IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 12, 2019, accepted December 3, 2019, date of publication December 5, 2019, date of current version December 26, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957881

A Big Data Semantic Driven Context Aware Recommendation Method for Question-Answer Items

JORGE CASTRO[1], RACIEL YERA TOLEDO[2], AHMAD A. ALZAHRANI[3], PEDRO J. SÁNCHEZ[1], MANUEL J. BARRANCO[1], AND LUIS MARTÍNEZ[1], (Senior Member, IEEE)

[1]Computer Science Department, University of Jaén, Jaén, Spain
[2]Knowledge Management Center, University of Ciego de Ávila, Ciego de Ávila 65100, Cuba
[3]Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

Corresponding author: Luis Martínez (martin@ujaen.es)

This work was supported by the Deanship of Scientific Research (DSR), King Abdulaziz University, Jeddah, under Grant DF-678-611-1441.

ABSTRACT Content-Based recommender systems (CB) filter relevant items to users in overloaded search spaces using information about their preferences. However, classical CB scheme is mainly based on matching between items descriptions and user profile, without considering that context may influence user preferences. Therefore, it cannot achieve high accuracy on user preference prediction. This paper aims to handle context-awareness (CA) to improve quality of recommendation taking contextual information as the trend in current trend interest, in which a stream of status updates can be analyzed to model the context. It proposes a novel CA-CB approach that recommends question/answer items by considering context awareness based on topic detection within current trend interest. A case study and related experiments were developed in the big data framework Spark to show that the context integration benefits recommendation performance.

INDEX TERMS Content-based recommender system, context-awareness, user profile contextualization, map-reduce, big data.

I. INTRODUCTION

The increasing amount of information available in World Wide Web scenarios, such as e-commerce, affects users' satisfaction when they search for items that meet their interests. This situation originates that users need to put significant effort for finding relevant pieces of information for them. In some scenarios it might not be possible for users to explore items in order to select the most suitable one. Hence, the information overload problem impacts users satisfaction. Recommender systems have been a powerful tool for alleviating information overload in large search spaces [1], [2]. Recommender systems have been proved to be successful in several domains, such as e-business [3], e-learning [4], [5], e-tourism [6], [7], e-commerce [8], web pages [9], [10] and financial investment [11], among others.

Several approaches have been explored for alleviating information overload problem with recommender systems. The most widespread ones are collaborative filtering (CF) [12] and Content Based (CB) [13]. The main difference between them is that CF focuses on users' interaction with items, i.e., user preferences, while CB focuses on the analysis of items descriptions, i.e., item content. Therefore, the performance of these recommendation approaches is subject to the quality and amount of available information of both types. In addition to these successful strategies, other approaches have been proposed, such as knowledge-based recommender systems that focus on employing expert information over the recommendation domain through ontologies [14], among others [4], [15], or social network recommender systems that use links between users to improve the recommendation [16]. Recent research lines also focus on integrating contextual information [17] or providing recommendations targeted to groups of users [18], [19]. Specifically, context-aware recommendation (CA) [20] has emerged as a relevant tool for leveraging the value of recommendations by exploiting context information with the goal of recommending items that are really relevant to changing user needs.

The associate editor coordinating the review of this manuscript and approving it for publication was Renato Ferrero.

In this paper, we propose a novel context-awareness content-based (CA-CB) recommendation approach whose main novelty is the introduction of context awareness based on topic detection within current trend interest. Furthermore, this approach introduces a general methodology to combine two different information sources (the content-based and the context-aware-related) in order to boost the recommendation performance. Specifically, the application domain is question-answering items (QA items), given that QA items have a strong component of textual information for both explaining the question and answering it [21]. We select the CB recommendation paradigm because even though CB suffers from user cold start because they need some input from user preferences and lack of diversity in recommendations [22], CB has demonstrated their utility when new items are introduced in the system, i.e., in scenarios with strong item cold-start [23]. This feature makes the application of CB approaches interesting in domains in which new items are constantly introduced, such as web pages or news. QA recommendation shares the features to apply CB with textual descriptions, hence, we focus on it. We remark that across this research work a QA item is viewed as a pair question-answer, being such pair the textual item to be recommended. Therefore, our current problem is different to the problem of finding an answer to a given question [24].

Within QA domain recommendation, it is interesting to focus on recommending answered questions that are in the target user's area of interest, and that are also relevant regarding the current trend interest. Therefore, it seems convenient and necessary to explore Context-Aware CB (CA-CB), which integrates contextual information to the content-based recommendation. As key works in this direction, Musto *et al.* [25] take as base the movie recommendation domain, and consider context as a weighting factor that influences the recommendation score of a user for a certain item, and Son *et al.* [26] consider the location context and define user by the articles read in the past along with his/her location, working over the movie recommendation domain. De Pessemier *et al.* [27] consider recommendations in mobile devices as very suitable to integrate context-awareness, and use devices sensors and time of the day to deliver contextualized news recommendations.

In QA recommendation, SeaHawk [28] and Prompter [29] provide CA-CB that supports programmers to complete issues and bugs using query completion and recommends StackOverflow questions, where the context is the specific part of the source code from which the recommendations are requested. In this direction, Libra [30] integrates recommendations but it also considers, in addition to context extracted from the integrated development environment, resources opened by the user such as URLs or documents to better understand his/her context. Other works consider contextual interest as current buzzwords to deliver currently relevant recommendations in e-commerce scenarios [31]. As it can be seen, there is no previous approach that focuses on context-aware recommendation regarding current trend interest in the QA domain.

In this paper, we propose a novel CA-CB approach for recommending QA items and introducing context awareness based on topic detection within current trend interest. Recent researches [32] have highlighted the immediacy of microblogging services such as Twitter, where users share short sentences or fragments of news. In this proposal, the context is extracted from microblogging systems to characterize current trends in current trend interest. The usage of such a context mainly helps recommending QA items related to topics of interest and indirectly overcomes the overfitting problem. However, the context extracted in this way is often noisy or several topics are mixed. With this regard, we propose to cluster context to identify the topics that are being discussed, and after, the most suitable context topic to target user's preferences is selected to build a contextualized user profile that combines preferences and context. This way, the proposal provides contextualized recommendations that are also adjusted to users' individual interests. Moreover, QA domain has large-scale data and microblogging systems generating data at high rate. In this regard, MapReduce approach within big data has been proved to be effective in high volume and high rate data scenarios [33], [34], therefore, our proposal is developed within Spark, a distributed framework for big data that takes advantage of in-memory operations.

The main novel contributions of this study are:

- A suitable way to study status updates, i.e. a user-provided free text, in the QA recommendation scenario to provide recommendations tailored to both the user preferences and current trend interest.
- Introduction of personalised contextualization of user preference profiles to better integrate current trend interest as the context of the recommendation.
- A proposal that applies semantic analysis of QA domain, dealing with mixture of topics in current trend interest and providing personalised context-awareness in the recommendation.
- A case study and experimentation developed within a big data framework that validates the proposal and determines that integration of contextual information extracted from current trend interest improves QA recommendation.
- Overall, a global methodology to integrate two different information sources coming from a content-based and a context-aware scenario, in order to build an integrated recommendation approach.

The remaining of this paper is structured as follows. Section II provides a background of CB, CA-CB and QA recommendation. Section III introduces in further detail our proposal of CA-CB for QA recommendation. Section IV shows a case study performed to evaluate the proposal and discuss the findings, and includes some brief references to

MapReduce and Spark. Finally, Section V concludes the paper.

II. PRELIMINARIES

This section provides the required background for the current research, including basics about CB, related works in content based recommendation with context awareness also in QA recommendation and eventually some references to QA recommendation.

A. CONTENT-BASED RECOMMENDER SYSTEMS

An accurate definition of recommender system, given by Burke [35], is ''any system that produces individualized recommendations as output or has the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options''. Within recommender systems, various techniques can be distinguished based on the knowledge source [27]: demographic, knowledge-based, community-based, content-based, collaborative, and hybrid recommendations. Among them, we focus on CB.

The various CB approaches can be classified regarding the item representation. Here we focus on those CB that use a vector space modeling to represent items. With this regard there are CB with (i) feature-based representations, where items are usually stored in a database table where rows are items and columns are the item features [27], [36], [37]; and (ii) free-text representation, where there is a natural language piece of text that describes the item [38].

The TFIDF approach [39] is usually applied when dealing with free-text representation items. In TFIDF, the unstructured data is converted in structured data stemming words [40] to keep their root. This process reduces the number of components of documents unifying words such as computer, compute and computing, which are different forms that share meaning. After that, for each document, a vector of weights of each term is generated multyplying the $tf_{t,d}$ by the idf_t [41], to consider the importance of the term on the document:

$$profile_d^{tfidf} = \{tf_{t,d} * idf_t \quad s.t. \quad t \in d\} \tag{1}$$

$$idf_t = -\log\left(\frac{|N|}{|N_t|}\right) \tag{2}$$

where $tf_{t,d}$ is the number of occurrences of term t in document d, N is the set of all documents and N_t is the set of documents that contain the term t at least once.

At this point the system contains a vector space model of items. User profiles can be generated by aggregating the profiles of the items that they liked in the past [36]. The recommendation is computed comparing user profiles with item profiles with the cosine correlation and the closest ones are recommended.

