

# A question answer-based context-aware recommendation method with semantic model within a big data framework

Jorge Castro , Jie Lu, Guangquan Zhang, Luis Martínez

## 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 collaborative interest, where a feed of status updates can be analyzed to model the context. It proposes a novel CA-CB approach that recommends question/answer items and introduces context awareness based on topic detection within collaborative interest. A case study and related experiments show that the context integration significantly benefits recommendation.

## Index Terms

content-based recommender system, context-aware recommendation, user profile contextualization, map-reduce

This research work was partially supported by the Research Project TIN-2015-66524-P, the Spanish FPU fellowship (FPU13/01151).

J. Castro is with the Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain, and also with the School of Software, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: jcastro@decsai.ugr.es).

J. Lu and G. Zhang are with the School of Software, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: jie.lu@uts.edu.au, guangquan.zhang@uts.edu.au)

L. Martínez is with the Computer Science Department, University of Jaén, 23008 Jaén, Spain, and also with the School of Management, Wuhan University of Technology, Wuhan 430070, China (e-mail: martin@ujaen.es)

## 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 information 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. Recommender systems have been proved to be successful in several domains, such as e-business [1], e-learning [2], [3], e-tourism [4], [5], e-commerce [6], web pages [7], [8] and financial investment [9], among others.

Several approaches have been explored for alleviating information overload problem with recommender systems. The most widespread ones are collaborative filtering (**CF**) [10] and Content Based (CB) [11]. 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 [12], among **others** [13], [2], or social network recommender systems that use links between users to improve the recommendation [14]. Recent research lines also focus on integrating contextual information [15] or providing recommendations targeted to groups of users[16], [17]. In this paper, we focus on **the** recommendation of question answering (QA) items with content-based approach and contextual information integration [18].

Within CB we distinguish two kinds: (i) based on item features, and (ii) based on items descriptions. In this paper we focus on the latter, given that QA items have a strong component of textual information for both explaining the question and answering it [19]. Although CB **suffers** from user cold start because they need some input from user preferences and lack of diversity in recommendations [20], CB **has** demonstrated their utility when new items are introduced in the system, i.e., in scenarios with strong item cold-start [21]. This feature makes the application of CB approaches interesting in domains where new items are constantly introduced, such as web pages or news. In this direction, QA recommendation shares the features to apply CB with

textual descriptions, hence, we focus on [it](#).

In the QA domain recommendation scenario there are several proposals for recommendation with CB approaches. Shao et al.[22] propose to apply Latent Dirichlet Allocation to label questions in a latent semantic feature category and find the most suitable answerers. In this direction, Zheng et al. [23] combine trust-based analysis of answerers with content analysis. While these approaches aim to reduce the answer time searching for adequate answerer for new posted questions, there are other recommender systems that aim to expand users' knowledge recommending already answered questions. Odiete et al. [24] analyze users' preferences and build a graph of expertise used to find gaps in their knowledge and suggest relevant questions.

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 collaborative interest. Therefore, it is interesting to explore Context-Aware CB (CA-CB), which integrate contextual information to the content-based recommendation. De Pessemier et al. [25] consider recommendations in mobile devices as very suitable to integrate context-awareness, and uses devices sensors and time of the day to deliver contextualized news recommendations. In QA recommendation, SeaHawk [26] and Prompter [27] provide CA-CB that support 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 [28] also 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 collaborative interest as current buzzwords to deliver currently relevant recommendations in e-commerce scenarios [29]. As it can be seen, there is no previous approach that focuses on context-aware recommendation regarding collaborative interest in the QA domain.

In this paper, we propose a novel CA-CB approach that recommends QA items and introduces context awareness based on topic detection within collaborative interest. Recent researches [30] 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 collaborative interest. The usage of such a context mainly helps recommend answers 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 that the context topic most suitable 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 generate 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 [31], [32], therefore, our proposal is developed within Spark, a distributed framework for big data that takes advantage of in-memory operations.

The contributions of this study are:

- A suitable way to study status updates in the QA recommendation scenario to provide recommendations tailored to both the user preferences and current collaborative interest.
- Introduction of personalised contextualization of user preference profiles to better integrate collaborative interest context in the recommendation.
- A proposal that applies semantic analysis of QA domain, deals with mixture of topics in collaborative interest and provides 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 collaborative interest improves QA recommendation.

The remaining of this paper is structured as follows. Section II provides a background of CB, CA-CB and the MapReduce approach. 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. 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 and basics in MapReduce and Spark.

### A. Content-based recommender systems.

An accurate definition of recommender system, given by R. Burke [33], 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

[25]: 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, and (ii) free-text representation.

In feature-based representation, items are usually stored in a database table where rows are items and columns are the item features (also called fields or properties) and each item might have a different value for each of these features. The recommender system learns a model that uses such information to approximate the rating function. There are proposals based on weighting features importance regarding the user's ratings using collaborative information [34] or using entropy and dependence between items' features and user's ratings [35]. Other approaches learn the rating function using linear regression [25].

In free-text representation there is a natural language piece of text that describes the item, such as movie synopsis or that is part of the item itself, such as the content of a web page or a news article. Such a kind of unstructured data provides information about the item (called document in these systems), however, they also have the complexity of dealing with natural language due to polysemous words and synonyms.

The TFIDF approach [36] is usually applied when dealing with free-text representation items. In TFIDF, the unstructured data is converted in structured data stemming words [37] 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 multiplying the  $tf_{t,d}$  by the  $idf_t$  [38], to consider the importance of the term on the document:

$$\text{profile}_d^{tfid} = \{tf_{t,d} * idf_t \quad s.t. \quad t \in d\} \quad (1)$$

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.

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

At this point the system contains a vector space model of items. User profiles can be generated aggregating the profiles of the items that they liked in the past [34]. The recommendation is

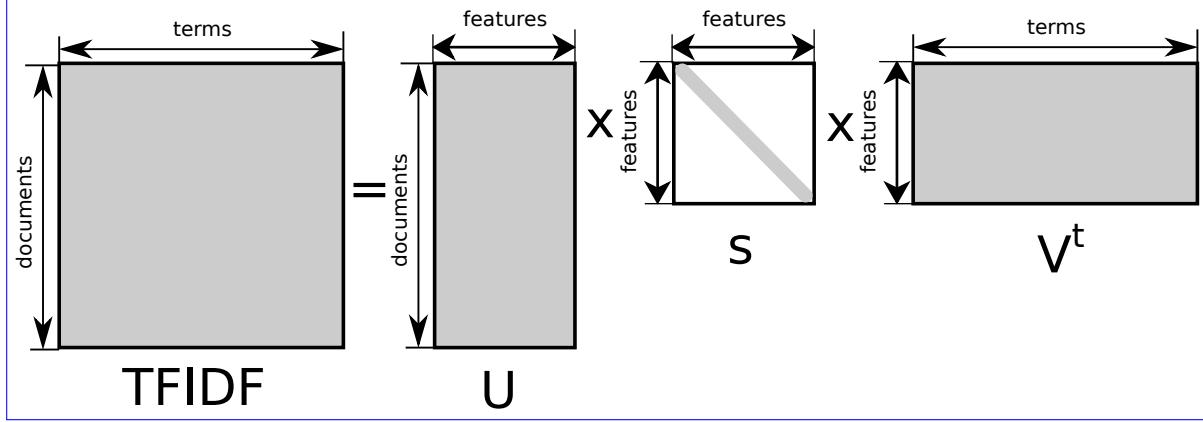


Figure 1: Decomposition of TFIDF matrix with singular Value decomposition. Note that  $s$  is a diagonal matrix with the singular values sorted in descending order.

