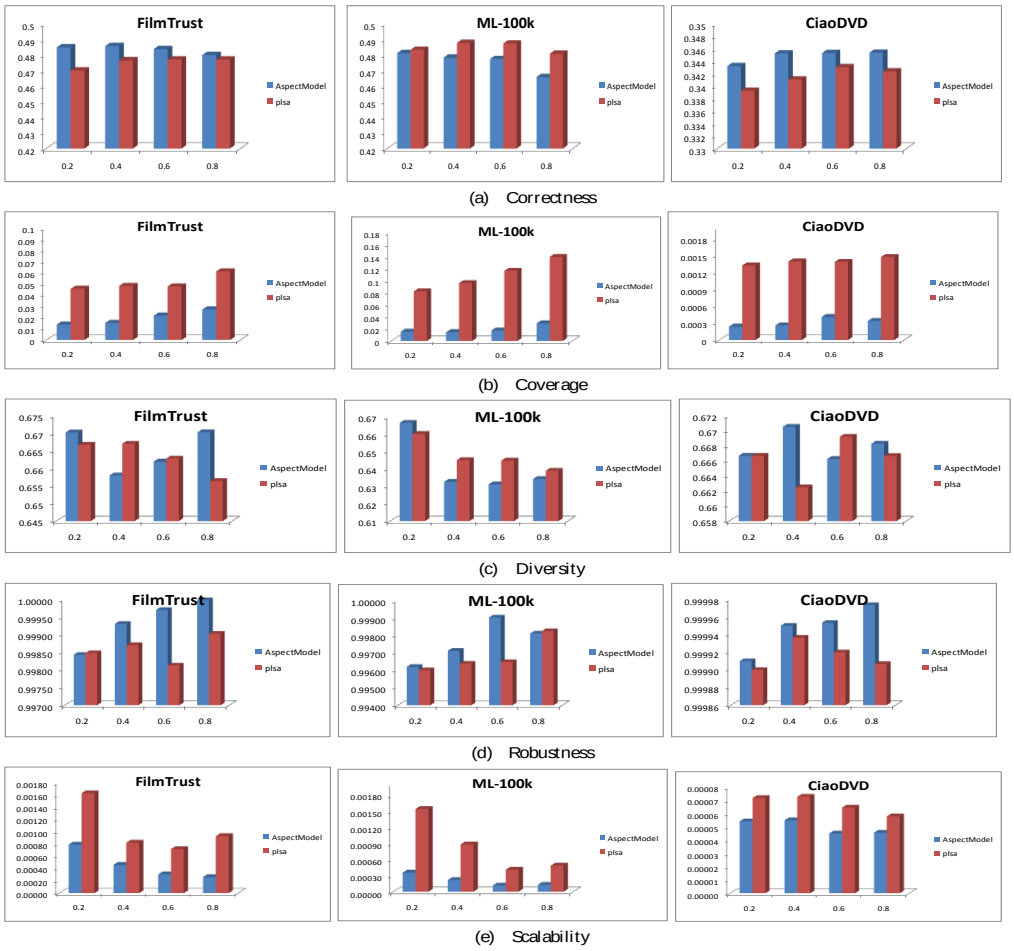

Fig.7. Presentation of conventional evaluation results. (a), (b), (c), (d), and (e) show the evaluation results of Correctness, Coverage, Diversity, Robustness, and Scalability, respectively over 3 datasets

As the tables depict, using averaged results reduces the high dimensionality issue. Hence, we can infer some conclusions from these tables; for instance, PLSA has absolute superiority over the AspectModel algorithm in terms of *Coverage* and *Scalability*, but the AspectModel overcomes PLSA in term of *Robustness*. Nevertheless, the most important note in this context is that even when we averaged the results, it is still not easy to decide on the best algorithm because of the contradictions and the high dimensionality. Even after considering the averaged values, the fluctuations between both algorithms even for a single dataset is still apparent; that is, no algorithm dominates in all dimensions over a single dataset. Also, the results of the same dimension are not replicated over different datasets, even after considering the averaged values. For instance, the results show that, in terms of *Correctness*, the AspectModel algorithm has superiority over PLSA on Filmtrust and CiaoDVD dataset while the PLSA beats the AspectModel in the ML-100k dataset. The opposite case happens in term of *Diversity*, where the aspect model overcomes PLSA twice. All these observations show how complicated the comparison between different algorithms using the conventional way is.