While TFIDF method is effective, it cannot deal with polisemy or synonym words. In order to overcome this issue, Latent Semantic Analysis (LSA) is applied [13], [42]. In LSA, the term-document matrix is factorized with Singular Value Decomposition (SVD) to reduce it to orthogonal dimensions and keep the f most relevant singular values (see Figure 1).

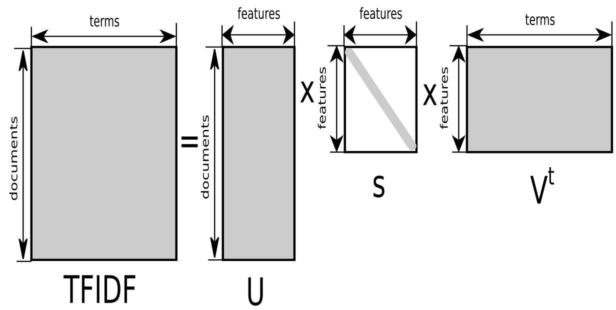

FIGURE 1. Factorization of TFIDF matrix with singular value decomposition. Note that *s* is a diagonal matrix with the singular values sorted in descending order.

$$TFIDF_{(|D|\times|T|)} = U_{(|D|\times f)} * s_{(f)} * V^t_{(f\times|T|)} \tag{3}$$

This way, a reduced feature space is defined, which properly manages noise and redundancy of terms. User profiles are generated from this feature-space definition through a linear combination of document profiles that they liked [43]. Then, recommendations are generated comparing user and document profiles with cosine correlation coefficient.

B. CONTEXT-AWARE RECOMMENDER SYSTEMS

In addition to the information traditionally used by recommender systems, as noted by Burke [35], other sources of information can be considered in the recommendation, such as the context in which the recommendation is received by users. Ricci [44] stated that the conditions or circumstances in which the recommendation is delivered significantly affect users' decision behavior. Therefore, the consideration of users' context is key to provide interesting recommendations.

With this regard, the various context-aware recommendation approaches can be classified into three classes [45]:

- Pre-filtering: The system selects and uses only the feedback gathered in the same context in which the recommendation is delivered to the user.
- Post-filtering: The recommendations are generated first without considering contextual information. After that, the item predictions are modified regarding the specific context of the users, possibly filtering out some items.
- Contextual modeling: The contextual information is directly integrated in the model that is used to recommend.

In contrast to pre- and post-filtering approaches, the research community have developed fewer research works focused on contextual modelling. A recent survey on context-aware recommendation developed by Villegas *et al.* [20] identified only few research works that combine content-based and context-aware recommendation, and are also based on contextual modelling.

Here, Hong *et al.* [46] propose a framework that analyses the relationship between user profiles and services under the same context situation to infer user preference rules using the decision tree algorithm, being points of interest and indoor shopping the recommendation domain. Musto *et al.* [25] take as base the movie recommendation domain, and consider context as a weighting factor that influences the recommendation score of a user for a certain item. Son *et al.* [26] consider the location context and define user by the articles read in the past along with his/her location. Similarly, Fang *et al.* [47] focuses on mobile recommender systems for assisting indoor shopping by considering location-context. On the other hand, Wang *et al.* [48] work over the song recommendation domain, and formulates the context-aware recommendation of songs as a two-step process: i) infers the user's current situation category given some contextual features sensed from a mobile phone, and ii) finds a song that matches the given situation. Over the same domain, in Shin *et al.* [49] the context refers to the time at which the user listens to a song, and such information is integrated into the recommendation model; and in Cheng and Shen [50] the authors implement a recommendation model where a set of latent topics is used to associate music content with a user's music preferences under certain location. Finally, Kuo *et al.* [51] also considers context as a weighting factor that influences the recommendation score in a location-based recommender system. In addition, De Pessemier *et al.* [27] consider recommendations in mobile devices as appropriate to integrate context-awareness, and uses devices sensors and time of the day to deliver contextualized news recommendations.

The overview of these previous works concludes that even though there is a common research scheme on using context as a weighting factor to adapt the recommendation results according to the corresponding scenario; we also identified that most of the research done is centered on the use of the location context, which suggests that this research branch needs further developments toward more generalized proposals.

The current paper is focused on filling this gap by presenting a content-based context-aware recommendation approach based on contextual modeling, which in contrast to the previous researches, considers the knowledge extraction for a specific scenario (Twitter) to be used in the recommendation domain, in this case the QA items recommendation.

C. RELATED WORKS ON QA RECOMMENDATIONS

The recommendation of QA items has been a hot topic in the last few years in the research community [24], [52]–[54]. The literature has identified two main streams of QA items recommendation. The first one is focused on helping users to find an answer on complex, subjective, or context-dependent questions. The second one is centered on recommending relevant question-answer pairs to the users, containing contents that they could be interested in.

Srba and Bielikova [24] have developed a comprehensive survey on community question answering where they reviewed 265 papers published between 2005 and 2014, considering research works belonging to both kinds of QA approaches. Such survey identifies the question lifecycle as question creation, question answering, question closing, and question search. The revised papers are grouped by: 1) exploratory studies, that concern with analyses of data which are recorded during the question answering process 2) researches focused on content and user modelling for managing various characteristics of users, questions and corresponding answers to derive high-level attributes from low-level question answering interactions, and 3) adaptive support approaches, which build on results from exploratory and content/user modeling studies in order to directly influence users' collaboration.

The analysis of the recent research works confirms that most of the research efforts are focused on content modeling and user modeling, as it was also identified by Srba and Bielikova [24]. In addition, there are few efforts focused on the second kind of QA recommendation works (e.g. recommending relevant question-answer pairs to the users), and most of them are focused on finding semantically related questions that reflect different aspects of the user query and provide supplementary information [55]. In this group, Wang *et al.* [56] also extended a language model with question popularity prediction to provide better question recommendations, and Zhou *et al.* [57] proposed a topic-enhanced translation-based language model which incorporates also answer information.

In software engineering domain, the systems SeaHawk [28] and Prompter [29] provide CA-CB recommendations that support programmers to complete issues and bugs by using query completion and recommends StackOverflow questions, in which the context is the specific part of the source code from which the recommendations are requested. In the same domain, Libra [30] integrates recommendations but it also considers, in addition to context extracted from the integrated development environment, resources opened by the user such as URLs or documents to better understand his/her context. Other works consider contextual interest as current buzzwords to deliver currently relevant recommendations in e-commerce scenarios [31].

Therefore, it is worthy to note the lack of the management of the users preferences over the question-answer pairs, as a relevant source to be employed in QA recommendation. The current paper aims at filling this gap by proposing a context-aware recommendation model that is able of recommending useful question-answer pairs to the final users, taking into account its preferences as well as contextual information based on current information trends.

III. SEMANTIC MODEL FOR RECOMMENDING QUESTIONS WITH CONTEXT AWARENESS BASED ON TOPIC DETECTION IN CURRENT TREND INTEREST

Here, a novel proposal, LSAContextCluster, for recommendation based on CA-CB is introduced. Recommendations might need to be targeted to specific contexts, e.g., when a system delivers recommendations of QA items in the history domain and currently people are posting about Colombus Day, then the system should promote QA items related to the

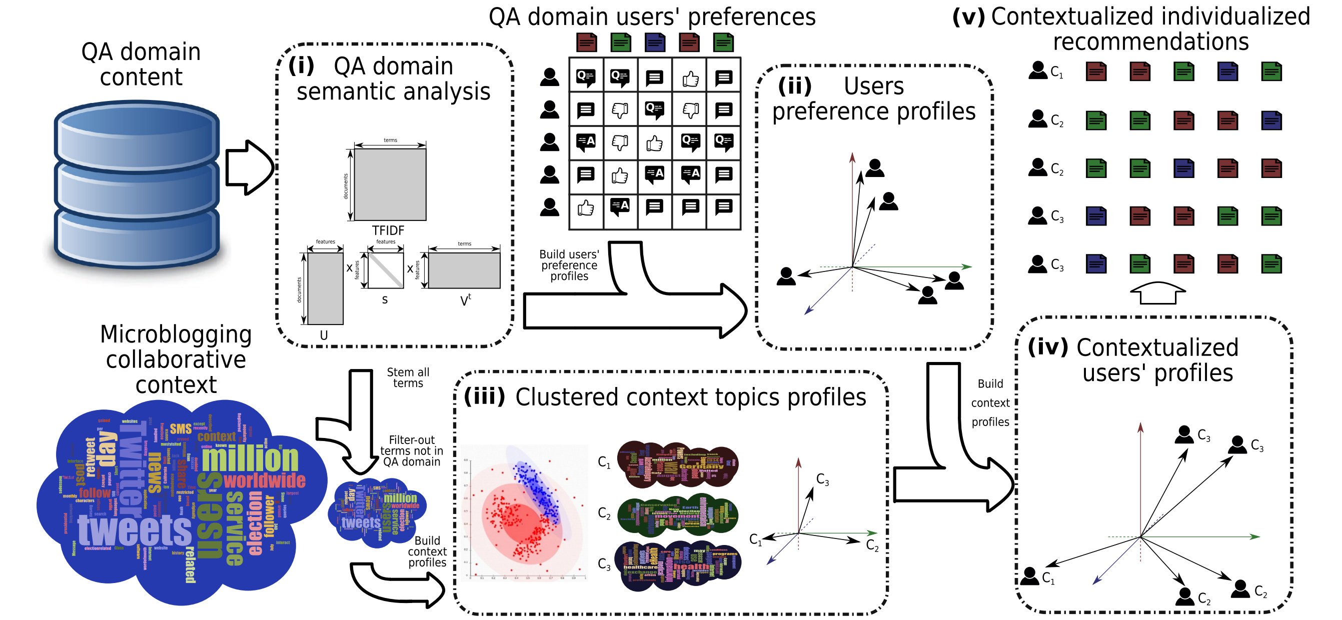

FIGURE 2. General scheme of the proposal.

discovery of the Americas. In this kind of cases, it is possible to modify user profiles to include contextual information in such a way that later recommendations are both targeted to user preferences and current context.