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 polysemy or synonym words. In order to overcome this issue, Latent Semantic Analysis (LSA) is applied [11]. 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).

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

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 [39]. 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 R.Burke [33], other sources of information can be considered in the recommendation, such as the context in which the recommendation is received by users. F. Ricci [40] 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 [41]:

- 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 pre-filtering, the approach is to filter out the information that was not gathered in the current context. In traditional recommender systems the information is viewed as a function  $R : User \times Item \rightarrow Rating$  that the recommender system tries to approximate. In CA, the information can be viewed as a three dimensional cube  $User \times Item \times Context \rightarrow Rating$ . Contextual pre-filtering selects only the information relative to the context, hence, it tries to approximate function  $R_{context} : User \times Item \rightarrow Ratings_{context}$ . This way, they only consider ratings generated in the context in which the recommendation is delivered to the user. Contextual pre-filtering is a simple and effective approach, but it is affected by data sparsity when there is not enough information generated in all contexts to provide accurate recommendations. Some researchers have tried to overcome this issue through context-generalization [41] when the information available in the current context is not enough and include information from the broader context.

In post-filtering, the recommendations are first computed overlooking the contextual information, as in traditional recommender systems. After this initial step, recommendations are adjusted to the current context either removing irrelevant items or through a weighting function that updates predictions regarding the suitability of items to target user's context. A previous work [42] evaluated them in the same scenarios and compared pre- and post-filtering approaches. It determined that neither pre-filtering nor post-filtering completely dominates the other and a study to determine the best approach is needed in each specific case.

Previous approaches try to reduce the CA problem to a two dimensional one that can be solved with traditional recommender systems. This is not the case in contextual modeling, in which contextual information is directly integrated in the recommendation model to recommend. With this regard, researchers have explored heuristic-based [43], probabilistic [44], or matrix factorization [45] contextual modeling approaches.

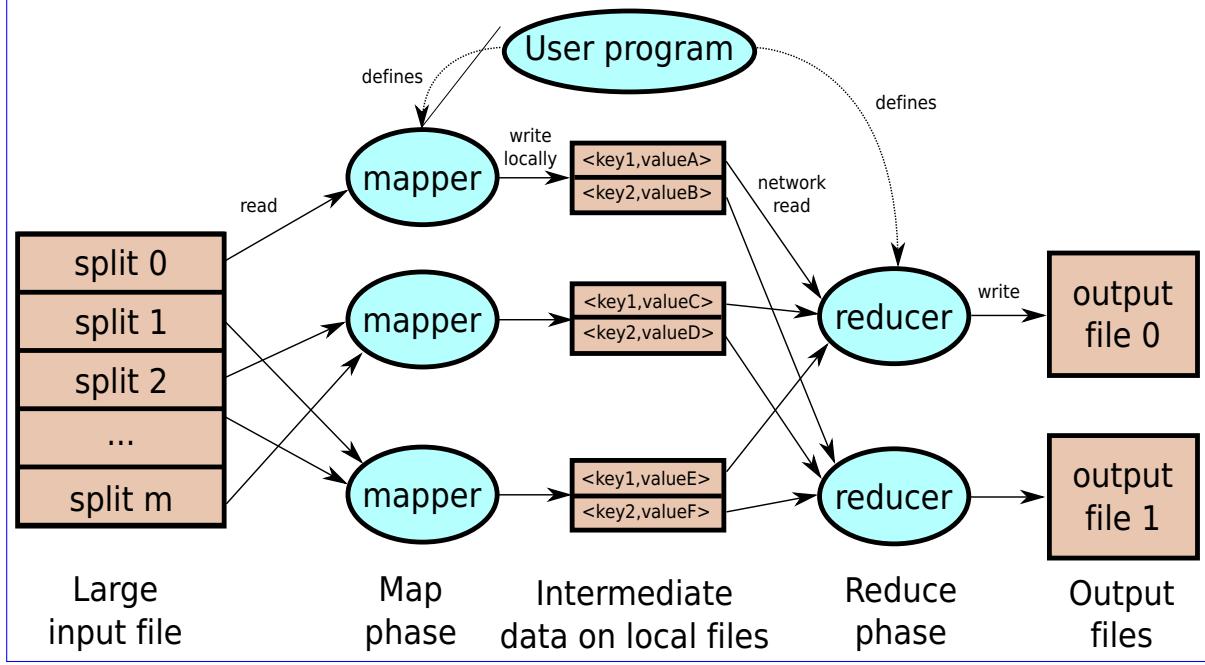


Figure 2: Scheme of the MapReduce approach [46].

### C. MapReduce and Spark

MapReduce [46] is a processing tool to manage data in a scalable way. It was originally designed by Google in 2003 aiming to support the most powerful search-engine on the internet, but later it was also applied for parallelization in general purpose applications. In MapReduce (see Figure 2), two main operations are defined by the user: *Map* and *Reduce*. The *Map* operation takes an input, which is given as a set of  $\langle \text{key}, \text{value} \rangle$  pairs, and produces an intermediate set of  $\langle \text{key}, \text{value} \rangle$  pairs that can be of different type and are defined by the user. An intermediate phase merges all pairs that have the same key and redistributes the work. The *Reduce* operation combines pairs by key to produce a smaller pair set.

There are several solutions that apply the MapReduce approach. Among them, Apache Spark [47], which was 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 [48] that was built to take advantage of Spark suitability for iterative tasks

and provides several machine learning techniques for classification, optimization, and data pre-processing, among others. Specifically, we use the tools that MLlib provides for regression and clustering. Spark Streaming [49] 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.

### III. SEMANTIC MODEL FOR RECOMMENDING QUESTIONS WITH CONTEXT AWARENESS BASED ON TOPIC DETECTION IN COLLABORATIVE 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 answers in the history domain and currently people are posting about Columbus Day, then the system should promote answers related to the 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 3, and it is composed of five phases:

- (i) QA domain semantic analysis: It applies LSA to reduce the dimensionality of the term-document matrix.
- (ii) 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.
- (iii) 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.
- (iv) Contextualize user profiles: It selects the context topic that is 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.
- (v) Prediction: It compares the document profiles and the contextualized user profile to recommend.