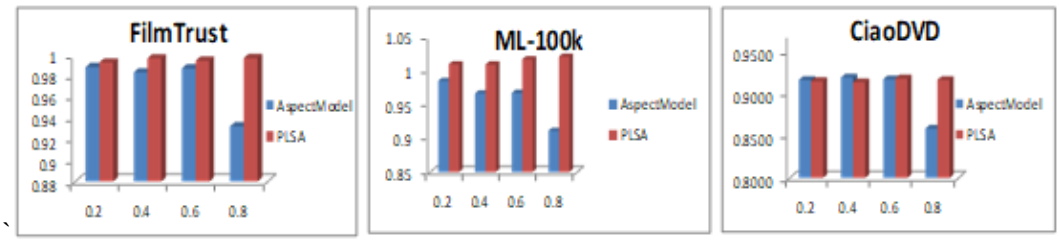

Fig.8. Presentation of ComPer results over three datasets

Table 10. Averaged Evaluation Results of The Filmtrust Dataset

	Correctness	Coverage	Diversity	Robustness	Scalability
AspectModel	**0.9361**	0.0199	**1.986**	**0.0065**	2630
PLSA	0.9053	**0.0534**	1.969	0.014	**1076**

Table 11. Averaged Evaluation Results of the Ml-100k Dataset

	Correctness	Coverage	Diversity	Robustness	Scalability
AspectModel	0.9072	0.019	1.790	**0.0024**	5938
PLSA	**0.9407**	**0.123**	**1.836**	0.0032	**1569.75**

Table 12. Averaged Evaluation Results of The CiaoDVD Dataset

	Correctness	Coverage	Diversity	Robustness	Scalability
AspectModel	**0.526**	0.0003	**2.011**	**0.0005**	20155
PLSA	0.518	**0.0014**	1.996	0.008	**15100**

Table **13**, in contrast, shows *ComPer* scores for both recommenders over the three datasets. These results show the superiority of PLSA over the AspectModel, in general. It assures that changing evaluation settings did not lead to incomparable *ComPer*'s results. Also, it is noticeable how easy making the decision based on *ComPer's* results is.

Table 13. *ComPer* Results of Both Recommenders for All Datasets

	Filmtrust	ML-100k	CiaoDVD
AspectModel	0.97	0.97	0.92
PLSA	**0.98**	**1.01**	0.92

4. Results Without Considering the Mapping:

The previous subsection demonstrated how *ComPer* simplifies the comparison by reducing the dimensionality, and how it supports fair evaluation by generating reproducible results that are independent of evaluation settings. This subsection investigates whether the proposed approach would really take effect (i.e., does use any other combination will take the same effect)? To do so, we combined the results of the five considered dimensions using two conventional ways; namely, summation and averaged. That is, we aggregated and averaged the results of the five dimensions into a single value without considering the inferred mapping. Fig.9-a and Fig.9-b depicts the aggregated and the averaged results, respectively.

As it is depicted in Fig.9, the results on the combination without taking into account the mapping does not solve the issues; both, the aggregated and the averaged results are affected by the setting change. Fig.9-a, for instance, shows that PLSA overcomes AspectModel on the Filmtrust dataset, but not on ML-100k.