The proposed model fits into the CA approach of contextual modeling, because it integrates contextual information in the model built by the recommender system. The general scheme of the proposal, LSAContextCluster, is depicted in Figure 2, and it is composed of five phases:

1) QA domain semantic analysis: It applies LSA to reduce the dimensionality of the term-document matrix.
2) Build user's preference profile: It analyses users' preferences and generates a profile for each of them based on the profiles of the document he/she liked in the past.
3) Build context model: It analyses the context, which consists on a stream of status updates within a given time frame, applies clustering to separate the various topics that the context contains, and generates feature-space profiles for each context topic.
4) Contextualize user profiles: It selects the context topic most suitable to target user's preferences and combines the preference-based user profile with the context topic profile to generate the contextualized user profile.
5) Prediction: It compares the document profiles and the contextualized user profile to recommend.

For a further detailed description, a pseudo-code description of the proposal (See Algorithm 1) and the notation used in the proposal (See Table 1) are provided.

Before the proposal presentation, it is also necessary to explain some important concepts that have been already referred at the mentioned phases:

- *QA item*: A QA item is viewed as a pair question-answer, being such pair the textual item to be recommended.
- *Term*: A simple word inside a document represented in this case by a QA item.
- *Feature*: In content-based recommendation, features are used to characterize items, and as the main criteria for recommendation generation. In this work, the LSA approach is used for identifying the set of features (i.e. the feature space) most relevant to the current set of QA items (i.e. documents). With this aim, it takes as input the TFIDF matrix of such set of QA items containing their terms
- *Document profile*: The QA items (i.e. documents) are represented through the vector $profile_d^{LSA}$, associated to the feature space obtained through the LSA method.
- *User profile*: The use profile in the feature space is represented by a vector $profile_u^{LSA}$, built by the combination of all the QA items $profile_d^{LSA}$ where user has expressed or not interest in a document either creating, commenting, or voting it.
- *Status update*: A user-provided free text associated to some information source (in this work Twitter).
- *Context*: In this research, the context is composed of a set of status updates in a given time window. Here this set of updates is processed in a similar way to the document profile, to obtain the context model $profile_c^{LSA}$ to be used in the subsequent phases.
- *Topic*: The feature vectors representing terms associated to the context $profile_c^{LSA}$, are clustered for grouping them into the most related ones. A context topic $profile_{c_i}^{LSA}$ can be then identified as one of each cluster.
- *Contextualized user profile*: It is the profile resulting from the combination of the user profile $profile_u^{LSA}$, and the profile $profile_{c_i}^{LSA}$ associated to a selected context topic.

Algorithm 1 Pseudocode

1: STEMMING: Reducing terms to their roots
1.1: For each document d in D
1.2: For each word w in d
1.3: $w^* \leftarrow stemming(w)$
1.4: $terms_d.add(w^*)$
1.5: $T.add(w^*)$
2: TFIDF method
2.1: For each document d in D
2.2: For each term t in $terms_d$
2.3: $tfidf_{t,d} \leftarrow tf_{t,d} * idf_t$
2.4: $profile_d^{tfidf}.add(tfidf_{t,d})$
2.5: $TFIDF.add(profile_d^{tfidf})$
3: LSA method: SVD matrix factorizing
3.1: $(U, s, V) \leftarrow SVD(TFIDF)$
$profile_d^{LSA} = \{u_{d,1} \dots u_{d,f}\}$
$profile_t^{LSA} = \{v_{t,1} \dots v_{t,f}\}$
4: USER Profile
4.1: For each feature x in *feature-space*
4.2: For each document d in R_u
4.3: $profile_x \leftarrow profile_x + profile_{d,x}^{LSA}$
4.4: $profile_u^{LSA}.add(profile_x)$
5: TWITTER context
5.1: For each Twitter status update d in MCC
5.2: For each word w in d
5.3: $w^* \leftarrow stemming(w)$
5.4: *if* $w^* \in T$ *then* $terms_{MCC}.add(w^*)$
6: CLUSTERING on Twitter terms
6.1: $\{c_i\} \leftarrow c\_means\_clustering(terms_{MCC})$
6.2: For each cluster c_i
6.3: For each feature x in *feature-space*
6.4: For each term t in c_i
6.3: $profile_x \leftarrow profile_x + profile_{t,x}^{LSA}$
6.4: $profile_{c_i}^{LSA}.add(profile_x)$
7: CONTEXTUALIZATION user profile
7.1: $c_i \leftarrow argmax_{c_j} cosine(profile_u^{LSA}, profile_{c_j}^{LSA})$
7.2: $profile_{C,u}^{LSA} \leftarrow \alpha * profile_u^{LSA} + (1-\alpha) * profile_{c_i}^{LSA}$
8: PREDICTION
8.1: $p_{u,d} \leftarrow profile_{C,u}^{LSA} * (profile_d^{LSA})^T$

Figure 2 clearly shows that our approach uses the content-based recommendation technique to perform its final goal. The content-based recommendation technique [2], [58], [59] comprises three steps, which are 1) Item profiling, 2) User profiling, and 3) Matching user with item profiles. In this research work the item profiling is implemented in two stages through the QA domain semantic analysis (Phase 1) as well as in the clustered context topic profiling (Phase 3). Meanwhile, the user profiling is also composed in two stages by the user preference profiling (Phase 2) and the contextualized user profiling (Phase 4). At last, the profile matching is done in the Phase 5, at the contextualized individualized recommendation. As it was pointed out in the Introduction section, the most relevant issue of our proposal is the incorporation of the contextual information across the three steps of the traditional content-based recommendation technique.

TABLE 1. Notation.

Notation	Meaning
D	Set of documents $\{d\}$ which contains questions and its related answers.
$terms_d$	Set of terms $\{t\}$ that appear in a document d, after stemming (to reduce to their roots).
T	Set of stemmed terms in the domain, $\bigcup_d terms_d$
$profile_d^{tfidf}$	Profile of a document d in the set $terms_d$, after applying TFIDF method.
$TFIDF$ matrix	Rows in this matrix are $profile_d^{tfidf}$, for each d. After factorizing $TFIDF$, applying LSA method matrices U, s and V appear.
$feature\text{-}space$	After applying LSA method to TFIDF matrix a new space of f features appears, where $f << \|T\|$, which will be used to build profiles.
U matrix	Rows in this matrix are $profile_d^{LSA}$, for each d
s matrix	Diagonal in this matrix contains singular values after factorizing TFIDF using the LSA method.
V matrix	Rows in this matrix are $profile_t^{LSA}$, for each $t \in T$
$profile_d^{LSA}$	Profile of a document d in the feature-space
$profile_t^{LSA}$	Profile of a term t in the feature-space
R	Set of ratings on documents of D
$r_{u,d}$	Rating of user u for document d
R_u	Subset of documents in D such that $r_{u,d}$ belong to R
$profile_u^{LSA}$	Profile of a user u in the feature-space .
MCC	"Microblogging collaborative context" is the set of Twitter status updates, used as context in the proposal.
$terms_{MCC}$	Set of terms $\{t\} \subset T$ that appear in MCC, after stemming.
c_i	A set $\{c_i\}$ of clusters results after applying a fuzzy c-means clustering algorithm to $terms_{MCC}$.
$profile_{c_i}^{LSA}$	Profile of a cluster c_i in the feature-space.
$profile_{C,u}^{LSA}$	Contextualized profile of a user u in the feature-space

A. QA DOMAIN SEMANTIC ANALYSIS

Our proposal assumes that the QA dataset contains textual information of the question and their related answers. In this proposal we consider the questions together with all their answers as the document, and the words used in their text as the terms. The terms are stemmed using the Porter Stemmer algorithm [40]. Once terms are stemmed, the TFIDF document profiles $profile_d^{TFIDF}$ are built according to Eq. 1.

Once the TFIDF document profiles are built, LSA [13] is performed to reduce the dimensionality of the matrix. LSA is proven to be effective through the description of both documents and terms in a feature space with a reduced number of features. Therefore, the aim of this step is to decompose the initial word-document matrix in a word-features matrix U, a singular value vector s, and a document-features matrix V (see Eq. 3).