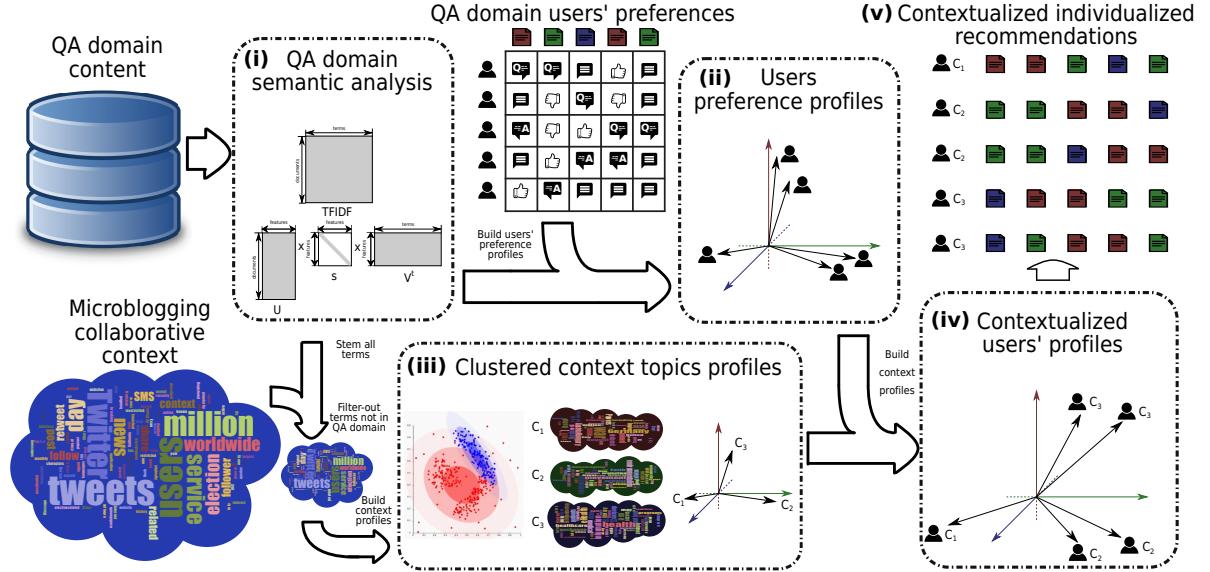


Figure 3: General scheme of the proposal.

### 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 [37]. 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 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, which allows to reduce the dimensionality of the original matrix keeping the  $f$  most relevant singular values of the original matrix. Hence, we obtain the profile of both terms and documents in the feature space, which compose the QA domain semantic model:

$$profile_d^{LSA} = \{u_{t,1}, \dots, u_{t,f}\} \quad (4)$$

Table I: Users' preferences over items, the rating matrix.

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

$$profile_t^{LSA} = \{v_{t,1}, \dots, v_{t,f}\} \quad (5)$$

### 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 unary matrix that states whether a given user has expressed interest in another document (see Table I) either creating, commenting, or voting it. In this table, the set of documents that user  $u$  has expressed interest in is defined as:

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

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 latent space:

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

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 collaborative interest. Formally, a status update consists of a

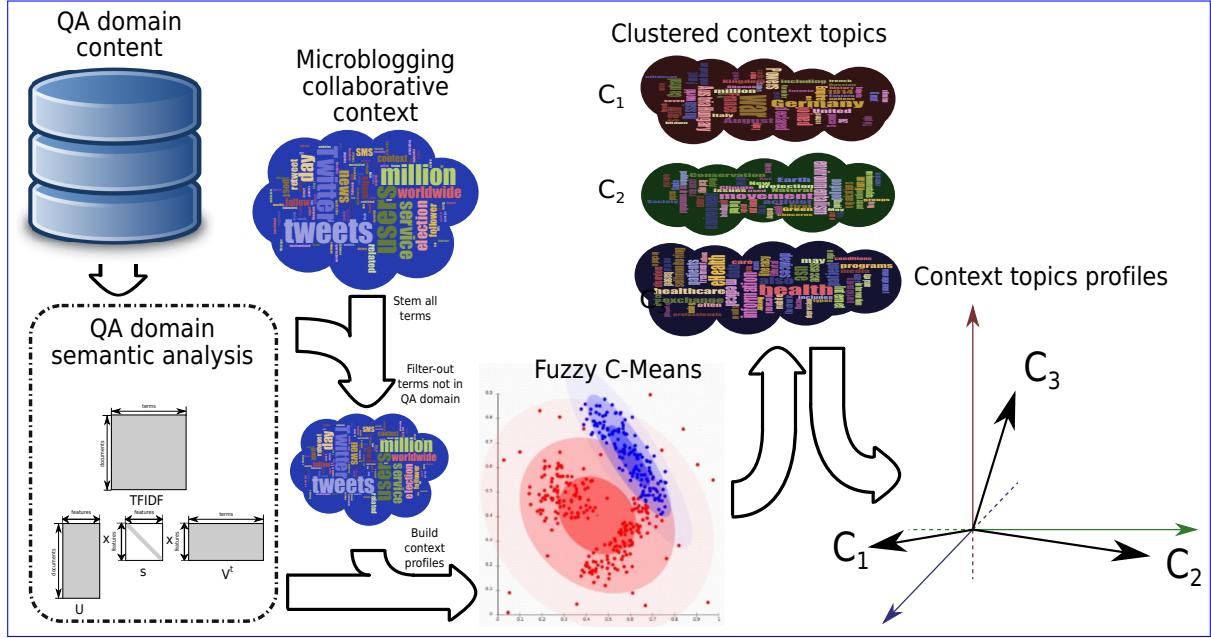


Figure 4: Context model building phase.

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. 4.

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 algorithm [50] 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.

$$\text{profile}_{c_i}^{\text{LSA}} = \sum_{t \in c_i} \text{profile}_t^{\text{LSA}} = \left\{ \sum_{t \in c_i} \text{profile}_{t,1}^{\text{LSA}}, \dots, \sum_{t \in c_i} \text{profile}_{t,f}^{\text{LSA}} \right\} \quad (8)$$

#### 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 a greater 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.

$$\underset{c_i}{\operatorname{argmax}} \quad \text{cosine}(\text{profile}_u^{\text{LSA}}, \text{profile}_{c_i}^{\text{LSA}}) \quad (9)$$

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  $\alpha$  parameter – the greater its value, the more importance of the user's preference profile over the profile of the selected context topic in the contextualized user profile.

$$\text{profile}_{C,u}^{\text{LSA}} = \alpha * \text{profile}_u^{\text{LSA}} + (1 - \alpha) * \text{profile}_{c_i}^{\text{LSA}} \quad (10)$$

#### 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  $p_{u,d}$ :

$$p_{u,d,c} = \text{profile}_{C,u}^{\text{LSA}} * s * \text{profile}_d^{\text{LSA}} \quad (11)$$

## IV. CASE STUDY AND EXPERIMENT

To evaluate the proposal, we 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 then detailed. After that, the evaluation measures are commented. Lastly, the results are analyzed.

### 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. [51] was performed with modifications to consider contextual information in the experiment:

- Split the dataset in training and test.
- Build the model with training data.
- Build the profile of each user including contextual information if applicable.
- Recommend to each user based on their profile and the model.
- Evaluate recommendations with the test set.

This procedure was repeated 20 times and 5-cross fold validation was used to split the data in training and test sets. Moreover, various contexts were considered, which are detailed in Section IV-C.

### B. Methods compared

The baseline method to compare with was the LSA method without contextual information. For the sake of fair comparison, the number of features was fixed in all models, and 30 features were considered in LSA.

We compared several ways for integrating contextual information in QA recommendation. Here, we show the results of three ways to characterize the context:

- 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.

$$profile_C = \sum_{t \in C} *profile_t \quad (12)$$

- 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:

$$profile_{c_i} = \sum_{t \in c_i} \mu_{t,c_i} * profile_t \quad (13)$$

Table II: 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

where  $\mu_{t,c_i}$  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 highest membership value:

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

where  $\mu_{t,c_i}^{max}$  is one if  $\mu_{t,c_i}$  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 parameter  $\alpha$ . In the experiment we explored several values for it, here, to make the results clearer, we show only  $\alpha \in [0.90, 1.00]$  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<sup>1</sup>. 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<sup>2</sup>, given the current interest on such topic. Some stats of the dataset are detailed in Table II.