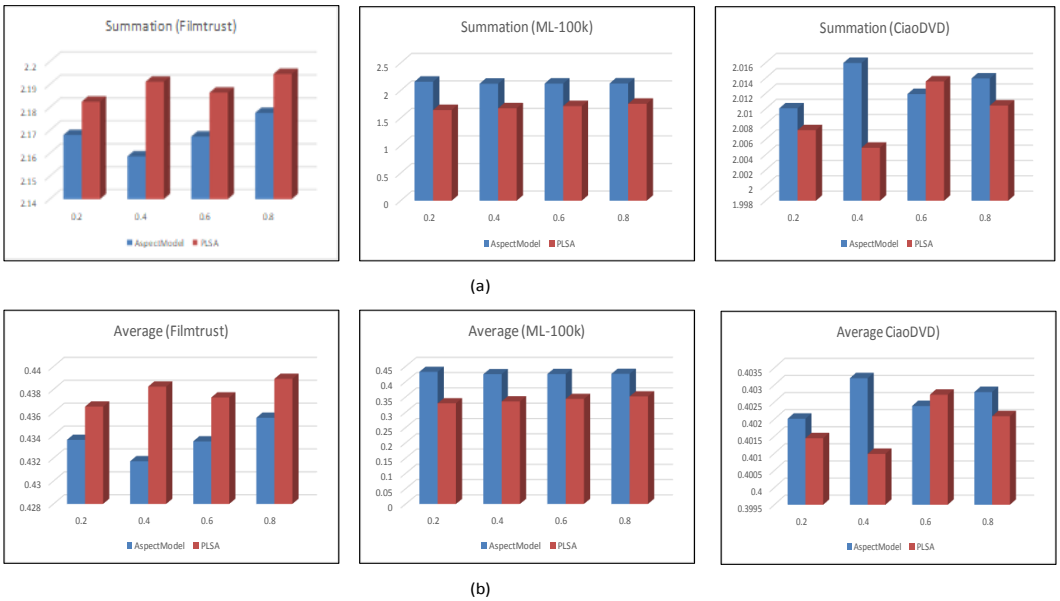

Fig.9. Conventional combination of evaluation dimensions. (a) summation of dimensions. (b) average of the dimensions

Having said that, we can say that the aforementioned issues cannot be resolved by combining the dimensions arbitrarily, and the proposed approach would really take effect.

Also, considering only one dimension is insufficient as there is a consensus in the literature that a single dimension is insufficient to evaluate recommenders. Accordingly, we can say that *ComPer* has the potential to provide a consistent and unified approach to evaluate RSs.

By comparing the final results of *ComPer* with the results obtained conventionally, we notice that *ComPer* achieves its three goals: thoroughness, simplicity, and independency. Thoroughness is presented by the *ComPer* evaluation methodology, in which it combines the results of the five dimensions into a single value. Simplicity is clear if we compared the final results of *ComPer* (Table 13) with the results of the five dimensions as presented in Table 10, Table 11, and Table 12. Independency is demonstrated in the previous subsections; these results show that *ComPer* is unbiased to a particular evaluation setting; which, in turn, allows for a fair evaluation for different recommenders.

VI. Discussion

The results presented in this paper demonstrate that: first, *ComPer* simplifies the comparison by reducing the high dimensionality (i.e., a single value that reflects the effect of five dimensions). Second, *ComPer* supports fairness by providing results that are not affected by experimental settings, and by taking into account different dimensions. Third, the proposed mapping (and the correlation matrix) would really take effect. Finally, based on the experimental results, we can say that *ComPer* has the potential to provide rational results.

Despite these promising results, wider experiments are still required to prove the applicability of the proposed approach. It is worthwhile to mention in this context that this article is meant to introduce *ComPer* to the literature; It describes the approach in detail, and it provides an initial demonstration of the approach. Therefore, the experimental results provided in the previous section serves as proof of concepts.