An approximated factorization of the TFIDF matrix is performed with Singular Value Decomposition [60], which allows to reduce the dimensionality of the original matrix keeping the top f singular values of the original matrix. Hence the two matrixes, U and V, provide the profiles of both terms and documents in the feature space, which compose the QA domain semantic model:

$$profile_d^{LSA} = \{u_{d,1}, \dots, u_{d,f}\} \tag{4}$$

$$profile_t^{LSA} = \{v_{t,1}, \dots, v_{t,f}\} \tag{5}$$

TABLE 2. Users' preferences over items, the rating matrix.

	d_1	$\dots$	d_k	$\dots$	d_n
u_1	r_{u_1,d_1}	$\dots$	r_{u_1,d_k}	$\dots$	r_{u_1,d_n}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
u_j	r_{u_j,d_1}	$\dots$	r_{u_j,d_k}	$\dots$	r_{u_j,d_n}
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
u_m	r_{u_m,d_1}	$\dots$	r_{u_m,d_k}	$\dots$	r_{u_m,d_n}

B. BUILDING USER PREFERENCE PROFILE

At this point LSAContextCluster has built a model with terms and documents profiles. In order to provide personalized recommendations to users, it is needed to build user profiles in the same feature-space. LSAContextCluster holds a binary matrix that states whether a given user has expressed or not interest in a document (see Table 2) either creating, commenting, or voting it. In this table, the set of documents that user u has expressed interest in is defined as:

$$R_u = \{d \quad s.t. \quad r_{u,d} \in R\} \tag{6}$$

This way, the user's profile is built upon the profiles of the documents that belong to R_u, and describes the user's preferences in terms of the feature space:

$$profile_u^{LSA} = \sum_{d \in R_u} profile_d^{LSA} = \{\sum_{d \in R_u} profile_{d,1}^{LSA}, \dots, \sum_{d \in R_u} profile_{d,f}^{LSA}\} \tag{7}$$

The user's profile does not need to be normalised because it is later compared with other profiles using cosine correlation coefficient, which only considers the angle of the vectors being compared.

C. CONTEXT MODEL BUILDING

To include contextual information in the recommendation process, it is needed to build the context model. In this proposal, we aim to promote questions that are relevant regarding current happenings. With this regard, our proposal uses status updates from microblogging services, such as Twitter, as the source of current trend interest. Formally, a status update consists of a free text input generated by a user with certain timestamp, among other meta-data. Analyzing these status updates, LSAContextCluster generates a model of the context that is later used to modify the user profile. The scheme of the context model building phase is depicted in Fig. 3.

The context is composed of the status updates that were generated in a given time window, which is set to 24 hours in this proposal, although it could be modified to adjust the sensitivity of the context model. First, all terms of the status updates of the current context are stemmed. After that, from all the terms that the context contains, LSAContextCluster filters out the terms that do not appear in the QA semantic model generated in phase one (see Section III-A).

Given that the context is composed of several status updates, there can be a mixture of topics. To determine the context topics, the proposal performs a fuzzy clustering of the terms used in the context. Hence, fuzzy c-means clustering

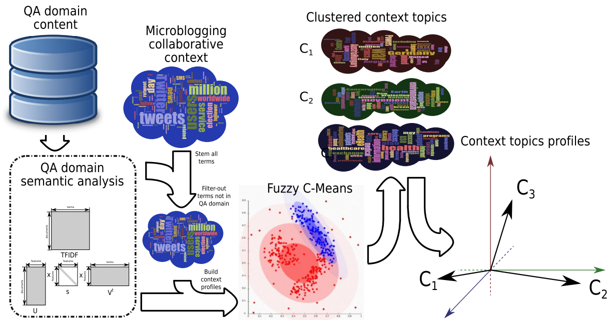

FIGURE 3. Context model building phase.

algorithm [61] groups the terms using their feature vector $profile_t^{LSA}$ as the term definition. The distance among terms used in the clustering is based on cosine correlation coefficient. The result is a set of clusters where each cluster c_i defines a context topic.

Once the terms of current context are grouped in context topics, the proposal generates a context profile for each topic combining the profiles of the terms that are included in each cluster. Therefore, LSAContextCluster builds a profile for each context topic using the feature representation of each term from the QA domain (see Eq 8). At the end of this phase, LSAContextCluster has generated a model of the context composed of several context profiles, one for each context topic detected by the clustering.

$$profile_{c_i}^{LSA} = \sum_{t \in c_i} profile_t^{LSA} = \{\sum_{t \in c_i} profile_{t,1}^{LSA}, \dots, \sum_{t \in c_i} profile_{t,f}^{LSA}\} \tag{8}$$

D. USER PROFILE CONTEXTUALIZATION

In this step, the target user's preference profile is combined with the context model to provide contextualized personalized recommendations. To do so, in a personalized way, from all the context topic profiles that the context model contains, LSAContextCluster selects the most similar to the user's preference profile. This way, the context topic c_i that has the greatest cosine coefficient with the target user preferences is used to modify his/her profile, hence, the contextualization of user profiles is personalized to user preferences.

$$\underset{c_i}{\text{argmax}} \quad cosine(profile_u^{LSA}, profile_{c_i}^{LSA}) \tag{9}$$

After this selection, the profile of the selected context topic c_i and the user's preference profile are combined to obtain the contextualized user profile. With this regard, the convex combination is applied, which allows to perform a weighted combination, regulated by α parameter. A greater value of α gives more importance to the user's preference profile over the profile of the selected context topic in the contextualized user profile. A lower value of α gives more importance to the profile of the selected context topic for the recommendation generation.

$$profile_{C,u}^{LSA} = \alpha * profile_u^{LSA} + (1 - \alpha) * profile_{c_i}^{LSA} \tag{10}$$

Future works will explore more sophisticated approaches such as matrix factorization, for integrating the contextual information in the initial content-based recommendation model. At this paper we have not considered this issue because our primary aim is to show that the direct integration of the context into the initial model, can directly influence the recommendation performance. In forthcoming works, we will be focused on enhancing such integration for optimizing the recommendation accuracy.

E. PREDICTION

Once the contextualized user profile is built, we can produce a prediction of the suitability for a given item regarding the profile. The recommendation is a list of documents sorted by $p_{u,d}$:

$$p_{u,d} = profile_{C,u}^{LSA} * (profile_d^{LSA})^T \tag{11}$$

IV. CASE STUDY AND EXPERIMENT

To evaluate the proposal, we have performed an experiment that simulates the recommendation of QA items in various contexts. The remainder of the section is structured as follows. First, the settings of the experiment are described. The datasets and methods for processing them are further detailed. After that, the evaluation measures are commented. Lastly, the results are analyzed. Overall, this section is a case study on the use of our proposal over the StackExchange QA dataset, and Tweeter as microblogging service for representing contexts.

A. EXPERIMENTAL PROCEDURE

In these experiments we compared several CB approaches based on LSA with contextual information. In order to do the experiment, the procedure proposed by Sarwar *et al.* [62] was performed with modifications to consider contextual information in the experiment:

- Split the dataset in training and test using the 5-cross fold validation as a splitting technique [63]–[65], over the QA items gathered from the StackExchange dataset through the procedure explain below at Section IV-C. We pointing out that 5-cross fold validation is a very popular procedure for evaluating recommender systems [65]. In this procedure, the data set is divided into k folds. One of the folds is used for testing the model and the remaining $k - 1$ folds are used for training. The cross validation process is repeated k times with each of the k subsamples used exactly once as test data and the remaining ones for training. Finally the average performance of the k evaluations, reached after the last step of this procedure, is then reported [65].
- Build the model with each training data obtained in the previous step, composed of QA items obtained from the StackExchange dataset.
- Build the profile of each user including contextual information if applicable. This contextual information is composed of tweets and was gathered according to the procedure explained below at Section IV-C.
- Recommend QA items to each user based on their profile and the model.
- Evaluate recommendations with the test set associated to the corresponding training data already referred at Steps 1 and 2, and considering NDCG [65] as evaluation metric. Further details on NDCG are described at Section IV-D.

This whole procedure was also repeated 20 times to avoid any bias in the splitting procedure. Moreover, various contexts were considered, which are detailed in Section IV-C. The experimental procedure was developed within the MapReduce approach [66] through the big data framework Spark, which provides abstractions for distributed computations and is able to process large amounts of data.

Apache Spark [67], introduced as part of the Hadoop Ecosystem, offers to the user a set of in-memory primitives that complement the MapReduce ones and that is suitable for iterative tasks. It is based on Resilient Distributed Datasets (RDDs), a structure that stores data in such a way that later computations can be easily parallelized in distributed machines. RDDs allow to cache or redistribute intermediate results, which enables the design of data processing pipelines.

Within Spark we use two libraries: MLlib and Spark Streaming. MLlib is a scalable machine learning library [68] that was built to take advantage of Spark suitability for iterative tasks and provides several machine learning techniques for classification, optimization, and data preprocessing, among others. Specifically, we use the tools that MLlib provides for regression and clustering. Spark Streaming [69] provides an scalable way to manage data produced at high rates, which allows us to handle the data provided by microblogging systems and compute the context model.

B. METHODS COMPARED

We compared several ways for integrating contextual information in QA recommendation. For the sake of fair comparison, the number of features was fixed in all models, and 30 features were considered in LSA step.

- No clustering (LSAContext): The terms are not separated in clusters, therefore the context profile is unique. There is a single profile of the context that is built combining the profiles of the terms that are included in the context.

$$profile_C = \sum_{t \in C} profile_t \tag{12}$$

- Weighted by membership (LSAContextClusterFuzzy): The cluster profiles are built combining the feature vector of each term weighted by the membership value of the term to the cluster:

$$profile_{c_i} = \sum_{t \in c_i} \mu_{t,c_i} * profile_t \tag{13}$$

  where μ_{t,c_i} is the membership of term t to cluster c_i.
- Max membership (LSAContextClusterMax): The terms are used only in the cluster to whom they have the

TABLE 3. Main features of some StackExchange sites.