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 5.

<sup>1</sup><http://data.stackexchange.com/>

<sup>2</sup><https://3dprinting.stackexchange.com/>

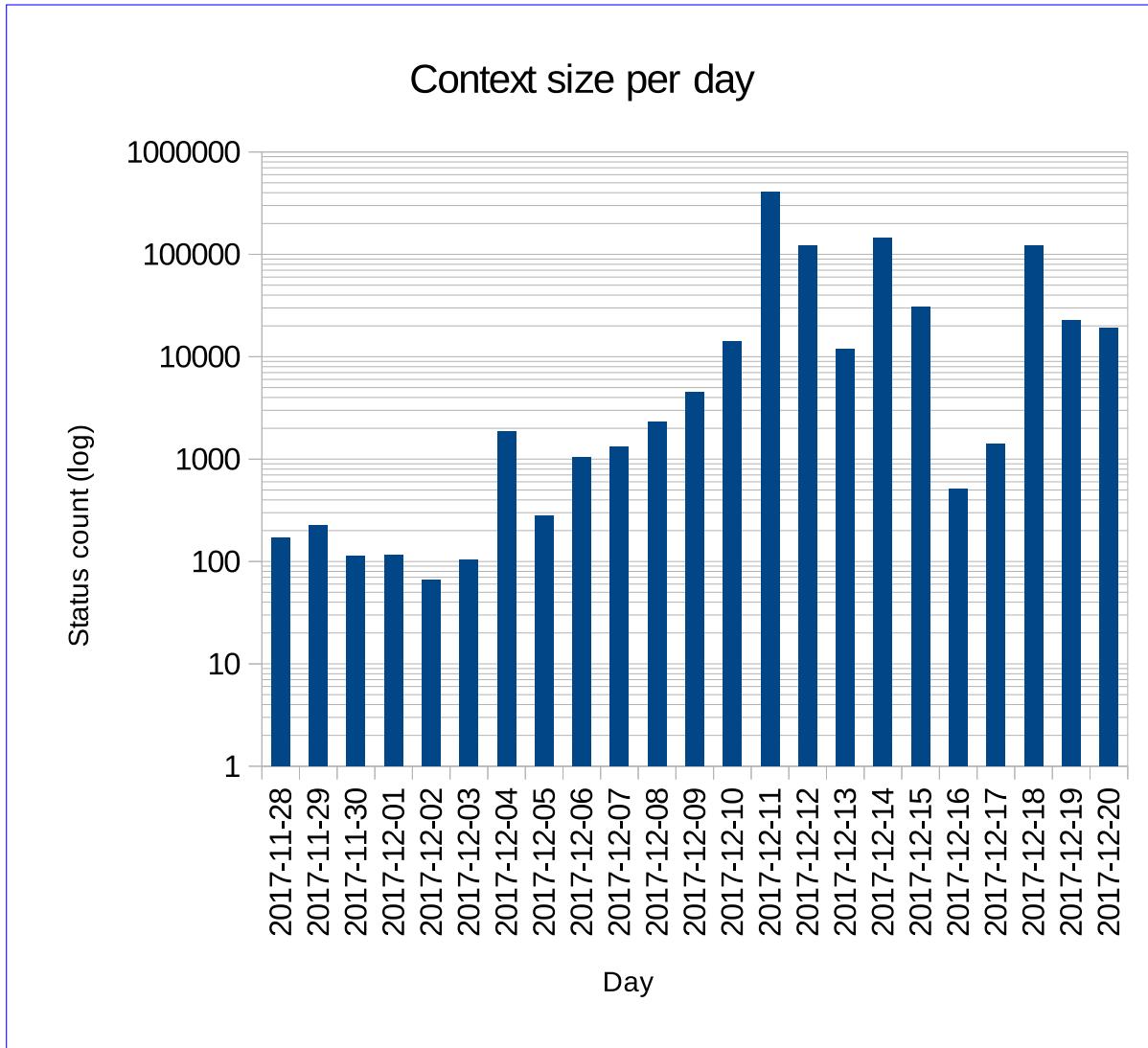


Figure 5: Contextual dataset used in the experiment, where each day has a different status count.

#### 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 [52]. In order to consider the quality of the sorting, the NDCG is used:

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

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

$$NDCG = \frac{DCG}{DCG_{perfect}} \quad (16)$$

where  $DCG_{perfect}$  is a perfect sorting of the items, i.e., the list of items sorted by their value in the test set.

### 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 6, 7, and 8 show the results of the 3 techniques compared in the 3dprinting QA dataset. The three figures show in X axis the alpha parameter and in Y axis the NDCG. Note that for the  $\alpha = 1.00$  setup the result of all approaches are equal to those of LSA approach because the context profile is ignored in this setup. The series denote the context, hence its position shows the results of the proposal with the corresponding *alpha* value for the day.

In Figure 6 it can be noticed that, although the proposal improves the results of LSA ( $\alpha = 1.00$ ) 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, LSAContext improves the results of LSA for all alpha values.

Figure 7 shows that LSAContextFuzzy improves the results of LSA ( $\alpha = 1.00$ ). It obtains better results than LSAContext, given that the results are distributed higher than those of the LSAContext approach. If we focus on the results on single days, it can be noticed that LSAContextFuzzy improves greatly for the context of day 2017-11-28. However, 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 value of the LSA approach. If we focus specifically on the day 2017-12-12, there is a decay for  $\alpha \in [0.90, 0.95]$  but it improves the results of LSA for  $\alpha \in [0.96, 0.99]$ .

Figure 8 shows that LSAContextClustering improves the results of LSA ( $\alpha = 1.00$ ) for most of the contexts explored. It consistently obtains better results than LSA, although the improvements obtained in some contexts is canceled with worst results in other contexts.