It is known that users have different expectations from RSs. According to Shahab et al. [32], "*we should opt for a recommender system that can identify the different users' requirements*." Accordingly, *ComPer* has been designed to evaluate multiple users' requirements (i.e., evaluation dimensions). However, since users' satisfaction is one of the main goals of any recommender, we will validate *ComPer* against users' satisfaction. The best way to evaluate users' satisfaction is through online user studies. These studies aim to answer the question: does *ComPer* provides results that reflect users' acceptance of the recommender? By answering this question, we can verify the efficiency of *ComPer* by investigating its consistency with users' satisfaction. In this context, we invite researchers and practitioners to deploy *ComPer* in their evaluations and to share their observations.

VII. Conclusion and Future Work

The area of recommendation systems has recently witnessed almost a consensus that comparing RSs requires a common method of evaluation [18]. In this paper, we proposed the ***Com****prehensive* ***Per****formance* (***ComPer***) evaluation which is a new evaluation method that deals with RS as a human. The proposed approach is inspired by the idea that an RS is analogous to a human being, and so the recommendation process can be analogized to the human learning process. Therefore, we inferred a correlation matrix between Bloom's taxonomy (a well-known classification of educational learning objectives) and RSs evaluation dimensions. We demonstrate, through experimental evaluation, that *ComPer* provides comprehensive and clear evaluation results. The experiments also show that *ComPer*'s results are unbiased to a particular evaluation setting.

The rationale behind *ComPer* is that users hold different expectations from RSs. So, it is insufficient to evaluate these systems based on one dimension. On the other side, as the experimental results show, evaluating multiple dimensions separately makes it difficult to decide on the best algorithm. Thus, evaluating all the existed dimensions is irrational due to the abundant number of RS's dimensions. Therefore, we propose *ComPer* as a booting step toward unifying the evaluation process to provide a fair comparison between recommenders.

Despite the effort done so far, there are still different directions for future work. As it is mentioned previously, the considered evaluation dimensions have been selected based on their popularity in the literature (i.e., how many times each dimension has been evaluated). Although this factor is a reasonable indication of the importance of different dimensions, it is insufficient because popularity does not necessarily mean importance. In addition, not all dimensions are of the same importance for a particular application area. For example, evaluating the *Correctness* of a programming code recommender is more important than evaluating its *Diversity*. Also, the same dimension has different significance for different application areas. For instance, while *Robustness* is so important for e-commerce websites, it is less important for code recommendations. Hence, as future work, an investigation of the relationship between each dimension and RS application areas is required. Also, further effort is still required to find the relationship between different evaluation dimensions, and to investigate how they affect each other.

Regarding the experimental evaluation, although the presented results demonstrate the simplicity, thoroughness, and independency of *ComPer*, more validation aspects are still required. Hence, we will expand the evaluation methodology to consider other evaluation metrics for the considered dimensions, as well as more datasets. Also, since users' acceptance of the

recommendations is the main goal of RSs, we will study the level of consistency of *ComPer* results in users' perceived acceptance. To do so, a user study will be conducted. The result of this user study will be compared with *ComPer* results. Related to this point, we also invite researchers and businesses who run actual recommenders to validate *ComPer* results on real systems in order to generalize our findings.

REFERENCES

[1] Meyer, F., Françoise F., Fabrice C., and Eric G. "Toward A New Protocol to Evaluate Recommender Systems." arXiv preprint arXiv:1209.1983. (2012).

[2] Avazpour, I., Teerat P., Lars G., and John G. "Dimensions and Metrics For Evaluating Recommendation Systems." In Recommendation systems in software engineering, pp. 245-273. Springer, Berlin, Heidelberg. (2014).

[3] Bellogin, A., Pablo C., and Ivan C. "Precision-oriented Evaluation of Recommender Systems: an algorithmic comparison." In Proceedings of the fifth ACM conference on Recommender systems, pp. 333-336. ACM. (2011).