	3dprinting	academia	android	history
Users	4025	48448	119810	13433
Questions	597	57967	41423	6127
Answers	1135	16737	49985	12212
Comments	2754	40319	123291	53510
Votes	7860	138416	359710	162809
Ratings	2458	119082	109644	38082
Sparsity	0.99898	0.99996	0.99998	0.99954

highest membership value:

$$profile_{c_i} = \sum_{t \in T} \mu_{t,c_i}^{max} * profile_t \tag{14}$$

where μ_{t,c_i}^{max} is one if μ_{t,c_i} is the maximum membership across clusters, and zero otherwise.

Moreover, in order to adjust the weight of preference profile over context profile, the methods compared have the parameter α, already specified at Section III.D. In the experiment we explored several values for it, here, to make the results clearer, we show only $\alpha \in [0.90, 0.99]$ with increments of 0.01.

C. DATASETS

In the experiment there are two sources of data: The QA domain and the contextual information.

The QA domain used in the experiment is the StackExchange dataset.[1] This dataset consists of the database dump of each site in the StackExchange ecosystem. Context influence may be vary across sites. In this case study, we focus on the StackExchange site devoted to 3D printing,[2] given the current interest on such a topic. Some stats of the dataset are detailed in Table 3.

Regarding the contextual dataset, a set of interesting keywords is defined based on the aim of the proposal for contextualizing recommendations. Given that the proposal focuses on selecting currently hot topics, we have selected the terms news, current and situation. From these seed terms, we extracted a dataset of tweets that contain any of these words from Twitter. The stats of the dataset extracted is depicted in Figure 4.

D. EVALUATION MEASURES

Usually, measures to evaluate the prediction errors in terms of rating deviation are used. However, the methods being compared do not provide a rating prediction, but a value that expresses the suitability of items regarding the user profile. Therefore, the measures that can be used are information retrieval ones, such as precision and recall. Researchers have remarked that, although they are useful, they are not sensible to the sorting of the items that the recommender systems does [70]. In order to consider the quality of the sorting, the Normalized Discounted Cumulative Gain (NDCG) is used. NDCG [71] is a measure from information retrieval,

[1] http://data.stackexchange.com/

[2] https://3dprinting.stackexchange.com/

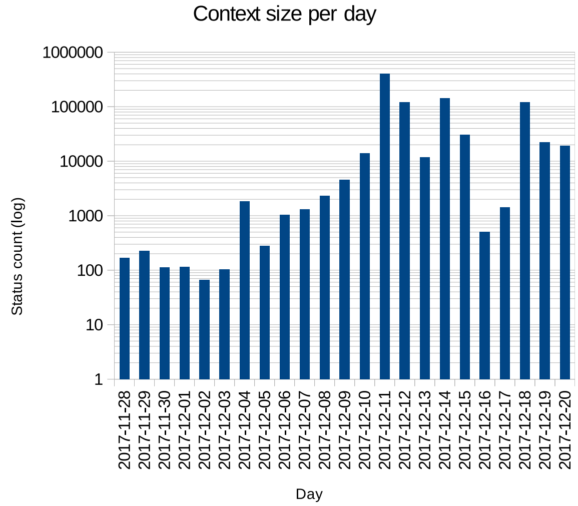

FIGURE 4. Contextual dataset used in the experiment, where each day has a different status count.

where positions are discounted logarithmically. It assumes that highly relevant documents are more useful when appearing earlier in a result list, and that highly relevant documents are more useful than marginally relevant documents, which are in turn more useful than non-relevant documents.

NDCG at first depends on the Discounted Cumulative Gain (DCG), which premise is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result. DCG is formalized as:

$$DCG_u = \sum_{k=1}^{N} \frac{r_{u,recom_{u,k}}}{log_2(k+1)} \tag{15}$$

where $recom_{u,k} \in I$ is the item recommended to user u in k position.

To obtain NDCG, this DCG value should be normalized by dividing it by the maximum DCG value, $DCG_{perfect}$, that can be reached [71]. $DCG_{perfect}$ is a perfect sorting of the items, i.e., the list of items sorted by their value in the test set. Specifically, the reaching of $DCG_{perfect}$ is done by sorting all relevant documents at the test set by their relative relevance, and therefore calculating the DCG value of such list, producing in this way the maximum possible DCG. In the specific case of our experimental scenario, the considered values of the items are the binary values of the initial matrix which assigns 1 to the items where a given user has expressed or not interest either creating, commenting, or voting it. Therefore, we sorted each list according to such values, by considering in the initial positions to those items evaluated as 1, and at the lower positions to those items evaluated as 0. The DCG value calculated to such list is the maximum possible DCG value, and therefore is considered as $DCG_{perfect}$.

Finally, the NDCG values for each user is calculated as:

$$NDCG = \frac{DCG}{DCG_{perfect}} \tag{16}$$

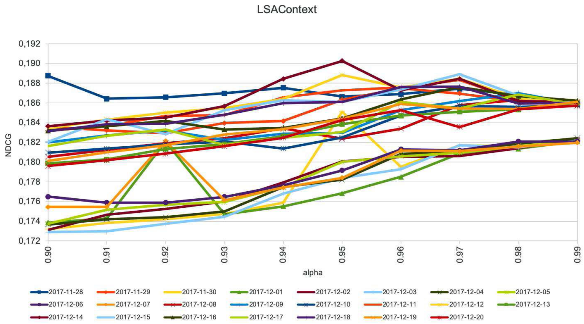

FIGURE 5. **Results managing context without clustering.**

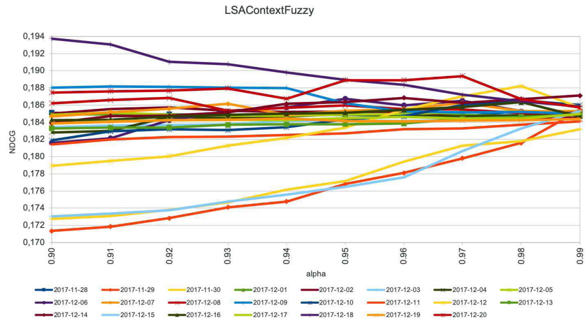

FIGURE 6. **Results managing context with fuzzy membership.**

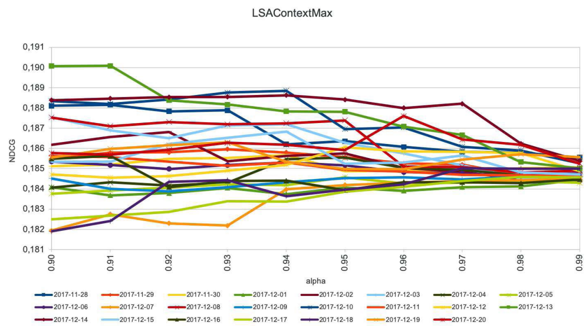

FIGURE 7. **Results managing context with max membership.**

Finally, such user-associated NDCG values are averaged to obtained the final NDCG value reported across this experimental section.

E. RESULTS

In this section, the results obtained for the different approaches compared are shown and analyzed to evaluate the performance of the proposal. Figures 5, 6, and 7 show the results of the three techniques compared in the 3dprinting QA dataset. The three figures show in X axis the α parameter and in Y axis the NDCG. The series denote the context, hence its position shows the results of the proposal with the corresponding α value for the day.

In Figure 5 can be noticed that, although the proposal reaches a higher performance in some days (contexts), the improvement does not compensates for the decay in performance in other days. Focusing in the context of 2017-11-28, it is clear that context plays a relevant role because LSAContext obtains the best performance for lower values of α (i.e.giving more importance to the context in the final recommendation). This initial behavior of the direct incorporation of the context was expected, regarding that there are some days where the *relevance* of the context can be higher or lower for the recommendation performance. Furthermore, for other days such as 2017-12-11, 2017-12-12, 2017-12-14, and 2017-12-18, the results show a clear positive influence of the context in the recommendation performance. However, for several days such as 2017-12-01 and 2017-12-03, the incorporation of the context (i.e. setting lower values of α) gets a worse recommendation performance.

Figure 6 shows that LSAContextClusteringFuzzy improves the results of LSAContext, regarding it reaches a higher stability for lower values of α. Specifically, for some specific scenarios such as the context of day 2017-12-18, LSAContextClusteringFuzzy improves the recommendation performance for such α values. On the other hand, there is a major decay in three contexts: 2017-11-29, 2017-11-30 and 2017-12-15 in which the proposal does not even reach the average value of the other approaches. In some context such as the day 2017-12-12, there is a decay for $\alpha \in [0.90, 0.95]$ but an improvement for $\alpha \in [0.96, 0.99]$. However, even though for LSAContextClusteringFuzzy there are some scenarios that LSA is not improved, here we remark that in contrast to LSAContext, for most of the days the results improve or lie around the average NDCG values, and there are only four days that for some values of α are clearly under such average. This improvement of LSAContextClusteringFuzzy over LSAContext is clearly introduced by the fuzzy clustering approach, that allows the detection and incorporation of context topics in the recommendation model. In this way, while for some days the whole context could not be relevant for the recommendation purpose, maybe some identified topic in this context could actually be relevant for the recommendation purpose. The day 2017-12-20 is a day where this assumption is clearly proved, regarding the for LSAContext its values tend to be under the average performance, but for LSAContextClusteringFuzzy its values tend to be over such average.