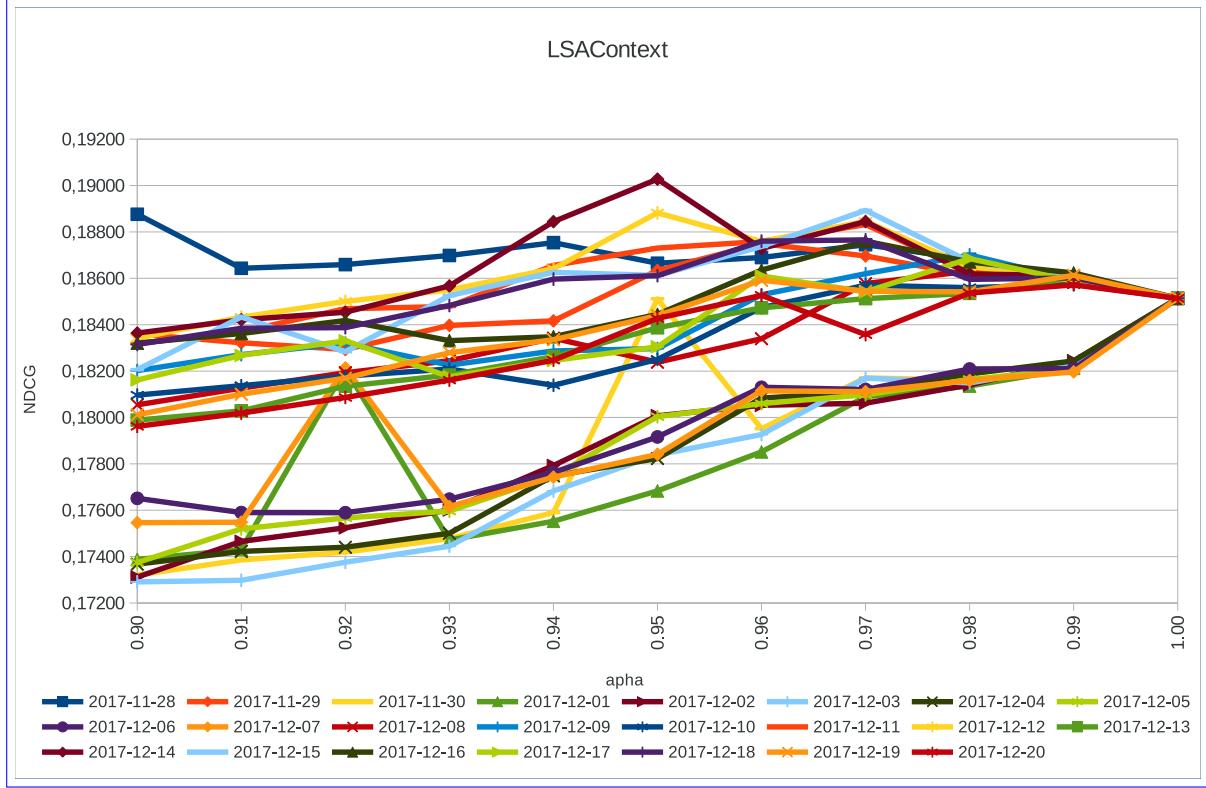


Figure 6: Results managing context without clustering

Figure 9 compares the results of each proposal with the best alpha. Here LSA has no variability across days because it does not consider context. LSAContext has a great variability in performance across days, however, the better results obtained from 2017-12-08 onwards are canceled by the low performance from 2017-11-30 to 2017-12-07. LSAContextClusteringMax and LSAContextClusteringFuzzy performances did not drop drastically in certain contexts as it happened for LSAContext. Instead, they show ups and downs in the contexts. It is worth to notice that LSAContextClusteringMax obtained greater peaks than LSAContextClusteringFuzzy.

Figure 10 summarizes the results of the three proposals as compared to the LSA approach. It shows that LSAContextFuzzy and LSAContext, although they obtain better results than LSA in some contexts, in average they do not provide improvement. In the case of LSAContextClustering, 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.

The best approaches of the compared ones is the LSAContextClustering with  $\alpha = 0.94$ . This

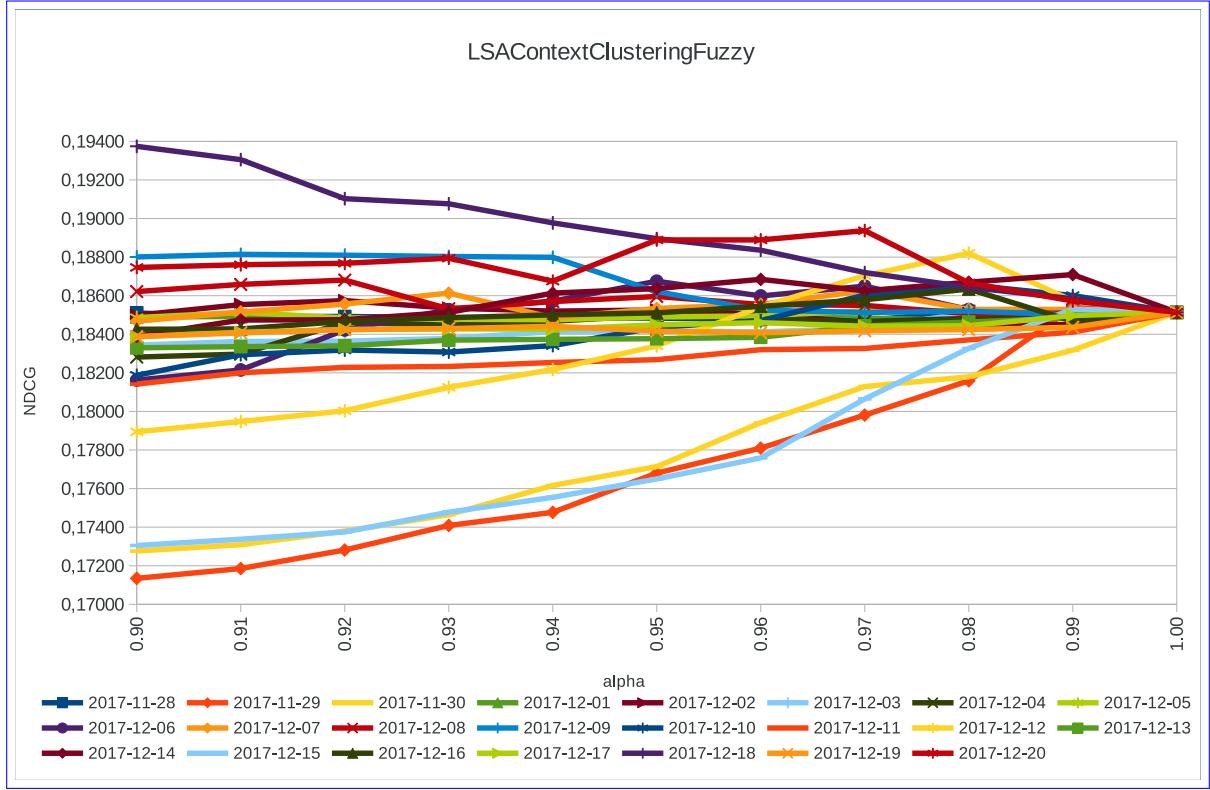


Figure 7: Results managing context with fuzzy membership.

value has been optimized for the 3dprinting QA dataset, hence, for other domains it needs to be adjusted. This parameter provides the LSAContextClustering with flexibility to adapt to different QA domains.