[4] Said, A., and Alejandro, B. "Comparative Recommender System Evaluation: Benchmarking Recommendation Frameworks". In: Proceedings of the 8th ACM Conference on Recommender systems. ACM. (2014)

[5] Del O., Félix H., and Elena G. "Evaluation of Recommender Systems: A New Approach." Expert Systems with Applications 35, no. 3: 790-804. (2008).

[6] Krathwohl, David R. "A Revision of Bloom's Taxonomy: An Overview." Theory into practice 41, no. 4: 212-218. (2002).

[7] Campos, P. G., Fernando D., and Iván C. "Time-Aware Recommender Systems: A Comprehensive Survey and Analysis of Existing Evaluation Protocols." User Modeling and User-Adapted Interaction 24, no. 1-2: 67-119. (2014).

[8] Arana-Llanes, J. Y., Rendón-Miranda, J. C., González-Serna, J. G., & Alejandres-Sánchez, H. O. "Design and User-Centered Evaluation of Recommender Systems for Mobile Devices-Methodology for User-Centered Evaluation of Context-Aware Recommender Systems". In International Conference on Computational Science and Computational Intelligence (CSCI), (Vol. 2, pp. 277-280). IEEE. (2014).

[9] Pu, P., Li C., and Rong H. "A User-Centric Evaluation Framework for Recommender Systems." In Proceedings of the fifth ACM conference on Recommender systems, pp. 157-164. ACM. (2011)

[10] Böhmer, M., Lyubomir G., and Antonio K. "Appfunnel: A Framework for Usage-Centric Evaluation of Recommender Systems That Suggest Mobile Applications." In Proceedings of the 2013 international conference on Intelligent user interfaces, pp. 267-276. ACM. (2013).

[11] Erdt, M., Florian J., Katja S., and Christoph R. "Investigating Crowdsourcing as an Evaluation Method For TEL Recommender Systems." In ECTEL meets ECSCW 2013: Workshop on Collaborative Technologies for Working and Learning, vol. 1047, pp. 25-29. (2013)

[12] Najjar, N. A., and David C. W. "Evaluating Group Recommendation Strategies in Memory-Based Collaborative Filtering." In Proceedings of the ACM recommender systems conference workshop on human decision making in recommender systems, pp. 43-51. New York, NY: ACM. (2011)

[13] Isinkaye, F. O., Folajimi Y. O., and Ojokoh B. A. "Recommendation Systems: Principles, Methods and Evaluation." Egyptian Informatics Journal 16, no. 3: 261-273. (2015).

[14] Guibing G., Jie Z., Zhu S., and Neil Y. "LibRec: A Java Library for Recommender Systems". in Posters, Demos, Late-breaking Results and Workshop Proceedings of the 23rd Conference on User Modelling, Adaptation and Personalization (UMAP). (2015)

[15] Hofmann, T., and Jan P. "Latent Class Models for Collaborative Filtering." In IJCAI, vol. 99, no. 1999. (1999)

[16] Hofmann, T. "Latent Semantic Models for Collaborative Filtering." ACM Transactions on Information Systems (TOIS) 22, no. 1: 89-115. (2004)

[17] Pang, Jiaona, Jun Guo, and Wei Zhang. "Using Multi-Objective Optimization to Solve the Long Tail Problem in Recommender System." Pacific-Asia Conference on Knowledge Discovery and Data Mining. Springer, Cham, (2019).

[18] Smyth, B., and Paul M. "Similarity vs. Diversity." In International Conference on Case-Based Reasoning, pp. 347-361. Springer, Berlin, Heidelberg. (2001)

[19] Said, A., Bellogín, A., and De V. A. "A Top-N Recommender System Evaluation Protocol Inspired By Deployed Systems". In LSRS Workshop at ACM RecSys. (2013).

[20] Chen, Mingang, and Pan Liu. "Performance Evaluation of Recommender Systems." International Journal of Performability Engineering 13, no. 8 (2017).