Figure 7 shows that LSAContextClusteringMax improves, for most of the explored contexts, the average NDCG value which is around 0.1840. This result contrasts to LSAContextClusteringFuzzy in the sense that for LSAContextClusteringFuzzy there are some days achieving a notably lower accuracy, while for LSAContextClusteringMax all the results were around and in many cases over LSAContextClusteringFuzzy. This behavior suggested that not all the information associated to the identified clusters is significant for the recommendation performance, and that if we incorporate only the most relevant information, such performance can be improved.

Figure 8 compares the results of each proposal with the best α. Here LSAContext has a great variability in performance across days, obtaining the better results from 2017-12-08 onwards, but with a low performance across

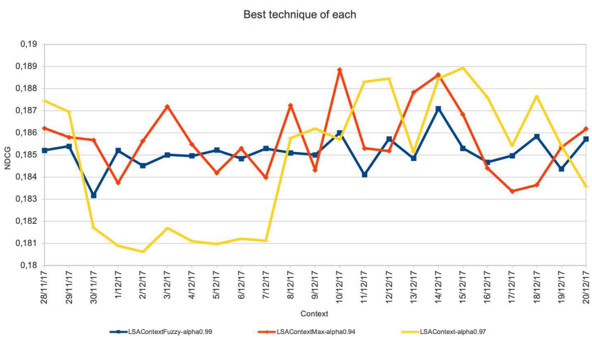

FIGURE 8. Results of the proposal to manage context with fuzzy membership.

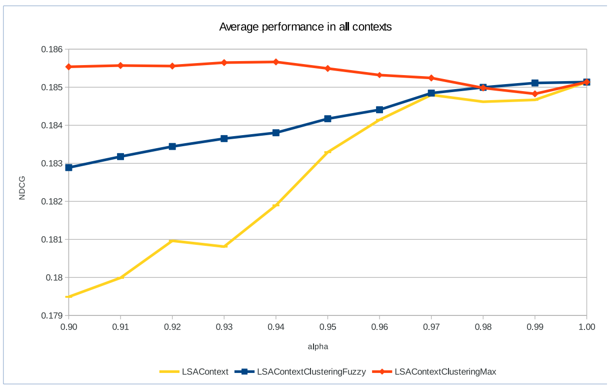

FIGURE 9. Average NDCG of the compared approaches in all contexts.

the days 2017-11-30 and 2017-12-07. However, the most sophisticated LSAContextClusteringMax and LSAContextClusteringFuzzy approaches introduce stability in the overall performance of the proposal. Furthermore, it is worth to notice that for some specific days, LSAContextClusteringMax swiftly obtained a greater accuracy in relation to the consequent days. In contrast, the improvement of LSAContextClusteringFuzzy was more uniformly distributed across the day sequences.

Figure 9 summarizes the results of the three proposals as compared to the LSA approach. In the case of LSAContextClusteringMax, it overcomes the results of all the remaining approaches for $\alpha \in [0.90, 0.97]$. For $\alpha = 0.94$ it reached the maximum average NDCG across all contexts explored, hence this value is the best one in this QA domain. On the other hand, in the case of LSAContextClusteringFuzzy and LSAContext, although in average they do not provide improvement, according to the previously presented results they notably obtain better results in some contexts and therefore are alternatives to be taking into account for the contextual modelling.

Summarizing, the best approaches of the compared ones is the LSAContextClusteringMax with $\alpha = 0.94$. This value has been optimized for the 3dprinting QA dataset, hence, for other domains it needs to be adjusted. This parameter provides the LSAContextClusteringMax with flexibility to adapt to different QA domains.

TABLE 4. Comparison between our proposal and the related works on context-awareness.

	Context Type	Domains	Working principle
[46]	Activity, age, gender	Indoor shopping, point-of-interest	Analyses the relation between user profiles and services under the same context
[26]	Location	News	Users are defined by the articles read in the past along with their location
[48]	Activity	Music	The context-aware songs recommendation is formulated by inferring the user's current situation category given some contextual features, and finding a song that matches the given situation
[47]	Location	Indoor shopping	User preferences are calculated by integrating the time spent in a brand store, frequency of visits to the store, and the matching between the special offers at the brand store and the user's preferences
[49]	Time	Music	Context refers to the time at which the user listens to a song.
[50]	Activity, Location	Music	A set of latent topics is used to associate music content with a user's music preferences under certain location
[51]	Location	Point of Interest	Context is a weighting factor that influences the recommendation score of a user for a certain item.
[25]	Time, Social, Location	Movies	Take as base the movie recommendation domain, and consider context as a weighting factor
This approach	Micro-blogging content	QA items	Consider context based on topic detection within current trend interest to be fused with the features on previously preferred QA items, being modeled as textual information both items and context

F. ON COMPARISON WITH RELATED WORKS

Regarding that the main contribution of this work is a global methodology to integrate two different information sources coming from a content-based and a context-aware scenario, the comparison with related works should consider the research antecedents in both context-aware and content-based recommendation paradigms.

To develop a comparison against both paradigms, we take as main reference a popular survey recently published by Villegas *et al.* [20]. Such survey analyzed 286 research papers on context-aware recommendation, and characterized them according to several criteria such as their working principle and the used information source. However this work, already referred at the Preliminaries section, only identifies too few works considering content-based context-aware recommendation supported by content modelling.

Table 4 presents a comparison between such few works and our proposal considering context type, domains, and working principles as comparison criteria. The table clearly shows that most of the previous works are focused on managing activity, location, and time as the context type; and indoor shopping, news, music, and point of interests as recommendation domains. This substantially differs from the context type and recommendation domain of our current proposal,

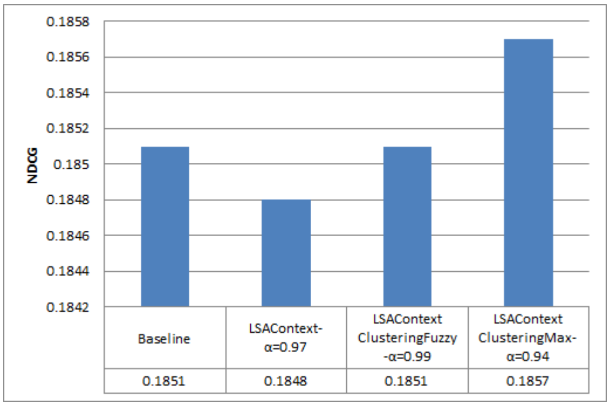

FIGURE 10. Average NDCG of the compared approaches in all contexts.

which are microblogging content and QA items respectively. Furthermore, a difference regarding the context type and recommendation domain, necessarily implies also a different working principle as could be appreciated at Table 4. This contrast disables of direct comparison between our proposal and previous works in terms of recommendation performance.

Even though these facts show that our proposal has some particularities in relation to previous works that disable a direct comparison, it is necessary to compare our work against some representative approach in the state-of-art, to prove that it outperforms some previously proposed method that can be applicable to the current problem. Considering that QA items are textual items, we choose the traditional version of the referred Latent Semantic Analysis as a baseline recommendation approach. [42]. The traditional Latent Semantic Analysis is a very popular and effective approach in content-based recommendation [13], and could be applied in the current scenario. We remark that in this case the context is not taken into account in the recommendation generation.

Figure 10 presents the results of the comparison between such baseline and the best values for each proposed recommendation approach evaluated in the previous subsection (i.e. LSAContext, LSAContextClusteringFuzzy, and LSAContextClusteringMax). In a similar way to the previous experimental scenarios, the average NDCG for all users is used as the evaluation criteria. The table shows that even though LSAContext and LSAContextClustering-Fuzzy reach a performance close to the baseline, LSAContextClusteringMax clearly outperforms the baseline and therefore proves that our research work introduces a recommendation approach that enhances the performance associated to a very representative previous work. Taking into account LSAContext and LSAContextClusteringFuzzy here we remark that Figure 10 presents the global average value, and that even though such average is close to the baseline, Figure 5 and 6 already show that both approaches obtained in some days results that are notably better than their final average performances.

Overall, our paper presented an approach that considers the role of the context in QA recommendations, showing that QA items are a key scenario where content can play a relevant role in the recommendation improvement. However, we point out that our approach can be generalizable to any textual item and beyond, being used as a general methodology to combine two different information sources in order to boost the recommendation performance. According to our viewpoint, this fact would be the main strength of our proposal.

However, our proposal still has some limitations that need to be covered in further research. Here a limitation is related to the fact that sometimes the use of the context brings a negative impact in the recommendation performance. Even though such behaviour is expected, further research is needed to do a best identification of such scenarios.

V. CONCLUSION AND FURTHER STUDY

In this paper, we have explored the application of contextual information in the QA domain recommendation. LSAContextClustering first builds the LSA model associated to the QA domain. After that, it builds the user profile combining the QA profiles with the user preferences. In parallel, it builds the profiles of the context, which is separated in a number of clusters and a context profile is built for each of them. The following step is to combine the user profile with the context profile that is more close to their preferences, which is achieved computing the cosine coefficient between the profiles. This combined profile allows the system to know user preferences and also consider contextual information in the recommendation.

We performed a case study and experimentation developed within the big data framework Spark to compare various configurations for the proposed approach. We found out that the best way to generate each context cluster profile is to select only the words whose membership value is the highest across clusters in the explored QA domain. We also shows that the proposal provides better results as compared to the baseline method (LSA).