## V. CONCLUSIONS 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. It shows that all improve-

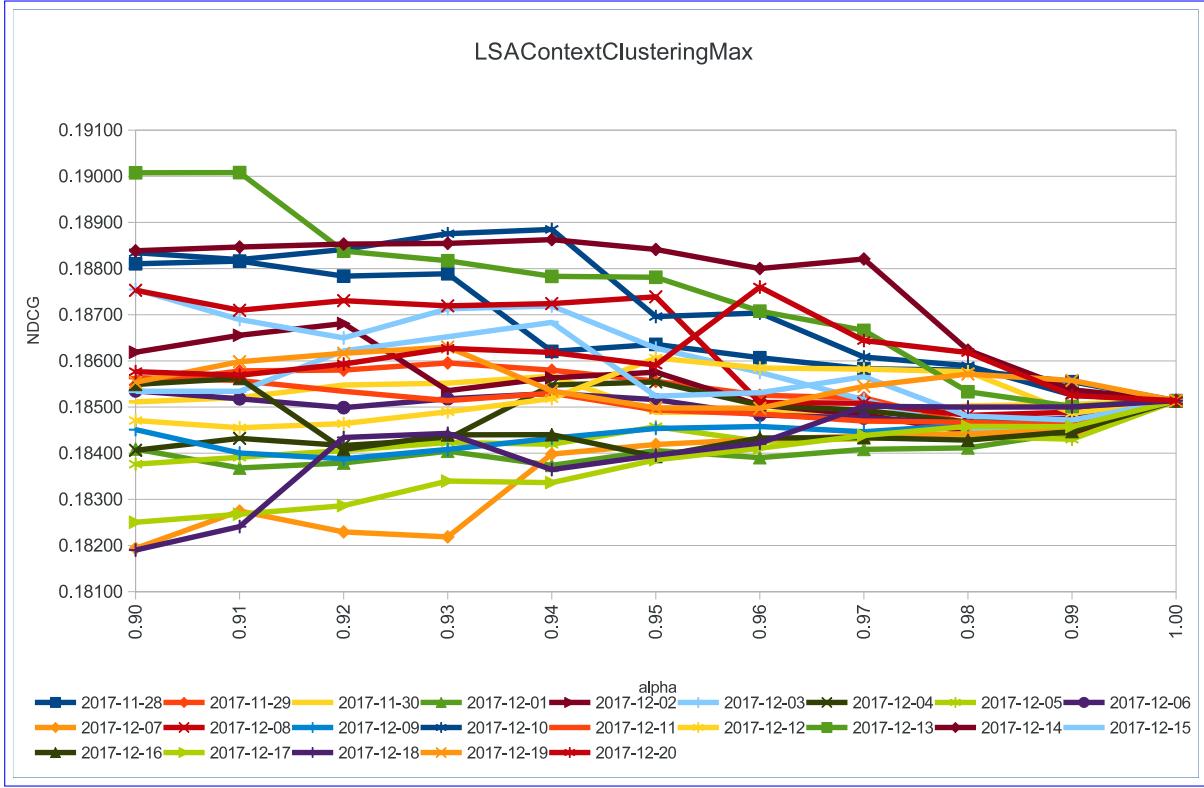


Figure 8: Results managing context with max membership.

ments done provide better results as compared to the baseline method (LSA). 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.

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.

## REFERENCES

- [1] J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, “A web-based personalized business partner recommendation system using fuzzy semantic techniques,” *Computational Intelligence*, vol. 29, no. 1, pp. 37–69, 2013.
- [2] D. Wu, J. Lu, and G. Zhang, “A fuzzy tree matching-based personalized e-learning recommender system,” *IEEE Transactions on Fuzzy Systems*, vol. 23, no. 6, pp. 2412–2426, 2015.
- [3] R. Yera Toledo and Y. Caballero Mota, “An e-learning collaborative filtering approach to suggest problems to solve in programming online judges,” *International Journal of Distance Education Technologies*, vol. 12, no. 2, pp. 51–65, 2014.
- [4] M. Al-Hassan, H. Lu, and J. Lu, “A semantic enhanced hybrid recommendation approach: A case study of e-government tourism service recommendation system,” *Decision Support Systems*, vol. 72, pp. 97 – 109, 2015.

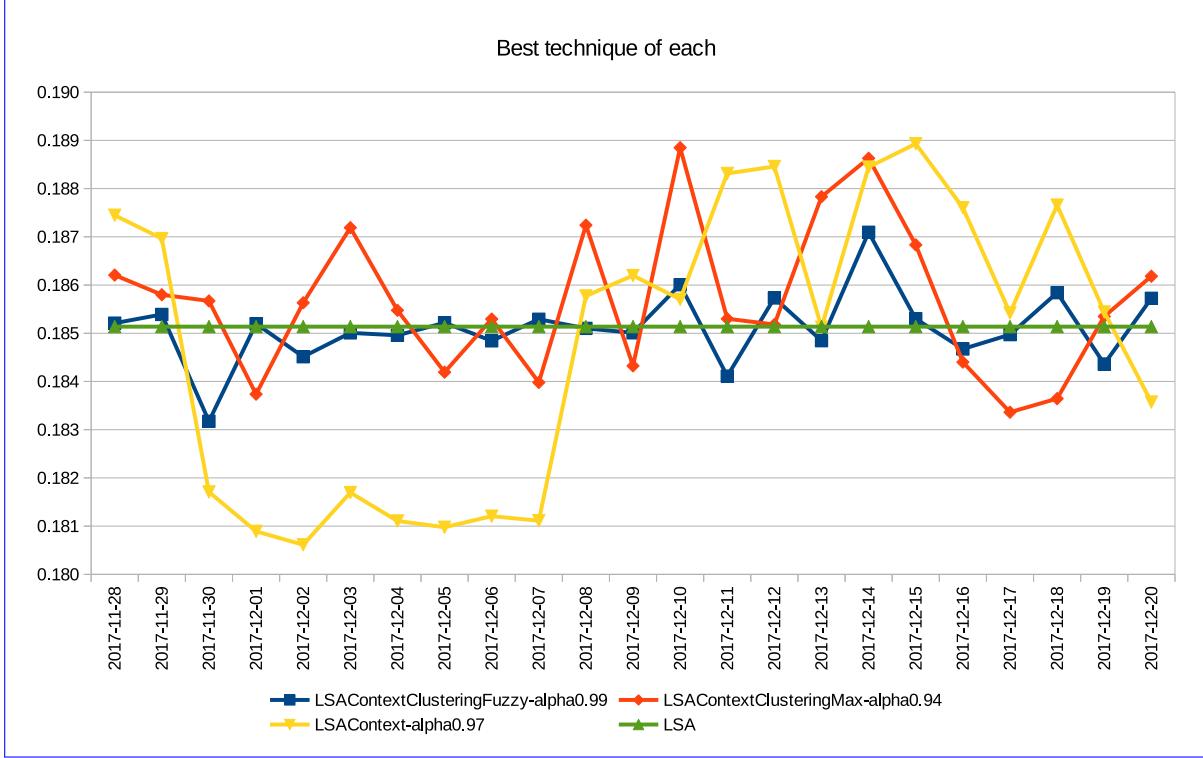


Figure 9: Results of the proposal to manage context with fuzzy membership.

- [5] J. Noguera, M. Barranco, R. Segura, and L. Martínez, “A mobile 3d-gis hybrid recommender system for tourism,” *Information Sciences*, vol. 215, pp. 37–52, 2012.
- [6] D. Rafailidis and A. Nanopoulos, “Modeling users preference dynamics and side information in recommender systems,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 782–792, 2016.
- [7] T. T. S. Nguyen, H. Y. Lu, and J. Lu, “Web-page recommendation based on web usage and domain knowledge,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 10, pp. 2574–2587, 2014.
- [8] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu, “Uncertainty analysis for the keyword system of web events,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 46, no. 6, pp. 829–842, 2016.
- [9] C. Musto, G. Semeraro, P. Lops, M. de Gemmis, and G. Lekkas, “Personalized finance advisory through case-based recommender systems and diversification strategies,” *Decision Support Systems*, vol. 77, pp. 100 – 111, 2015.
- [10] Y. Koren and R. Bell, *Advances in Collaborative Filtering*. Boston, MA: Springer US, 2015, pp. 77–118.
- [11] 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. Springer US, 2015, pp. 119–159.
- [12] M. Nilashi, O. Ibrahim, and K. Bagherifard, “A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques,” *Expert Systems with Applications*, vol. 92, pp. 507 – 520, 2018.
- [13] 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 Systems with Applications*, vol. 42, no. 3, pp. 1202–1222, 2015.
- [14] S. Deng, L. Huang, and G. Xu, “Social network-based service recommendation with trust enhancement,” *Expert Systems with Applications*, vol. 41, no. 18, pp. 8075 – 8084, 2014.