[21] Silveira, T., Zhang, M., Lin, X., Liu, Y. and Ma, S. "How Good Your Recommender System Is? A Survey on Evaluations in Recommendation". International Journal of Machine Learning and Cybernetics, 10(5), pp.813-831. (2019)

[22] Zheng, Y., Mobasher, B., & Burke, R. "Carskit: A Java-Based Context-Aware Recommendation Engine". In 2015 IEEE International Conference on Data Mining Workshop (ICDMW) (pp. 1668-1671). IEEE. (2015)

[23] Garimella, K., et al. "Reducing Controversy by Connecting Opposing Views." Proceedings of the Tenth ACM International Conference on Web Search and Data Mining. ACM. (2017)

[24] Christakopoulou, E., and George K. "Local Item-Item Models for Top-N Recommendation." Proceedings of the 10th ACM Conference on Recommender Systems. ACM. (2016)

[25] Cheng, W., Guisheng Y., Yuxin D., Hongbin D., and Wansong Z. "Collaborative Filtering Recommendation On Users' Interest Sequences." PloS one 11, no. 5 (2016): e0155739.

[26] Duwairi, R. M. "Machine Learning For Arabic Text Categorization." Journal of the American Society for Information Science and Technology 57.8: 1005-1010. (2006)

[27] Yu, T., Guo, J., Li, W., Wang, H. J., & Fan, L. "Recommendation With Diversity: An Adaptive Trust-Aware Model". Decision Support Systems, 113073. (2019)

[28] Bag, S., Abhijeet G., and Manoj K. T. "An Integrated Recommender System For Improved Accuracy and Aggregate Diversity." Computers & Industrial Engineering 130, p.p: 187-197. (2019)

[29] Navigli, R., and Federico M. "An Overview of Word And Sense Similarity." Natural Language Engineering. p.p : 1-22. (2019)

[30] Ge, M., Carla D., and Dietmar, J. "Beyond Accuracy: Evaluating Recommender Systems By Coverage And

Serendipity." In Proceedings of the fourth ACM conference on Recommender systems, pp. 257-260. ACM, (2010).
[31] Champiri, D., Adeleh A., and Salim S. "Meta-Analysis of Evaluation Methods and Metrics Used in Context-Aware Scholarly Recommender Systems." Knowledge and Information Systems p.p: 1-32. (2019)
[32] Sohail, S., Jamshed S., and Rashid A. "A Comprehensive Approach For the Evaluation of Recommender Systems Using Implicit Feedback." International Journal of Information Technology 11, no. 3 (2019): 549-567.
[33] Del O., Fđix H., and Elena G. "Evaluation of Recommender Systems: A New Approach." Expert Systems with Applications 35, no. 3: 790-804. (2008)

Authors' Profiles

Alaa Alslaity received his M.Sc. and bachelor's degrees in computer science from the Jordan University of Science and Technology, Jordan. Currently, Alslaity is a Ph.D. candidate and a Research Assistant at the School of Electrical Engineering and Computer Science, University of Ottawa, Canada. His research interests include Recommender System and its evaluation issues, and Persuasive Technology.

Thomas Tran received his Ph.D. in Computer Science from the University of Waterloo in 2004. He is currently a Full Professor at the School of Electrical Engineering and Computer Science, University of Ottawa. His research interests include Artificial Intelligence, Electronic Commerce, Intelligent Agents and Multi Agent Systems, Trust and Reputation Modeling, Reinforcement Learning, Recommender Systems, Knowledge-Based Systems, Architecture for Mobile E-Business, and Vehicular Ad-hoc Networks.

How to cite this paper: Alaa Alslaity, Thomas Tran, "ComPer: A Comprehensive Performance Evaluation Method for Recommender Systems", International Journal of Information Technology and Computer Science(IJITCS), Vol.11, No.12, pp.1-18, 2019. DOI: 10.5815/ijitcs.2019.12.01