In this scenario, contextual information is a key source of information to provide users with relevant recommendations that allow them to better understand the current scenario. The provided system is a relevant tool in the completion of user knowledge through the recommendation of QA items. Future works will be focused on mitigating some of the proposal's limitation as well as developing some natural extensions of the current proposal. They will be focused on: 1) Proposing approaches for measuring the quality of the information provided by the current context to incorporate it into the recommendation approach, taking as base some concepts previously developed such as relevant contextual information [72] and differential context relaxation and weighting [73], 2) Extending the proposal to be used in the group recommendation scenario [74], [75], 3) Proposing new approaches following the presented contextual modelling scenario, for boosting recommendation diversity, 4) Exploring the effect of natural noise management approaches [76], [77], in the current recommendation scenario, and 5) Considering the use of more sophisticated tools such as ontologies for improving the semantic information processing across the proposal.

ACKNOWLEDGMENT

The authors, therefore, gratefully acknowledge DSR technical and financial support.

REFERENCES

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, "Recommender systems survey," *Knowl.-Based Syst.*, vol. 46, pp. 109–132, Jul. 2013.
[2] R. Yera and L. Martínez, "Fuzzy tools in recommender systems: A survey," *Int. J. Comput. Intell. Syst.*, vol. 10, no. 1, pp. 776–803, 2017.
[3] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, "A Web-based personalized business partner recommendation system using fuzzy semantic techniques," *Comput. Intell.*, vol. 29, no. 1, pp. 37–69, 2013.
[4] D. Wu, J. Lu, and G. Zhang, "A fuzzy tree matching-based personalized e-learning recommender system," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 2412–2426, Dec. 2015.
[5] R. Y. Toledo and Y. C. Mota, "An e-learning collaborative filtering approach to suggest problems to solve in programming online judges," *Int. J. Distance Educ. Technol.*, vol. 12, no. 2, pp. 51–65, 2014.
[6] M. Al-Hassan, H. Lu, and J. Lu, "A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system," *Decis. Support Syst.*, vol. 72, pp. 97–109, Apr. 2015.
[7] J. Noguera, M. Barranco, R. Segura, and L. Martínez, "A mobile 3D-GIS hybrid recommender system for tourism," *Inf. Sci.*, vol. 215, pp. 37–52, Dec. 2012.
[8] D. Rafailidis and A. Nanopoulos, "Modeling users preference dynamics and side information in recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 782–792, Jun. 2016.
[9] T. T. S. Nguyen, H. Y. Lu, and J. Lu, "Web-page recommendation based on Web usage and domain knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 10, pp. 2574–2587, Oct. 2014.
[10] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu, "Uncertainty analysis for the keyword system of Web events," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 829–842, Jun. 2016.
[11] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, and G. Lekkas, "Personalized finance advisory through case-based recommender systems and diversification strategies," *Decis. Support Syst.*, vol. 77, pp. 100–111, Sep. 2015.
[12] Y. Koren and R. Bell, *Advances in Collaborative Filtering*. Boston, MA, USA: Springer, 2015, pp. 77–118.
[13] M. de Gemmis, P. Lops, C. Musto, F. Narducci, and G. Semeraro, "Semantics-aware content-based recommender systems," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 119–159.
[14] M. Nilashi, O. Ibrahi, and K. Bagherifard, "A recommender system based on collaborative filtering using Ontology and dimensionality reduction techniques," *Expert Syst. Appl.*, vol. 92, pp. 507–520, Feb. 2018.
[15] L. O. Colombo-Mendoza, R. Valencia-García, A. Rodríguez-González, G. Alor-Hernández, and J. J. Samper-Zapater, "RecomMetz: A context-aware knowledge-based mobile recommender system for movie showtimes," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1202–1222, 2015.
[16] S. Deng, L. Huang, and G. Xu, "Social network-based service recommendation with trust enhancement," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8075–8084, Dec. 2014.
[17] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2015, pp. 191–226.
[18] J. Castro, R. Yera, and L. Martínez, "An empirical study of natural noise management in group recommendation systems," *Decis. Support Syst.*, vol. 94, pp. 1–11, Feb. 2017.
[19] J. Masthoff, "Group recommender systems: Aggregation, satisfaction and group attributes," in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Boston, MA, USA: Springer, 2015, pp. 743–776.
[20] N. M. Villegas, C. Sánchez, J. Díaz-Cely, and G. Tamura, "Characterizing context-aware recommender systems: A systematic literature review," *Knowl.-Based Syst.*, vol. 140, pp. 173–200, Jan. 2018.
[21] A. Figueroa and G. Neumann, "Context-aware semantic classification of search queries for browsing community question–answering archives," *Knowl.-Based Syst.*, vol. 96, pp. 1–13, Mar. 2016.
[22] A. B. Barragáns-Martínez, M. Rey-López, E. Costa-Montenegro, F. A. Mikic-Fonte, J. C. Burguillo, and A. Peleteiro, "Exploiting social tagging in a Web 2.0 recommender system," *IEEE Internet Comput.*, vol. 14, no. 6, pp. 23–30, Nov./Dec. 2010.
[23] C. C. Aggarwal, *Content-Based Recommender Systems*. Cham, Switzerland: Springer, 2016, pp. 139–166.
[24] I. Srba and M. Bielikova, "A comprehensive survey and classification of approaches for community question answering," *ACM Trans. Web*, vol. 10, no. 3, p. 18, 2016.
[25] C. Musto, P. Semeraro, P. Lops, and M. de Gemmins, "Contextual eVSM: A content-based context-aware recommendation framework based on distributional semantics," in *E-Commerce and Web Technologies*. Berlin, Germany: Springer, 2013, pp. 125–136.
[26] J.-W. Son, A.-Y. Kim, and S.-B. Park, "A location-based news article recommendation with explicit localized semantic analysis," in *Proc. 36th ACM SIGIR Int. Conf. Res. Develop. Inf. Retr.*, Springer, 2013, pp. 293–302.
[27] T. De Pessemier, C. Courtois, K. Vanhecke, K. Van Damme, L. Martínez, and L. De Marez, "A user-centric evaluation of context-aware recommendations for a mobile news service," *Multimedia Tools Appl.*, vol. 75, no. 6, pp. 3323–3351, Mar. 2016.
[28] L. Ponzanelli, "Holistic recommender systems for software engineering," in *Proc. Companion ACM 36th Int. Conf. Softw. Eng. (ICSE)*, New York, NY, USA, 2014, pp. 686–689.
[29] L. Ponzanelli, G. Bavota, M. D. Penta, R. Oliveto, and M. Lanza, "Prompter: A self-confident recommender system," in *Proc. IEEE Int. Conf. Softw. Maintenance Evol.*, Sep. 2014, pp. 577–580.
[30] L. Ponzanelli, S. Scalabrino, G. Bavota, A. Mocci, R. Oliveto, M. Di Penta, and M. Lanza, "Supporting software developers with a holistic recommender system," in *Proc. 39th Int. Conf. Softw. Eng. (ICSE)*, Piscataway, NJ, USA, May 2017, pp. 94–105.
[31] N. Parikh and N. Sundaresan, "Buzz-based recommender system," in *Proc. ACM 18th Int. Conf. World Wide Web (WWW)*, New York, NY, USA, 2009, pp. 1231–1232
[32] A. Hermida, "Twittering the news," *Journalism Pract.*, vol. 4, no. 3, pp. 297–308, 2010.
[33] J. Maillo, S. Ramírez, I. Triguero, and F. Herrera, "kNN-IS: An iterative spark-based design of the k-nearest neighbors classifier for big data," *Knowl.-Based Syst.*," vol. 117, pp. 3–15, Feb. 2016.
[34] L. R. Nair and S. D. Shetty, "Streaming Twitter data analysis using spark for effective job search," *J. Theor. Appl. Inf. Technol.*, vol. 80, no. 2, p. 349, 2015.
[35] R. Burke, "Hybrid recommender systems: Survey and experiments," *User Model. User-Adapted Interact.*, vol. 12, no. 4, pp. 331–370, Nov. 2002.
[36] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, "Feature-weighted user model for recommender systems," in *Proc. 11th Int. Conf. User Modeling*. Berlin, Germany: Springer-Verlag, 2007, pp. 97–106.
[37] J. Castro and R. M. Rodr "iguez, and M. J. Barranco, "Weighting of features in content-based filtering with entropy and dependence measures," *Int. J. Comput. Intell. Syst.*, vol. 7, no. 1, pp. 80–89, 2014.
[38] M. Terzi, M. Ferrario, and J. Whittle, "Free text in user reviews: Their role in recommender systems," in *Proc. Workshop Recommender Syst. Social Web, 3rd ACM Conf. Recommender Syst.*, 2011, pp. 45–48.
[39] U. Erra, S. Senatore, F. Minnella, and G. Caggianese, "Approximate TF–IDF based on topic extraction from massive message stream using the GPU," *Inf. Sci.*, vol. 292, pp. 143–161, Jan. 2015.
[40] M. F. Porter, "An algorithm for suffix stripping," *Program*, vol. 14, no. 3, pp. 130–137, 1980.
[41] P. Lops, M. Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, ch. 3, pp. 73–105.
[42] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[43] R. Bambini, P. Cremonesi, and R. Turrin, *A Recommender System for an IPTV Service Provider: A Real Large-Scale Production Environment*. Boston, MA, USA: Springer, 2011, pp. 299–331.
[44] F. Ricci, "Contextualizing recommendations," in *Proc. ACM RecSys Workshop Context-Aware Recommender Syst. (CARS), Conjunct 6th ACM Conf. Recommender Syst. (RecSys)*, 2012.