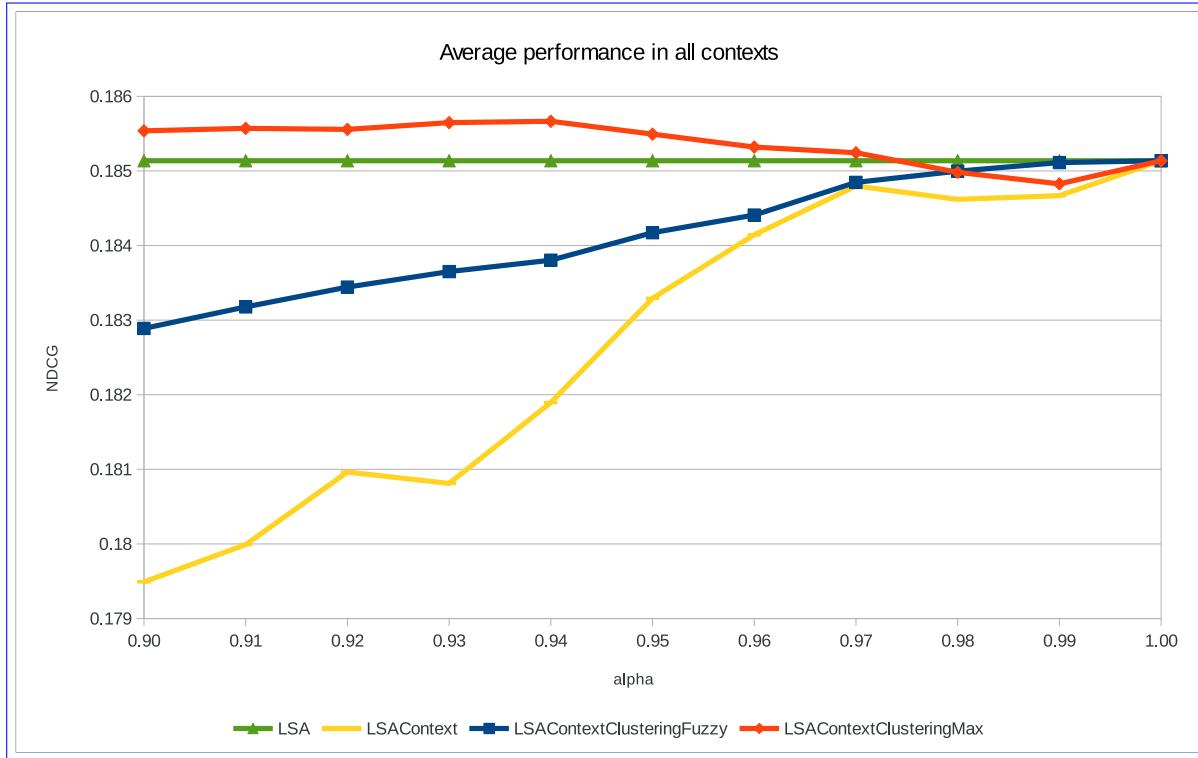


Figure 10: Average NDCG of the compared approaches in all contexts.

- [15] G. Adomavicius and A. Tuzhilin, *Context-Aware Recommender Systems*. Springer US, 2015, pp. 191–226.
- [16] J. Castro, R. Yera, and L. Martínez, “An empirical study of natural noise management in group recommendation systems,” *Decision Support Systems*, vol. 94, pp. 1 – 11, 2017.
- [17] J. Masthoff, “Group recommender systems: Aggregation, satisfaction and group attributes,” in *Recommender Systems Handbook*, F. Ricci, L. Rokach, and B. Shapira, Eds. Springer US, 2015, pp. 743–776.
- [18] N. Kuchmann-Beauger, M.-A. Aufaure, and R. Thollot, “Context-aware question answering system,” Jan. 13 2015, uS Patent 8,935,277.
- [19] A. Figueiroa and G. Neumann, “Context-aware semantic classification of search queries for browsing community question–answering archives,” *Knowledge-Based Systems*, vol. 96, pp. 1 – 13, 2016.
- [20] A. Belen Barragans-Martínez, M. Rey-Lopez, 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 COMPUTING*, vol. 14, no. 6, pp. 23–30, NOV-DEC 2010.
- [21] C. C. Aggarwal, *Content-Based Recommender Systems*. Springer International Publishing, 2016, pp. 139–166.
- [22] B. Shao and J. Yan, “Recommending answerers for stack overflow with lda model,” in *Proceedings of the 12th Chinese Conference on Computer Supported Cooperative Work and Social Computing*, ser. ChineseCSCW ’17. New York, NY, USA: ACM, 2017, pp. 80–86.
- [23] X. L. Zheng, C. C. Chen, J. L. Hung, W. He, F. X. Hong, and Z. Lin, “A hybrid trust-based recommender system for online communities of practice,” *IEEE Transactions on Learning Technologies*, vol. 8, no. 4, pp. 345–356, Oct 2015.
- [24] O. Odiete, T. Jain, I. Adaji, J. Vassileva, and R. Deters, “Recommending programming languages by identifying skill gaps using analysis of experts. a study of stack overflow,” in *Adjunct Publication of the 25th Conference on User Modeling*,