[45] G. Adomavicius and A. Tuzhilin, "Context-aware recommender systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, ch. 3, pp. 217–253.
[46] J. Hong, E. H. Suh, J. Kim, and K. S. Kim, "Context-aware system for proactive personalized service based on context history," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 7448–7457, 2009.
[47] B. Fang, S. Liao, K. Xu, H. Cheng, C. Zhu, and H. Chen, "A novel mobile recommender system for indoor shopping," *Expert Syst. Appl.*, vol. 39, no. 15, pp. 11992–12000, 2012.
[48] X. Wang, D. Rosenblum, and Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 99–108.
[49] D. Shin, J. W. Lee, J. Yeon, and S. G. Lee, "Context-aware recommendation by aggregating user context," in *Proc. IEEE Conf. Commerce Enterprise Comput.*, Jul. 2009, pp. 423–430.
[50] Z. Cheng and J. Shen, "Just-for-Me: An adaptive personalization system for location-aware social music recommendation," in *Proc. ACM Int. Conf. Multimedia Retr.*, 2014, p. 185.
[51] M. H. Kuo, L. C. Chen, and C. W. Liang, "Building and evaluating a location-based service recommendation system with a preference adjustment mechanism," *Expert Syst. Appl.*, vol. 36, no. 2, pp. 3543–3554, 2009.
[52] Y. Liu, C. Zhang, X. Yan, Y. Chang, and P. S. Yu, "Generative question refinement with deep reinforcement learning in retrieval-based QA system," 2019, *arXiv:1908.05604*. [Online]. Available: https://arxiv.org/abs/1908.05604
[53] H. H. Fang, F. Wu, Z. Zhao, X. Duan, Y. Zhuang, and M. Ester, "Community-based question answering via heterogeneous social network learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 122–128.
[54] Z. Zhao, H. Lu, V. W. Zheng, D. Cai, X. He, and Y. Zhuang, "Community-based question answering via heterogeneous social network learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3532–3538.
[55] Y. Cao, H. Duan, C.-Y. Lin, Y. Yu, and H.-W. Hon, "Recommending questions using the MDL-based tree cut model," in *Proc. ACM 17th Int. Conf. World Wide Web*, 2008, pp. 81–90.
[56] J. Wang, X. Hu, Z. Li, W. Chao, and B. Hu, "Learning to recommend questions based on public interest," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 2029–2032.
[57] T. C. Zhou, M. R. Lyu, I. King, and J. Lou, "Learning to suggest questions in social media," *Knowl. Inf. Syst.*, vol. 43, no. 2, pp. 389–416, 2015.
[58] G. Adomavicius and A. T. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
[59] L. Martínez, L. G. Pérez, and M. Barranco, "A multigranular linguistic content-based recommendation model: Research articles," *Int. J. Intell. Syst.*, vol. 22, no. 5, pp. 419–434, May 2007.
[60] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of dimensionality reduction in recommender systems—A case study," in *Proc. ACM WebKDD Workshop*, 2000.
[61] J. C. Bezdek, W. Full, and R. Ehrlich, "FCM: The fuzzy c-means clustering algorithm," *Comput. Geosci.*, vol. 10, nos. 2–3, pp. 191–203, 1984.
[62] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. ACM 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[63] J. Herlocker, J. Konstan, K. Terveen, and J. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004.
[64] A. Gunawardana and G. Shani, "A survey of accuracy evaluation metrics of recommendation tasks," *J. Mach. Learn. Res.*, vol. 10, pp. 2935–2962, Dec. 2009.
[65] G. Shani and A. Gunawardana, "Evaluating recommendation systems," in *Recommender Systems Handbook*. Boston, MA, USA: Springer, 2011, ch. 8, pp. 257–297.
[66] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
[67] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, and M. J. Franklin, "Apache spark: A unified engine for big data processing," *Commun. ACM*, vol. 59, no. 11, pp. 56–65, 2016.
[68] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, and S. Owen, "MLlib: Machine learning in apache spark," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1235–1241, 2016.
[69] S. Chintapalli, D. Dagit, B. Evans, R. Farivar, T. Graves, M. Holderbaugh, Z. Liu, K. Nusbaum, K. Patil, and B. J. Peng, "Benchmarking streaming computation engines: Storm, flink and spark streaming," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops*, May 2016, pp. 1789–1792.
[70] A. Gunawardana and G. Shani, *Evaluating Recommender Systems*. Boston, MA, USA: Springer, 2015, pp. 265–308.
[71] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, Oct. 2002, doi: 10.1145/582415.582418.
[72] A. Odić, M. Tkalčič, J. F. Tasič, and A. Košir, "Predicting and detecting the relevant contextual information in a movie-recommender system," *Interacting Comput.*, vol. 25, no. 1, pp. 74–90, 2013.
[73] Y. Zheng, R. Burke, and B. Mobasher, "Differential context relaxation for context-aware travel recommendation," in *E-Commerce and Web Technologies* (Lecture Notes in Business Information Processing), vol. 123. Berlin, Germany: Springer, 2012, pp. 88–99.
[74] J. Castro, J. Lu, G. Zhang, Y. Dong, and L. Martínez, "Opinion dynamics-based group recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 12, pp. 2394–2406, Dec. 2018. [Online]. Available: http://ieeexplore.ieee.org/document/7919222/
[75] J. Castro, M. J. Barranco, R. M. Rodríguez, and L. Martínez, "Group recommendations based on hesitant fuzzy sets," *Int. J. Intell. Syst.*, vol. 33, no. 10, pp. 2058–2077, 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/int.21922
[76] R. Y. Toledo, Y. C. Mota, and L. Martínez, "Correcting noisy ratings in collaborative recommender systems," *Knowl.-Based Syst.*, vol. 76, pp. 96–108, Mar. 2015.
[77] L. Martínez, J. Castro, and R. Yera, "Managing natural noise in recommender systems," in *Proc. 5th Int. Conf. Theory Pract. Natural Comput. (TPNC)*, Sendai, Japan, C. Martín-Vide, T. Mizuki, and M. A. Vega-Rodríguez, Eds. Cham, Switzerland: Springer, Dec. 2016, pp. 3–17.

JORGE CASTRO received the B.Sc. and M.Sc. degrees in computer science from the University of Jaén, Spain, and the Ph.D. degree from the University of Granada, in 2018. During the Ph.D. degree, he was developing a software library focused on experimenting with recommendation algorithms. His research interests include recommender systems, group recommendation, and context-aware recommendation.

RACIEL YERA TOLEDO received the B.Sc. degree in computer science from the Informatics Science University, Cuba, in 2010, the M.Sc. degree from the University of Ciego de Ávila, Cuba, in 2012, and the Ph.D. degree from Las Villas Central University, Cuba, in 2015. He is currently a Professor of computer science with the University of Ciego de Ávila. His current interest includes the improvement of recommender systems performance through the applications of computational intelligence techniques. He has received three Annual Provincial Awards from the Cuban Academy of Sciences in 2013 for his research results related to recommender systems.

AHMAD A. ALZAHRANI received the Ph.D. degree in computer science from La Trobe University, Australia, in 2014. He is currently an Assistant Professor with the Faculty of Computing and Information Technology, King Abdulaziz University. His research interests include pervasive computing and human–computer interaction.

PEDRO J. SÁNCHEZ received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1993 and 2007, respectively. From 2004 to 2008, he was an Assistant Head and the Head of the Computer Science Department, University of Jaén, Jaén, Spain, from 2008 to 2016. He is currently a Senior Lecturer with the Computer Science Department, University of Jaén. His current research interests include decision making, fuzzy logic-based systems, computing with words, and recommender systems.

MANUEL J. BARRANCO received the Ph.D. degree in computer science from the Polytechnic University of Madrid, Spain, in 1995. He is currently a Full Professor with the Computer Science Department, University of Jaén, Spain. His research interests include recommender systems.

LUIS MARTÍNEZ received the M.Sc. and Ph.D. degrees in computer science from the University of Granada, Granada, Spain, in 1993 and 1999, respectively. He is currently a Full Professor with the Computer Science Department, University of Jaén, Jaén, Spain. He is also a Visiting Professor with the University of Technology Sydney, the University of Portsmouth (Isambard Kingdom Brunel Fellowship Scheme), and the Wuhan University of Technology (Chutian Scholar), a Guest Professor with Southwest Jiaotong University, Chengdu, China, and an Honourable Professor with Xihua University, Chengdu, China. He has co-edited 11 journal special issues on fuzzy preference modeling, soft computing, linguistic decision making, and fuzzy sets theory. He has been a main researcher of 14 research and development projects. He has also published more than 100 articles in journals indexed by the SCI and more than 150 contributions in international conferences related to his research areas. He has been appointed as Highly Cited Researcher 2017 in computer science. His current research interests include decision making, fuzzy logic-based systems, computing with words, and recommender systems. He is a member of the European Society for Fuzzy Logic and Technology. He was a recipient of the IEEE Transactions on Fuzzy Systems Outstanding Paper Award 2008 and 2012 (bestowed in 2011 and 2015). He is the Co-Editor-in-Chief of the *International Journal of Computational Intelligence Systems* and an Associate Editor of the journals, including the IEEE TRANSACTIONS ON FUZZY SYSTEMS, *Information Fusion*, the *International Journal of Fuzzy Systems*, and the *Journal of Intelligent & Fuzzy Systems*.