- Adaptation and Personalization*, ser. UMAP '17. New York, NY, USA: ACM, 2017, pp. 159–164.
- [25] T. De Pessemier, C. Courtois, K. Vanhecke, K. Van Damme, L. Martens, and L. De Marez, “A user-centric evaluation of context-aware recommendations for a mobile news service,” *Multimedia Tools and Applications*, vol. 75, no. 6, pp. 3323–3351, Mar 2016.
  - [26] L. Ponzanelli, “Holistic recommender systems for software engineering,” in *Companion Proceedings of the 36th International Conference on Software Engineering*, ser. ICSE Companion 2014. New York, NY, USA: ACM, 2014, pp. 686–689.
  - [27] L. Ponzanelli, G. Bavota, M. D. Penta, R. Oliveto, and M. Lanza, “Prompter: A self-confident recommender system,” in *2014 IEEE International Conference on Software Maintenance and Evolution*, Sept 2014, pp. 577–580.
  - [28] 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 *Proceedings of the 39th International Conference on Software Engineering*, ser. ICSE '17. Piscataway, NJ, USA: IEEE Press, 2017, pp. 94–105.
  - [29] N. Parikh and N. Sundaresan, “Buzz-based recommender system,” in *Proceedings of the 18th International Conference on World Wide Web*, ser. WWW '09. New York, NY, USA: ACM, 2009, pp. 1231–1232.
  - [30] A. Hermida, “Twittering the news,” *Journalism Practice*, vol. 4, no. 3, pp. 297–308, 2010.
  - [31] 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,” *Knowledge-Based Systems*, vol. 117, pp. 3 – 15, 2017, volume, Variety and Velocity in Data Science.
  - [32] L. R. Nair and S. D. Shetty, “Streaming twitter data analysis using spark for effective job search,” *Journal of Theoretical and Applied Information Technology*, vol. 80, no. 2, p. 349, 2015.
  - [33] R. Burke, “Hybrid recommender systems: Survey and experiments,” *User Modelling and User-Adapted Interaction*, vol. 12, no. 4, pp. 331–370, 2002.
  - [34] P. Symeonidis, A. Nanopoulos, and Y. Manolopoulos, “Feature-weighted user model for recommender systems,” in *Proceedings of the 11th international conference on User Modeling*. Springer-Verlag, 2007, pp. 97–106.
  - [35] J. Castro, R. M. Rodríguez, and M. J. Barranco, “Weighting of features in content-based filtering with entropy and dependence measures,” *International Journal of Computational Intelligence Systems*, vol. 7, no. 1, pp. 80–89, 2014.
  - [36] U. Erra, S. Senatore, F. Minnella, and G. Caggianese, “Approximate tf-idf based on topic extraction from massive message stream using the gpu,” *Information Sciences*, vol. 292, pp. 143 – 161, 2015.
  - [37] M. F. Porter, “An algorithm for suffix stripping,” *Program*, vol. 14, no. 3, pp. 130–137, 1980.
  - [38] P. Lops, M. Gemmis, and G. Semeraro, “Content-based recommender systems: State of the art and trends,” in *Recommender Systems Handbook*. Springer US, 2011, ch. 3, pp. 73–105.
  - [39] R. Bambini, P. Cremonesi, and R. Turrin, *A Recommender System for an IPTV Service Provider: a Real Large-Scale Production Environment*. Springer US, 2011, pp. 299–331.
  - [40] F. Ricci, “Contextualizing recommendations,” in *ACM RecSys Workshop on Context-Aware Recommender Systems (CARS 2012)*. In: *Conjunction with the 6th ACM Conference on Recommender Systems (RecSys 2012)*. ACM, 2012.
  - [41] G. Adomavicius and A. Tuzhilin, “Context-aware recommender systems,” in *Recommender Systems Handbook*. Springer US, 2011, ch. 3, pp. 217–253.
  - [42] U. Panniello, A. Tuzhilin, M. Gorgoglione, C. Palmisano, and A. Pedone, “Experimental comparison of pre- vs. post-filtering approaches in context-aware recommender systems,” in *Proceedings of the third ACM conference on Recommender systems*, ser. RecSys '09. New York, NY, USA: ACM, 2009, pp. 265–268.
  - [43] U. Panniello, A. Tuzhilin, and M. Gorgoglione, “Comparing context-aware recommender systems in terms of accuracy and diversity,” *User Modeling and User-Adapted Interaction*, vol. 24, no. 1-2, pp. 35–65, 02 2014.

- [44] G. Adomavicius, R. Sankaranarayanan, S. Sen, and A. Tuzhilin, "Incorporating contextual information in recommender systems using a multidimensional approach," *ACM Transactions on Information Systems*, vol. 23, no. 1, pp. 103–145, 2005.
- [45] L. Baltrunas, B. Ludwig, and F. Ricci, "Matrix factorization techniques for context aware recommendation," in *Proceedings of the Fifth ACM Conference on Recommender Systems*, ser. RecSys '11. New York, NY, USA: ACM, 2011, pp. 301–304.
- [46] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008.
- [47] M. Zaharia, R. S. Xin, P. Wendell, T. Das, M. Armbrust, A. Dave, X. Meng, J. Rosen, S. Venkataraman, M. J. Franklin *et al.*, "Apache spark: a unified engine for big data processing," *Communications of the ACM*, vol. 59, no. 11, pp. 56–65, 2016.
- [48] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen *et al.*, "Mllib: Machine learning in apache spark," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 1235–1241, 2016.
- [49] S. Chintapalli, D. Dagit, B. Evans, R. Farivar, T. Graves, M. Holderbaugh, Z. Liu, K. Nusbaum, K. Patil, B. J. Peng *et al.*, "Benchmarking streaming computation engines: storm, flink and spark streaming," in *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International*. IEEE, 2016, pp. 1789–1792.
- [50] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2-3, pp. 191–203, 1984.
- [51] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 285–295.
- [52] A. Gunawardana and G. Shani, *Evaluating recommender systems*. Springer US, 2015, pp. 265–308.



**Jorge Castro** received the BSc. and MSc. degrees in Computer Scientifience from the University of Jaén, Spain. In 2011 he started in research in the same university, in which he collaborated in several research projects. He is currently a PhD student in the Department of Computer Science and Artificial Intelligence in the University of Granada, Spain. During his PhD, he is also developing a software library focused on experimenting with recommendation algorithms. His research interests are recommender systems, group recommendation and context-aware recommendation.



**Jie Lu** (F'17) is a Distinguished Professor, the Director of the Centre for Artificial Intelligence, and the Associate Dean (Research Excellence) with the Faculty of Engineering and Information Technology at the University of Technology Sydney, Australia. She received her PhD in information systems from the Curtin University of Technology, Australia, in 2000.

Her research expertise spans fuzzy transfer learning, decision support systems, recommender systems, concept drift, and their applications in e-business. She has published 10 research books and over 400 papers in refereed journals and conference proceedings, with over 170 papers in IEEE Transactions and other international journals. She has been awarded eight Australian Research Council (ARC) Discovery Project grants and many other research grants. She is a member of the ARC College of Experts.

She serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for the International Journal on Computational Intelligence Systems (Atlantis), Associate Editor for IEEE Transactions on Fuzzy Systems, Editor for a book series on Intelligent Information Systems (World Scientific), and has served as a guest editor of 12 special issues, general/PC/organization chairs for ten international conferences as well as having delivered 16 keynote/plenary speeches at IEEE and other international conferences.



**Guangquan Zhang** is an Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory at the Center for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia. He received his PhD in applied mathematics from Curtin University of Technology, Australia, in 2001.

His research interests include fuzzy sets and systems, fuzzy optimization, fuzzy transfer learning, and fuzzy modelling in machine learning and data analytics. He has authored four monographs, five textbooks, and 300 papers including 154 refereed international journal papers.

Dr. Zhang has won seven Australian Research Council (ARC) Discovery Projects grants and many other research grants. He was awarded an ARC QEII fellowship in 2005. He has served as a member of the editorial boards of several international journals, as a guest editor of eight special issues of IEEE Transactions and other international journals, and has cochaired several international conferences and workshops in the area of fuzzy decision-making and knowledge engineering.



**Luis Martínez** (M'10) received the M.Sc. and Ph.D. degrees in computer sciences 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 Visiting Professor in University of Technology Sydney, University of Portsmouth (Isambard Kingdom Brunel Fellowship Scheme), and in the Wuhan University of Technology (Chutian Scholar), Guest Professor in the Southwest Jiaotong University and Honourable professor in Xihua University both in Chengdu (China). He has co-edited eleven journal special issues on fuzzy preference modelling, soft computing, linguistic decision making and fuzzy sets theory and has been main researcher in 14 R&D projects, also has published more than 100 papers in journals indexed by the SCI and more than 150 contributions in International Conferences related to his areas. His current research interests include decision making, fuzzy logic-based systems, computing with words and recommender systems. He was a recipient of the IEEE Transactions on fuzzy systems Outstanding Paper Award 2008 and 2012 (bestowed in 2011 and 2015, respectively). He is a 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. He is a member of IEEE and of the European Society for Fuzzy Logic and Technology. Eventually, he has been appointed as Highly Cited Researcher 2017 in Computer sciences